CIM Series in Mathematical Sciences 3

Raquel Pinto
Paula Rocha Malonek
Paolo Vettori *Editors*

Coding Theory and Applications

4th International Castle Meeting, Palmela Castle, Portugal, September 15–18, 2014

Springer

CIM Series in Mathematical Sciences

Volume 3

The CIM Series in Mathematical Sciences is published on behalf of and in collaboration with the Centro Internacional de Matemática (CIM) in Coimbra, Portugal. Proceedings, lecture course material from summer schools and research monographs will be included in the new series.

More information about this series at
http://www.springer.com/series/11745

Raquel Pinto • Paula Rocha Malonek • Paolo Vettori
Editors

Coding Theory and Applications

4th International Castle Meeting, Palmela Castle, Portugal, September 15–18, 2014

Springer

Editors

Raquel Pinto
Department of Mathematics
University of Aveiro
Aveiro, Portugal

Paula Rocha Malonek
Department of Electrical and
Computer Engineering
University of Porto
Porto, Portugal

Paolo Vettori
Department of Mathematics
University of Aveiro
Aveiro, Portugal

ISSN 2364-950X ISSN 2364-9518 (electronic)
CIM Series in Mathematical Sciences
ISBN 978-3-319-17295-8 ISBN 978-3-319-17296-5 (eBook)
DOI 10.1007/978-3-319-17296-5

Library of Congress Control Number: 2015945097

Mathematics Subject Classification (2010): 11T71, 68P30, 94A60, 94B05, 94B10, 94B12, 94B15, 94B20, 94B25, 94B27, 94B30, 94B35, 94B40, 94B50, 94B60, 94B65, 94B70, 94B75, 94B99

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

It is a distinct pleasure for us to present this volume, which gathers the results of the Proceedings of the 4th International Castle Meeting on Coding Theory and its Applications (4ICMCTA) held at Palmela Castle, Portugal, on September 15–18, 2014.

The 4ICMCTA meeting was organized under the auspices of the Research & Developmente Center for Mathematics and Applications (CIDMA) from the University of Aveiro. Following in the spirit of the previous installments held at La Mota Castle, Spain, in 1999 and 2008, and in Cardona Castle, Spain, in 2011, the meeting was a good opportunity for communicating new results, exchanging ideas, strengthening international cooperation, and introducing young researchers into the Coding Theory community.

The event's scientific program consisted of four invited talks and 39 regular talks by authors from 24 different countries. This volume contains the contribution of one invited speaker, as well as 37 communications presented at the meeting. The topics represent some of the most relevant research areas in modern Coding Theory: codes and combinatorial structures, algebraic geometric codes, group codes, quantum codes, convolutional codes, network coding and cryptography. We thank all the authors for their participation. We are also grateful to the scientific committee and to all the external reviewers who implemented a careful reviewing process that guaranteed the high quality of the accepted contributions. Moreover, we would like to mention the valuable support of Ángela Barbero and Øyvind Ytrehus from the steering committee.

We also thank all the people who made this meeting possible, namely the Organizing Committee (Paulo Almeida, Isabel Brás, Diego Napp, Ricardo Pereira and Rita Simões), Edubox SA and the University of Aveiro for the administrative support. In particular, we thank CIDMA, the Portuguese Foundation for Science and Technology (FCT) and the Portuguese International Center for Mathematics (CIM) for their financial support.

Last but not least, we would like to thank CIM and Springer-Verlag for giving us the opportunity to publish these proceedings in the Springer CIM Series in Mathematical Sciences.

February 2015

Raquel Pinto
Paula Rocha
Paolo Vettori

Contents

Part I Invited Talk

Part II Communications

Part I
Invited Talk

Capacity of Higher-Dimensional Constrained Systems

Brian Marcus

Abstract One-dimensional constrained systems, also known as discrete noiseless channels and sofic shifts, have a well-developed theory and have played an important role in applications such as modulation coding for data recording. Shannon found a closed form expression for the capacity of such systems in his seminal paper, and capacity has served as a benchmark for the efficiency of coding schemes as well as a guide for code construction. Advanced data recording technologies, such as holographic recording, may require higher-dimensional constrained coding. However, in higher dimensions, there is no known general closed form expression for capacity. In fact, the exact capacity is known for only a few higher-dimensional constrained systems. Nevertheless, there have been many good methods for efficiently approximating capacity for some classes of constrained systems. These include transfer matrix and spatial mixing methods. In this article, we will survey progress on these and other methods.

Keywords Constrained systems • Capacity • Entropy • Constrained coding

1 Introduction

In contrast to error correction coding, the main philosophy of constrained coding is to avoid patterns that are more prone to error rather than to correct error patterns. In some systems, there are certain patterns that will likely lead to failure and so a constrained encoder encodes user data sequences in a way that avoids the most problematic patterns. In practice a constrained encoder is used in cascade with an error correction encoder, and in recent years there has been much work done on

B. Marcus (✉)
Department of Mathematics, University of British Columbia, Vancouver, BC, Canada
e-mail: marcus@math.ubc.ca

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_1

codes that are both constrained codes and error correcting codes. In fact, the line between constrained coding and error correction code has become rather blurred. However, in this article we will not consider questions of error correction.

2 1D Constraints (Sequences)

2.1 *Motivation: Magnetic Recording*

Historically, the main motivation for constrained codes was the magnetic recording channel. The classical system is illustrated in Fig. 1. At every tick of a clock cycle, a ‘1’ is recorded by changing the direction of current in the write-head and a ‘0’ is recorded by keeping the direction the same. This effectively encodes the data into a sequence of magnets. When reading, a ‘1’ is sensed by a change in magnetization and converted into a readback voltage and a ‘0’ is sensed by the absence of voltage.

It is desirable that the 1’s occur sufficiently frequently but not too frequently. The reason for the former is clock drift: the clock is not perfect and thus must be adjusted in order to keep synchronization with the data patterns; this is done by observing a sufficient set of nonzero voltage samples. The reason for the latter is intersymbol interference: if 1’s are too close to each other, then they could interfere and cancel out or mislead the detector to thinking that at least one of the 1’s is in the wrong position (see Fig. 2).

Clock drift and intersymbol interference can be mitigated by encoding data into binary sequences with lower and upper bounds d and k on the runlengths of zeros between successive 1’s. This constraint is called the RLL(d, k) constraint (see Fig. 3). Such a constraint effectively controls separation between peaks/troughs in the read waveform.

As an example, the following sequence satisfies $RLL(1, 3)$:

$$1\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,1\,0\,0\,0\,1$$

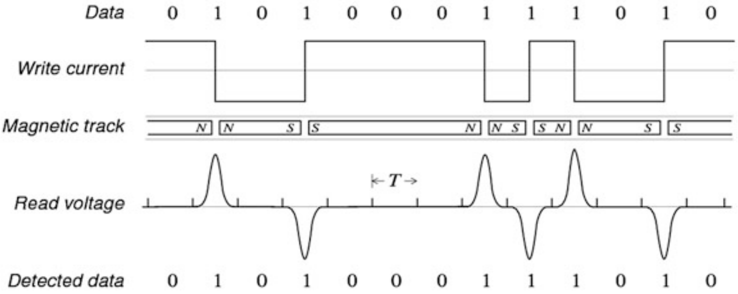


Fig. 1 Magnetic recording channel

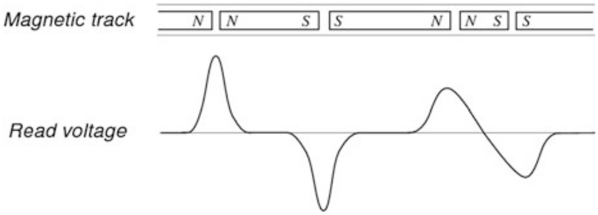


Fig. 2 Intersymbol interference

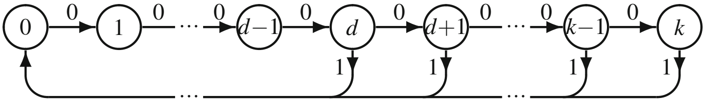


Fig. 3 RLL constraint

A constrained encoder (also known as a modulation encoder) encodes arbitrary user data sequences into constrained sequences. For example, the following table represents an encoder for the $RLL(1,3)$ constraint at rate 1:2:

Previous input	Present input	Present output
0	0	10
1	0	00
–	1	01

Here a ('present') 1-bit input is encoded into a 2-bit output as a function of the input and the previous input. The reader can check that all encoded sequences satisfy RLL(1,3). Note that the input bit can be recovered form the 2-bit output by simply reading off the 2nd bit. In particular, the decoder operates independently block to block, and this feature avoids error propagation in decoding. This particular encoder, known as Modified Frequency Modulation, was one of the first encoders to be used in magnetic recording.

Constrained coding continues to enjoy widespread interest beyond data recording. We refer to [13] for a recent example where constrained codes are used to provide a wide range of trade-offs between rate of information transmission and performance of energy transfer in certain wireless communication systems.

2.2 Definitions and Examples

A *1D constraint* (or *1D constrained system*) is the set of sequences obtained by sequentially reading the labels of a finite directed labeled graph. The labels are chosen from a finite set called an *alphabet*. The labeled graph is often called a *presentation* and the sequences so obtained are often called the *allowed sequences*. Given a 1D constrained system, one can always find a presentation that is deterministic in the sense that at each state, the outgoing edges are labeled distinctly.

The classical examples are the *RLL*(d,k) constraint (Fig. 3), and the *CHG*(b) constraint consisting of sequences of symbols ± 1 where the absolute value of the running sum is required to be bounded by b for all sequences independent of length (Fig. 4); this constraint was imposed to control the spectral content of the set of encoded signals.

Other examples, which are not necessarily of practical interest, include the golden mean constraint where 1's are required to be isolated (Fig. 5), the even (resp., odd) constraint where runlengths of zeros are required to be even (resp., odd) (Figs. 6 and 7). For instance, the following sequence satisfies the even constraint:

0 1 0 0 0 0 1 0 0 1 0 0 1 1 0 0 1

For more background on 1D constraints, see [31]. For background on the related concept of sofic shift, see [25].

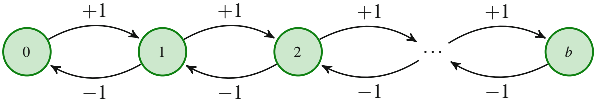


Fig. 4 CHG(b) constraint

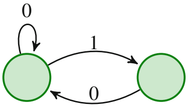


Fig. 5 Golden mean constraint

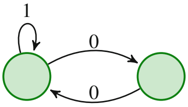
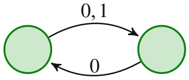


Fig. 6 EVEN constraint

Fig. 7 ODD constraint



2.3 1D Capacity, Exact Computation

Let X be a 1D constraint and $B_n(X)$ be the number of allowed sequences of length n. Define the *capacity* of X as:

$$c(X) := \lim_{n\to\infty} \frac{\log B_n(X)}{n}$$

So, the capacity is the asymptotic growth rate of the number of allowed sequences. In that sense, it is a measure of the size of a constraint. Its operational importance derives from the classical result of Shannon:

Theorem 1 ([39]) *Given a 1D constraint X, $c(X)$ is the supremum of rates of all possible decodable constrained encoders for X.*

Here, decodable means decodable in any sense, including state dependent decoders. The result does not necessarily yield codes at rate exactly equal to $c(X)$. The following results go further.

Theorem 2 ([1, 19]) *Any rate $\leq c(X)$ can be achieved with a finite-state encoder and non-catastrophic decoder.*

We won't give a precise definition of 'non-catastrophic,' but the rough idea is that the decoder will not propagate errors. In most cases of interest, this can be achieved by a sliding-window decoder. In fact, the proof is constructive and in many cases gives efficient encoders by the so-called state-splitting algorithm. However, in today's disk drives, rates of encoding are very high and efficient encoding methods are much different.

The capacity of a 1D constraint is explicitly computable. As a simple example, consider $X = G$, the golden mean constraint. It is easy to show that

$$B_{n+1}(X) = B_n(X) + B_{n-1}(X)$$

and so the sequence $B_n(X)$ is Fibonacci, up to some initial conditions. This sequence grows like powers of the golden mean and hence $c(X) = \log \frac{1+\sqrt{5}}{2} \approx 0.69$. This also explains the origin of the name of the golden mean constraint.

In general, the computation proceeds using simple linear algebra, which we describe as follows. Let A be the adjacency matrix of a deterministic presentation of X; this means that the rows and columns of A are indexed by the vertices of the presentation and A_{uv} is the number of edges from u to v. Note that the information given by the labels of the graph is not incorporated in A.

Theorem 3 (**[39]**) *$c(X) = \log \lambda(A)$, where A is the adjacency matrix of a deterministic presentation of X and $\lambda(A)$ is the largest eigenvalue of A.*

For the golden mean constraint, we see that the adjacency matrix of the presentation in Fig. 5 is:

$$A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$$

An easy computation shows that $\lambda(A) = \frac{1+\sqrt{5}}{2}$, agreeing with the computation of $c(X)$ above.

3 2D Constraints (Arrays)

3.1 Motivation: 2D Recording, Statistical Mechanics

In magnetic and optical storage disks, data is recorded on parallel tracks in a 2D medium. In analogy with intersymbol interference, a constraint may be imposed to limit *inter-track* interference; this could potentially increase track density and overall information density.

Over the past few decades, there has been particular interest in the development of holographic data storage systems. While holographic recording still holds promise, it has not advanced sufficiently to compete with the consistent advances of more conventional magnetic and optical recording. Nevertheless, the system setup illustrates how constrained coding may be used in storage devices of the future.

As indicated in Fig. 8, in such a system a laser illuminates a programmable array, called a spatial light modulator (SLM). A given 2D array of 0's and 1's is represented by a so-called object beam. The interference pattern between the object beam and a simple plane wave, called a reference beam, at given angle of propagation, is

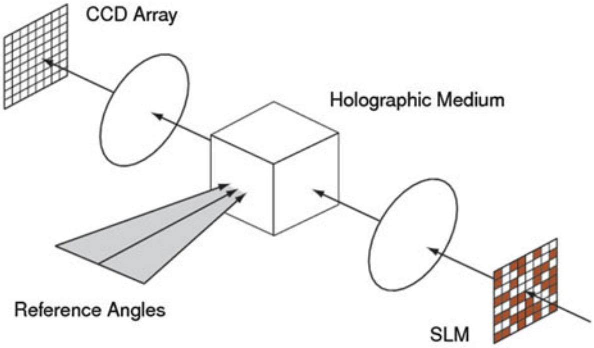


Fig. 8 Holographic recording channel

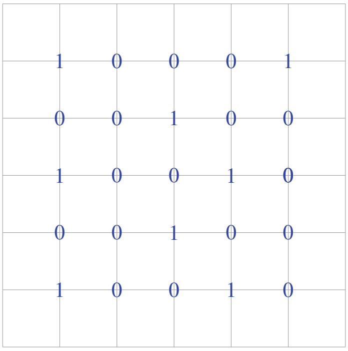
recorded in a three-dimensional medium, such as a crystal. The object beam can be reproduced by illuminating the crystal with the same reference beam used to record it; the object beam can be effectively recovered on an array known as a charge coupled device (CCD). By varying the angle of the reference beam, many arrays can be recorded in the same medium.

While data is physically recorded in a 3D medium, the data really is a collection of 2D arrays. Since light from two adjacent pixels may interfere, inter-symbol interference can be a problem, and for this reason a constraint may be imposed to limit the distance between two 1's. For instance, 1's may be required to be isolated both horizontally and vertically. See Fig. 9 for a typical array that satisfies this constraint. Often this is known as the *hard square* constraint, because it models the hard square lattice gas in statistical mechanics: arrays of "hard" particles, positioned at sites labelled 1, that cannot overlap [40] (see Fig. 10).

Since local variations of light intensity can occur from pixel to pixel, another constraint of interest is the imposition of a balance between the number of 0's and 1's within small regions; this enables the selection of a good local threshold for detection. For more information on the holographic channel, see [2].

Another application of 2D constrained coding is that of 2D barcodes (Fig. 11), used for individual product identification, such as PDF417 for airline tickets and QR for automobiles [42].

Fig. 9 Hard square (HS) constraint



1	0	0	0	1
0	0	1	0	0
1	0	0	1	0
0	0	1	0	0
1	0	0	1	0

Fig. 10 Hard square lattice gas

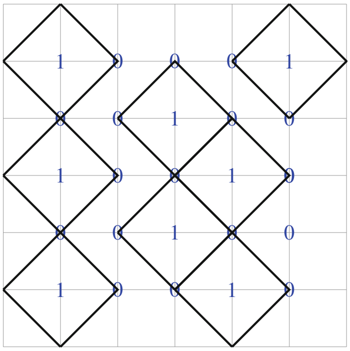
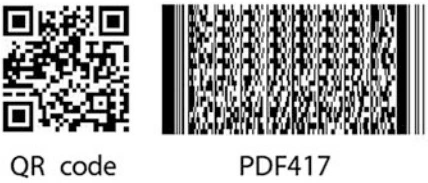


Fig. 11 Bar codes



3.2 Definitions and Examples

A *2D constraint* is the set of all 2D arrays on the square lattice defined by a pair of finite directed labelled graphs ("horizontal" and "vertical"). This can be made precise in many different ways: a pair of vertex-labeled graphs with the same labeled vertices (but different edges), a pair of edge-labeled graphs with the same labeled edges (but different vertices), and a collection of labeled square tiles with colored edges such that the colors of adjacent tiles match up (see, for example, [27]).

However, for our purposes, we find it useful to focus on two special classes of 2D constraints: the square of a 1D constraint, and finite type constraints, described as follows.

Given a 1D constraint X, we define the *square*, $X^{\otimes 2}$, to be the set of all arrays that satisfy X in both the horizontal and vertical direction. For example, observe that the hard square constraint, HS, defined above is the same as the square of the golden mean constraint: $HS = G^{\otimes 2}$

As another example consider, *EVEN*$^{\otimes 2}$, the 2D EVEN constraint, where run-lengths of 0's are required to be even both horizontally and vertically; a typical pattern in this constraint is:

$$\begin{array}{cccccccccccccc}
0&1&0&0&0&0&1&1&0&0&1&1&1&0\\
0&0&1&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&1&0&0&0&0&1&0&0&0&0&0\\
0&1&0&0&0&0&0&0&1&0&0&0&0&0\\
1&1&0&0&1&1&0&0&1&0&0&0&0&1\\
0&1&0&0&0&0&1&1&1&1&0&0&1&0
\end{array}$$

Similarly, we have *ODD*$^{\otimes 2}$, the 2D ODD constraint.

Given a finite list $\mathscr{F}$ of finite patterns, we define the *constraint of finite type* (also called shift of finite type) $X_{\mathscr{F}}$ as the set of all arrays which do not contain any sub-array from $\mathscr{F}$. As an example, the hard square constraint $HS = G^{\otimes 2} = X_{\mathscr{F}}$ where

$$\mathscr{F} = \{\, 11,\ \begin{smallmatrix}1\\1\end{smallmatrix}\}.$$

A variation of HS is the Non-attacking kings constraint (NAK), where 1's are isolated horizontally, vertically and diagonally: $NAK = X_{\mathscr{F}}$ where

$$\mathscr{F} = \left\{ 11,\ \begin{smallmatrix}1\\1\end{smallmatrix},\ \begin{smallmatrix}1&\\&1\end{smallmatrix},\ \begin{smallmatrix}&1\\1&\end{smallmatrix} \right\},$$

with typical allowed pattern:

$$\begin{matrix} 0&1&0&1&0&1&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&0&1\\ 0&0&0&0&1&0&1&0&0&1&0&0&0&0\\ 1&0&1&0&0&0&0&0&0&0&0&1&0&1 \end{matrix}.$$

Such a constraint may be imposed if intersymbol interference is not adequately handled by the hard square constraint.

Finally, we mention the Read/Write Isolated Memory constraint [17], where 1's are required to be isolated horizontally and diagonally, but not necessarily vertically: $RWIM = X_{\mathscr{F}}$ where

$$\mathscr{F} = \left\{ 11,\ \begin{smallmatrix}1&\\&1\end{smallmatrix},\ \begin{smallmatrix}&1\\1&\end{smallmatrix} \right\}$$

with a typical allowed pattern:

$$\begin{matrix} 0&1&0&1&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&1&0&1&0&0&0&0&0\\ 1&0&1&0&0&0&0&0&1&0&0&1&0&1 \end{matrix}$$

Here, instead of a physical 2D array, the array represents a sequence of rewrites of a physical 1D memory. The horizontal constraint is imposed to mitigate intersymbol interference (thereby aiding the read process) and the diagonal constraint is imposed to eliminate the possibility that the write-head will need to re-write two adjacent memory cells in one rewrite (thereby aiding the write process).

3.3 2D Capacity

The definition of capacity of a 2D constraint naturally generalizes the definition of capacity of a 1D constraint.

For a 2D constraint X, let $B_{n\times n}(X)$ denote the number of allowed arrays of size $n \times n$ and define the *capacity* of X as:

$$c(X) := \lim_{n\to\infty} \frac{\log B_{n\times n}(X)}{n^2}$$

In contrast to a 1D constraint X, where the capacity is exactly the log of the largest eigenvalue of a specific matrix associated to X, there is no known general, tractable expression for capacity of a 2D constraint (say in terms of a pair of labelled graphs or a finite list of forbidden finite patterns). In fact, even for constraints as simple as the hard square constraint, the exact capacity is not known. The quest for $c(HS)$ has become somewhat of a "holy grail," in part because of its interpretation as the free energy of a hard square lattice gas [40].

3.4 Examples of Exact Computation

There are some constraints for which capacity is known exactly. One example is a class of constraints with algebraic structure: a *group shift* is a shift-invariant, closed subgroup of G^{Z^2} for some finite group G. The capacity of a group shift is known to be the log of a positive integer, which can be viewed as the size of a basis in some sense [26]. A special case, of particular interest in coding theory, is the class of 2D convolutional codes, which are group shifts where $G = F^n$ for a finite field F and positive integer n.

There are other isolated examples, where capacity is known exactly; however, in each case, constraints obtained by seemingly small variations do not have known capacity:

1. $c(\mathit{RLL}(d, d+1)^{\otimes 2}) = 0$, but $c(\mathit{RLL}(d, d+2)^{\otimes 2})$ is unknown [21].
2. $c(\mathit{CHG}(2)^{\otimes 2}) = 1/4$, but for $b \geq 3$, $c(\mathit{CHG}(b)^{\otimes 2})$ is unknown [27].
3. $c(\mathrm{ODD}^{\otimes 2}) = 1/2$, but $c(\mathrm{EVEN}^{\otimes 2})$ is unknown [27].
4. The hard hexagon model is the set of binary configurations on the 2-dimensional triangular lattice where adjacent vertices cannot both be 1. The capacity of this model is known to be $\log(\lambda)$ where λ is the largest root of a specific degree-24 polynomial [3]. Yet, for similar constraints such as HS, NAK, and RWIM, the capacity is unknown.
5. The q-colored checkerboard C_q is the set of all q-ary configurations on the 2-dimensional square lattice such that adjacent sites have different symbols (each of the q symbols is viewed as a 'color;" so this constraint is the set of configurations of q colors such that adjacent sites have different colors). C_q can be viewed as a constraint of finite type defined by forbidden list:

$$\mathscr{F} = \{ab, \begin{matrix} a \\ b \end{matrix} : a = b\}$$

For instance, a typical element of C_3 is

$$\begin{array}{cccccccccccccc}
0&1&0&2&1&0&1&0&1&2&1&2&1&2\\
2&0&2&1&2&1&0&2&0&1&0&1&0&1\\
1&2&1&0&1&0&1&0&1&2&1&2&1&0\\
0&1&0&2&0&2&0&1&0&1&0&1&2&1
\end{array}$$

It is easy to see that $c(C_2) = 0$. It is known [23] that $c(C_3) = (3/2)\log(4/3)$, yet for $q \geq 4$, $c(C_q)$ is unknown.

6. The dimer model is the set of all tilings of the plane by dominos (1×2 and 2×1 rectangles). See Fig. 12.
This set can be viewed as the set of all configurations on an alphabet of four symbols $\{L, R, T, B\}$ subject to the constraint that to the right of an L must be an R, to the top of a B must be a T, etc. In this way, one sees that this is a constraint of finite type defined by the forbidden list

$$\mathscr{F} = \{LL, LT, LB, RR, TR, BR, \genfrac{}{}{0pt}{}{T}{L}, \genfrac{}{}{0pt}{}{T}{R}, \genfrac{}{}{0pt}{}{T}{T}, \genfrac{}{}{0pt}{}{B}{B}, \genfrac{}{}{0pt}{}{L}{B}, \genfrac{}{}{0pt}{}{R}{B}\}$$

See Fig. 13.
It is known [20] that

$$c(\text{ Dimers }) = \frac{1}{16\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log(4 + 2\cos\theta + 2\cos\phi)\, d\theta d\phi$$

However, for the monomer-dimer model (i.e., tilings by dominos and 1×1 squares), the capacity is unknown.

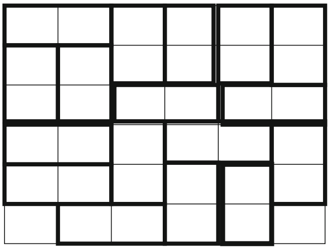
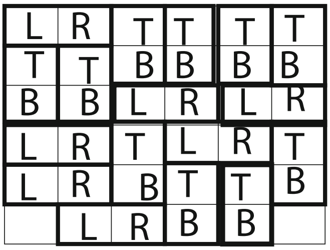


Fig. 12 Dimer tiling



L	R	T	T	T	T
T	T	B	B	B	B
B	B	L	R	L	R
L	R	T	L	R	T
L	R	B	T	T	B
	L	R	B	B	

Fig. 13 Dimer tiling as a constraint of finite type

3.5 Strip Systems

While only a handful of 2D constraints have known capacity, there has been much success in obtaining excellent approximations to capacity. In the following sections, we discuss some approaches to approximations. There are many other techniques, such as to [5, 7, 11, 14, 15, 34, 38] and many more.

Many numerical approximations are based, in one way or another, on the notion of a *strip system*, $S_n := S_n(X)$, defined for any given 2D constraint X and positive integer n, as the set of allowed arrays of X on an n-high strip. A typical configuration for $X = HS$ and $n = 4$ is:

$$\begin{array}{c|c} \hline \uparrow & \cdots 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1 \cdots \\ n & \cdots 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0 \cdots \\ | & \cdots 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0 \cdots \\ \downarrow & \cdots 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0 \cdots \\ \hline \end{array}$$

Observe that S_n itself can be viewed as a 1D constraint, with alphabet consisting of all allowed columns configurations of height n (over the original alphabet of the 2D constraint X). The following result is an easy consequence of the definition.

Theorem 4 *Let X be a 2D constraint and $c_n(X) := c(S_n)$. Then*

$$\lim_{n\to\infty} \frac{c_n(X)}{n} = c(X).$$

Thus, 2D capacities can be approximated by 1D capacities. However, the convergence is typically very slow because generally the time to compute $c_n(X)$ is exponential in n, owing to the exponential size of the alphabet of S_n.

Now assume, for simplicity, that X is a *nearest neighbour constraint*; this means that X is a constraint of finite type, defined by forbidden patterns only on rectangles of size 1×2 and 2×1. For instance, the hard square constraint, HS, is nearest neighbour. For such a constraint, let A_n denote the set of all allowed n-letter columns

$$\begin{matrix} a_n \\ \vdots \\ a_2 \\ a_1 \end{matrix}$$

in X. Then the pair $c = \begin{matrix} a_n \\ \vdots \\ a_2 \\ a_1 \end{matrix}$, $d = \begin{matrix} b_n \\ \vdots \\ b_2 \\ b_1 \end{matrix}$ is allowed in S_n if and only if

$$\begin{matrix} a_n b_n \\ \vdots \\ a_2 b_2 \\ a_1 b_1 \end{matrix}$$

is allowed in the original 2D constraint X. Defining the *horizontal transfer matrix* H_n, indexed by A_n, by:

$$(H_n)_{c,d} = 1 \text{ iff } cd \text{ is allowed}$$

we see that $c_n(X) = \log(\lambda(H_n))$. It follows that

$$c(X) = \lim_{n\to\infty} \frac{\log(\lambda(H_n))}{n}.$$

3.6 Numerical Approximations in Symmetric Case

In this section we give a rough idea of an efficient way to find lower bounds on the capacity of certain 2D constraints.

For this, assume that the constraint is not only nearest neighbor but also the square of a 1D constraint that is *symmetric* in the sense that ab is allowed iff ba is allowed (note that the hard square constraint satisfies all of these properties). Then each H_n is a symmetric matrix. It follows that $\lambda(H_n)$ is lower bounded by its *Rayleigh quotients*; in particular, letting $\mathbf{1}_n$ denote the vector of all 1's, we have

$$\lambda(H_n) \geq \frac{\mathbf{1}_n H_n \mathbf{1}'_n}{\mathbf{1}_n \cdot \mathbf{1}'_n},$$

where v' denotes transpose. It follows that for all p,

$$\lambda((H_n)^p) \geq \frac{\mathbf{1}_n (H_n)^p \mathbf{1}'_n}{\mathbf{1}_n \cdot \mathbf{1}'_n},$$

Thus,

$$c(X) = \lim_{n\to\infty} \frac{\log(\lambda((H_n)^p))}{pn} \geq \lim_{n\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (H_n)^p \mathbf{1}'_n}{\mathbf{1}_n \cdot \mathbf{1}'_n}$$

Since the limit is over n, we must, at the very least, construct H_n for ever increasing n in order to obtain good lower bounds. However, observe that the numerator can be interpreted as the number of allowed $n \times (p+1)$ arrays, and so we can count patterns from top to bottom instead of from left to right:

	← $p+1$ →		
↑	1	0	0
|	0	1	0
n	1	0	1
|	0	1	0
↓	1	0	0

Letting V_p denote the vertical transfer matrix for a strip of width $p+1$, we have

$$\mathbf{1}_n (H_n)^p \mathbf{1}'_n = \mathbf{1}_p (V_p)^{n-1} \mathbf{1}'_p$$

and so

$$c(X) \geq \lim_{n\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_p (V_p)^{n-1} \mathbf{1}'_p}{\mathbf{1}_p \mathbf{1}'_p}$$

This lower bound is much easier to compute: instead of calculating $(H_n)^p$ for a fixed power p and different matrices H_n, we calculate $(V_p)^n$ for a fixed matrix V_p and different powers n. Since the powers of a nonnegative matrix grow like powers of the largest eigenvalue, we then obtain

Theorem 5 ([32]) *Let V_p denote the vertical transfer matrix for a strip of width $p+1$.*

$$c(X) \geq (1/p)(\log(\lambda(V_p)) - \log(\lambda(V_0))).$$

This result was later rediscovered and improved:

Theorem 6 ([6, 9]) *Given p and q,*

$$c(X) \geq (1/p)(\log(\lambda(V_{p+2q})) - \log(\lambda(V_{2q})))$$

Calkin and Wilf also obtained upper bounds using similar considerations.

Further improvements on lower bounds were obtained in Louidor-Marcus [27] by replacing the sequence $\mathbf{1}_n$ with other sequences of vectors $\mathbf{y}_n$, such that $\mathbf{y}_n (H_n)^p \mathbf{y}'_n$ represent "weighted counts." Using these techniques, $c(NAK)$ was determined to within 7 digits, $c(RWIM)$ to within 4 digits, and $c(EVEN^{\otimes 2})$ to within 2 digits.

Further improvements, especially on lower bounds, have been recently obtained by Chan and Rechnitzer [8], using a variation of transfer matrices (called "Corner Transfer Matrices"), originally developed by Baxter in his solution to the capacity of the hard hexagon model. Using these methods, they have improved the estimates above and determined the capacity of the hard square constraint to within 17 digits.

3.7 Asymptotic Rate of Approximation

Recall that for 1D constraints, the capacity can be computed exactly as the log of the largest eigenvalue of a matrix associated with a given constraint. One might ask if the same holds for 2D constraints (possibly with an infinite dimensional matrix)? A sobering thought is that it is algorithmically undecidable to decide whether a constraint has strictly positive capacity [4].

But there is another way to think about this. One can ask what real numbers occur as the capacity of a 2D constraint? The analogue for 1D capacities is known. To describe this result, we need the notion of an *algebraic integer*, which is defined as a root of a polynomial with integer coefficients and leading coefficient $= 1$. For each algebraic integer, there is a unique such polynomial, called the *minimal polynomial*, of minimal degree. The roots of such a polynomial are known as the *algebraic conjugates* of an algebraic integer. A *Perron number* is an algebraic integer which strictly dominates in absolute value all of its algebraic conjugates.

Theorem 7 ([24]) *A real number $c \geq 0$ is the capacity of a 1D constrained system iff it is the log of a root of a Perron number.*

The "only if" part is a fairly easy consequence of the fact that the capacity is the log of the largest eigenvalue of a matrix with nonnegative integer entries. The "if" part is much more subtle, but is constructive given the minimal polynomial.

In contrast, the characterization of numbers which occur as the capacity of a 2D constraint is given by a computational-theoretic, rather than algebraic, characterization as follows.

Theorem 8 ([18]) *A real number $c \geq 0$ is the capacity of a 2D constrained system if and only if c is* right recursively enumerable (rre)*, i.e. there is an algorithm (Turing machine), which, on input n, produces a rational number t_n such that $t_n \geq c$ and $\lim_{n\to\infty} t_n = c$*

The "only if" part can be proved using the approximations given in the definition of capacity, and in fact $t_n := \frac{c_n(X)}{n}$ works. The "if" part is again far more subtle, but somewhat constructive, in the sense fact that for any given rre number, a constraint can be constructed given the associated Turing machine.

Unfortunately, rre numbers, and hence 2D capacities, can be arbitrarily poorly approximable. However, some are better than others. We say that a real number $c \geq 0$ is *approximable in polynomial time* if there is an algorithm (i.e., Turing machine), which, on input n, produces a rational number t_n, computed in polynomial time, s.t. $|t_n - c| < 1/n$. If the capacity of a 2D constraint is approximable in polynomial time, then we can regard its capacity as being "within reach," if not exactly known. The capacity of the hard square constraint has this property:

Theorem 9 ([16, 35]) *$c(HS)$ is approximable in polynomial time.*

The proofs both rely on a notion of strong spatial mixing but otherwise are quite different. Pavlov's approximations are easier to describe: let $r_n := c_{n+1}(HS) - c_n(HS)$; he shows that r_n converges to $c(HS)$ exponentially fast and then "trades off" this exponential convergence with the exponential time to compute r_n, yielding polynomial approximability.

Since then, techniques from both proofs have been applied to other constraints. Marcus and Pavlov [29, 30] have used techniques based on Pavlov's approach to prove polynomial approximability for some general classes of constraints. And

Wang-Yin-Zhong [43] used the Gamarnik-Katz technique to prove polynomial approximability of $c(RWIM)$. However, at present, it is unknown whether $c(NAK)$ is polynomially approximable.

3.8 Higher Dimensions

One can consider d-dimensional constrained systems for any dimension d, in particular the d-th power $X^{\otimes d}$ of a 1D constraint. which generalizes the square of a 1D constraint. Most capacity approximation techniques extend to higher dimensions, but are generally less accurate. There are also a few cases where capacities are known exactly. For instance, exact capacity is known for group shifts in all dimensions. And for all d, $c(CHG(2)^{\otimes d}) = (1/2)^d$, and $c(ODD^{\otimes d}) = 1/2$ [27].

It is easy to see that for any 1D constraint X, the capacity $c(X^{\otimes d})$ is monotonically non-increasing in d, and so the limit as $d \to \infty$ exists. Can one compute this limit?

The *independence entropy* of a 1D constraint measures the contribution to capacity attributed to sequences of positions whose values can be freely switched without violating the constraint. This concept was developed in [28] (where the reader can find a precise definition), based on the precursor notion of maximal insertion rate developed in [36]. The independence entropy of a 1D constraint turns out to be explicitly computable, and has the following significance:

Theorem 10 ([33]) *For a 1D constraint X, $\lim_{d\to\infty} c(X^{\otimes d})$ equals the independence entropy of X.*

So, while the computation of capacity for d-dimensional constraints, $d \geq 2$, is extremely challenging, it is very easy to compute in the limit as $d \to \infty$.

3.9 Encoding for 2D Constraints

Recall from Sect. 2.3 that the capacity of a 1D constraint is the maximal rate of an invertible encoder and that general efficient encoding methods exist. While in some sense capacity of a 2D constraint is the maximal rate of an encoder, the theory and practice of 2D encoding is far behind. Here, we mention just a few ideas.

One simple idea is to use 1D encoding techniques to encode into strip systems and then insert buffer rows between strips to ensure that the constraint is satisfied across the strips. This works only for certain constraints such as the hard square, where one can insert a buffer row of all 0's. As a typical example, consider

$$\begin{array}{cccccccccccccc}
0&1&0&1&0&0&1&0&1&0&0&0&1&0\\
1&0&0&0&0&1&0&0&0&1&0&1&0&1\\
0&1&0&0&1&0&1&0&1&0&1&0&1&0
\end{array}$$

$$\begin{array}{cccccccccccccc}
0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}$$

$$\begin{array}{cccccccccccccc}
0&0&1&0&0&0&1&0&0&1&0&0&1&0\\
1&0&0&0&0&1&0&0&0&0&0&1&0&1\\
0&0&1&0&1&0&1&0&1&1&0&0&0&0
\end{array}$$

A more interesting class of encoders is the class of 2D Bit Stuffing Encoders [37]. Again this works only for certain constraints. For example, for the hard square constraint, one encodes an information sequence in a row-by-row rastering fashion: for each information bit ‘0’, write ‘0’ in a given position; for each information bit ‘1’, write ‘1’ in a given position and a ‘stuffed’ ‘0’ immediately below and to the right in order to guarantee that the constraint is satisfied. One decodes sequentially by discarding ‘stuffed’ ‘0’s. This scheme is then improved upon as follows.

1. First encode an information sequence by a variable-rate *transformer* into a biased *IID*$(p, 1-p)$ sequence (i.e., in the output sequences of the transformer, 0 appears with probability p and 1 appears with probability $1 - p$).
2. Then write encoded bits as described above.
3. Optimize over p: increasing p from 1/2 to 1 reduces the number of ‘stuffed’ 0’s and therefore increases the number of available encoding positions, but it also reduces the rate of the transformer.
4. The optimal p can be determined analytically and yields an encoder at rate within $1\,\%$ of $c(HS)$.

Further improvements and applications of bit stuffing to other constraints are given in Tal-Roth [41].

Finally we mention that for some 2D group shifts, there are constructions of 2D algebraic encoders. We define an *invertible algebraic 2D encoder* for a group shift $X \subseteq G^{Z^2}$ as a map

$$f : H^{Z^2} \to X$$

(for some finite group H) which is simultaneously a group isomorphism and a sliding-block map (the latter in the sense that there is a positive integer N such that the value, $f(x)_s$, of a configuration of X at a given site $s \in Z^2$ depends only on the values of x in a square of size N centered at s)

For 2D convolutional codes, necessary and sufficient conditions for invertible algebraic 2D encoders were given by Fornasini-Valcher [12]. There has been considerable further work on both the structure of and encoding for group shifts, within both the symbolic dynamics and systems theory communities; see, for example, [10, 22].

Acknowledgements This article is a summary of the lecture I gave, by the same title, at the 4th International Castle Conference on Coding Theory held in Palmela, Portugal. It is a pleasure to thank the organizers for putting together such a stimulating conference, cutting across many topics of both theoretical and practical importance.

References

1. Adler, R., Coppersmith, D., Hassner, M.: Algorithms for sliding block codes – an application of symbolic dynamics to information theory. IEEE Trans. Inf. Theory **29**, 5–22 (1983)
2. Ashley, J., et al.: Holographic data storage. IBM J. Res. Dev. **44**, 341–366 (2000)
3. Baxter, R.: Hard hexagons: exact solution. Physics A **13**, 1023–1030 (1980)
4. Berger, R.: The undecidability of the domino problem. Mem. Am. Math. Soc. **66**, 1–72 (1966)
5. Blahut, R., Weeks, W.: The capacity and coding gain of certain checkerboard codes. IEEE Trans. Inf. Theory **44**, 1193–1203 (1998)
6. Calkin, N., Wilf, H.: The number of independent sets in a grid graph. SIAM J. Discret. Math. **11**, 54–60 (1998)
7. Censor, K., Etzion, T.: The positive capacity region of two-dimensional run-length-constrained channels. IEEE Trans. Inf. Theory **52**, 5128–5140 (2006)
8. Chan, Y., Rechnitzer, A.: Accurate lower bounds on two-dimensional constraint capacities from corner transfer matrices. IEEE Trans. Inf. Theory **60**, 3845–3858 (2014)
9. Engel, K.: On the Fibonacci number of an m by n lattice. Fibonacci Q. **28**, 72–78 (1990)
10. Fagnani, F., Zampieri, S.: Minimal and systematic convolutional codes over finite Abelian groups. Linear Algebra Appl. **378**, 31–59 (2004)
11. Forchhammer, S., Justesen, J.: Entropy bounds for constrained two-dimensional random fields. IEEE Trans. Inf. Theory **45**, 118–127 (1999)
12. Fornasini, E., Valcher, M.: Algebraic aspects of two-dimensional convolutional codes. IEEE Trans. Inf. Theory **40**, 1068–1082 (1994)
13. Fouldadgar, A., Someone, O., Erkip, E.: Constrained codes for joint energy and information transfer with receiver energy utilization requirements. In: Proceedings of IEEE International Symposium on Information Theory, Honolulu, pp. 991–995 (2014)
14. Friedland, S.: On the entropy of Zd subshifts of finite type. Linear Algebra Appl. **252**, 199–220 (1997)
15. Friedland, S., Lundow, P., Markstrom, K.: The 1-vertex transfer matrix and accurate estimation of channel capacity. IEEE Trans. Inf. Theory **56**, 3692–3699 (2010)
16. Gamarnik, D., Katz, D.: Sequential cavity method for computing free energy and surface pressure. J. Stat. Phys. **137**, 205–232 (2009)
17. Golin, M.J., Yong, X., Zhang, Y., Sheng, L.: New upper and lower bounds on the channel capacity of read/write isolated memory. Discret. Appl. Math. **140**, 35–48 (2004)
18. Hochman, M., Meyerovitch, T.: A characterization of the entropies of multidimensional shifts of finite type. Ann. Math. **171**(3), 2011–2038 (2012)
19. Karabed, R., Marcus, B.: Sliding-block coding for input-restricted channels. IEEE Trans. Inf. Theory **34**, 2–26 (1988)
20. Kastelyn, P.: The statistics of dimers on a lattice. Physica A **27**, 1209–1225 (1961)
21. Kato, A., Zeger, K.: On the capacity of two-dimensional run length constrained channels. IEEE Trans. Inf. Theory **45**, 1527–1540 (1999)
22. Kitchens, B.: Multidimensional convolutional codes. SIAM J. Discret. Math. **15**, 367–381 (2002)
23. Lieb, E.: Residual entropy of square ice. Phys. Rev. **162**, 162–172 (1967)
24. Lind, D.: The entropies of topological Markov shifts and a related class of algebraic integers. Ergod. Theory Dyn. Syst. **4**, 283–300 (1984)

25. Lind, D., Marcus, B.: An Introduction to Symbolic Dynamics and Coding. Cambridge University Press, Cambridge (1995, reprinted 1999)
26. Lind, D., Schmidt, K., Ward, T.: Mahler measure and entropy for commuting automorphisms of compact groups. Invent. Math. **101**, 593–629 (1990)
27. Louidor, E., Marcus, B.: Improved lower bounds on capacities of symmetric 2-dimensional constraints using rayleigh quotients. IEEE Trans. Inf. Theory **56**, 1624–1639 (2010)
28. Louidor, E., Marcus, B., Pavlov, R.: Independence entropy of Zd shift spaces. Acta Appl. Math. **126**, 297–317 (2013)
29. Marcus, B., Pavlov, R.: Computing bounds on entropy of $\mathbb{Z}^d$ stationary Markov random fields. SIAM J. Discret. Math. **27**, 1544–1558 (2013)
30. Marcus, B., Pavlov, R.: An integral representation for topological pressure in terms of conditional probabilities (2013, to appear). Isr. J. Math. arXiv:1309.1873v2
31. Marcus, B., Roth, R., Siegel, P.: Constrained systems and coding for recording channels. In: Pless, V.S., Huffman, W.C. (eds.) Handbook of Coding Theory, vol. II, chapter 20. Elsevier Press, Amsterdam/New York (1998)
32. Markley, N., Paul, M.: Maximal measures and entropy for $\mathbb{Z}^\nu$ subshifts of finite type. In: Devaney, R., Nitecki, Z. (eds.) Classical Mechanics and Dynamical Systems. Dekker Notes, vol. 70, pp. 135–157. Dekker, New York (1981)
33. Meyerovitch, T., Pavlov, R.: Entropy and measures of maximal entropy for axial powers of subshifts. Proc. Lond. Math. Soc **109**(4), 921–945 (2014)
34. Ordentlich, E., Roth, R.: Two-dimensional weight-constrained codes through enumeration bounds. IEEE Trans. Inf. Theory **46**, 1292–1301 (2000)
35. Pavlov, R.: Approximating the hard square entropy constant with probabilistic methods. Ann. Probab. **40**, 2362–2399 (2012)
36. Poo, T.L., Chaichanavong, P., Marcus, B.: Trade-off functions for constrained systems with unconstrained positions. IEEE Trans. Inf. Theory **52**, 1425–1449 (2006)
37. Roth, R., Siegel, P., Wolf, J.: Efficient coding scheme for the hard-square model. IEEE Trans. Inf. Theory **47**, 1166–1176 (2001)
38. Schwartz, M., Vardy, A.: New bounds on the capacity of multidimensional run-length constraints. IEEE Trans. Inf. Theory **57**, 4373–4382 (2011)
39. Shannon, C.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423, 623–656 (1948)
40. Simon, B.: The Statistical Mechanics of Lattice Gases. Princeton University Press, Princeton (1993)
41. Tal, I., Roth, R.: Bounds on the rate of 2-d bit-stuffing encoders. IEEE Trans. Inf. Theory **56**, 2561–2567 (2010)
42. Wang, Y.: System for encoding and decoding data in machine readable graphic form. US Patent 5,243,655 (1993)
43. Wang, Y., Yin, Y., Zhong, S.: Approximate capacities of two-dimensional codes by spatial mixing. In: Proceedings of IEEE International Symposium on Information Theory, Honolulu, pp. 1061–1065 (2014)

Part II
Communications

From 1D Convolutional Codes to 2D Convolutional Codes of Rate $1/n$

Paulo Almeida, Diego Napp, and Raquel Pinto

Abstract In this paper we introduce a new type of superregular matrices that give rise to novel constructions of two-dimensional (2D) convolutional codes with finite support. These codes are of rate $1/n$ and degree δ with $n \geq \delta + 1$ and achieve the maximum possible distance among all 2D convolutional codes with finite support with the same parameters.

Keywords 1D and 2D convolutional codes • MDS codes • Superregular matrix

1 Introduction

When considering data recorded in two dimensions, like pictures and video, two-dimensional (2D) convolutional codes [2–5, 7, 8] seem to be a better framework to encode such data than one-dimensional (1D) codes, since it takes advantage of the interdependence of the data in more than one direction. In [3] the distance properties of 2D convolutional codes of rate $1/n$ and degree δ were studied, and constructions of such codes with maximum possible distance (MDS) were given for $n \geq \frac{(\delta+1)(\delta+2)}{2}$. In this paper we relax this restriction and present 2D MDS convolutional codes of rate $1/n$ with $n \geq \delta + 1$. The idea is to consider 1D convolutional codes obtained as the projection of the 2D code on the vertical lines. For that we need to introduce a new type of matrices of a particular structure and show that they are superregular.

P. Almeida • D. Napp • R. Pinto (✉)
Department of Mathematics, CIDMA – Center for Research and Development in Mathematics and Applications, University of Aveiro, Campus Universitario de Santiago, 3810-193 Aveiro, Portugal
e-mail: palmeida@ua.pt; diego@ua.pt; raquel@ua.pt

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_2

2 1D and 2D Convolutional Codes

In this section we give some basic results on 1D and 2D convolutional codes that will be useful throughout the paper.

Let $\mathbb{F}$ be a finite field and $\mathbb{F}[z_2]$ the ring of polynomials in one indeterminate with coefficients in $\mathbb{F}$. A 1**D (finite support) convolutional code** $\mathscr{C}$ of rate k/n is an $\mathbb{F}[z_2]$-submodule of $\mathbb{F}[z_2]^n$, where k is the rank of $\mathscr{C}$ (see [6]). A full column rank matrix $\hat{G}(z_2) \in \mathbb{F}[z_2]^{n\times k}$ whose columns constitute a basis for $\mathscr{C}$ is called an **encoder** of $\mathscr{C}$. So,

$$\begin{aligned}\mathscr{C} &= \mathrm{im}_{\mathbb{F}[z_2]}\, \hat{G}(z_2)\\ &= \left\{\hat{\mathbf{v}}(z_2) \in \mathbb{F}[z_2]^n \mid \hat{\mathbf{v}}(z_2) = \hat{G}(z_2)\hat{\mathbf{u}}(z_2) \text{ with } \hat{\mathbf{u}}(z_2) \in \mathbb{F}[z_2]^k\right\}.\end{aligned}$$

The **weight** of a word $\hat{\mathbf{v}}(z_2) = \sum_{i\geq 0} v(i)z_2^i \in \mathbb{F}[z_2]^n$ is given by

$$\mathrm{wt}(\hat{\mathbf{v}}(z_2)) = \sum_{i\in\mathbb{N}} \mathrm{wt}(v(i)),$$

where the weight of a constant vector $v(i)$ is the number of nonzero entries of $v(i)$ and the **distance** of a 1D convolutional code $\mathscr{C}$ is defined as

$$\mathrm{dist}(\mathscr{C}) = \min\left\{\mathrm{wt}(\hat{\mathbf{v}}(z_2)) \mid \hat{\mathbf{v}}(z_2) \in \mathscr{C}, \text{ with } \hat{\mathbf{v}}(z_2) \neq \mathbf{0}\right\}.$$

If $\mathscr{C}$ is a 1D convolutional code of rate $1/n$, then all its encoders differ by a nonzero constant. The degree of $\mathscr{C}$ is defined as the column degree of any encoder of $\mathscr{C}$. The next result follows immediately.

Corollary 1 ([6]) *Let $\mathscr{C}$ be a 1D convolutional code of rate $1/n$ with degree ν. Then*

$$\mathrm{dist}(\mathscr{C}) \leq n(\nu + 1).$$

A 1D convolutional code of rate $1/n$ with degree ν and distance $n(\nu + 1)$ is said to be Maximum Distance Separable (MDS). In [3] constructions of such codes were given for $n \geq \nu + 1$ as stated in the next theorem.

Theorem 2 *Let $\nu, n \in \mathbb{N}$ with $n \geq \nu + 1$ and $\mathscr{G} = [G_0\ G_1\ \cdots G_\nu]$, with $G_i \in \mathbb{F}^n$, $i = 0, 1, \ldots, \nu$, be a matrix such that all its minors of any order are different from zero. Then $\mathscr{C} = \mathrm{Im}_{\mathbb{F}[z_2]} \sum_{i=0}^{\nu} G_i z_2^i$ is an MDS 1D convolutional code of rate $1/n$ and degree ν.*

We are going to consider now convolutional codes whose codewords belong to $\mathbb{F}[z_1, z_2]^n$, where $\mathbb{F}[z_1, z_2]$ is the ring of polynomials in two indeterminates with coefficients in $\mathbb{F}$. Such codes are called 2D (finite support) convolutional codes. More precisely, a **2D (finite support) convolutional code** $\mathscr{C}$ of rate k/n is a free $\mathbb{F}[z_1, z_2]$-submodule of $\mathbb{F}[z_1, z_2]^n$ of rank k (see [7, 8]). An encoder of $\mathscr{C}$ is a full column rank matrix $\hat{G}(z_1, z_2) \in \mathbb{F}[z_1, z_2]^{n\times k}$ whose columns constitute a basis for $\mathscr{C}$. Therefore,

$$\begin{aligned}\mathscr{C} &= \operatorname{im}_{\mathbb{F}[z_1,z_2]} \hat{G}(z_1, z_2)\\ &= \left\{\hat{\mathbf{v}}(z_1, z_2) \in \mathbb{F}[z_1, z_2]^n \mid \hat{\mathbf{v}}(z_1, z_2)=\hat{G}(z_1, z_2)\hat{\mathbf{u}}(z_1, z_2) \text{ with } \hat{\mathbf{u}}(z_1, z_2)\in\mathbb{F}[z_1, z_2]^k\right\}.\end{aligned}$$

The **weight** of a word $\hat{\mathbf{v}}(z_1, z_2) = \sum_{(i,j)\in\mathbb{N}^2} v(i, j)z_1^i z_2^j \in \mathbb{F}[z_1, z_2]^n$ is defined in a similar way to the 1D case as

$$\operatorname{wt}(\hat{\mathbf{v}}(z_1, z_2)) = \sum_{(i,j)\in\mathbb{N}^2} \operatorname{wt}(v(i, j)),$$

and the **distance** of $\mathscr{C}$ as

$$\operatorname{dist}(\mathscr{C}) = \min\{\operatorname{wt}(\hat{\mathbf{v}}(z_1, z_2)) \mid \hat{\mathbf{v}}(z_1, z_2) \in \mathscr{C}, \text{ with } \hat{\mathbf{v}}(z_1, z_2) \neq \mathbf{0}\}.$$

In this paper, we restrict to 2D convolutional codes of rate $1/n$. Given an encoder

$$\hat{G}(z_1, z_2) = \sum_{(i,j)\in\mathbb{N}^2} G(i, j)z_1^i z_2^j \in \mathbb{F}[z_1, z_2]^n$$

of a 2D convolutional code $\mathscr{C}$ of rate $1/n$, we define the **degree** of $\hat{G}(z_1, z_2)$ as $\delta = \max\{i + j \mid G(i, j) \neq 0\}$. Since two encoders of $\mathscr{C}$ differ by a nonzero constant, all encoders of $\mathscr{C}$ have the same degree and we define the **degree** of $\mathscr{C}$ as the degree of any of its encoders.

If $\mathscr{C}$ is a 2D convolutional code of rate $1/n$ and degree δ, then

$$\operatorname{dist}(\mathscr{C}) \leq \frac{(\delta + 1)(\delta + 2)}{2}n. \tag{1}$$

Such bound is called the 2D generalized Singleton bound and if the distance of $\mathscr{C}$ equals such bound, $\mathscr{C}$ is said to be MDS (see [3]).

3 Superregular Matrices

In [1] a new type of superregular matrices was introduced. The superregular matrices we are going to construct in this work have similar entries and, therefore, some properties are the same, even if the structure of these new matrices is different. Before we develop our new construction, we will recall some definitions pertinent to this type of superregular matrices.

Let $A = [\mu_{i\ell}]$ be a square matrix of order m over $\mathbb{F}$ and S_m the symmetric group of order m. The determinant of A is given by

$$|A| = \sum_{\sigma \in S_m} (-1)^{\operatorname{sgn}(\sigma)} \mu_{1\sigma(1)} \cdots \mu_{m\sigma(m)}.$$

Whenever we use the word *term*, we will be considering one product of the form $\mu_{1\sigma(1)} \cdots \mu_{m\sigma(m)}$, with $\sigma \in S_m$, and the word *component* will be reserved to refer to each of the $\mu_{i\sigma(i)}$, with $1 \leq i \leq m$ in a term. Denote $\mu_{1\sigma(1)} \cdots \mu_{m\sigma(m)}$ by μ_σ.

A *trivial term* of the determinant is a term μ_σ, with at least one component $\mu_{i\sigma(i)}$ equal to zero. If A is a square submatrix of a matrix B with entries in $\mathbb{F}$, and all the terms of the determinant of A are trivial, we say that $|A|$ is a **trivial minor** of B (if $B = A$ we simply say that $|A|$ is a trivial minor). We say that B is **superregular** if all its nontrivial minors are different from zero.

The next theorem states that square matrices over $\mathbb{F}$ of a certain form are superregular.

Theorem 3 *Let α be a primitive element of a finite field $\mathbb{F} = \mathbb{F}_{p^N}$ and $A = [\mu_{i\ell}]$ be a square matrix over $\mathbb{F}$ of order m, with the following properties*

1. *If $\mu_{i\ell} \neq 0$ then $\mu_{i\ell} = \alpha^{\beta_{i\ell}}$ for a positive integer $\beta_{i\ell}$;*
2. *If $\hat{\sigma} \in S_m$ is the permutation defined by $\hat{\sigma}(i) = m - i + 1$, then $\mu_{\hat{\sigma}}$ is a nontrivial term of $|A|$;*
3. *If $\ell \geq m - i + 1$, $\ell < \ell'$, $\mu_{i\ell} \neq 0$ and $\mu_{i\ell'} \neq 0$ then $2\beta_{i\ell} \leq \beta_{i\ell'}$;*
4. *If $\ell \geq m - i + 1$, $i < i'$, $\mu_{i\ell} \neq 0$ and $\mu_{i'\ell} \neq 0$ then $2\beta_{i\ell} \leq \beta_{i'\ell}$.*

Suppose $|A|$ is a nontrivial minor and N is greater than any exponent of α appearing as a nontrivial term of $|A|$. Then $|A| \neq 0$.

Proof Let $\sigma \in S_m$ such that μ_σ is a nontrivial term of $|A|$. By property 1, we have $\mu_\sigma = \alpha^{\beta_\sigma}$, for a positive integer β_σ.

Let $T_m = \{\sigma \in S_m \mid \sigma \neq \hat{\sigma}$ and μ_σ is a nontrivial term of $|A|\}$. If $T_m = \emptyset$ then $|A| = \mu_{\hat{\sigma}} = \alpha^{\beta_{\hat{\sigma}}} \neq 0$.

If $T_m \neq \emptyset$, let $\sigma \in T_m$. We are going to prove that if $\sigma \neq \hat{\sigma}$ then $\beta_{\hat{\sigma}} < \beta_{\sigma}$. Since μ_σ in a nontrivial term of $|A|$, for any $1 \leq i \leq m$, there exists $\ell \geq i$ such that $\sigma(\ell) \geq m - i + 1$. Now, for any $1 \leq \ell \leq m$ define

$$S_\ell = \{i \mid i \leq \ell \text{ and } \sigma(\ell) \geq m - i + 1\}.$$

Notice that $\bigcup\limits_{1 \leq j \leq m} S_\ell = \{1, 2, \ldots, m\}$ and, since $\sigma \neq \hat{\sigma}$, exists at least one ℓ_0, such that $1 \leq \ell_0 \leq m$ and $S_{\ell_0} = \emptyset$. By properties 3 and 4, we have that $\sum_{i \in S_\ell} \beta_{i\ m-i+1} \leq \beta_{\ell\ \sigma(\ell)}$. Therefore $\beta_{\hat{\sigma}} = \sum_{i=1}^{m} \beta_{i\ m-i+1} \leq \sum_{\substack{\ell=1 \\ S_\ell \neq 0}}^{m} \beta_{\ell\ \sigma(\ell)} < \sum_{\ell=1}^{m} \beta_{\ell\ \sigma(\ell)}$. So $|A| = \alpha^{\beta_{\hat{\sigma}}} + \sum_{h=\beta_{\hat{\sigma}}+1}^{N-1} \epsilon_h \alpha^h$, where $\epsilon_h \in \mathbb{F}_p$. Hence $|A| \neq 0$. □

Let us now construct specific types of superregular matrices, which will be useful in the next section. Let p be a prime number, N a positive integer and α be a primitive element of a finite field $\mathbb{F} = \mathbb{F}_{p^N}$. For $0 \leq a \leq \delta$ and $0 \leq b \leq \delta - a$ define the $n \times 1$ matrix $G(a, b)$ as

$$G(a,b) = \begin{bmatrix} \alpha^{2^{(\delta-a-b)n(\delta+1)+a}} & \alpha^{2^{((\delta-a-b)n+1)(\delta+1)+a}} & \ldots & \alpha^{2^{((\delta-a-b+1)n-1)(\delta+1)+a}} \end{bmatrix}^T \tag{2}$$

For example, if $\delta = 2$

$$G(0,2) = \begin{bmatrix} \alpha^{2^0} \\ \alpha^{2^3} \\ \vdots \\ \alpha^{2^{3(n-1)}} \end{bmatrix} \qquad G(0,0) = \begin{bmatrix} \alpha^{2^{6n}} \\ \alpha^{2^{3(2n+1)}} \\ \vdots \\ \alpha^{2^{3(3n-1)}} \end{bmatrix} \qquad G(1,1) = \begin{bmatrix} \alpha^{2^1} \\ \alpha^{2^4} \\ \vdots \\ \alpha^{2^{3(n-1)+1}} \end{bmatrix}$$

The following technical lemmas will be useful in the next section.

Lemma 4 *Let $\delta \geq 0$, $0 \leq j \leq \delta$ and $n \geq \delta - j + 1$. Then for $N \geq 2^{9n-2}$, the matrices*

$$\mathcal{G}_j = [G(j,0)\ \ G(j,1) \cdots G(j,\delta - j)] \in \mathbb{F}_{p^N}^{n \times (\delta - j + 1)}$$

have all its minors of any dimension, different from zero.

Note that all elements of $\mathcal{G}_j$ are different from zero, which means that all its minors are nontrivial. Moreover, up to column permutations, the minors of $\mathcal{G}_j$ satisfy Theorem 3.

Lemma 5 *Let $N \geq 2^{9n-1}$ and α a primitive element of a finite field $\mathbb{F} = \mathbb{F}_{p^N}$. Let $n \geq 3$ and $G(a, b) \in \mathbb{F}^n$, with $0 \leq a \leq 2$ and $0 \leq b \leq 2 - a$, be defined as in (2). Then the following matrices are superregular:*

$$A_1 = \begin{bmatrix} G(0,2) & G(0,1) & G(0,0) \\ G(0,1) & G(0,0) & 0 \\ G(0,0) & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & G(0,2) \\ 0 & G(0,2) & G(0,1) \\ G(0,2) & G(0,1) & G(0,0) \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 & 0 & G(0,2) \\ 0 & 0 & G(0,2) & G(0,1) \\ G(0,2) & G(0,1) & G(0,0) & 0 \\ G(0,1) & G(0,0) & 0 & 0 \\ G(0,0) & 0 & 0 & 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 0 & G(0,2) \\ 0 & G(0,2) & G(0,1) \\ 0 & G(0,1) & G(0,0) \\ G(0,2) & G(0,0) & 0 \\ G(0,1) & 0 & 0 \\ G(0,0) & 0 & 0 \end{bmatrix},$$

$$\text{and } A_5 = \begin{bmatrix} 0 & 0 & G(0,2) \\ 0 & G(0,2) & G(0,1) \\ G(0,2) & G(0,1) & G(0,0) \\ G(0,1) & G(0,0) & 0 \\ G(0,0) & 0 & 0 \end{bmatrix}.$$

4 Constructions of MDS 2D Convolutional Codes of Rate 1/n and Degree $\delta \leq 2$ for $n \geq \delta + 1$

In this section we will consider 2D convolutional codes of rate $1/n$ and degree δ and we will give constructions of MDS codes for $\delta \leq 2$. Let $\hat{G}(z_1, z_2) = \sum_{0 \leq i+j \leq \delta} G(i, j) z_1^i z_2^j$ be an encoder of $\mathscr{C}$, with $G(i, j) \in \mathbb{F}_{p^N}$ defined as in (2). We can write

$$\hat{G}(z_1, z_2) = \sum_{j=0}^{\delta} G_j(z_2) z_1^j, \tag{3}$$

where $G_j(z_2) = \sum_{i=0}^{\delta-j} G(j, i) z_2^i \in \mathbb{F}[z_2]^n$, $j = 0, 1, \ldots, \delta$. Analogously, given $\hat{\mathbf{u}}(z_1, z_2) \in \mathbb{F}[z_1, z_2]$ and $\hat{\mathbf{v}}(z_1, z_2) = \hat{G}(z_1, z_2)\hat{\mathbf{u}}(z_1, z_2)$, we can write them in the same way, i.e.,

$$\hat{\mathbf{u}}(z_1, z_2) = \sum_{j=0}^{\ell} \hat{\mathbf{u}}_j(z_2) z_1^j \quad \text{and} \quad \hat{\mathbf{v}}(z_1, z_2) = \sum_{j=0}^{\delta+\ell} \hat{\mathbf{v}}_j(z_2) z_1^j, \tag{4}$$

with $\ell \in \mathbb{N}$, where $\hat{\mathbf{u}}_j(z_2) = \sum_{i \geq 0} u(j,i) z_2^i \in \mathbb{F}[z_2]$, for any $j = 0, \dots, \ell$ and

$$\hat{\mathbf{v}}_s(z_2) = \sum_{i=0}^{\delta} G_i(z_2) \hat{\mathbf{u}}_{s-i}(z_2), \text{ for any } s = 0, \dots, \delta + \ell$$

(we consider $\hat{\mathbf{u}}_a(z_2) = 0$ if $a < 0$ or if $a > \ell$). Note that $\hat{\mathbf{v}}_s(z_2)$ are codewords of 1D convolutional codes.

We conjecture that for $n \geq \delta + 1$ and N sufficiently large, the $2D$ convolutional code $\mathscr{C} = \mathrm{Im}_{\mathbb{F}[z_1,z_2]} \hat{G}(z_1,z_2)$ is MDS. Next theorem considers such codes for $\delta \leq 2$.

Theorem 6 *Let $N \geq 2^{9n-1}$, $\delta \leq 2$, $n \geq \delta + 1$ and $\hat{G}(z_1,z_2)$ as defined in (3). Then $\mathscr{C} = \mathrm{Im}_{\mathbb{F}[z_1,z_2]} \hat{G}(z_1,z_2)$ is a 2D MDS convolutional code of rate $1/n$ and degree δ.*

Proof It is obvious that $\mathscr{C}$ has rate $1/n$ and degree δ. Let us consider first $\delta = 2$. To prove that $\mathscr{C}$ is MDS we have to show that all nonzero codewords of $\mathscr{C}$, $\hat{\mathbf{v}}(z_1,z_2)$, have weight greater or equal than $6n$. Let $\hat{\mathbf{v}}(z_1,z_2)$ be a nonzero codeword of $\mathscr{C}$ and $\hat{\mathbf{u}}(z_1,z_2) \in \mathbb{F}_{p^N}[z_1,z_2]$ such that $\hat{\mathbf{v}}(z_1,z_2) = \hat{G}(z_1,z_2)\hat{\mathbf{u}}(z_1,z_2)$ and let us represent both vectors as in (4). It is obvious that $\hat{\mathbf{u}}(z_1,z_2) \neq 0$ and in order to calculate the weight of $\hat{\mathbf{v}}(z_1,z_2)$ we can assume without loss of generality that $\hat{\mathbf{u}}_0(z_2) \neq 0$. Thus $\hat{\mathbf{v}}_0(z_2) = G_0(z_2)\hat{\mathbf{u}}_0(z_2)$, $\hat{\mathbf{v}}_1(z_2) = [G_0(z_2) \quad G_1(z_2)] \begin{bmatrix} \hat{\mathbf{u}}_1(z_2) \\ \hat{\mathbf{u}}_0(z_2) \end{bmatrix}$ and $\hat{\mathbf{v}}_2(z_2) = [G_0(z_2) \quad G_1(z_2) \quad G_2(z_2)] \begin{bmatrix} \hat{\mathbf{u}}_2(z_2) \\ \hat{\mathbf{u}}_1(z_2) \\ \hat{\mathbf{u}}_0(z_2) \end{bmatrix}$, are all nonzero vectors, i.e., the weight of any of these vectors is at least one. By definition, $\hat{\mathbf{u}}(z_1,z_2) = \hat{\mathbf{u}}_0(z_2) + \hat{\mathbf{u}}_1(z_2)z_1 + \cdots + \hat{\mathbf{u}}_\ell(z_2)z_1^\ell$, with $\hat{\mathbf{u}}_\ell(z_2) \neq 0$ for some $\ell \in \mathbb{N}$. Then, since $\hat{\mathbf{v}}_{2+\ell}(z_2) = G(2,0)\hat{\mathbf{u}}_\ell(z_2)$ it follows that $\mathrm{wt}(\hat{\mathbf{v}}_{2+\ell}(z_2)) = n \ \mathrm{wt}(\hat{\mathbf{u}}_\ell(z_2)) \geq n$. Since $\hat{\mathbf{v}}_0(z_2) = G_0(z_2)\hat{\mathbf{u}}_0(z_2)$ then, by Lemma 4, $\mathrm{wt}(\hat{\mathbf{v}}_0(z_2)) \geq 3n$. Now, if $\hat{\mathbf{u}}_1(z_2) = 0$ we have $\hat{\mathbf{v}}_1(z_2) = G_1(z_2)\hat{\mathbf{u}}_0(z_2)$ and again by Lemma 4, $\mathrm{wt}(\hat{\mathbf{v}}_1(z_2)) \geq 2n$. Hence

$$\mathrm{wt}(\hat{\mathbf{v}}(z_1,z_2)) \geq \mathrm{wt}(\hat{\mathbf{v}}_0(z_2) + \hat{\mathbf{v}}_1(z_2)z_1 + \hat{\mathbf{v}}_{2+\ell}(z_2)z_1^{2+\ell}) \geq 6n.$$

Next, we consider $\hat{\mathbf{u}}_1(z_2) \neq 0$. Suppose $\mathrm{wt}(\hat{\mathbf{u}}_0(z_2)) \geq 4$ and let i_1 be the minimum of the support of $\hat{u}_0(z_2)$ and i_2 be the maximum of the support of $\hat{u}_0(z_2)$, i.e., min Supp $(\hat{\mathbf{u}}_0(z_2)) = i_1$ and max Supp $(\hat{\mathbf{u}}_0(z_2)) = i_2$. Since $i_2 \geq i_1 + 3$, we have that

$$\begin{bmatrix} v(0,i_1+2) \\ v(0,i_1+1) \\ v(0,i_1) \end{bmatrix} = \begin{bmatrix} G(0,2) & G(0,1) & G(0,0) \\ G(0,1) & G(0,0) & 0 \\ G(0,0) & 0 & 0 \end{bmatrix} \begin{bmatrix} u(0,i_1) \\ u(0,i_1+1) \\ u(0,i_1+2) \end{bmatrix}$$

and

$$\begin{bmatrix} v(0,i_2+2) \\ v(0,i_2+1) \\ v(0,i_2) \end{bmatrix} = \begin{bmatrix} 0 & 0 & G(0,2) \\ 0 & G(0,2) & G(0,1) \\ G(0,2) & G(0,1) & G(0,0) \end{bmatrix} \begin{bmatrix} u(0,i_2-2) \\ u(0,i_2-1) \\ u(0,i_2) \end{bmatrix}.$$

Since the matrices A_1 and A_2 in Lemma 5 are superregular, we obtain, for $s \in \{i_1, i_2\}$, $\mathrm{wt}(\mathbf{v}(0,s)z_2^s + \mathbf{v}(0,s+1)z_2^{s+1} + \mathbf{v}(0,s+2)z_2^{s+2}) \geq 3n-2$. Then $\mathrm{wt}(\hat{\mathbf{v}}_0(z_2)) \geq 6n-4$. Therefore, $\mathrm{wt}(\hat{\mathbf{v}}(z_1,z_2)) \geq \mathrm{wt}(\hat{\mathbf{v}}_0(z_2) + \hat{\mathbf{v}}_1(z_2)z_1 + \hat{\mathbf{v}}_{2+\ell}(z_2)z_1^{2+\ell}) \geq 6n - 4 + 1 + n \geq 6n$, since $n \geq 3$.

Assume now that $\mathrm{wt}(\hat{\mathbf{u}}_0(z_2)) = 3$. If $\mathrm{Supp}(\hat{\mathbf{u}}_0(z_2)) = \{i, i+1, i+2\}$, for some $i \in \mathbb{N}$, then $\hat{\mathbf{v}}_0(z_2) = v(0,i)z_2^i + v(0,i+1)z_2^{i+1} + v(0,i+2)z_2^{i+2} + v(0,i+3)z_2^{i+3} + v(0,i+4)z_2^{i+4}$, where

$$\begin{bmatrix} v(0,i+4) \\ v(0,i+3) \\ v(0,i+2) \\ v(0,i+1) \\ v(0,i) \end{bmatrix} = \begin{bmatrix} 0 & 0 & G(0,2) \\ 0 & G(0,2) & G(0,1) \\ G(0,2) & G(0,1) & G(0,0) \\ G(0,1) & G(0,0) & 0 \\ G(0,0) & 0 & 0 \end{bmatrix} \begin{bmatrix} u(0,i) \\ u(0,i+1) \\ u(0,i+2) \end{bmatrix},$$

and, since A_5 is superregular, by Lemma 5, $\mathrm{wt}(\widehat{\mathbf{v}_0}(z_2)) \geq 5n-2$. Let $j = \min\ \mathrm{Supp}(\hat{\mathbf{u}}_1(z_2))$. If $j < i$, then $v(1,j) = G(0,0)u(1,j)$, if $j > i$ then $v(1,i) = G(1,0)u(0,i)$ and if $j = i$, then $v(1,i) = G(1,0)u(0,i) + G(0,0)u(1,i)$, so, in any case, we get $\mathrm{wt}(\hat{\mathbf{v}}(z_1,z_2)) = \mathrm{wt}(\hat{\mathbf{v}}_0(z_2) + \hat{\mathbf{v}}_1(z_2)z_1 + \hat{\mathbf{v}}_{2+\ell}(z_2)z_1^{2+\ell}) \geq 5n - 2 + n - 1 + n = 7n - 3 \geq 6n$, since $n \geq 3$.

Suppose now that $\mathrm{Supp}(\hat{\mathbf{u}}_0(z_2)) = \{i_1, i_2, i_3\}$ with $i_1 < i_2 < i_3$ and $i_2 - i_1 > 1$ or $i_3 - i_2 > 1$. In this case, we will always obtain $\mathrm{wt}(\widehat{\mathbf{v}_0}(z_2)) \geq 6n-2$. For example, If $i_2 - i_1 = 2$ and $i_3 - i_2 = 1$, we have that

$$\begin{bmatrix} v(0,i_1+5) \\ v(0,i_1+4) \\ v(0,i_1+3) \\ v(0,i_1+2) \\ v(0,i_1+1) \\ v(0,i_1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & G(0,2) \\ 0 & G(0,2) & G(0,1) \\ 0 & G(0,1) & G(0,0) \\ G(0,2) & G(0,0) & 0 \\ G(0,1) & 0 & 0 \\ G(0,0) & 0 & 0 \end{bmatrix} \begin{bmatrix} u(0,i_1) \\ u(0,i_2) \\ u(0,i_3) \end{bmatrix},$$

so $\mathrm{wt}(\hat{\mathbf{v}}_0(z_2)) \geq \mathrm{wt}\left(\mathbf{v}(0,i_1)z_2^{i_1} + \mathbf{v}(0,i_1+1)z_2^{i_1+1} + \cdots + \mathbf{v}(0,i_1+5)z_2^{i_1+5}\right) \geq 6n-2$. Thus $\mathrm{wt}(\hat{\mathbf{v}}(z_1,z_2)) = \mathrm{wt}(\hat{\mathbf{v}}_0(z_2) + \hat{\mathbf{v}}_{2+\ell}(z_2)z_1^{2+\ell}) \geq 6n - 2 + n \geq 6n$, since $n \geq 3$.

Suppose now $\mathrm{wt}(\hat{\mathbf{u}}_0(z_2)) = 2$ and $\mathrm{Supp}(\hat{\mathbf{u}}_0(z_2)) = \{i, j\}$ with $i < j$, with similar arguments as before we get $\mathrm{wt}(\hat{\mathbf{v}}_0(z_2)) \geq 4n-1$. The worst case is when $j = i+1$ where

$$\begin{bmatrix} v(0,i_1+3) \\ v(0,i_1+2) \\ v(0,i_1+1) \\ v(0,i_1) \end{bmatrix} = \begin{bmatrix} 0 & G(0,2) \\ G(0,2) & G(0,1) \\ G(0,1) & G(0,0) \\ G(0,0) & 0 \end{bmatrix} \begin{bmatrix} u(0,i) \\ u(0,i+1) \end{bmatrix},$$

which implies $\mathrm{wt}(\hat{\mathbf{v}}_0(z_2)) \geq 4n-1$. We also have $\mathrm{wt}(\hat{\mathbf{v}}_1(z_2)) \geq 2n-2$ always. Thus $\mathrm{wt}(\hat{\mathbf{v}}(z_1, z_2)) = \mathrm{wt}(\hat{\mathbf{v}}_0(z_2) + \hat{\mathbf{v}}_1(z_2)z_1 + \hat{\mathbf{v}}_{2+\ell}(z_2)z_1^{2+\ell}) \geq 4n - 1 + 2n - 2 + n = 7n - 3 \geq 6n$, since $n \geq 3$.

Finally, assume that $\mathrm{wt}(\hat{\mathbf{u}}_0(z_2)) = 1$. Here, we obtain $\mathrm{wt}(\hat{\mathbf{v}}_0(z_2)) \geq 3n$ and $\mathrm{wt}(\hat{\mathbf{v}}_1(z_2)) \geq 3n - 1$. Hence, $\mathrm{wt}(\hat{\mathbf{v}}(z_1, z_2)) \geq 6n$, for $n \geq 3$.

For $\delta \leq 1$ the theorem follows immediately. □

Acknowledgements This work was partially supported by Portuguese funds through the CIDMA – Center for Research and Development in Mathematics and Applications, and the Portuguese Foundation for Science and Technology ("FCT-Fundação para a Ciência e a Tecnologia"), within project UID/MAT/04106/2013.

References

1. Almeida, P., Napp, D., Pinto, R.: A new class of superregular matrices and MDP convolutional codes. Linear Algebra Appl. **439**, 2145–2157 (2013)
2. Charoenlarpnopparut, C.: Applications of Gröbner bases to the structural description and realization of multidimensional convolutional code. Sci. Asia **35**, 95–105 (2009)
3. Climent, J.J., Napp, D., Perea, C., Pinto, R.: A construction of MDS 2D convolutional codes of rate $1/n$ based on superregular matrices. Linear Algebra Appl. **437**, 766–780 (2012)
4. Fornasini, E., Valcher, M.E.: Algebraic aspects of two-dimensional convolutional codes. IEEE Trans. Inf. Theory **40**(4), 1068–1082 (1994)
5. Napp, D., Perea, C., Pinto, R.: Input-state-output representations and constructions of finite support 2D convolutional codes. Adv. Math. Commun. **4**(4), 533–545 (2010)
6. Rosenthal, J., Smarandache, R.: Maximum distance separable convolutional codes. Appl. Algebra Eng. Commun. Comput. **10**(1), 15–32 (1999)
7. Valcher, M.E., Fornasini, E.: On 2D finite support convolutional codes: an algebraic approach. Multidimens. Syst. Signal Process **5**, 231–243 (1994)
8. Weiner, P.A.: Multidimensional convolutional codes. Ph.D. thesis, Department of Mathematics, University of Notre Dame, Notre Dame (1998)

A Coding-Based Approach to Robust Shortest-Path Routing

Ángela I. Barbero and Øyvind Ytrehus

Abstract Robust and distributed creation of routing tables is essential for the functioning of a modern communication network. One of the two main types of routing algorithms in use in today's Internet is made up of variations of the so-called *distance-vector* or Bellman-Ford algorithm (Bellman, Quart Appl Math 16:87–90, 1958; Ford (1956) Network flow theory paper P-923. RAND Corporation, Santa Monica; Moore (1959) The shortest path through a maze. In: Proceedings of an international symposium on the theory of switching 1957, Cambridge. Part II. Harvard University Press, pp 285–292). These algorithms suffer from two main deficiencies: (i) The amount of data exchanged for the algorithms to function is considered excessive for some applications, and (ii) the algorithms respond slowly to "bad news" in the network. This is known as the count-to-infinity ($c2\infty$) problem. In order to address (ii), protocol designers (RFC1058 – routing information protocol (1988) Internet Engineering Task Force) have introduced heuristics such as the "split horizon" and the "route poisoning" techniques. It can be shown by simple examples that these heuristics do not solve the $c2\infty$ problem completely. In this paper we describe a simple routing algorithm, the Tree Routing algorithm, that exchanges no more data than existing algorithms, and that at the same time provides routing agents with no less (and often more) insight into the topology of the network. The Tree Routing algorithm is inspired by techniques used in information theory and coding theory. We explain why the Tree Routing algorithm will never respond slower, and will often respond faster, than existing algorithms.

Keywords Communication networks • Routing • Distance vector algorithm • Information theory • Coding • Belief propagation

Á.I. Barbero (✉)
Department of Applied Mathematics, Universidad de Valladolid, 47011 Valladolid, Spain
e-mail: angbar@wmatem.eis.uva.es

Ø. Ytrehus (✉)
Department of Informatics, University of Bergen, N-5020 Bergen, Norway
e-mail: oyvind@ii.uib.no

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_3

1 Introduction

Distance vector routing (DVR) is a classic technique [2–4] for obtaining minimum distance routing tables in a communication network in a distributed way. The algorithm, described in Sect. 2.1, relies on message exchange between neighbour routers. DVR is used in the Internet in the form of the Routing Information Protocol (RIP) [5] and its extensions [6, 7].

Shortest-path routing is simple (e.g. using Dijkstra's algorithm) if nodes have sufficient correct knowledge of the network topology. In this work, we elaborate on a more general discussion initiated in [1] and proceed to present a new algorithm that appears to represent a good tradeoff between data exchange and performance. We observe that the core of the problem of the DVR family is that the structure of its messages does not provide nodes with enough information about the network topology. The new algorithm provides nodes with information about the relevant parts of the network topology, at no increase in communication cost.

The connection to coding can be justified as follows: The core function carried out by any routing algorithm is to collect information about the network structure, and to use this collected information to calculate routing tables that are optimum according to some pre-specified criteria. The collection of information implies an information transfer, and this in turn requires that the information is encoded in a suitable manner. Thus, coding is a subtask in the routing process, although it is often not considered to be.

The structure of this abstract is as follows: Sect. 2 provides notation, the network model, and describes previous work. The new algorithm is described in Sect. 3. Application of the algorithm on a toy network is given in Sect. 4, while properties of the algorithms are discussed in Sect. 5.

2 Background, Notation, and Network Model

A communication network is described in this paper by an undirected graph $G = G(\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of nodes and $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ is the set of undirected edges. For convenience, we will use the terms *edge* and *link* as synonyms, and by *node* we will mean what in network terminology is usually called a *router*. For a given node $v \in \mathscr{V}$, we denote by $\mathscr{N}(v) = \{u : \{v, u\} \in \mathscr{E}\}$ the set of neighbours of v.

Each edge is assumed to have *unit capacity*. This corresponds to the case of distance being measured in terms of the number of hops in the graph. Although this may seem a crude measure of distance, this is actually what is used in many network routing protocols.

We emphasize that the results in this paper may be adapted to more general models that also represent, for example, broadcast links or more diverse link capacities, and that take into account other distance measures.

Table 1 Example of a routing table

Destination node id	$\mathbf{id}_1$	$\mathbf{id}_2$	$\cdots$	$\mathbf{id}_{n-1}$
Distance	d_1	d_2	$\cdots$	d_{n-1}
Outgoing link	L_1	L_2	$\cdots$	L_{n-1}

2.1 Distance Vector Routing

Distance-Vector routing (DVR) is an example of a distributed version of the Bellman-Ford algorithm. The purpose of the algorithm is to determine the shortest path from each node to every other node. This path is represented in each node by a *routing table*. For the case of a network with n nodes, this routing table at each node has $n - 1$ columns and three rows, in the form of Table 1.

When a node has a data packet to forward to a destination node with identifier $\mathbf{id}_k$, the sending node sends it to the corresponding outgoing link L_k.

In this distributed version of the BF algorithm, the nodes exchange information (only) with its neighbours. This information is conveyed in messages that contain the sending node's current routing table (Table 1) *except for the last row*, which contains information which is relevant only to the node itself.[1] In the standardized protocols used in the Internet, like RIP in different versions, this routing table exchange takes place either at regular intervals, or message transmission is triggered by the arrival of new information in a node.

In addition, each node will monitor the distance $d(\mathbf{id})$ to each of its neighbours **id** on the corresponding link, (which, as noted above, in our case by definition is 1 to any neighbour node). Each time a routing message arrives, the current node will update each column (i.e. each destination node $\mathbf{id}_k$) of its routing table according to the following rules:

1. Let $d_k(old)$ be the distance to node $\mathbf{id}_k$ according to the old table, let $d_k(new)$ be the distance according to the incoming table, and let $d(neighbour) = 1$ be the length of the link.
2. If the incoming routing table message arrives on a link L that is *not* used for node $\mathbf{id}_k$, then update the routing table if $d(neighbour) + d_k(new) < d_k(old)$ with the new distance $d(neighbour) + d_k(new)$.
3. If the incoming routing table message arrives on a link L that *is* currently used for node $\mathbf{id}_k$, then update the routing table with the new distance $d(neighbour) + d_k(new)$.

In summary, in Distance Vector routing, each node executes Algorithm 1.

Distance Vector routing is simple and is known to converge fast to a set of shortest paths for each node pair when good news arrive, i.e. when new links become available.

[1] Nevertheless, in some implementations even the last row is contained in the messages.

Algorithm 1 The DVR algorithm (DVR)

```
Determine who the neighbours are
Initialize routing table: For each neighbour node, set the distance to 1 and the outgoing link to the obvious value
while network is active do
    Determine who are the neighbours now
    (A) Send a message to each neighbour (if there are changes), consisting of the first two rows of Table 1.
    (B) Based on the current input from the neighbours, calculate new minimum distances to each destination node, and update routing tables accordingly
end while
```

On the other hand, the algorithm responds slowly to bad news, i.e. when links disappear, in some cases, notably for cyclic networks. This is known as the count-to-infinity ($c2\infty$) problem. In order to alleviate this, protocol designers have applied heuristic techniques like "split horizon" and "route poisoning" [5]. Using the "split horizon" heuristic, a node a will not report to a neighbour b (in step (A) of the loop) a path to node c if a's best path to c passes through b. In "split horizon" with "route poisoning", nodes will deliberately advertise to its neighbours infinite distances for destinations for which the best paths have failed. This will trigger the neighbours to suspend the split horizon strategy.

These heuristics still do not completely solve the ($c2\infty$) problem for all network topologies. In the next section we propose a related algorithm which improves the response to bad news in many cases.

3 A New Algorithm

The new algorithm, the *Tree Routing algorithm* (TR), has two components, inspired by information theoretic and coding theoretic arguments.

1. Instead of transferring distance vectors, in the *Tree Routing algorithm* a message $M(u, v)$ from a node u to a neighbour $v \in \mathscr{N}(u)$ is comprised of the first two rows of a table in the form of Table 2. In the second row, Parent, represents the destination's predecessor in a *particular routing tree* with a root in u. As argued in [1], in terms of size of the exchanged messages, we may just as well transfer the Parent instead of the Distance. This allows the recipient v to reconstruct the particular routing tree that the sending neighbour u has built. Also observe that, in consequence, the third row of Table 2, Distance, is implied by the first two rows [1], and so is the fourth row.
2. Drawing from the similarity to belief propagation or message passing decoders for LDPC codes, a message a $M(u, v)$ from a node u to a neighbour $v \in \mathscr{N}(u)$ is computed based on the input to u from the *other* neighbours, $\{w : w \in \mathscr{N}(u), w \neq v\}$. Thus, the *particular routing tree* mentioned above is the one that u creates without using node v.

Table 2 An extended routing table

Destination node id	$\mathbf{id}_1$	$\mathbf{id}_2$	$\cdots$	$\mathbf{id}_{n-1}$
Parent node id	p_1	p_2	$\cdots$	p_{n-1}
Distance	d_1	d_2	$\cdots$	d_{n-1}
Outgoing link	L_1	L_2	$\cdots$	L_{n-1}

The procedure is described in Algorithm 2.

Algorithm 2 The tree routing algorithm

```
Determine who the neighbours are
Initialize routing table: For each neighbour node, set the distance to 1 and the outgoing link to the obvious value
while network is active, each node $u$ will do
    Determine $\mathcal{N}(u) :=$ who the neighbours are now
    (A) Send a message $M(u, v)$ to each neighbour $v \in \mathcal{N}(u)$ (if there are changes with respect to most recent message transmitted from $u$ to $v$ ), consisting of the first two rows of Table 1. $M(u, v)$ is based on merging the most recent messages, $\{M(w, u) : w \in \mathcal{N}(u), w \neq v\}$.
    (B) Receiving nodes should disregard a message advertising any path that uses a link that according to more recent information does not exist (anymore).
    (C) Based on the current input from all the neighbours, calculate the new routing table.
end while
```

The mechanism in (B) is necessary to counter the $c2\infty$ problem. Due to space limitations we will not discuss in detail what we mean by *most recent information*, beyond the following: In an implementation where message transmission is synchronous, the age of information corresponds to the length of the path over which information has travelled. In case of asynchronous communication, time stamps may be necessary.

3.1 Variations

The description of the TR algorithm is by necessity general and rudimentary. We observe that different versions of the TR can be designed, suited to the particular emphasis on memory use, node computation requirements, and convergence speed, and that it can be generalized to multipath applications.

Thus, *in this paper*, by the TR algorithm we mean the generic family of routing algorithms which is distinguished from the DVR family by the nature of information exchanged: The TR algorithms exchange tree structures while the DV algorithms exchange distance vectors.

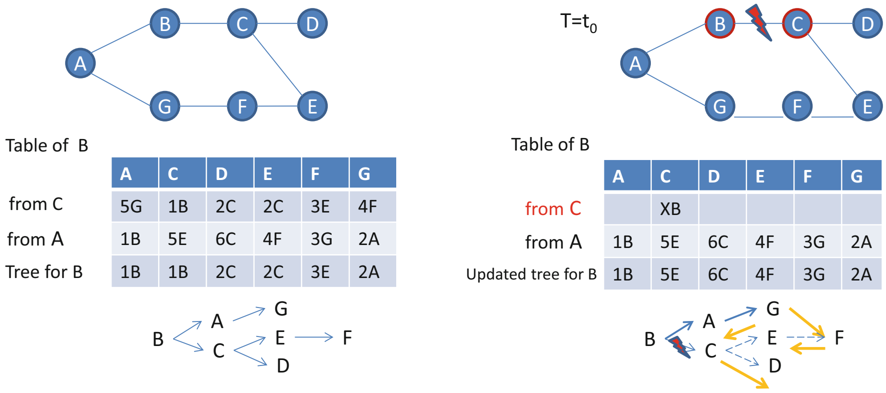


Table of B

	A	C	D	E	F	G
from C	5G	1B	2C	2C	3E	4F
from A	1B	5E	6C	4F	3G	2A
Tree for B	1B	1B	2C	2C	3E	2A

Table of B

	A	C	D	E	F	G
from C		XB				
from A	1B	5E	6C	4F	3G	2A
Updated tree for B	1B	5E	6C	4F	3G	2A

Fig. 1 Response to bad news. The *left part* of the figure shows the network at time $t_0 - 1$, the messages sent to B at that time, and the routing tree constructed for B. For convenience, we have included Distance information, although this is implied by the Parent information and therefore does not need to be sent. Thus C will report to B that A's parent is G; this implies that the distance to A is 5. The *right part* of the figure shows the network at time t_0, after the link $\{B, C\}$ has collapsed

4 Example

Figure 1 shows an example of how the Tree Routing algorithm responds to bad news. Figure 2 shows, by way of an example, a comparison of the network topology information conveyed in Tree Routing and in Distance Vector Routing.

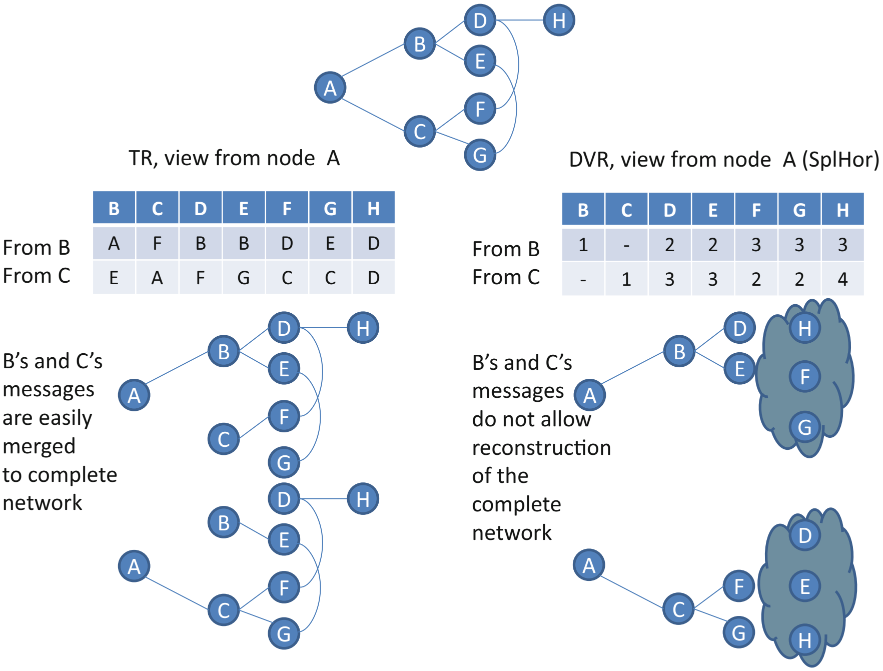
5 Properties of the Tree Routing Algorithm

We are not aware of a *formal* definition of the ($c2\infty$) problem. Therefore it is also not clear how to provide formal statements of how the algorithms perform with respect to this problem. We can however, prove the following statements (but do not include the proofs here, for lack of space).

Proposition 1 *The amount of data exchanged in the TR is not larger than in the obvious representation of the DVR, and not significantly larger than an optimum representation of the DVR [1].*

Proposition 2 *The TR conveys at least as much information (in the information theoretic sense) about the network topology among the neighbours, as the DVR and its variants do, and often more.*

The more information a deciding entity is in possession of, the better are the *optimum* decisions it can make. Thus, the last proposition suggests that, on average and with optimum use of the collected information, a TR algorithm will converge



TR, view from node A

	B	C	D	E	F	G	H
From B	A	F	B	B	D	E	D
From C	E	A	F	G	C	C	D

DVR, view from node A (SplHor)

	B	C	D	E	F	G	H
From B	1	-	2	2	3	3	3
From C	-	1	3	3	2	2	4

B's and C's messages are easily merged to complete network

B's and C's messages do not allow reconstruction of the complete network

Fig. 2 A comparison of the TR and the DVR. The actual network is shown at the *top* of the figure. On the *left* are shown the messages that B and C, respectively, send to A in the TR algorithm. For each node in the respective routing trees of B and C, the parent node is given. This allows A to reconstruct the two routing trees and, in this case, the entire network. The *right part* shows the corresponding messages sent in the DVR algorithm. We observe that for the DVR algorithm, A is unable to reconstruct a clear picture of the graph. Moreover, although it is still possible to deduce that A is on a cycle, the current versions of the DVR do not attempt to do this, and it is difficult to establish the exact structure of that cycle. This complicates the design of a simple rule to avoid the $c2\infty$ problem

faster than a DVR algorithm on dynamic networks where nodes may move or links may appear and disappear.

We will provide simulation results for the average convergence times and packet delay values for different classes of graphs, including Gilbert random graphs and random geometric graphs.

References

1. Barbero, Á.I., Ytrehus, Ø.: Information exchange for routing protocols. In: Proceedings of ITA 2014, San Diego (2014)
2. Bellman, R.: On a routing problem. Quart. Appl. Math. **16**, 87–90 (1958)
3. Ford, L.R.J.: Network Flow Theory Paper P-923. RAND Corporation, Santa Monica (1956)

4. Moore, E.M.: The shortest path through a maze. In: Proceedings of an International Symposium on the Theory of Switching 1957, Cambridge, pp. 285–292. Part II. Harvard University Press (1959)
5. RFC1058 – routing information protocol (1988) Internet Engineering Task Force
6. RFC1723 – RIP version 2 carrying additional information (1994) Internet Engineering Task Force
7. RFC2453 – RIP version 2 (1998) Internet Engineering Task Force

Constructions of Fast-Decodable Distributed Space-Time Codes

Amaro Barreal, Camilla Hollanti, and Nadya Markin

Abstract Fast-decodable distributed space-time codes are constructed by adapting the iterative code construction introduced in [6] to the N-relay multiple-input multiple-output channel, leading to the first fast-decodable distributed space-time codes for more than one antenna per user. Explicit constructions are provided alongside with a performance comparison to non-iterated (non-) fast-decodable codes.

Keywords Distributed space-time codes • Fast-decodability • Half-duplex relay channel • Cyclic division algebras

1 Introduction

The increasing interest in *cooperative diversity* techniques as well as rapid growth in the field of multi-antenna communications motivates the investigation of flexible coding techniques for the multiple-input multiple-output (MIMO) cooperative channel. The tools developed in [4] and [5] provide the necessary tools to construct fast-decodable space-time (ST) codes for the N-relay non-orthogonal amplify-and-forward (NAF) cooperative channel with a single antenna at both the source and the relays. Our work extends these methods to the N-relay MIMO NAF channel, that is the relays are allowed to employ multiple antennas for transmission and reception. In addition, many ST codes for the relay scenario exhibit a high rate and hence require a high number of receive antennas at the destination, whereas in this work a single antenna suffices.

A. Barreal (✉) • C. Hollanti (✉)
Department of Mathematics and Systems Analysis, Aalto University, P.O. Box 11100, FI-00076 Aalto, Espoo, Finland
e-mail: amaro.barreal@aalto.fi; camilla.hollanti@aalto.fi

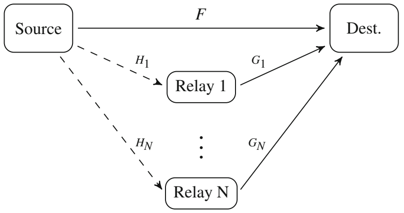
N. Markin (✉)
School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371, Singapore
e-mail: nadyaomarkin@gmail.com

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_4

2 The MIMO Amplify-and-Forward Channel

We consider the case of a single user communicating with a single destination over a wireless network. N intermediate relays participate in the transmission process. In addition, we assume the *half-duplex* constraint, that is the relays can only either receive or transmit a signal in a given time instance.

Denote by n_s, n_d and n_r the number of antennas at the source, destination, and relays, respectively. A superframe consisting of N consecutive cooperation frames of length T, each composed of two partitions of $T/2$ symbols, is defined, and all channels are assumed to be static during the transmission of the entire superframe.



F, H_i and G_i, $1 \leq i \leq N$ denote the Rayleigh distributed channels from the source to the destination, relays, and from the relays to the destination, respectively.

In a realistic scenario, $n_r \leq n_s$, the transmission process can be modeled as

$$\begin{aligned} Y_{i,1} &= \gamma_{i,1} F X_{i,1} + V_{i,1}\,, & i &= 1,\ldots,N \\ Y_{i,2} &= \gamma_{i,2} F X_{i,2} + V_{i,2} + \gamma_{R_i} G_i B_i (\gamma'_{R_i} H_i X_{i,1} + W_i)\,, & i &= 1,\ldots,N \end{aligned}$$

where $Y_{i,j}$ and $X_{i,j}$ are the received and sent matrices, $V_{i,j}$ and W_i represent additive white Gaussian noise, the matrices B_i are needed for normalization and $\gamma_{i,j}$, γ_{R_i}, γ'_{R_i} are signal-to-noise (SNR) related scalars.

From the destinations point of view, the above transmission model can be viewed as a virtual single-user MIMO channel as

$$Y_{n_d \times n} = H_{n_d \times n} X_{n \times n} + V_{n_d \times n},$$

where $n = N(n_s + n_r)$, X and Y are the (overall) transmitted and received codewords whose structure will take a particular form, and the channel matrix H is determined by the different relay paths. For more details, as well as for the remaining case $n_r > n_s$, we refer to [8].

2.1 Optimal Space-Time Codes for MIMO NAF Relay Channels

Let $\doteq$ denote exponential equality, i.e., we write

$$f(\mathrm{SNR}) \doteq \mathrm{SNR}^b \Leftrightarrow \lim_{\mathrm{SNR}\to\infty} \frac{\log f(\mathrm{SNR})}{\log \mathrm{SNR}} = b$$

and similar for $\dot{\leq}$. Consider an $n_d \times n$ MIMO channel and let $\mathscr{A}$ be a scalably dense alphabet [8, p.651, Definition 2], that is for $\mathscr{A}(\mathrm{SNR})$ its value at a given SNR and for $0 \leq r \leq \min\{n_d, n\}$ we require

$$|\mathscr{A}(\mathrm{SNR})| \doteq \mathrm{SNR}^{\frac{r}{n}}$$

$$|a|^2 \dot{\leq} \mathrm{SNR}^{\frac{r}{n}} \ \text{ for } a \in \mathscr{A}(\mathrm{SNR}),$$

for instance PAM or (rotated) QAM constellations. An $n \times n$ ST code $\mathscr{X}$ such that

1. The entries of any $X \in \mathscr{X}$ are linear combinations of elements in $\mathscr{A}$.
2. On average, R complex symbols from $\mathscr{A}$ are transmitted per channel use.
3. $\min\limits_{X_i \neq X_j \in \mathscr{X}} |\det(X_i - X_j)| \geq \kappa > 0$ for a constant κ independent of the SNR.

is called a *rate-R non-vanishing determinant* (NVD) code, and we say the code is *full-rate* if $R = n_d$. This is the largest rate that still allows for the use of a linear decoder, e.g., a sphere decoder, with n_d antennas at the destination.

Consider an N-relay MIMO NAF channel. It was shown in [8] that given a rate-$2n_s$ NVD block-diagonal code $\mathscr{X}$, thus where each $X \in \mathscr{X}$ takes the form $X = \mathrm{diag}\{\Xi_i\}_{i=1}^N$ with $\Xi_i \in \mathrm{Mat}(2n_s, \mathbb{C})$, the equivalent code $C = \left[C_1 \cdots C_N\right]$, $C_i = \left[\Xi_i\,[1 : n_s, 1 : 2n_s]\ \Xi_i\,[n_s + 1 : 2n_s, 1 : 2n_s]\right]$ achieves the optimal diversity-multiplexing tradeoff (DMT) for the channel, transmitting C_i in the ith cooperation frame.

It would thus be desirable to have block-diagonal ST codes which in addition achieve:

1. *Full rate n_d*: The number of independent complex symbols (e.g., QAM) per codeword equals $n_d(n_s + n_r)N$.
2. *Full rank*: $\min\limits_{X_i \neq X_j \in \mathscr{X}} \mathrm{rank}(X_i - X_j) = (n_s + n_r)N$.
3. *NVD*: $\min\limits_{X_i \neq X_j \in \mathscr{X}} |\det(X_i - X_j)|^2 \geq \kappa > 0$ for a constant κ.

The last condition can be abandoned at the low SNR regime without compromising the performance. For very low SNR, even relaxing on the full-rank condition does not have adverse effect, since the determinant criterion is asymptotic in nature.

2.2 On Fast-Decodability

Consider a ST lattice code $\mathscr{X} = \{\sum_{i=1}^k z_i \cdot B_i \mid z_i \in \mathbb{Z} \cap J\}$, where $\{B_i\}_{i=1}^k$, $k \leq 2n^2$, are lattice basis matrices of a rank-k lattice $\Lambda \subseteq \mathrm{Mat}(n, \mathbb{C})$, and $J \subset \mathbb{Z}$ is finite and referred to as the *signaling alphabet*. Maximum-likelihood decoding of ST codes amounts to finding the codeword in $\mathscr{X}$ that achieves

$$Z = \arg\min\{||Y - HX||_F^2\}_{X \in \mathscr{X}}.$$

Writing b_i for the vectorization of HB_i, that is stacking its columns followed by separating the real and imaginary parts, define $B = (b_1, \ldots, b_k)$ and $z = (z_1, \ldots, z_k)^T$. Each received codeword can thus be represented as $B \cdot z$. Performing QR-decomposition on B, where $QQ^\dagger = I$, R upper triangular, leads to finding

$$\arg\min\{||Y - HX||_F^2\}_{X \in \mathscr{X}} \rightsquigarrow \arg\min\{||Q^\dagger y - Rz||_E^2\}_{z \in J^k} = \arg\min\{||y' - Rz||_E^2\}_{z \in J^k},$$

where $y' = Q^\dagger y$. This search can be simplified by using a sphere decoder, the complexity of which is upper bounded by that of exhaustive search, i.e., by $|J|^k$. The structure of the matrix R can however reduce the complexity of decoding. A ST code whose decoding complexity is $|J|^{k'}$, $k' < k - 1$, due to the structure of R is called *fast-decodable* [2]. A more extensive review on fast-decodability can be found in [5].

3 Iterated Space-Time Codes

Recently, an iterative ST code construction has been proposed in [6]. By choosing the maps and elements involved in the construction carefully, the resulting code can inherit some good properties from the original code, such as fast-decodability or full-diversity. This makes the proposed method an interesting tool for constructing bigger codes from well-performing ones.

Consider a tower of field extensions $\mathbb{Q} \subseteq \mathrm{F} \subseteq \mathrm{K}$, with $\mathrm{F}/\mathbb{Q}$ finite Galois and K/F cyclic Galois of degree n with Galois group $\Gamma(\mathrm{K}/\mathrm{F}) = \langle\sigma\rangle$. Let $\gamma \in \mathrm{F}^\times$ be such that $\gamma^i \notin \mathrm{Nm}_{\mathrm{K}/\mathrm{F}}(\mathrm{K}^\times)$, $i = 1, \ldots, n$, and let $\mathscr{C} = (\mathrm{K}/\mathrm{F}, \sigma, \gamma) = \bigoplus_{i=0}^{n-1} e^i \mathrm{K}$, where $e^n = \gamma$ and $ke = e\sigma(k)$ for all $k \in \mathrm{K}$, be a cyclic division algebra of dimension n^2 over its center F. Given $c = \sum_{i=0}^{n-1} e^i c_i \in \mathscr{C}$, the representation over its maximal subfield is

$$\lambda : c \mapsto \begin{bmatrix} c_0 & \gamma\sigma(c_{n-1}) & \cdots & \gamma\sigma^{n-1}(c_1) \\ c_1 & \sigma(c_0) & \cdots & \gamma\sigma^{n-1}(c_2) \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & \sigma(c_{n-2}) & \cdots & \sigma^{n-1}(c_0) \end{bmatrix}$$

Let $\tau \in \mathrm{Aut}_{\mathbb{Q}}(\mathrm{K})$ such that $\tau\sigma = \sigma\tau$ and commutes also with complex conjugation, $\tau^2 = 1$ and $\tau(\gamma) = \gamma$. Fix $\theta \in \mathrm{F}^\times$ such that $\tau(\theta) = \theta$. Setting $X = \lambda(x)$, $Y = \lambda(y) \in \mathrm{Mat}(n, \mathscr{C})$, define a map

$$\alpha_\theta : (X, Y) \mapsto \begin{bmatrix} X & \theta\tau(Y) \\ Y & \tau(X) \end{bmatrix}.$$

The conditions imposed on τ and θ ensure that the image of α_θ is an F-algebra, and is division if and only if $\theta \neq c\tau(c)$ for all $c \in \mathscr{C}$. For more details on this construction method, the reader may consult [6].

4 Distributed Iterated Space-Time Codes

Consider an N-relay MIMO channel. Given a ST code $\mathscr{X} \subset \mathrm{Mat}(n, \mathbb{C})$, we define the following map $f : \mathscr{X} \to \mathrm{Mat}(nN, \mathbb{C})$

$$f_\eta^N : X \mapsto \mathrm{diag}\{\eta^i(X)\}_{i=0}^{N-1}$$

for a suitable function η such that $\eta^N = \mathrm{id}$. In the following, we will make use of this function to construct distributed ST codes from iterated and non-iterated codes. Often the map η is chosen to be a field automorphism, so that the determinant will correspond to a field norm and be non-vanishing with a suitable choice of fields. Here, we choose $\eta = \mathrm{id}$, as in our specific examples the blocks composing the codeword matrices will already have the NVD property, thus no special modifications will be necessary. Note that if the blocks $\eta^i(X)$ are fast-decodable, the resulting block-diagonal code will also be fast-decodable [5].

4.1 Explicit Constructions for $N = 2$ Relays and $n_r + n_s = 4$

In the following, let $N = 2$ be the number of relays, both equipped with n_r antennas and such that $n_r + n_s = 4$. We construct three different codes, each of them with different characteristics, arising from the following towers of extensions:

$$\begin{tikzcd}
\mathscr{C}_s = (K_s/F_s, \sigma_s, -1) \arrow[d, dash, "2"] & \mathscr{C}_g = (K_g/F_g, \sigma_g, i) \arrow[d, dash, "2"] & \mathscr{C}_m = (K_m/\mathbb{Q}, \sigma_m, -\frac{8}{9}) \arrow[d, dash, "2"] \\
K_s = F_s(i) \arrow[d, dash, "2"] & K_g = F_g(\sqrt{5}) \arrow[d, dash, "2"] & K_m = \mathbb{Q}(\zeta_5) \arrow[dd, dash, "4"] \\
F_s = \mathbb{Q}(\sqrt{-7}) \arrow[d, dash, "2"] & F_g = \mathbb{Q}(i) \arrow[d, dash, "2"] & \\
\mathbb{Q} & \mathbb{Q} & \mathbb{Q} \\
\sigma_s : i \mapsto -i & \sigma_g : \sqrt{5} \mapsto -\sqrt{5} & \sigma_m : \zeta_5 \mapsto \zeta_5^3
\end{tikzcd}$$

1. The *Silver code*, well known to be fast-decodable [2, 3], is constructed from the cyclic division algebra $\mathscr{C}_s$ and is a finite subset of

$$\left\{ \frac{1}{\sqrt{7}} \begin{bmatrix} x_1\sqrt{7}+(1+i)x_3+(-1+2i)x_4 & -x_2^*\sqrt{7}-(1-2i)x_3^*-(1+i)x_4^* \\ x_2\sqrt{7}-(1+2i)x_3-(1-i)x_4 & x_1^*\sqrt{7}-(1-i)x_3^*-(-1-2i)x_4^* \end{bmatrix} \middle| x_j \in \mathbb{Z}[i], 1 \le j \le 4 \right\}.$$

Choosing $\theta_s = -17$, $\tau_s = \sigma_s$, and given two set elements $X = X(x_1, x_2, x_3, x_4), Y = Y(y_1, y_2, y_3, y_4)$, we construct a distributed iterated Silver code via the map

$$f(\alpha_{\theta_s}(X, Y))_{\mathrm{id}}^2 = \begin{bmatrix} \alpha_{\theta_s}(X,Y) & 0 \\ 0 & \alpha_{\theta_s}(X,Y) \end{bmatrix} = \begin{bmatrix} X & \theta_s\tau_s(Y) & 0 & 0 \\ Y & \tau_s(X) & 0 & 0 \\ 0 & 0 & X & \theta_s\tau_s(Y) \\ 0 & 0 & Y & \tau_s(X) \end{bmatrix} \in \mathrm{Mat}(8, K_s).$$

Set $\mathscr{X}_s = \{\sum_{j=1}^{16} z_j \cdot S_j \mid z_j \in J \cap \mathbb{Z}\}$, where a lattice basis $\Lambda_s = \{S_j\}_{j=1}^{16}$ is given by

$$\{f(\alpha_{\theta_s}(X(1,0,0,0), Y(0,0,0,0)))_{id}^2, \ldots, f(\alpha_{\theta_s}(X(0,0,0,0), Y(0,0,0,1)))_{id}^2,$$
$$f(\alpha_{\theta_s}(X(i,0,0,0), Y(0,0,0,0)))_{id}^2, \ldots, f(\alpha_{\theta_s}(X(0,0,0,0), Y(0,0,0,i)))_{id}^2\}.$$

2. The *Golden code*, a well-performing ST code introduced in [1], is constructed from $\mathscr{C}_g$ and consists of codewords taken from the set

$$\left\{ \frac{1}{\sqrt{5}} \begin{bmatrix} \nu(x_1+x_2\omega) & \nu(x_3+x_4\omega) \\ i\sigma_g(\nu)(x_3+x_4\sigma_g(\omega)) & \sigma_g(\nu)(x_1+x_2\sigma_g(\omega)) \end{bmatrix} \middle| x_j \in \mathbb{Z}[i], 1 \leq j \leq 4 \right\},$$

where $\omega = (1+\sqrt{5})/2$ and $\nu = 1 + i - i\omega$. The Golden code, although very good in performance, is not fast-decodable without modifying the sphere decoder used and has higher decoding complexity than the Silver code.

We set $\theta_g = 1 - i$, $\tau_g = \sigma_g$. Then, for two elements $X = X(x_1, x_2, x_3, x_4), Y = Y(y_1, y_2, y_3, y_4)$, the distributed iterated Golden code is constructed as

$$f(\alpha_{\theta_g}(X,Y))_{\text{id}}^2 = \begin{bmatrix} \alpha_{\theta_g}(X,Y) & 0 \\ 0 & \alpha_{\theta_g}(X,Y) \end{bmatrix} = \begin{bmatrix} X & \theta_g\tau_g(Y) & 0 & 0 \\ Y & \tau_g(X) & 0 & 0 \\ 0 & 0 & X & \theta_g\tau_g(Y) \\ 0 & 0 & Y & \tau_g(X) \end{bmatrix} \in \text{Mat}(8, K_g).$$

Set $\mathscr{X}_g = \{\sum_{j=1}^{16} z_j \cdot G_j \mid z_j \in J \cap \mathbb{Z}\}$, where a lattice basis $\Lambda_g = \{G_j\}_{j=1}^{16}$ is

$$\{f(\alpha_{\theta_g}(X(1,0,0,0), Y(0,0,0,0)))_{id}^2, \ldots, f(\alpha_{\theta_g}(X(0,0,0,0), Y(0,0,0,1)))_{id}^2,$$
$$f(\alpha_{\theta_g}(X(i,0,0,0), Y(0,0,0,0)))_{id}^2, \ldots, f(\alpha_{\theta_g}(X(0,0,0,0), Y(0,0,0,i)))_{id}^2\}.$$

3. Finally we also consider the fast-decodable MIDO$_{A4}$ code constructed in [7], using $\mathscr{C}_m$ as the algebraic structure. Write $\zeta = \zeta_5$ and choose $\{1-\zeta, \zeta-\zeta^2, \zeta^2-\zeta^3, \zeta^3-\zeta^4\}$ a basis of $\mathbb{Z}[\zeta]$. Setting $r = |-8/9|^{1/4}$, codewords are taken from

$$\left\{ \begin{bmatrix} x_1 & -r^2x_2^* & -r^3\sigma_m(x_4) & -r\sigma_m(x_3)^* \\ r^2x_2 & x_1^* & r\sigma_m(x_3) & -r^2\sigma_m(x_4)^* \\ rx_3 & -r^3x_4^* & \sigma_m(x_1) & -r^2\sigma(x_2)^* \\ r^3x_4 & rx_3^* & r^2\sigma_m(x_2) & \sigma_m(x_1)^* \end{bmatrix} \middle| x_j \in \mathbb{Z}[\zeta], 1 \leq j \leq 4 \right\},$$

where for $1 \leq j \leq 4$, $x_j = x_j(l_{4j-3}, l_{4j-2}, l_{4j-1}, l_{4j}) = l_{4j-3}(1-\zeta) + l_{4j-2}(\zeta - \zeta^2) + l_{4j-1}(\zeta^2-\zeta^3) + l_{4j}(\zeta^3-\zeta^4)$. Given an element $X = X(x_1, x_2, x_3, x_4)$ from this set, the adaptation to the cooperative channel is

$$f(X)_{\text{id}}^2 = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \in \text{Mat}(8, K_m).$$

Set $\mathscr{X}_m = \{\sum_{j=1}^{16} z_j \cdot M_j \mid z_j \in J \cap \mathbb{Z}\}$. A lattice basis $\Lambda_m = \{M_j\}_{j=1}^{16}$ is

$$\{X(x_1(1,0,0,0),0,0,0),\ldots,X(0,0,0,x_4(0,0,0,1))\}.$$

4.2 Determinant and Performance Comparison

For the carried out simulations we fix $J = \{\pm 1\}$, the 2-PAM signaling constellation. Further, comparison between the constructed codes requires some kind of normalization, and we choose to normalize the volume of the fundamental parallelotope of the underlying lattices to be $\delta(\Lambda) = 1$. We can then compare the distribution of the normalized determinants among all codewords, as illustrated below. In addition to the previously introduced codes, we further consider a modified version of the distributed iterated Silver code using $\theta_s = -1$. Although this choice does not guarantee full-diversity in general, with 2-PAM the resulting code is still fully diverse (Figs. 1 and 2).

	Golden	Silver$_{-17}$	Silver$_{-1}$	MIDO$_{A4}$
Minimum det.	$\mathbf{4.445 \cdot 10^{-3}}$	$1.553 \cdot 10^{-5}$	$4.16 \cdot 10^{-4}$	$3.871 \cdot 10^{-7}$
Maximum det.	13.871	4.099	14.268	**80.500**
Average det.	1.819	0.493	2.007	**7.485**

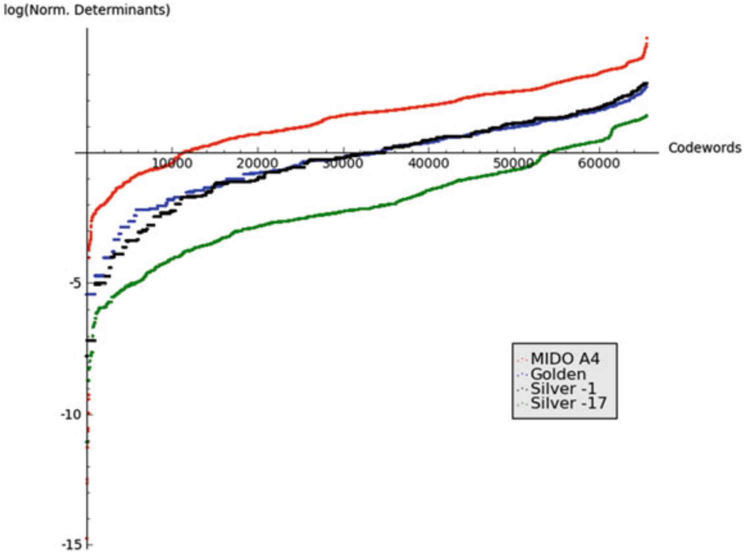


Fig. 1 The logarithmic distribution of the normalized determinants of all the 2^{16} codewords in $\mathscr{X}_g$, $\mathscr{X}_m$ and $\mathscr{X}_s$ for both $\theta_s = -17$ and $\theta_s = -1$

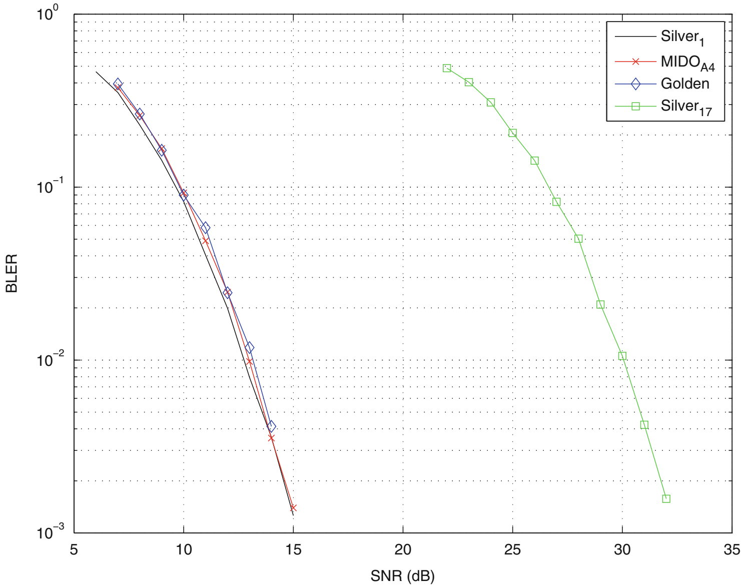


Fig. 2 Performance comparison of the above four example codes with 2-PAM signaling. The data rate is $16/8 = 2$ bits per channel use (bpcu). The Silver code with $\theta_s = -17$ has the worst performance, which is to be expected due to high peak-to-average power ratio stemming from the fact that $|17|$ is not close to one. The other codes perform more or less equally

The exact complexity reduction of the iterated distributed codes remains to be examined. It is also not necessarily obvious, that the proposed construction achieves the DMT, since the conditions in [8] require that the code rate is $2n_s$, while our example constructions all have code rate $n_d = 1 < 2n_s$. However, since they are full-rate (similarly to the codes in [8]) for n_d antennas at the destination, we expect that they do achieve the DMT.

Acknowledgements N. Markin was supported by the Singapore National Research Foundation under Research Grant NRF-RF2009-07

References

1. Belfiore, J.C., Rekaya, G., Viterbo, E.: The golden code: a 2 × 2 full-rate space-time code with nonvanishing determinants. IEEE Trans. Inf. Theory **51**(4), 1432–1436 (2005)
2. Biglieri, E., Hong, Y., Viterbo, E.: On fast-decodable space-time block codes. IEEE Trans. Inf. Theory **55**(2), 524–530 (2009)
3. Hollanti, C., Lahtonen, J., Ranto, K., Vehkalahti, R., Viterbo, E.: On the algebraic structure of the silver code: a 2 × 2 perfect space-time block code. In: 2008 IEEE ITW, Porto, pp. 91–94 (2008)

4. Hollanti, C., Markin, N.: Algebraic fast-decodable relay codes for distributed communications. In: 2012 IEEE ISIT, Cambridge, pp. 935–939 (2012)
5. Hollanti, C., Markin, N.: A unified framework for constructing fast-decodable codes for n relays. In: 2012 MTNS, Melbourne (2012)
6. Markin, N., Oggier, F.: Iterated space-time code constructions from cyclic algebras. IEEE Trans. Inf. Theory **59**(9), 5966–5979 (2013)
7. Vehkalahti, R., Hollanti, C., Oggier, F.: Fast-decodable asymmetric space-time codes from division algebras. IEEE Trans. Inf. Theory **58**(4), 2362–2385 (2011)
8. Yang, S., Belfiore, J.C.: Optimal space-time codes for the MIMO amplify-and-forward cooperative channel. IEEE Trans. Inf. Theory **53**(2), 647–663 (2007)

Cyclic Generalized Separable (L, G) Codes

Sergey Bezzateev

Abstract A new class of cyclic generalized separable (L, G) codes is constructed.

Keywords Generalized (L,G) codes • Goppa codes • Cyclic codes

1 Introduction

A classical Goppa code [1] is determined by two objects: a Goppa polynomial $G(x)$ with coefficients from $GF(q^m)$ and location set L of codeword positions

$$L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subseteq GF(q^m), G(\alpha_i) \neq 0, \ \forall \alpha_i \in L.$$

Definition 1 A q-ary vector $\boldsymbol{a} = (a_1 a_2 \ldots a_n)$ is a codeword of (L, G)-code if and only if the following equality is satisfied

$$\sum_{i=1}^{n} a_i \frac{1}{x - \alpha_i} \equiv 0 \mod G(x).$$

Definition 2 Goppa code is called separable if the polynomial $G(x)$ does not have multiple roots.

In [1] V.D. Goppa proved that the primitive BCH codes are the only subclass of Goppa codes that are cyclic with $G(x) = (x - \gamma)^t, \gamma \in GF(q^m)$, $L \subseteq GF(q^m) \setminus \{\gamma\}$. Accordingly, the only one class of separable Goppa codes with $G(x) = (x - \gamma), \gamma \in GF(q^m), L \subseteq GF(q^m) \setminus \{\gamma\}$ defined as cyclic.

In 1973 in [2] and later in [3–11] a subclasses of extended separable Goppa codes and subclasses of separable Goppa codes with Goppa polynomials of degree 2 and additional parity check were proposed. It was proved that these codes are cyclic.

S. Bezzateev (✉)
Saint Petersburg State University of Aerospace Instrumentation, Bolshaya Morskaya st., 67, Saint Petersburg, 190000 Russia
e-mail: bsv@aanet.ru

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_5

However, the existence among separable Goppa codes any subclass of cyclic codes remained an open problem ([12] Ch.12, Corollary 9, Research Problem 12.3).

In 2013 in [13] the subclass of cyclic separable Goppa codes with a special choice of location set L and and Goppa polynomial $G(X)$ of degree 2 was suggested.

$$\begin{aligned}
&L = \{\alpha_1, \alpha_2 \ldots, \alpha_{n-1}, \alpha_n\} \subset \{GF(q^{2m}) \setminus GF(q^m)\} \bigcup \{1\},\\
&\alpha_n = 1, \alpha_i^{q^m} = \alpha_i^{-1} = \alpha_{n-i}, n = q^m \pm 1,\\
&G(x) = (x-\beta)(x-\beta^{-1}),\ \beta \in GF(q^{2m}),\ \beta + \beta^{-1} \in GF(q^m),\\
&G(\alpha_i) \neq 0, \alpha_i \neq \alpha_j,\ \forall i, j \in \{1, \ldots, n\},\ i \neq j.
\end{aligned}$$

A generalized Goppa code [14] can be constructed by using the following generalization of location set L:

$$L = \left\{ \frac{f_1'(x)}{f_1(x)}, \frac{f_2'(x)}{f_2(x)}, \ldots, \frac{f_n'(x)}{f_n(x)} \right\}, \tag{1}$$

where $f_i'(x)$ is a formal derivative of $f_i(x)$ in $GF(q)$ and

$$\begin{aligned}
&f_i(x) = x^\ell + a_{i,\ell-1}x^{\ell-1} + \ldots + a_{i,1}x + a_{i,0}, a_{i,j} \in GF(q^\mu),\\
&\gcd(f_i(x), f_j(x)) = 1,\ \gcd(f_i(x), G(x)) = 1,\ \forall i, j,\ i \neq j.
\end{aligned}$$

Definition 3 q-ary vector $\boldsymbol{a} = (a_1 a_2 \ldots a_n)$ is a codeword of generalized (L, G)-code if and only if the following equality is satisfied

$$\sum_{i=1}^{n} a_i \frac{f_i'(x)}{f_i(x)} \equiv 0 \mod G(x). \tag{2}$$

Generalized Goppa codes have allowed to expand a class of cyclic Goppa codes with $G(x) = (x-\gamma)^t$. Many cyclic (n, k, d) codes can be described as generalized Goppa codes [15] with

$$\begin{aligned}
&f_i(x) = f(\alpha^i x),\ f(x) = x^\ell + a_{\ell-1}x^{\ell-1} + \ldots + a_1 x + a_0,\ \alpha, a_j \in GF(q^\mu),\\
&a_0 \neq 0, \alpha^n = 1,\ n|(q^\mu - 1),\ \gcd(f_i(x), f_j(x)) = 1,\ \forall i, j,\ i \neq j
\end{aligned}$$

and

$$G(x) = x^t.$$

For such codes the design bound for minimum distance $d_G \geq \frac{t+1}{\ell}$ and the corresponding decoding algorithm were determined [16, 17]. However, a subclass of cyclic generalized separable Goppa codes is still remained limited by polynomial $G(x) = (x-\gamma), \gamma \in GF(q^\mu)$.

2 Two Subclasses of Binary Cyclic Generalized Separable Goppa Codes

In this paper we will consider a binary case with two variants of separable Goppa polynomial

$$G(x) = x^n - 1 \text{ and } \hat{G}(x) = x(x^n - 1). \tag{3}$$

We will need the following definitions.

Definition 4 For any integers $n,\ n|(2^m - 1)$ and $l,\ 0 \leq l < n$ a cyclotomic coset m_l is given by

$$m_l = \{l2^j \mod n, \forall j = 0, 1, \ldots, \lambda_l - 1\},$$

where λ_l is the smallest integer greater than 0 such that $l2^{\lambda_l} \equiv l \mod n$.

Definition 5 The minimal polynomial $M_l(x)$ of element $\alpha^l \in GF(2^m)$ is given by

$$M_l(x) = \prod_{j \in m_l} (x - \alpha^j),\ \deg M_l(x) = \lambda_l.$$

Definition 6 The generator polynomial of a cyclic (n, k, d) code C is given by

$$g(x) = \prod_{j \in D} (x - \alpha^j),\ D = \bigcup_{j=1}^{\nu} m_{l_j} \text{ and } g(x) = \prod_{j=1}^{\nu} M_{l_j}(x),\ \deg g(x) = \prod_{j=1}^{\nu} \lambda_{l_j} = n - k,$$

where D is the set containing the indices of the zeros of the generator polynomial $g(x)$. The size of set D is equal to $n - k$.

For some D let's consider a binary linear (η, κ, τ) code C_L with the length η, dimension κ, minimum distance τ and parity-check matrix H_L

$$H_L = \begin{bmatrix} \frac{\beta_1^{j1}}{G(\beta_1)} & \cdots & \frac{\beta_\eta^{j1}}{G(\beta_\eta)} \\ \frac{\beta_1^{j2}}{G(\beta_1)} & \cdots & \frac{\beta_\eta^{j2}}{G(\beta_\eta)} \\ \vdots & \ddots & \vdots \\ \frac{\beta_1^{jk}}{G(\beta_1)} & \cdots & \frac{\beta_\eta^{jk}}{G(\beta_\eta)} \end{bmatrix}, \begin{matrix} \beta_i \in GF(2^\mu) \setminus \{0, 1\},\ GF(2^\mu) \cap GF(2^m) = \{0, 1\}, \\ N = \{j_1, j_2, \ldots, j_k\}, N \cup D = \{0, 1, \ldots, n - 1\}. \end{matrix} \tag{4}$$

Let $\mathbf{b} = (b_1\ b_2 \ldots b_\tau\ b_{\tau+1} \ldots b_\eta)$ with $b_i = 1, \forall i = 1, \ldots \tau$ and $b_i = 0, \forall i = \tau + 1, \ldots, \eta$ be a codeword of this code. Then for this vector $\mathbf{b}$ and parity-check matrix H_L we obtain

$$\mathbf{b}\cdot H^T = 0 \text{ and } \sum_{i=1}^{\eta} b_i \frac{\beta_i^{j_l}}{G(\beta_i)} = \sum_{i=1}^{\tau} \frac{\beta_i^{j_l}}{G(\beta_i)} = 0, \forall l = 1, \ldots, k. \tag{5}$$

As in [17] we will call C_L as non-zero-locator code for cyclic code C with the set D if for any $m_i \subset D$ exists $j : j \in m_i, \sum_{i=1}^{\tau} \frac{\beta_i^j}{G(\beta_i)} \neq 0$. We associate with codeword $\mathbf{b}$ of this non-zero-locator code C_L the following locator polynomial

$$\begin{aligned} &f(x) = (x-\beta_1)(x-\beta_2)\cdots(x-\beta_\tau),\ \beta_j \in GF(2^\mu),\ j = 1, \ldots, \tau,\\ &f_i(x) = (x-\alpha^i\beta_1)(x-\alpha^i\beta_2)\cdots(x-\alpha^i\beta_\tau),\ \alpha \in GF(2^m),\ \alpha^n = 1,\\ &\gcd(f_i(x), f_j(x)) = 1,\ \forall i \neq j,\ \ i, j = 1, \ldots, n. \end{aligned} \tag{6}$$

Theorem 7 *Generalized (L, G) code with Goppa polynomial $G(x)$ (3) and locator set L (1) defined by non-zero-locator code C_L (4),(5) and by associated locator polynomial $f(x)$ (6) is a cyclic code C with the set D of indices of zeroes of generator polynomial.*

Proof Parity-check matrix H_G for this code is:

$$H_G = \begin{bmatrix} \alpha_1^{\ell_1}\sum_{i=1}^{\tau}\frac{\beta_i^{\ell_1}}{G(\beta_i)} & \ldots & \alpha_n^{\ell_1}\sum_{i=1}^{\tau}\frac{\beta_i^{\ell_1}}{G(\beta_i)} \\ \alpha_1^{\ell_2}\sum_{i=1}^{\tau}\frac{\beta_i^{\ell_2}}{G(\beta_i)} & \ldots & \alpha_n^{\ell_2}\sum_{i=1}^{\tau}\frac{\beta_i^{\ell_2}}{G(\beta_i)} \\ \vdots & \ddots & \vdots \\ \alpha_1^{\ell_\delta}\sum_{i=1}^{\tau}\frac{\beta_i^{\ell_\delta}}{G(\beta_i)} & \ldots & \alpha_n^{\ell_\delta}\sum_{i=1}^{\tau}\frac{\beta_i^{\ell_\delta}}{G(\beta_i)} \end{bmatrix} = \begin{bmatrix} \alpha_1^{\ell_1} & \ldots & \alpha_n^{\ell_1} \\ \alpha_1^{\ell_2} & \ldots & \alpha_n^{\ell_2} \\ \vdots & \ddots & \vdots \\ \alpha_1^{\ell_\delta} & \ldots & \alpha_n^{\ell_\delta} \end{bmatrix}, \tag{7}$$

where $\{\ell_1, \ell_2, \ldots, \ell_\delta\} \subseteq D$.

□

Note 8 By Definition 6 dimension of this code is $k = n - \|D\|$, where $\|D\|$ is a size of the set D.

For the case $\hat{G}(x) = x(x^n - 1)$ we will obtain a similar theorem.

Theorem 9 *Generalized $(L, \hat{G})$ code with Goppa polynomial $\hat{G}(x)$ (3) and locator set L (1) defined by non-zero-code C_L (4),(5) is a cyclic code $\hat{C}$ with the set $\hat{D} \subseteq D \cup m_{-1}$ of indices of zeroes of generator polynomial.*

Proof Parity-check matrix $H_{\hat{G}}$ for this code is:

$$H_{\hat{G}} = \begin{bmatrix} \alpha_1^{-1} \sum\limits_{i=1}^{\tau} \frac{1}{G(\beta_i)} & \ldots & \alpha_n^{-1} \sum\limits_{i=1}^{\tau} \frac{1}{G(\beta_i)} \\ \alpha_1^{\ell_1} \sum\limits_{i=1}^{\tau} \frac{\beta_i^{\ell_1}}{G(\beta_i)} & \ldots & \alpha_n^{\ell_1} \sum\limits_{i=1}^{\tau} \frac{\beta_i^{\ell_1}}{G(\beta_i)} \\ \alpha_1^{\ell_2} \sum\limits_{i=1}^{\tau} \frac{\beta_i^{\ell_2}}{G(\beta_i)} & \ldots & \alpha_n^{\ell_2} \sum\limits_{i=1}^{\tau} \frac{\beta_i^{\ell_2}}{G(\beta_i)} \\ \vdots & \ddots & \vdots \\ \alpha_1^{\ell_\delta} \sum\limits_{i=1}^{\tau} \frac{\beta_i^{\ell_\delta}}{G(\beta_i)} & \ldots & \alpha_n^{\ell_\delta} \sum\limits_{i=1}^{\tau} \frac{\beta_i^{\ell_\delta}}{G(\beta_i)} \end{bmatrix} = \begin{cases} H_G, \text{ if } -1 \in D \text{ or } 0 \in N, \\ \begin{bmatrix} \alpha_1^{-1} \ldots \alpha_n^{-1} \\ H_G \end{bmatrix}, \text{ if } -1 \notin D \text{ and } 0 \notin N. \end{cases} \tag{8}$$

□

Theorem 10 *From* (2), (3) *and* (6) *we obtain the following estimation for minimal distance of binary cyclic generalized separable Goppa code:*

$$d_G \geq \frac{2n+1}{\tau} \textit{ for } G(x) = x^n - 1$$

and

$$d_{\hat{G}} \geq \frac{2n+3}{\tau} \textit{ for } \hat{G}(x) = x(x^n - 1).$$

3 Trace Non-zero-Locator Code

As example of non-zero-locator code let's consider a binary linear code with length η , parity-check matrix

$$H_L = \begin{bmatrix} \frac{\beta^{j_1}}{G(\beta)} & \frac{\beta^{2j_1}}{G(\beta^2)} & \cdots & \frac{\beta^{\eta j_1}}{G(\beta^\eta)} \\ \frac{\beta^{j_2}}{G(\beta)} & \frac{\beta^{2j_2}}{G(\beta^2)} & \cdots & \frac{\beta^{\eta j_2}}{G(\beta^\eta)} \\ \vdots & \ddots & \vdots & \\ \frac{\beta^{j_k}}{G(\beta)} & \frac{\beta^{2j_k}}{G(\beta^2)} & \cdots & \frac{\beta^{\eta j_k}}{G(\beta^\eta)} \end{bmatrix}, \quad \begin{array}{l} \beta - \text{ primitive element in } GF(2^\mu), \\ tr(\beta^{j_i}) = 0, \forall i = 1, \ldots, k, \\ N = \{j_1, j_2, \ldots, j_k\}, \\ N \cup D = \{0, 1, \ldots, n-1\}. \end{array} \tag{9}$$

and codeword

$$\mathbf{b} = (b_1 b_2 \ldots b_\eta), wt(\boldsymbol{b}) = \mu \text{ and } b_1 = b_2 = b_{2^2} = \ldots = b_{2^{\mu-1}} = 1.$$

Now we can rewrite matrix H_G (7) in the following form

$$H_G = \begin{bmatrix} \alpha_1^{\ell_1} tr\left(\frac{\beta^{\ell_1}}{G(\beta)}\right) & \dots & \alpha_n^{\ell_1} tr\left(\frac{\beta^{\ell_1}}{G(\beta)}\right) \\ \alpha_1^{\ell_2} tr\left(\frac{\beta^{\ell_2}}{G(\beta)}\right) & \dots & \alpha_n^{\ell_2} tr\left(\frac{\beta^{\ell_2}}{G(\beta)}\right) \\ \vdots & \ddots & \vdots \\ \alpha_1^{\ell_\delta} tr\left(\frac{\beta^{\ell_\delta}}{G(\beta)}\right) & \dots & \alpha_n^{\ell_\delta} tr\left(\frac{\beta^{\ell_\delta}}{G(\beta)}\right) \end{bmatrix} = \begin{bmatrix} \alpha_1^{\ell_1} & \dots & \alpha_n^{\ell_1} \\ \alpha_1^{\ell_2} & \dots & \alpha_n^{\ell_2} \\ \vdots & \ddots & \vdots \\ \alpha_1^{\ell_\delta} & \dots & \alpha_n^{\ell_\delta} \end{bmatrix}, \tag{10}$$

where $tr\left(\frac{\beta^{\ell_i}}{G(\beta)}\right) \neq 0, i = 1, \dots, \delta, \ \{\ell_1, \ell_2, \dots, \ell_\delta\} \subseteq D.$

For such trace non-zero-locator code we have locator polynomial $f(x)$ from (6):

$$f(x) = (x-\beta)(x-\beta^2)\cdots(x-\beta^{2^{\mu-1}}) = \Omega_1(x),\ \Omega_1(x) \in \mathbb{F}_2[x],\ \deg \Omega_1(x) = \mu,$$

$\Omega_1(x)$ is a minimal polynomial of element $\beta \in GF(2^\mu)$.

From Theorem 10 we obtain the following estimation for minimal distance of binary cyclic generalized separable Goppa code with trace non-zero-locator code:

$$d_G \geq \frac{2n+1}{\mu} \text{ for } G(x) = x^n - 1$$

and

$$d_{\hat{G}} \geq \frac{2n+3}{\mu} \text{ for } \hat{G}(x) = x(x^n - 1).$$

4 Examples

1.

$$\begin{aligned} & n = 21, \hat{G}(x) = x(x^{21}-1), \alpha \in GF(2^6), \alpha^{21} = 1, \beta \in GF(2^7), \\ & f(x) = x^7 + x^6 + x^4 + x + 1, f_i(x) = \alpha^{7i}x^7 + \alpha^{6i}x^6 + \alpha^{4i}x^4 + \alpha^i x + 1, \\ & L = \{\tfrac{x^6+1}{x^7+x^6+x^4+x+1}, \tfrac{\alpha^7x^6+\alpha}{\alpha^7x^7+\alpha^6x^6+\alpha^4x^4+\alpha x+1}, \dots, \tfrac{\alpha^{14}x^6+\alpha^{20}}{\alpha^{14}x^7+\alpha^{15}x^6+\alpha^{17}x^4+\alpha^{20}x+1}\}, \\ & tr\left(\frac{\beta^i}{\hat{G}(\beta)}\right) = 1,\ i = 0, 3, 4, 6, 7, 12, 14, 21, \\ & tr\left(\frac{\beta^i}{\hat{G}(\beta)}\right) = 0,\ i = 1, 2, 5, 8, 9, 10, 11, 13, 15, 16, 17, 18, 19, 20. \end{aligned}$$

Therefore from Theorem 9 we have $(21, 6, 7)$ cyclic code with generator polynomial $g(x) = m_1(x)m_3m_5(x)$. From Theorem 10 we obtain the following estimation for minimum distance for this generalized separable $(L, \hat{G})$ code:

$$d_G \geq \frac{2n+3}{\mu} = \frac{45}{7} > 6 \text{ and we have } d_G = d = 7.$$

2.

$$n = 21, \hat{G}(x) = x^{21} - 1, \alpha \in GF(2^6), \alpha^{21} = 1, \beta \in GF(2^7),$$
$$f(x) = x^7 + x^6 + x^4 + x^2 + 1,\ f_i(x) = \alpha^{7i}x^7 + \alpha^{6i}x^6 + \alpha^{4i}x^4 + \alpha^{2i}x^2 + 1,$$
$$L = \{\frac{x^6}{x^7+x^6+x^4+x^2+1}, \frac{\alpha^7x^6}{\alpha^7x^7+\alpha^6x^6+\alpha^4x^4+\alpha^2x^2+1}, \dots, \frac{\alpha^{14}x^6+\alpha^{20}}{\alpha^{14}x^7+\alpha^{15}x^6+\alpha^{17}x^4+\alpha^{19}x^2+1}\},$$
$$tr\left(\frac{\beta^i}{\hat{G}(\beta)}\right) = 1,\ i = 2, 3, 5, 6, 11, 13, 20,$$
$$tr\left(\frac{\beta^i}{\hat{G}(\beta)}\right) = 0,\ i = 0, 1, 4, 7, 8, 9, 10, 12, 14, 15, 16, 17, 18, 19.$$

From Eq. (10) and Theorem 7 we have $(21, 6, 7)$ cyclic code with generator polynomial $g(x) = m_1(x)m_3(x)m_5(x)$. From Theorem 10 we obtain the following estimation for minimum distance for this generalized separable (L, G) code:

$$d_{\hat{G}} \geq \frac{2n+1}{\mu} = \frac{43}{7} > 6 \text{ and we have } d_{\hat{G}} = d = 7.$$

5 Conclusion

The new subclasses of cyclic generalized separable Goppa codes with Goppa polynomials $x^n - 1$ and $x(x^n - 1)$ are proposed. The parameters and examples of the codes from these subclasses are shown.

References

1. Goppa,V.D.: A new class of linear error-correcting codes. Probl. Inf. Trans. **6**(3) , 24–30 (1970)
2. Berlecamp, E.R., Moreno, O.: Extended double-error-gorrecting binary Goppa codes are cyclic. IEEE Trans. Inf. Theory **19**(6), 817–818 (1973)
3. Tzeng, K.K., Zimmermann, K.: On extending Goppa codes to cyclic codes. IEEE Trans. Inf. Theory **21**(6), 712–716 (1975)
4. Tzeng, K.K., Yu, C.Y.: Characterization theorems for extending Goppa codes to cyclic codes. IEEE Trans. Inf. Theory **25**(2), 246–250 (1979)
5. Moreno, O.: Symmetries of binary Goppa codes. IEEE Trans. Inf. Theory **25**(5), 609–612 (1979)
6. Vishnevetskii, A.L.: Cyclicity of extended Goppa codes. Probl. Pered. Inf. **18**(3), 14–18 (1982)
7. Stichenoth, H.: Wich extended Goppa codes are cyclic? J. Comb. Theory **A 51**, 205–220 (1989)
8. B erger, T.P.: Goppa and related codes invariant under a prescribed permutation. IEEE Trans. Inf. Theory **46**(7), 2628–2633 (2000)
9. Berger, T.P.: On the cyclicity of Goppa codes, parity-check subcodes of Goppa codes, and extended Goppa codes. Finite Fields Appl. **6**, 255–281 (2000)
10. Berger, T.P.: Quasi-cyclic Goppa codes. In: Proceedings of ISIT2000, Sorrente, p. 195 (2000)
11. Berger, T.P.: New classes of cyclic extended Goppa codes. IEEE Trans. Inf. Theory **45**(4), 1264–1266 (1999)

12. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
13. Bezzateev, S., Shekhunova, N.: Subclass of cyclic Goppa codes. IEEE Trans. Inf. Theory **59**(11), 7379–7385 (2013)
14. Shekhunova, N.A., Mironchikov, E.T.: Cyclic (L, g)-codes. Probl. Pered. Inf. **17**(2), 3–9 (1981)
15. Bezzateev, S.V., Shekhunova, N.A.: One generalization of Goppa codes. In: Proceedings of ISIT-97, Ulm, p. 299 (1997)
16. Zeh, A., Wachter-Zeh, A., Bezzateev, S.: Decoding cyclic codes up to a new bound on the minimum distance. IEEE Trans. Inf. Theory **58**(6), 3951–3960 (2012)
17. Zeh, A., Bezzateev, S.: A new bound on the minimum distance of cyclic codes using small-minimum-distance cyclic codes. Designs Codes Cryptogr. **71**, 229–246 (2014)

The One-Out-of-k Retrieval Problem and Linear Network Coding

Giuseppe Bianchi, Lorenzo Bracciale, Keren Censor-Hillel, Andrea Lincoln, and Muriel Médard

Abstract In this paper we show how linear network coding can reduce the number of queries needed to retrieve *one specific message* among k distinct ones replicated across a large number of randomly accessed nodes storing one message each. Without network coding, this would require k queries on average. After proving that no scheme can perform better than a straightforward lower bound of $0.5k$ average queries, we propose and asymptotically evaluate, using mean field arguments, a few example practical schemes, the best of which attains $0.82k$ queries on average. The paper opens two complementary challenges: a systematic analysis of practical schemes so as to identify the best performing ones and design guideline strategies, as well as the need to identify tighter, nontrivial, lower bounds.

Keywords Delay tolerant network • Linear network coding • Fluid approximations

1 Introduction

This paper introduces a new problem, which we call *one-out-of-k retrieval*. Suppose there are k distinct messages $X = \{x_1, \ldots, x_k\}$, where $x_i \in 0, 1^m \ \forall i \in [1, k]$. A receiver wishes to learn all m bits of one *specific* target message, $x_r \in X$. We can produce some new set of messages $Y = \{y_1, y_2, \ldots\}$ of arbitrary size and contents. Each round, the receiver can request a message selected over a pre-determined probability distribution from Y. We wish to come up with a set of linearly coded messages for Y and a probability distribution over these such that the average

G. Bianchi (✉) • L. Bracciale (✉)
University of Roma Tor Vergata, Roma, Italy
e-mail: giuseppe.bianchi@uniroma2.it; lorenzo.bracciale@uniroma2.it

K. Censor-Hillel (✉)
Technion – Israel Institute of Technology, Haifa 32000, Israel
e-mail: ckeren@cs.technion.ac.il

A. Lincoln (✉) • M. Médard (✉)
MIT, Cambridge, MA 02139-4307, USA
e-mail: andreali@mit.edu,medard@mit.edu

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_6

number of rounds in which a message must be requested by the receiver from Y to learn all m bits of x_r is minimized.

This scenario is practically encountered in Delay Tolerant Networks (DTN). In such networks, data replication across the moving terminals is at the core of most proposed data access or data delivery solutions, as the likelihood that a user interested in a specific data item "physically" meets only the single data producer becomes rapidly negligible as the network size scales.

1.1 Contribution

PMost network coding research has focused on retrieving and decoding *all* the messages instead of a specific subset Even for the case of $k = 2$, schemes exist which take an average of ≈ 1.828 coded messages from Y, outperforming the naive average of 2. This raises some questions: how much reduction in average numbers of messages from Y can we gain? And with which practical constructions?

In the paper, we present a lower bound of $0.5k$ for the average number of rounds the receiver must request messages. Then, we propose some initial example schemes where the selection of the probability distribution over Y results in a lower average number of requests than the naive average of k messages needed from the set Y.

Moreover, we provide a general methodology to analyze such schemes. We specifically show how to apply mean field arguments to derive the asymptotic performance of the proposed approaches. We concretely apply our methodology to two example schemes, the best of which attains an average of $0.82k$ rounds of communication.

1.2 Previous Work

Previous work on network coding in DTNs has not considered the problem of solving for *one* out of k messages. In our model, the protocol does not allow for the receiver to request the specific information it wants and nor do we treat it as wanting all information. For instance, LT codes [6] are designed with the different goal of optimizing the decoding procedures. Many papers [2, 7, 8, 10] investigate routing protocols in DTNs. These papers attempt to decode all messages, as opposed to just *one* of k. Yoon and Hass consider application of linear network coding to DTNs but, unlike this work, investigate the case of sparse networks [9].

2 Network Model and Problem Statement

In our model there are k messages $X = \{x_1, \dots, x_k\}$, each of which is a can be represented by a binary vector of length m bits. There is a receiver node, r, which wants to know the contents of the one message, we will call this message x_r. The receiver, r, travels throughout the network and will receive messages from the nodes it contacts in close proximity. We model this as r contacting a random node, which transmits its output. These contacts cannot be commanded so messages may be repeated and r can not query for a particular message. In each round, the receiver node r receives exactly one coded message, y, from one of the transmitting nodes. Each round has a constant duration. The nodes in this network can store linear combinations of messages over some field F_f.

Definition 1 The type (or degree) of a coded message is the number of message linearly combined in that data message.

These linear combinations are stored with header data that specifies which messages were summed with what multiplicative constants.

Definition 2 Solving for message x_j means determining all m bits in the message x_j.

Definition 3 The *one-out-of-k* retrieval problem is determining what coding scheme produces the lowest expected time for r to solve for x_r where a coding scheme is the proportion $p_1, p_2 \dots p_k$ of the codeword degrees distributed in the networks.

In other words, we want to find $p_1 \cdots p_k$ that minimize the time for retrieving only one message, given that the receiver collects at each round an uncoded message with probability p_1, a “pair” (codeword with degree 2) with probability p_2, a “triplet” with probability p_3 etc.

Thus, Y is the set of all linear combinations of the k messages in X. Each coded message, $y \in Y$, is a linear combination of n messages and has a probability $\frac{p_n}{\binom{k}{n}}$ of being sent to the receiver.

2.1 A Trivial Example: k = 2

Consider the simple case where we have only two kind of different message that we call A and B. If we do not use coding ($p_1 = 1, p_2 = 0$) it is trivial to show that the average time spent from the receiver for collecting A (or equivalently B) is 2, i.e. k. Similarly if all nodes carry a random linear combination of both A and B ($p_1 = 0, p_2 = 1$) the expected retrieval time is *exactly* 2 encounters, so once again the average is 2. Now let AB be the linear combination of A and B so that at each encounter the receiver can collect A with probability $p/2$, B with probability $p/2$,

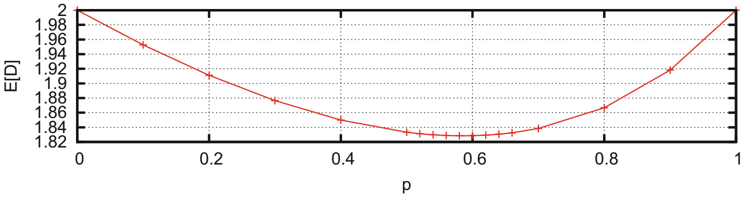


Fig. 1 Average retrieval delay for the case of $k = 2$

and AB with probability $1 - p$. The average delay to retrieve item A is:

$$\text{Delay} = 1 - p + \frac{1}{1 - p/2}$$

Then it is trivial to show that when $p = 2 - \sqrt{2}$ the expected time to retrieve A is minimized and equals to $2\sqrt{2} - 1 \approx 1.828$, i.e. about 9 % lower than both previous cases. Hence this problem is solved adopting the coding scheme $p_1 = 2 - \sqrt{2}$, $p_2 = \sqrt{2} - 1$. The delay versus p is shown in Fig. 1.

2.2 *Lower Bound*

For the problem of determining the contents of one message out of k we prove that $0.5k$ messages is the lowest achievable average cost.

Intuitively level to solve for c messages we must receive at least c coded messages.

Lemma 4 *On average there are greater than* $\frac{1}{2}k$ *messages solved for before or in the same round as* x_r.

Proof First let us define m_j as the number of messages solved before or at the same time that x_j is solved. If a message x_j never has its contents solved then define $m_j = k$. For convenience, let m_r be the number of messages solved before or at the same time as the message of interest x_r. x_r is randomly selected from $\{x_j | j \in [1, k]\}$. Thus, the average number of message solved before or at the same round as x_r is the average vale of m_j i.e. $\frac{\sum_{i=1}^{k} m_j}{k}$.

If all the values of m_j are distinct then the minimum value they can have is the integers from 1 to k thus the average value of m_j is:

$$\frac{(k+1)k}{2k} = \frac{k+1}{2}.$$

If some messages are solved at the same time (in the same round) then this sum is strictly greater because having multiple messages solved at the same time causes double counting.

Thus, on average there are greater than $\frac{1}{2}k$ messages solved before or in the same time step as x_r. □

Next we use this lemma to prove a lower bound on the average number of rounds needed.

Theorem 5 *There exists no scheme in our model such that the contents of a message x_r selected at random can be solved with fewer than $\frac{1}{2}k$ coded messages on average.*

Proof Given Lemma 4 when the receiver, r, has solved for x_r, having received t_r coded messages, r has also solved for more than $\frac{1}{2}k$ messages.

To solve for m_j messages the receiver must receive at least m_j coded messages. Thus the average number of coded messages needed to solve for x_r must be greater than or equal to the average value of m_j. Thus a lower bound for the average number of coded messages needed is $k/2$. □

3 Methodology

Determining whether the set of received messages fully specifies the target *one-out-of-k* message, is the major difficulty. Since messages are retrieved at random, differently coded messages are collected (e.g. uncoded messages, linear combination of two messages, linear combination of all k messages, and so on depending on the construction). The set of collected messages also depends on time, requiring a *transient* stochastic process to model a chosen strategy, which usually exhibits a non-trivial space state.

To avoid such stochastic modeling complexity, the methodology employed hereafter consists of three steps: (i) model a proposed coding strategy via a discrete time (vector) stochastic process; this is arguably the most complex step, as discussed later on; (ii) approximate the proposed coding strategy's transient solution with the deterministic mean trajectory specified by the drift (vector) differential equation of a conveniently rescaled stochastic process, and (iii) derive the average number of queries needed to retrieve the target message from a relevant probability distribution, which is derived from the knowledge of the drift equation solutions.

The approximation in step (ii) above is motivated by the fact that practical values of k are relatively large. It consists of using mean field techniques widely established in the literature since [5], which have been successfully applied to a variety of problems [1, 3], and which guarantee asymptotic convergence to *exact* results for finite state space systems under mild assumptions (see e.g., [3]). Our own results show a very accurate matching with simulation even for relatively small values of k.

Details and a simple example of the proposed methodology are presented in Appendix 1.

4 Practical Example Cases

In order to understand the *asymptotic* nature of the gain, and show how the proposed methodology can be concretely applied we show two example constructions. In both cases, we compare analytical results with simulation.

4.1 All-or-Nothing Scheme

This scheme is extremely simple in terms of states, permits a simple analysis, and can be used as a reference to gauge the improvements brought about by more complex schemes. The *all-or-nothing* scheme comprises only two possible types of messages, defined below.

Definition 6 A *singleton* is a message x_i for $i \in [1, k]$ sent in plain text.

Definition 7 A *fully coded message* is a random linear combination $\sum_{i=1}^{k} \alpha_i x_i$ of all k messages over a large field size $\mathbb{F}$, with $\alpha_i \in \mathbb{F}$.

We assume that all messages x_i, with $i \in [1, k]$, are equiprobable. Under this assumption, the *all-or-nothing* scheme is characterized by a single parameter p, where p is the singleton reception probability and $1 - p$ is the complementary fully coded message reception probability. The state space thus comprises two state variables: (i) the number of singletons received at a given time, and (ii) the number of fully coded messages received at the same time.

Theorem 8 *The* all-or-nothing *scheme achieves a best possible performance of* $0.86k$*; which corresponds to the value* $p \approx 0.6264$.

Proof Using the methodology presented above, let we define the following two density processes:

- $s(t) \in (0, 1)$ is the fraction of singletons accumulated until time t;
- $d(t) \in (0, 1)$ is the fraction of fully coded messages accumulated until time t.

In this case, the drift differential equation reduces to two independent ordinary differential equations. For the case of singletons, operating in a similar way to the example in Appendix 1, we have:

$$s'(t) = 1 - ps(t), \tag{1}$$

which, when solved with initial conditions $s(0) = 0$, yields

$$s(t) = 1 - e^{-pt}. \tag{2}$$

For the case of fully coded messages, we have:

$$d'(t) = (1-p)d(t) \quad \text{with } d(0) = 0. \tag{3}$$

Therefore

$$d(t) = (1-p)t. \tag{4}$$

We now note that a target message is decoded when either the corresponding singleton is received, or when the number of received singletons plus the number of fully coded messages is equal to the total number k of distinct messages. In terms of density processes, this latter condition is expressed by the equation

$$s(t) + d(t) = 1 \quad \rightarrow \quad e^{-pt} + t = 1. \tag{5}$$

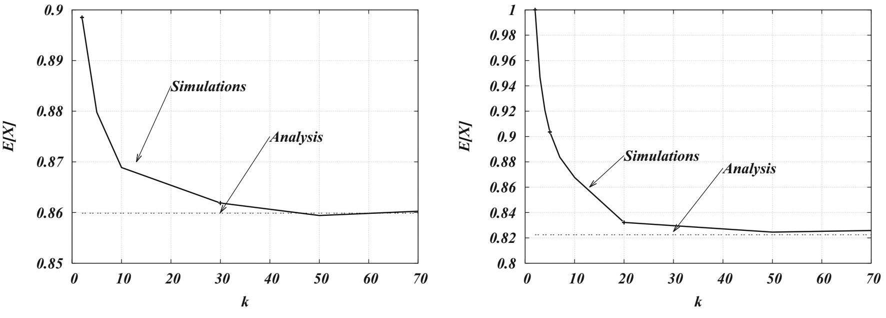
Let us call t^* the solution of this transcendental equation. By introducing the Lambert W function, we can express t^* in closed form as $t^* = \frac{W(\frac{p}{1-p})}{p}$.

Finally, the average number of messages $E[X]$ needed to decode the target message can be computed:

$$E[X] = \int_0^{t^*} s(\tau)d\tau = \frac{1 - e^{-W\left(\frac{p}{1-p}\right)}}{p} \tag{6}$$

This expression is minimized when $p = 0.626412$, and yields a minimum (normalized) number of retrieved messages $E[X] = 0.859884$. □

In order to verify the correctness of the analysis, Fig. 2a shows that simulations vary the number of messages from k = 2 to k = 70. Note that the theoretical results have an asymptotic nature, hence our choice of running simulations with small



(a) (b)

Fig. 2 Average retrieval delay varying the number of messages: mean field approximation vs simulation. (**a**) All-or-Nothing. (**b**) Pairs

values of k. Every point in the figure is the delay to retrieve a data message averaged on 50,000 samples. Even though the proposed methodology obtains an exact solution only for large values of k, already after $\mathrm{k} = 20$ the error is below 1 %.

4.2 Pairs-Only Scheme

This scheme shows how the state space can become extremely complex (actually an infinite set of state variables) even when considering an apparently very simple approach. Moreover, it can be solved using an alternative methodology, because its emerging decoding structure can be cast as an Erdős-Rényi random graph; thus it permits us to verify that our methodology, despite being extended to the case of infinite state variables (hence violating the assumptions in [3]), nevertheless yields the same results derived in the relevant random graph literature.

As the name suggests, the *pairs-only* scheme includes only one type of coded message, namely the random linear combination of two randomly chosen messages. This type of message is called *pair* and is formally defined as follows.

Definition 9 A *pair* is a random linear combination of two randomly chosen messages over a large field size in the form $\{(\alpha x_i + \beta x_j) | i \neq j$ and $i, j \in [1, k]\}$ where $\alpha, \beta \in \mathbb{F}$ and $\mathbb{F}$ is a large field.

In analyzing this scheme, the difficulty lies in defining an appropriate state space. Once this is done, the remaining analysis reduces to the conceptually straightforward application of our methodology. The state space definition and justification is presented in Appendix 2, along with the proof of the following theorem:

Theorem 10 *The* pairs-only *scheme achieves a performance of* $\frac{\pi^2}{12}k \approx 0.8224k$.

Our results confirm those found in random graphs literature. However, our approach can be extended to coding schemes which cannot be directly cast as a random graph problem, such as, the combination of singletons and pairs, which yields a performance slightly below $0.8k$ (we postpone analysis to a later extended version of this work). Comparison with simulation results averaged over 50.000 realizations is reported in Fig. 2b. Again, results show that convergence to the asymptotic result is very fast, with an error lower than 1 % for $k > 20$.

5 Conclusion

In this work we explore efficient solutions to *one-out-of-k retrieval*. We prove a lower bound of $0.5k$ and upper bound of $0.8224k$ on the number of coded messages needed on average to solve for the message of interest. Current simulation results suggest that the true minimum value for *one-out-of-k retrieval* should be higher than $0.5k$. The machinery given in Sect. 3 can be used to analyze various proposed schemes to produce upper bounds. Generalizing *one-out-of-k retrieval* to *m-out-of-k retrieval* is another interesting extension.

Acknowledgements This research is supported by NSF award CCF-1217506 and by the Israel Science Foundation (grant number 1696/14). Keren Censor-Hillel is a Shalon Fellow.

Appendix 1

Let's assume a discrete time scale, clocked by message arrivals, i.e., time $n \in \{1, 2, \cdots\}$ is defined as the time of arrival of the n-th element. Let us now identify a model for the *receiver state*. This is a critical step (as will appear in the construction examples discussed later on), as the relation between receiver state and the different "types" of messages collected (and how many) is in general not trivial and specific for every scheme considered; For instance, the reception of two different "types" of coded message, say a linear combination of messages A and B (called "pair"), and an uncoded message A (called "singleton") yields the decoding of message B, and suggests to use as state variables the number of message "types" resulting *after* decoding, in this case the two singletons A and B, rather than the actually received message types (a pair an a singleton).

In most generality, the status of the receiver at an arbitrary discrete time n is summarized by means of a state vector:

$$\bar{\psi}(n) = \{\psi_1(n), \psi_2(n), \cdots\} \tag{7}$$

where $\psi_i(n)$ is defined as the number of messages of "type" i stored by the receiver at time n.

Under the assumption of independent random messages being retrieved at each time step, and appropriate choice of the space state, $\bar{\psi}(n)$ introduced in (7) is a discrete-time Markov chain. Let us now write the relevant time-dependent state transition probabilities as functions of the vector state components normalized with respect to k, i.e.:

$$P\left\{\bar{\psi}(n+1)|\bar{\psi}(n)\right\} = f_{\bar{\psi}(n+1)}\left(\frac{\bar{\psi}(n)}{k}\right) \tag{8}$$

The conditional expectation, namely the *drift* of the considered Markov chain, is readily given by the vector

$$E\left[\bar{\psi}(n+1) - \bar{\psi}(n)|\bar{\psi}(n)\right] = \sum_{\bar{v} \in \text{all states}} \left(\bar{v} - \bar{\psi}(n)\right) f_{\bar{v}}\left(\frac{\bar{\psi}(n)}{k}\right) = \bar{d}\left(\frac{\bar{\psi}(n)}{k}\right), \tag{9}$$

where we conveniently express the state vector components as normalized with respect to k. We now introduce a new stochastic process which is a *doubly-rescaled* version of (7) in terms of both state (normalized with respect to k, i.e., a *density process* [1]) as well as time (also normalized with respect to k, i.e. $t = n/k$):

$$\bar{\sigma}(t) = \frac{\bar{\psi}\,(t \cdot k)}{k}$$

The conditional expectation (9) is readily rewritten for the rescaled process as:

$$E\left[k \cdot \bar{\sigma}(t + 1/k) - k \cdot \bar{\sigma}(t) | \bar{\sigma}(t)\right] = \frac{E\left[\bar{\sigma}(t + 1/k) - \bar{\sigma}(t) | \bar{\sigma}(t)\right]}{1/k} = \bar{d}\,(\bar{\sigma}(t)) \tag{10}$$

For large k, and under quite general assumptions (it suffices the drift $\bar{d}(.)$ to be a Lipschitz vector field [3]), the density process $\bar{\sigma}(t)$ converges in probability to a deterministic trajectory, computed by solving the system of differential equations obtained by replacing the left side of equation (10) with the derivative $\bar{\sigma}'(t)$:

$$\bar{\sigma}'(t) = \bar{d}\,(\bar{\sigma}(t)) \tag{11}$$

at last, from the knowledge of $\bar{\sigma}(t)$, the average number of messages needed to decode the target message is readily computed.

In order to better clarify, we present a trivial example.

Let us consider the simplest possible case of all messages being uncoded (singletons). Note that the final result, i.e., k messages retrieved on average, could be trivially derived from straightforward direct arguments; however, in addition to show how the above methodology can be cast in practical cases, this derivation will also recur as building block for the constructions discussed next, which instead do not appear readily tractable with direct arguments.

The first step is to define a convenient state space. In this case, the obvious state variable is the number $S(n)$ of distinct singletons received at time n. The process $S(n)$ is a discrete time markov chain, with the only non null transition probabilities being $P\{S(n+1) = S(n)|S(n)\} = S(n)/k$ (probability that the new retrieved singleton message is already stored), and $P\{S(n+1) = S(n) + 1|S(n)\} = 1 - S(n)/k$ (probability that the retrieved message is a new one). Hence, the drift of the chain is given by $E\left[S(n+1) - S(n)|S(n)\right] = 1 - S(n)/k$.

The second step consists in rescaling the process, and write, for the resulting density process $s(t) = S(tk)/k$, the differential drift equation $s'(t) = 1 - s(t)$. Since, at time $t = 0$, no messages are received, the differential equation shall be solved with the initial condition $s(0) = 0$, which yields $s(t) = 1 - e^{t}$.

Finally, in order to derive the average number of messages needed to retrieve a randomly chosen target message, we note that $s(t)$ is the fraction of messages retrieved at time t, and hence can be interpreted as the cumulative probability distribution function of the random variable X representing the retrieval (rescaled) time. Thus, $E[X] = \int_0^\infty [1 - s(t)]\,dt = \int_0^\infty e^{t} dt = 1$. Rescaling back to the original discrete time scale, we get the final result of k average messages needed to retrieve the target one.

Appendix 2

To avoid an overly long presentation, we directly operate over re-scaled state variables, i.e., densities (the transformation from discrete state variables to densities being readily performed as in the example presented in Sect. 4).

Since pairs are selected at random, the tracking of all the possible combination of messages would yield state space explosion. To circumvent such issue, we resort to the following convenient definition of an infinite, but numerable, set of state variables $s_i(t)$, where

- $s_1(t)$ is the fraction of messages (normalized with respect to k) which, at (normalized) time t, do *not* belong to any so far received pair;
- $s_2(t)$ is the fraction of messages which are covered by one and only one pair;
- $s_3(t)$ is the fraction of messages which belong to a group of three messages "connected" by two pairs;
- And, in most generality, $s_i(t)$ is the fraction of messages which belong to a group of i messages "connected" by $i-1$ pairs.

For an illustrative example, assume the node has so far received the pairs AB, AC, AD, EF, FG, HI, and JK. According to our definition, we have 1 group of 4 "connected" messages (A, B, C, D), 1 group of three connected messages (E,F,G), two groups of two connected messages (H,I) and (J,K), and all remaining messages not yet covered by any pair. Being k the total number of distinct messages, the state representation for the above example would be: $\{s_1(t) = 1 - 11/k, s_2(t) = 4/k, s_3(t) = 3/k, s_4(t) = 4/k, s_5(t) = 0, \cdots\}$. Note that $s_i(t) \cdot k/i$ yields the number of groups having cardinality i.

Suppose now that a pair AI is received: as a result, the two groups (A,B,C,D) and (H,I) merge in a new group of cardinality 6. This corresponds to the transition to the following state: $\{s_1(t) = 1 - 11/k, s_2(t) = 2/k, s_3(t) = 3/k, s_4(t) = 0, s_5(t) = 0, s_6(t) = 6/k, \cdots\}$.

For $k \to \infty$, the probability that a pair arrives in an already formed group *of finite size* vanishes; as such, a state transition can occur only because two different groups are merged via a random pair arrival. We can thus write the drift differential equations as follows:

$$\begin{aligned}
s_1'(t) &= -2s_1(t) \\
s_2'(t) &= 2\left[-2s_2(t) + s_1(t)^2\right] \\
s_3'(t) &= 3\left[-2s_3(t) + s_1(t)s_2(t) + s_2(t)s_1(t)\right] \\
s_4'(t) &= 4\left[-2s_4(t) + s_1(t)s_3(t) + s_2(t)^2 + s_3(t)s_1(t)\right]
\end{aligned}$$

$$\begin{aligned}
\cdots &= \cdots \\
s_i'(t) &= i\left[-2s_i(t) + \sum_{j=1}^{i-1} s_j(t)s_{i-j}(t)\right] \\
\cdots &= \cdots
\end{aligned} \tag{12}$$

These equations are readily explained as follows. Let us first focus on the set of messages so far not yet covered by any pair, i.e. those accounted by the state variable $s_1(t)$. Let us also remark that, owing to the normalization, $s_1(t)$ *also* corresponds to the probability to pick one of such messages as a component of an arriving pair. A state transition involving s_1 thus comprises two possible cases: (i) with probability $s_1(t)^2$, an arriving pair removes two of such messages and add them to the group of non overlapping pairs, namely those accounted in the state variable s_2, or (ii) with probability $2s_1(t) \cdot (1 - s_1(t))$ only one of the messages is removed. This corresponds to a negative *drift* for the state variable $s_1(t)$ given by the *average* state variable decrement:

$$s_1'(t) = -2 \cdot s_1(t)^2 - 2s_1(t)(1 - s_1(t)) = -2s_1(t),$$

as stated by the first equation in the above system.

Let us now focus on the set of messages accounted by the state $s_2(t)$. We recall that these are messages covered by exactly one pair, only. On one side, $s_2(t)$ can increase, with the addition of two new messages, only when an arriving pair covers two messages belonging to the set s_1 (this occurs with probability $s_1(t)^2$ as discussed above). On the other side, it decreases of (i) four messages, whenever a new arriving pair "hits" two messages in the set s_2 (hence "connects" the two pre-existing pairs forming a 4-messages group, this event has probability $s_2(t)^2$), or (ii) connects one pair in s_2 with a message outside the set s_2, this event occurs with probability $2s_2(t) \cdot (1 - s_2(t))$. By averaging the resulting state variations, we obtain the second drift equation. The remaining equations are derived via identical considerations.

It only remains to solve this differential system, using as initial conditions $s_1(0) = 1$, $s_i(0) = 0, \forall i > 1$. This is a purely calculus problem, not anymore related to our specific modeling problem, which is addressed as follows. First, we note that equations can be solved recursively, starting from the top. The following set of solutions is readily obtained:

$$\begin{aligned}
s_1(t) &= e^{-2t} \\
s_2(t) &= 2e^{-4t}t \\
s_3(t) &= 6e^{-6t}t^2 \\
s_4(t) &= \frac{64}{3}e^{-8t}t^3
\end{aligned}$$

$$\begin{aligned} s_5(t) &= \frac{250}{3}e^{-10t}t^4 \\ s_6(t) &= \frac{1728}{5}e^{-12t}t^5 \\ &\cdots \end{aligned} \tag{13}$$

Where the general solution pattern can be easily determined, besides a multiplicative constant C_i, as:

$$s_i(t) = C_i e^{-2it}t^{i-1} \tag{14}$$

Given that we are interested in the sum of all the $s_i(t)$ we can easily recognize that:

$$\sum_{i=1}^{\infty} s_i(t) = t^{-1}\sum_{i=1}^{\infty} C_i \left(e^{2t}t^{-1}\right)^{-i} = t^{-1}C_z\left(e^{2t}t^{-1}\right) \tag{15}$$

Where $C_z(e^{2t}t^{-1})$ is the Z-Trasform of sequence C_i calculated in the point $e^{2t}t^{-1}$.

Combining Eqs. 12 and 14 we obtain:

$$C_i e^{-2it}t^{i-2}(i-1-2it) = -2iC_i e^{-2it}t^{i-1} + e^{-2it}t^{i-2}i\sum_{j=1}^{i-1} C_j C_{i-j}$$

That after algebraic simplifications becomes:

$$C_i(i-1) = i\sum_{j=1}^{i-1} C_j C_{i-j}$$

We can transform this equation using the Z-Transform on i, so that:

$$-C_z - zC_z' = -z(C_z^2)'$$

and finally:

$$C_z = 2zC_zC_z' - zC_z'$$

Solving the above differential equation in z we have:

$$C_z = -\frac{1}{2}W\left(-\frac{2e^{-P}}{z}\right) \tag{16}$$

where W is the Lambert Function and P is a constant.

If we antitrasform the last expression, and after that we setted the constant P (knowing that $C_1 = 1$), we have:

$$C_i = \frac{(2i)^{i-1}}{i!} \tag{17}$$

C_i assumes the following values: 1, 2, 6, 64/3, 250/3, 1728/5, 67228/45, 2097152/315, 1062882/35, 80000000/567 . . .

We can then express $s_i(t)$ as:

$$s_i(t) = \frac{(2i)^{i-1}}{i!} t^{i-1} E^{-2it}$$

And finally calculate the average delay as:

$$E[D] = \sum_{i=i}^{\infty} \int_{t=0}^{\infty} s_i(t) = \sum_{i=1}^{\infty} \frac{1}{2i^2} = \frac{\pi^2}{12}$$

This is equal to $0.822467k$.

If we analyse the timeline of decoding process, we can notice a sharp threshold (corresponding to the receiving of $k/2$ pairs) that separates a phase in which the decoding of messages is negligible by a phase in which it is significant. Indeed, this problem can be also modelled as a random graph with k vertices where we randomly add edges. As Erdos pointed out in its seminal paper [4], after that vertices reach a degree $c = 1$ there is the emerging of a "giant component" whose size in the supercritical part (i.e. $c > 1$) is $\sim y(c)k$ where y is the solution of $e^{-cy} = 1 - y$.

References

1. Buckley, F.M., Pollett, P.K., et al.: Limit theorems for discrete-time metapopulation models. Probab. Surv. **7**, 53–83 (2010)
2. Chen, L., Ho, T., Low, S., Chiang, M., Doyle, J.: Optimization based rate control for multi-cast with network coding. In: Proceedings of IEEE Infocom, Anchorage (2007)
3. Darlings, R.W.R., Norris, J.R.: Differential equation approximations for markov chains. Probab. Surv. **5**, 37–79 (2008)
4. Erdős, P., Rényi, A.: On the evolution of random graphs. In: Publication of the Mathematical Institute of the Hungarian Academy of Sciences, pp. 17–61 (1960)
5. Kurtz, T.G.: Solutions of ordinary differential equations as limits of pure jump markov processes. J. Appl. Probab. **7**(1), 49–58 (1970)
6. Luby, M.: Lt codes. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, pp. 271–280 (2002). http://dx.doi.org/10.1109/SFCS.2002.1181950, doi:10.1109/SFCS.2002.1181950
7. Sassatelli, L., Medard, M.: Inter-session network coding in delay-tolerant networks under spray-and-wait routing. Model. Optim. Mobile Ad Hoc Wirel. Netw. **10**, 103–110 (2012)

8. Widmer, J., Le Boudec, J.: Network coding for efficient communication in extreme networks. In: ACM SIGCOMM Workshop on Delay-Tolerant Networking, Philadelphia, pp. 284–291 (2005)
9. Yoon, S., Haas, Z.: Application of linear network coding in delay tolerant networks. IEEE Ubiquitous and Future Networks (ICUFN), Jeju Island, pp. 338–343 (2010)
10. Zhang, X., Neglia, G., Kurose, J., Towsley, D.: On the benefits of random linear coding for unicast applications in disruption tolerant networks. Model. Optim. Mobile Ad Hoc Wirel. Netw. **4**, 1–7 (2006)

On the Error-Correcting Radius of Folded Reed–Solomon Code Designs

Joschi Brauchle

Abstract A general formula for the error-correcting radius of linear-algebraic multivariate interpolation decoding of folded Reed–Solomon (FRS) codes is derived. Based on this result, an improved construction of FRS codes is motivated, which can be obtained by puncturing Parvaresh–Vardy codes. The proposed codes allow decoding for all rates, remove the structural loss in decoding radius of the original FRS design and maximize the fraction of correctable errors.

Keywords Decoding radius • Low-order folded Reed–Solomon Codes • Multivariate interpolation • Parvaresh–Vardy codes

1 Introduction

Decoding Reed–Solomon (RS) codes can be seen as reconstructing a message polynomial of limited degree from a set of noisy evaluation points. There exist a multitude of univariate and multivariate interpolation decoding (MID) algorithms solving this problem. The classical bounded minimum distance decoder of Berlekamp–Welch (BW) [11] can be interpreted [1, 6] as a starting point for most MID algorithms. For an RS code of rate R the BW decoder recovers a list-of-1 candidate polynomial at minimum Hamming distance from the received word up to a fraction of $(1 - R)/2$ errors. Extending this idea, Sudan in [9] allowed for a list of $l \geq 1$ candidate polynomials to be recovered up to a radius of $1 - \sqrt{2R}$ via bivariate interpolation. Guruswami–Sudan [3] increased the radius to $1 - \sqrt{R}$ by means of multiplicities of the interpolation points. Parvaresh–Vardy (PV) codes [8] improve upon this value using MID of multiple algebraically correlated polynomials.

Simple linear-algebraic decoding of ℓ-order PV codes allows to correct up to a fraction of $\ell/(\ell + 1)(1 - \ell R)$ errors, but with a strong rate limitation. Through a puncturing pattern, Guruswami–Rudra [2] deduced m-folded Reed–Solomon (FRS) codes from PV codes which are linear-algebraically decodable [10] up to a fraction of $s/(s + 1)(1 - mR/(m - s + 1))$ errors with a lesser rate restriction. By allowing

J. Brauchle (✉)
Institute for Communications Engineering, Technische Universität München, Munich, Germany
e-mail: joschi.brauchle@tum.de

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_7

higher degree interpolation and interpolation points with multiplicities (which shall not be considered here for the sake of simple linear-algebraic decoding), a fraction of $1 - (mR/(m - s + 1))^{s/(s+1)}$ errors may be corrected using these codes.

In this paper, a general formula for the decoding radius of linear-algebraic MID in terms of code and decoder parameters is derived, showing that FRS codes are not designed optimally. An improved version called "low-order" m-folded Reed–Solomon (LOFRS) codes with a decoding radius up to $m/(m + 1)(1 - R)$ is motivated by this result. The term LOFRS was coined in a recent paper by Guruswami–Wang [4], who present a similar design that we wish to explicitly acknowledge. The contribution of this paper is to present LOFRS codes from a different perspective and to illustrate their relationship to PV codes.

The paper is organized as follows. Section 2 introduces notation and defines RS, PV and FRS codes. In Section 3, a general formula for the decoding radius of linear-algebraic MID is derived and applied to RS, PV and FRS codes. Section 4 analyzes the parameters of this formula such that an optimal decoding radius is achieved. It is shown that FRS codes can not make use of these parameters, so an improved design for FRS codes is presented and evaluated. Section 5 concludes the paper.

2 Review of (Folded) Reed–Solomon and Parvaresh–Vardy Codes

Let $\mathbb{F}_q$ be a finite field of order q, and $\mathbb{F}_q^* := \mathbb{F}_q \backslash \{0\}$ its multiplicative group with generating element $\alpha \in \mathbb{F}_q^*$. A vector space of dimension ℓ over $\mathbb{F}_q$ is denoted by $\mathbb{F}_q^\ell$ and the ring of polynomials in indeterminate x with coefficients in $\mathbb{F}_q$ by $\mathbb{F}_q[x]$. Let $\mathbb{F}_q[x]_{<k} := \{f \in \mathbb{F}_q[x] : \deg f < k\}$ be the vector space of polynomials in $\mathbb{F}_q[x]$ of degree less than k. The set of integers $\{1, \ldots, n\} =: [1, n]$ and let $\mathscr{E}(j, i, \ell) := \{\alpha^j, \alpha^{j+i}, \ldots, \alpha^{j+i(\ell-1)}\}$ be an ordered set of ℓ distinct multiples of α^i, starting at α^j.

Definition 1 (Evaluation Map) The evaluation map $\mathrm{ev}_{\mathscr{E}(j,i,\ell)}: \mathbb{F}_q[x]_{<k} \to \mathbb{F}_q^\ell$ is defined as $f \mapsto \left(f\left(\alpha^j\right), f\left(\alpha^{j+i}\right), \ldots, f\left(\alpha^{j+(\ell-1)i}\right)\right)$.

Definition 2 (Reed–Solomon Code) Let $\mathscr{E}$ be an evaluation set of n distinct elements from $\mathbb{F}_q$ called *code locators*. A RS code of length $n = |\mathscr{E}|$ and dimension $k \in [1, n]$ is the image of all message polynomials $f(x) = f_0 + f_1 x + \cdots + f_{k-1} x^{k-1} \in \mathbb{F}_q[x]_{<k}$ under the evaluation map $\mathrm{ev}_{\mathscr{E}}$, i.e.,

$$\mathrm{RS}[q, \mathscr{E}, k] := \left\{(\mathrm{ev}_{\mathscr{E}}(f)) : \forall f \in \mathbb{F}_q[x]_{<k}\right\} \tag{1}$$

and rate $R_{\mathrm{RS}} = k/n =: R$. If $\mathscr{E} = \mathbb{F}_q^*$, $n = \left|\mathbb{F}_q^*\right| = q - 1$, the code is called *primitive*.

In this paper, all considered RS codes are primitive.

Parvaresh–Vardy Codes generalize RS codes by evaluating more than just one polynomial of degree less than k at a common set of evaluation points $\mathcal{E}$.

Definition 3 (General Parvaresh–Vardy Code) Let $e(x) \in \mathbb{F}_q[x]$ be an irreducible polynomial over $\mathbb{F}_q$ of degree $a \geq k$. Let $f_0(x) \in \mathbb{F}_q[x]_{<k}$ denote the message polynomial as for RS codes. Choose integers a, d, ℓ so that $f_i(x) := (f_{i-1}(x))^d \mod e(x) \in \mathbb{F}_q[x]_{<k}$. A PV code of dimension k is defined as all matrices in $\mathbb{F}_q^{\ell \times n}$ such that

$$\mathrm{PV}\big[q, \mathcal{E}, k, d, \ell, e(x)\big] := \left\{ \begin{pmatrix} \mathrm{ev}_{\mathcal{E}}(f_0) \\ \vdots \\ \mathrm{ev}_{\mathcal{E}}(f_{\ell-1}) \end{pmatrix} : \forall f_0 \in \mathbb{F}_q[x]_{<k} \right\}. \tag{2}$$

If $\ell = 1$, only the message polynomial $f_0 \in \mathbb{F}_q[x]_{<k}$ is used and PV codes (2) reduce to RS codes as in (1). For $\ell > 1$, additional algebraically correlated polynomials $f_i \in \mathbb{F}_q[x]_{<k}$ carrying no further data are evaluated for $i \in [1, \ell - 1]$, so the rate of PV codes is limited to $R_{\mathrm{PV}} = k/(\ell n) = R_{\mathrm{RS}}/\ell$.

In the following, the parameters of PV codes are restricted: Let $e(x) = x^{q-1} - \alpha$ and $d = q$, such that $f_i(x) = (f_{i-1}(x))^q \mod \left(x^{q-1} - \alpha\right) = f_{i-1}(\alpha x) = f_0(\alpha^i x)$, $i \in [1, \ell - 1]$, allowing for a simplified definition of PV codes:

Definition 4 (Simple Parvaresh–Vardy Code) For the choice of $\mathcal{E} = \mathbb{F}_q^*$, $e(x) = x^{q-1} - \alpha$ and $d = q$, the PV codeword symbols

$$\underline{c}_j = \left[\mathrm{ev}_{\alpha^j}(f), \mathrm{ev}_{\alpha^{j+1}}(f), \ldots, \mathrm{ev}_{\alpha^{j+\ell-1}}(f)\right]^T = \left[\mathrm{ev}_{\mathcal{E}(j,1,\ell)}(f)\right]^T \in \mathbb{F}_q^{\ell} \tag{3}$$

are based on evaluating a single polynomial $f \in \mathbb{F}_q[x]_{<k}$ at $\mathcal{E}(j, 1, \ell)$.

Note that simple PV codes use repeated evaluation points in their codewords. For example, the neighboring symbols

$$\begin{aligned} \underline{c}_j &= \left[\mathrm{ev}_{\mathcal{E}(j,1,\ell)}(f)\right]^T &&= \left[f\left(\alpha^j\right),\ f\left(\alpha^{j+1}\right), \ldots, f\left(\alpha^{j+\ell-1}\right) \qquad\qquad \right]^T \quad \text{and} \\ \underline{c}_{j+1} &= \left[\mathrm{ev}_{\mathcal{E}(j+1,1,\ell)}(f)\right]^T &&= \left[\qquad\qquad f\left(\alpha^{j+1}\right), \ldots, f\left(\alpha^{j+\ell-1}\right), f\left(\alpha^{j+\ell}\right) \right]^T \end{aligned}$$

have $\ell - 1$ evaluation points $\alpha^{j+1}, \ldots, \alpha^{j+\ell-1}$ in common. In general, every evaluation point is used in ℓ consecutive code symbols, therefore reducing the rate by a factor of $1/\ell$.

Folded Reed–Solomon codes avoid this rate loss by transmitting only codeword symbols at code locators $\alpha^{j\ell}$, $j \in [0, n/\ell - 1]$, eliminating repeated evaluation points. Hence, FRS codes are PV codes where symbols $\underline{c}_i$, $i \neq j\ell$, are *punctured*.

Definition 5 (Folded Reed–Solomon Code) Let the *folding parameter* $m \geq 1$ be an integer satisfying $m|n$ and α be a primitive element of $\mathbb{F}_q$. Choose $N = n/m$ disjoint ordered sets of evaluation points $\mathcal{E}(jm, 1, m) = \left\{\alpha^{jm}, \alpha^{jm+1}, \ldots, \alpha^{jm+m-1}\right\}$, $j \in [0, N - 1]$. An m-FRS code of dimension k and length N consists of symbols

$$\underline{c}_j = \left[f(\alpha^{jm}), \ldots, f(\alpha^{jm+m-1})\right]^T = \left[\mathrm{ev}_{\mathcal{E}(jm,1,m)}(f)\right]^T \in \mathbb{F}_q^m. \tag{4}$$

In case of $m = 1$, FRS codes reduce to RS codes. For $n = q - 1$ and $\bigcup_{j=0}^{N-1} E(jm, 1, m) = \mathbb{F}_q^*$, FRS codes are essentially primitive RS codes, where m consecutive RS symbols are grouped into vectors from $\mathbb{F}_q^m$. Due to this close relationship, the rate $R_{\mathrm{FRS}} = {}^k\!/\!_n = R_{\mathrm{RS}}$ and minimum distance $d_{\min} = n - k + 1$ of FRS and RS codes are identical.

3 Decoding Radius of Linear-Algebraic MID

This section reviews linear-algebraic MID of PV, RS and FRS codes and derives a general formula for an achievable decoding radius (fraction of correctable errors) τ in terms of code and decoder parameters. The restriction to linear-algebraic algorithms allows for a much simpler presentation, but naturally leads to a slightly reduced decoding radius as well as an exponential list-size of the decoder output. The former consequence is negligible for high rate codes of practical interest and mitigation of the latter is possible [2, Sec. 4], but outside the scope of this paper.

Let $s \in [1, \ell]$ be an integer and $(s + 1)$ the dimension of an interpolation point $(x, y_1, \ldots, y_s)$. Let $Q(x, y_1, \ldots, y_s) = Q_0(x) + Q_1(x)y_1 + \cdots + Q_s(x)y_s \in \mathbb{F}_q[x, y_1,\ldots, y_s]$ be an $(s + 1)$-variate interpolation polynomial of degree 1 in the indeterminates $y_1, \ldots, y_s$. The codeword length shall be denoted by N and the received matrix by $\mathbf{r} = (\underline{r}_0, \underline{r}_1, \ldots, \underline{r}_{N-1}) \in \left(\mathbb{F}_q^\ell\right)^N$, with symbols $\underline{r}_j = [r_{j\ell}, r_{j\ell+1}, \ldots, r_{j\ell+\ell-1}]^T \in \mathbb{F}_q^\ell$, for $j \in [0, N-1]$. Let the $w = (1, k-1, \ldots, k-1)$-weighted degree of a monomial $x^{d_0} y_1^{d_1} \cdots y_s^{d_s}$ be defined as $\deg_w\!\left(x^{d_0} y_1^{d_1} \cdots y_s^{d_s}\right) := d_0 + (k - 1)\sum_{i=1}^{s} d_i$. Consequently, $\deg_w(Q)$ is the w-weighted degree of its leading monomial under w-weighted lexicographic ordering. Let $\sigma \in [1, \ell]$ denote the step size between two consecutive interpolation points within the received vector $\mathbf{r}$. Let $\mathscr{I} \subseteq [0, n - 1]$ denote the set of indices $i \in \mathscr{I}$ of all $(s + 1)$-dimensional interpolation points $(\alpha^i, r_{\sigma i}, \ldots, r_{\sigma i+s-1})$ used by the decoding algorithm and let $I := |\mathscr{I}|$ be its cardinality.

Linear-algebraic MID consists of an interpolation step and a root-finding step: In the **interpolation step,** the decoder finds a nonzero $(s + 1)$-variate polynomial $Q \in \mathbb{F}_q[x, y_1,\ldots, y_s]$ of minimal $\deg_w(Q) = D$ and degree 1 in $y_1, \ldots, y_s$, satisfying

$$Q(\alpha^i, r_{\sigma i}, \ldots, r_{\sigma i+s-1}) = 0, \quad \forall i \in \mathscr{I}, \tag{5}$$

giving a total of I constraints on at most $(s + 1)(D + 1) - s(k - 1)$ coefficients of Q. If D is large enough, the resulting homogeneous linear system of (5) has a nonzero solution for Q. The minimal such w-weighted degree is given in terms of I and s as

$$D(I, s) := \left\lfloor \frac{I + s(k - 1)}{s + 1} \right\rfloor. \tag{6}$$

In the **root-finding step** a list of candidate message polynomials $f \in \mathbb{F}_q[x]_{<k}$ is recovered from $\mathbf{r}$. For the codes and algorithms considered in this paper, it suffices to find all y-roots $f_0 \in \mathbb{F}_q[x]_{<k}$ of Q, such that f_i, $i \in [0, s-1]$, satisfies

$$Q(x, f_0(x), f_1(x), \ldots, f_{s-1}(x)) = 0. \tag{7}$$

Denote by $E := |\{j \in [0, N-1] : \underline{r}_j \neq \underline{c}_j\}|$ the total number of received symbols in error. An interpolation point $(\alpha^i, r_{\sigma i}, \ldots, r_{\sigma i+s-1})$, $i \in \mathcal{I}$, is said to agree with a message polynomial $f_0 \in \mathbb{F}_q[x]_{<k}$ if $r_t = f_{t \bmod \ell}(\alpha^i)$, $t \in [\sigma i, \sigma i + s - 1]$.

Lemma 6 *The maximum number of interpolation points corrupted by a single received symbol error* $\underline{r}_j \neq \underline{c}_j$, $j \in [0, N-1]$, *is given by*

$$A(\mathcal{I}, \ell, s, \sigma) := \max_j \left|\left\{i \in \mathcal{I} : [\sigma i, \sigma i + s - 1] \cap [j\ell, j\ell + \ell - 1] \neq \emptyset\right\}\right|. \tag{8}$$

Proof An $(s+1)$-dimensional interpolation point $(\alpha^i, r_{\sigma i}, \ldots, r_{\sigma i+s-1})$, $i \in \mathcal{I}$, corresponds to indices $[\sigma i, \sigma i + s - 1]$ in the received vector $\mathbf{r}$. A received symbol $\underline{r}_j \in \mathbb{F}_q^\ell$ uses indices $[j\ell, j\ell + \ell - 1]$. The j-th symbol $\underline{r}_j$ affects the i-th interpolation point if and only if the intersection $[\sigma i, \sigma i + s - 1] \cap [j\ell, j\ell + \ell - 1]$ is not empty. □

Lemma 7 *An $(s+1)$-variate polynomial $Q \in \mathbb{F}_q[x, y_1, \ldots, y_s]$ of* $\deg_w(Q) = D(I, s)$ *satisfying (5) will satisfy (7) if the number of agreements* $I - EA(\mathcal{I}, \ell, s, \sigma) > D(I, s)$.

Proof According to the *Polynomial Factor Theorem* [7, Cor. X.1.4], the number of roots of a non-constant univariate polynomial $P(x) = Q(x, f_0(x), \ldots, f_{s-1}(x)) \in \mathbb{F}_q[x]_{\leq D(I,s)}$ cannot exceed its degree, implying $Q(x, f_0(x), f_1(x), \ldots, f_{s-1}(x)) \equiv 0$. □

Theorem 8 *In case (7) suffices to recover all candidate polynomials $f_0 \in \mathbb{F}_q[x]_{<k}$, a fraction of correctable errors τ is achievable if*

$$\tau \leq \left(\frac{s}{s+1}\right)\left(\frac{I-k}{A(\mathcal{I}, \ell, s, \sigma)N}\right). \tag{9}$$

Proof Combining (6) and Lemma 7, we can choose any

$$\tau \leq \frac{E}{N} < \frac{1}{A(\mathcal{I}, \ell, s, \sigma)N}\left\lceil \frac{s(I-k)+1}{s+1} \right\rceil. \qquad \square$$

The result of Theorem 8 is applied to the following codes and decoding algorithms:

(1) Decoding of RS Codes (BW Algorithm): RS codes use $\ell = 1$ message polynomial, $N = n$ symbols $\underline{c}_j \in \mathbb{F}_q$ and $\sigma = 1$. The decoder finds a bivariate polynomial $Q(x, y) = Q_0(x) + Q_1(x)y \in \mathbb{F}_q[x, y]$ of minimal $w = (1, k-1)$-weighted degree, passing through all $(s+1) = 2$-dimensional points (α^i, r_i), $i \in \mathcal{I} = [0, n-1]$. Due to (6), $\deg_w(Q) \geq D(n, 1) = \left\lfloor \frac{n+k-1}{2} \right\rfloor$ is needed to satisfy all $I = n$ conditions. A symbol error affects $A(\mathcal{I}, 1, 1, 1) = 1$

interpolation point. Theorem 8 guarantees a fraction of correctable errors

$$\tau_{\mathrm{BW}} = (1 - R)/2 \tag{10}$$

up to which the message polynomial $f \in \mathbb{F}_q[x]_{<k}$ can be recovered [6, Thm. 5.2.2].

(2) Decoding of PV Codes: A rate R_{PV} PV code uses $N = n$ symbols $\underline{c}_j \in \mathbb{F}_q^{\ell}$. An $(\ell + 1)$-variate polynomial $Q(x, y_1, \dots, y_\ell) = Q_0(x) + Q_1(x)y_1 + \dots + Q_\ell(x)y_\ell \in \mathbb{F}_q[x, y_1, \dots, y_\ell]$ of $w = (1, k-1, \dots, k-1)$-weighted degree is found, passing through $(\ell + 1)$-dimensional interpolation points $(\alpha^i, r_{i\ell}, \dots, r_{i\ell+\ell-1})$, for $i \in \mathscr{I} = [0, n-1]$.

Hence, $I = n$ and $\deg_w(Q) \geq D(n, \ell) = \lfloor \frac{n+\ell(k-1)}{\ell+1} \rfloor$. In case $\underline{r}_j \neq \underline{c}_j$, $A(\mathscr{I}, \ell, \ell, \ell) = 1$ due to $\sigma = \ell$. Theorem 8 states that an achievable fraction of correctable errors is

$$\tau_{\mathrm{PV}} = \ell/(\ell + 1)\,(1 - \ell R_{\mathrm{PV}}) \tag{11}$$

such that a list of message polynomials $f_0 \in \mathbb{F}_q[x]_{<k}$ can be recovered [8, Lem. 6–9].

(3) FRS Decoding Scheme A: FRS codes are punctured PV codes with $\ell = m$, $\sigma = 1$, symbols $\underline{c}_j \in \mathbb{F}_q^m$ and $N = {}^{n}/_{m}$. In the linear-algebraic decoding scheme by Vadhan [5, 10], $\mathscr{I}$ is chosen so that all points $(\alpha^i, r_i, \dots, r_{i+s-1})$ are strictly contained inside the received symbols, i.e., $i \in \bigcup_{j=0}^{N-1}[mj, mj + m - 1]$, see Fig. 1(left) for example with s = 3. Therefore, $I = N(m - s + 1)$ and $A(\mathscr{I}, m, s, 1) = m - s + 1$. An $(s+1)$-variate polynomial $Q \in \mathbb{F}_q[x, y_1, \dots, y_s]$ is found if $\deg_w(Q) \geq D(N(m - s + 1), s) = \left\lfloor \frac{N(m-s+1)+s(k-1)}{s+1} \right\rfloor$.
Due to [5, Lem. 6] and (9), we can achieve a fraction of correctable errors

$$\tau_{\mathrm{FRS_A}} = \frac{s}{s+1}\left(1 - \left(\frac{m}{m-s+1}\right)R\right) \quad \text{for } 0 \leq R \leq (m - s + 1)/m. \tag{12}$$

(4) FRS Decoding Scheme B: Another scheme suggested by Justesen [2, Sec. 3.2] uses all $I = n$ interpolation points, thus requiring $\deg_w(Q) \geq D(n, s) = \left\lfloor \frac{n+s(k-1)}{s+1} \right\rfloor$. Due to the FRS code design, the cost of increasing I is that interpolation points overlap into neighboring code symbols, see Fig. 1(right). A symbol error affects $A(\mathscr{I}, m, s, 1) = m + s - 1$, i.e., an extra $2(s - 1)$ points over Scheme A. An achievable decoding radius is

$$\tau_{\mathrm{FRS_B}} = \frac{s}{s+1}\left(\frac{m}{m+s-1}\right)(1 - R) \qquad \text{for } 0 \leq R \leq 1. \tag{13}$$

Note that $\tau_{\mathrm{FRS_B}} > \tau_{\mathrm{FRS_A}}$ if $R > (m - s + 1)/(2m)$.

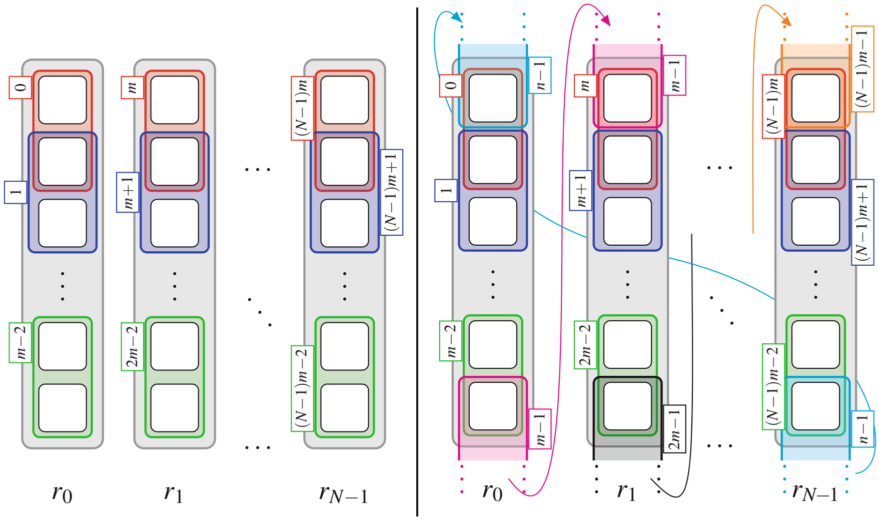


Fig. 1 FRS decoding schemes A (*left*) and B (*right*) using 3-variate interpolation. *White boxes* represent symbols from $\mathbb{F}_q$, grouped into interpolation points with code locator exponent in *square boxes*. *Gray boxes* denote received symbols $\underline{r}_j \in \mathbb{F}_q^m$, $j \in [0, N-1]$. *Dotted borders* with *arrows* depict overlapping interpolation points between neighboring symbols

4 Optimal Design of FRS Codes in Terms of Decoding Radius

According to Theorem 8, τ is a function of the interpolation parameter $s \in [1, m]$, the number of interpolation points $I \in [1, n]$ used, code length $N = {}^{n}/_{m}$ and maximum number of interpolation points $A(\mathscr{I}, m, s, \sigma) \in [{}^{I}/_{N}, m+s-1]$ affected by one symbol error. The range of parameters s and I is straightforward. The minimum value of $A(\mathscr{I}, m, s, \sigma)$ results from uniformly distributing I interpolation points among N code symbols. The maximum results from the maximum number of distinct $(s+1)$-dimensional points touching a symbol from $\mathbb{F}_q^m$. In order to maximize the decoding radius in (9), the optimal parameters are $s_{\text{opt}} = m$, $I_{\text{opt}} = n$ and $A_{\text{opt}} = m$, such that

$$\tau_{\text{opt}} = m/(m+1)(1-R). \tag{14}$$

Neither FRS decoding scheme A (using only $I = N(m-s+1) < I_{\text{opt}}$ interpolation points) nor scheme B (with $A(\mathscr{I}, m, s, 1) = m+s-1 > A_{\text{opt}}$ interpolation points affected by a symbol error) use these optimal values. In order to achieve the optimal parameters $I_{\text{opt}} = n$ and $A_{\text{opt}} = m$, FRS codes shall be adapted as follows: Based on FRS scheme B, the $2(s-1)$ transboundary $(s+1)$-dimensional interpolation points shall "wrap around" into the *same* code symbol instead of a neighboring one. This prevents crosstalk of erroneous interpolation points between neighboring symbols in case of symbol errors. The rate increase of an $\ell = m$-FRS over an ℓ-order PV code is due to the use of disjoint evaluation sets in (4), which shall be retained.

Definition 9 (Low-Order FRS Code) Let the *folding parameter* $m \geq 1$ be such that $m|n$ for $n = q - 1$ and let α be a primitive element of $\mathbb{F}_q$. Choose $N = {}^{n}\!/_{m}$ disjoint sets of evaluation points $\mathscr{E}(j, N, m) = \left\{\alpha^j, \alpha^{j+N}, \alpha^{j+2N}, \dots, \alpha^{j+n-N}\right\}$, $j \in [0, N-1]$ such that $\bigcup_{j=0}^{N-1} \mathscr{E}(j, N, m) = \mathbb{F}_q^*$. Note that α^N is an element of low order. A *low-order* m-FRS (LOFRS) code of dimension k and length N consists of symbols

$$\underline{c}_j = \left[f(\alpha^j), f(\alpha^{j+N}), \dots, f(\alpha^{j+n-N})\right]^T = \left[\mathrm{ev}_{\mathscr{E}(j,N,m)}(f)\right]^T \in \mathbb{F}_q^m. \tag{15}$$

For $m = 1$, LOFRS codes reduce to RS codes.

As for regular m-FRS codes, the decoder finds an $(s+1)$-variate interpolation polynomial $Q \in \mathbb{F}_q[x, y_1, \dots, y_s]$ passing through all $I = n$ interpolation points, i.e., satisfying (5) for $i \in \mathscr{I} = \bigcup_{j=0}^{N-1}\{j + N[0, m-1]\}$. Using (6), a $w = (1, k-1, \dots, k-1)$-weighted degree $\deg_w(Q) \geq D(n, s) = \left\lfloor \frac{n+s(k-1)}{s+1} \right\rfloor$ is required. Due to the choice of $\mathscr{I}$ and $\sigma = 1$, inter-symbol crosstalk of interpolation points is avoided in case of $\underline{r}_j \neq \underline{c}_j$ and so $A(\mathscr{I}, m, s, 1) = m = A_{\mathrm{opt}}$, see Fig. 2 for $(s+1) = 3$. In the root-finding step, a list of message polynomials $f \in \mathbb{F}_q[x]_{<k}$ is recovered [4, Prop. 5.3] from $Q(x, f(x), f(\alpha^N x), \dots, f(\alpha^{(s-1)N})) = 0$ if the

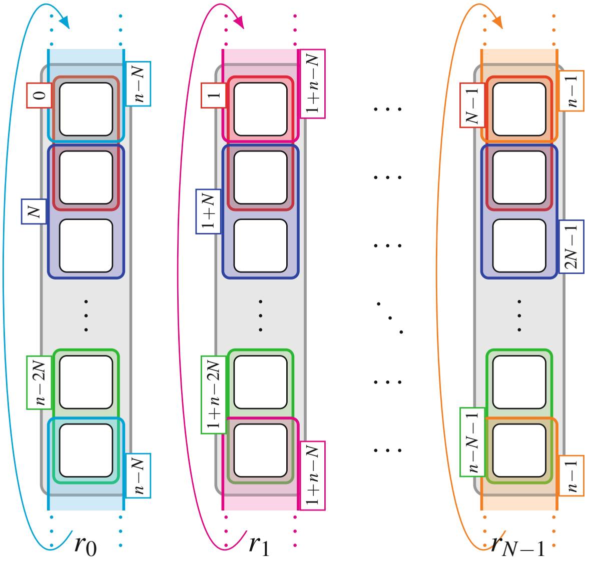


Fig. 2 MID of m-LOFRS code using 3-dimensional interpolation points. *Dotted boxes* with *arrows* depict interpolation points wrapping around into the *same* codeword symbol

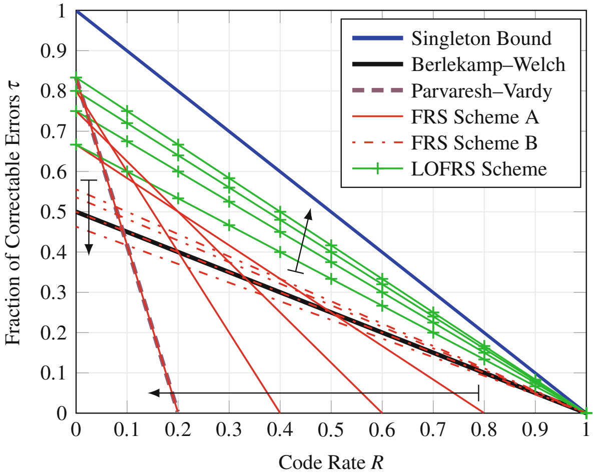


Fig. 3 Linear-algebraic MID radius τ versus code rate R of $\ell = 5$ PV codes, $m = 5$-FRS decoding schemes A and B, and proposed $m = 5$-LOFRS codes for parameter $s \in [2, m]$ (increasing along *arrows*)

agreement between $\mathbf{r}$ and f is larger than $D(n, s)$. Hence, it is possible to achieve a fraction of correctable errors

$$\tau_{\text{LOFRS}} = s/(s+1)\,(1-R) > \max\{\tau_{\text{FRS}_\text{A}}, \tau_{\text{FRS}_\text{B}}\} \qquad \text{for } 0 \leq R \leq 1. \tag{16}$$

By choosing the maximum interpolation parameter $s_{\text{opt}} = m$, the optimal decoding radius $\tau_{\text{LOFRS}} = \tau_{\text{opt}}$ is reached. In contrast, for FRS schemes A and B we have

$$\tau_{\text{FRS}_\text{A}} = m/(m+1)(1-mR) \qquad = \tau_{\text{PV}} \tag{17}$$

$$\tau_{\text{FRS}_\text{B}} = m/(m+1)(m)/(2m-1)(1-R) \approx \tau_{\text{BW}}. \tag{18}$$

Figure 3 compares the decoding radius of the BW algorithm for RS codes (10), linear-algebraic decoding of order $\ell = 5$ PV codes (11), $m = 5$-FRS decoding scheme A (12), decoding scheme B (13) and $m = 5$-LOFRS codes (16) for $s \in [2, m]$.

5 Conclusion

A general formula for the error-correcting radius of linear-algebraic MID of RS, PV and FRS codes was derived and analyzed. Through this formula, an improved design of m-FRS codes called *Low-Order m-FRS* codes was motivated, which was recently introduced by [4]. The proposed codes can be viewed in the context of punctured PV codes and they reach the optimal decoding radius $\tau_{\mathrm{opt}} = m/(m+1)(1-R)$.

Acknowledgements The author is supported by the German Ministry of Education and Research in the framework of the Alexander von Humboldt-Professorship and thanks G. Kramer, F. Kschischang, and V. Sidorenko as well as C. Senger, H. Bartz for their comments and discussions.

References

1. Gemmell, P., Sudan, M.: Highly resilient correctors for polynomials. Inform. Process. Lett. **43**(4), 169–174 (1992)
2. Guruswami, V., Rudra, A.: Explicit codes achieving list decoding capacity: error-correction with optimal redundancy. IEEE Trans. Inf. Theory **54**(1), 135–150 (2008)
3. Guruswami, V., Sudan, M.: Improved decoding of Reed–Solomon and algebraic-geometry codes. IEEE Trans. Inf. Theory **45**(6), 1757–1767 (1999)
4. Guruswami, V., Wang, C.: Explicit rank-metric codes list-decodable with optimal redundancy. CoRR abs/1311.7084 (2013)
5. Guruswami, V., Wang, C.: Linear-algebraic list decoding for variants of Reed–Solomon codes. IEEE Trans. Inf. Theory **59**(6), 3257–3268 (2013)
6. Justesen, J., Høholdt, T.: A course in error-correcting codes. Eur. Math. Soc. Zürich (2004). doi:10.4171/001
7. Lang, S.: Algebra. Graduate Texts in Mathematics, vol. 211, 3rd edn. Springer, New York (2002)
8. Parvaresh, F., Vardy, A.: Correcting errors beyond the Guruswami–Sudan radius in polynomial time. In: Proc. 46th IEEE Symp.Found.Comp.Sci.Pittsburgh, pp. 285–294 (2005)
9. Sudan, M.: Decoding of Reed–Solomon codes beyond the error-correction bound. J. Complex. **13**(1), 180–193 (1997)
10. Vadhan, S.: Pseudorandomness. Found. Trends Theor. Comput. Sci. **7**, 1–336 (2011)
11. Welch, L., Berlekamp, E.: Error correction for algebraic block codes. US Patent 4,633,470 (1986)

SPC Product Codes over the Erasure Channel

Sara D. Cardell and Joan-Josep Climent

Abstract SPC product codes are suitable for recovering lost symbols over erasure channels. These codes have a small minimum distance. However, they are capable of recovering a high number of erasures in some special cases, so the error correcting capability is higher than the minimum distance. In this work, we count the number of possible patterns that are uncorrectable when a number of erasures (up to 8) have occurred.

Keywords Erasure channel • Product code • Single parity-check code

1 Introduction

The erasure channel was introduced by Elias [2]. It is a communication channel where each sent symbol is either correctly received or considered as erased. In this model, each codeword symbol is lost with a fixed independent probability and an $[n, k, d]$-code can recover up to $d-1$ erasures. Given a fixed redundancy, maximum distance separable (MDS) codes (i.e., codes with $d - 1 = n - k$) are often the best adapted codes, since they offer maximal reliability.

The single parity-check (SPC) code is a very popular error detection code, since it is very easy to implement [1, 3]. The encoder appends 1 bit to an information sequence of $n-1$ bits, such that the resultant codeword has an even number of ones. Two or more SPC codes can be used jointly to obtain an SPC product code. Product codes are powerful codes that can be used to correct errors or recover erasures. SPC product codes have been proposed for applications such as cell loss recovery in ATM networks [4, 7], since they achieve a good performance under various decoding schemes [6]. The simplest form of an SPC product code is that where every row and

S.D. Cardell • J.-J. Climent (✉)
Departament d'Estadística i Investigació Operativa, Universitat d'Alacant, Ap. Correus 99, E-03080 Alacant, Spain
e-mail: s.diaz@ua.es; jcliment@ua.es

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_8

every column is terminated by a single parity bit. This code has four as minimum distance and is thus guaranteed to recover all erasure patterns with one, two and three erasures. However, the code is capable of recovering many higher erasure patterns, so the error correcting capability is higher than the minimum distance [1, 3]. It can be proven that, in some cases, up to $2n - 1$ erasures can be corrected. In [1, 3], the authors tried to count the number of patterns that cannot be corrected when no more than $2n - 1$ erasures have occurred. They obtained an upper bound for the number of uncorrectable configurations when 6 erasures have occurred. In this work we count the number of uncorrectable configurations for t erasures, with $t = 4, 5, 6, 7, 8$.

2 Preliminaries

Let $\mathbb{F}_q$ be the Galois field with q elements. A linear product code $\mathscr{C}$ over $\mathbb{F}_q$ is formed from two other linear codes $\mathscr{C}_h$ and $\mathscr{C}_v$ with parameters $[n_h, k_h, d_h]$ and $[n_v, k_v, d_v]$ over $\mathbb{F}_q$, respectively. The product code $\mathscr{C}$ will have parameters $[n_h n_v, k_h k_v, d_h d_v]$ over $\mathbb{F}_q$ (see [6]). The codewords of length $n_h n_v$ can be seen as arrays with size $n_v \times n_h$ in a way that the columns are codewords of $\mathscr{C}_v$ and the rows are codewords of $\mathscr{C}_h$. Over the erasure channel, the product code corrects up to $d_h d_v - 1$ erasures. However, we know that in some cases these codes can correct more erasures.

Let G_h and G_v be the generator matrices of the codes $\mathscr{C}_h$ and $\mathscr{C}_v$, respectively. The generator matrix G of the code $\mathscr{C}$ can be constructed by taking the Kronecker product of the matrices G_h and G_v, that is, $G_h \otimes G_v$ [5, page 569]. A codeword C in the product code can be generated either by multiplying a q-ary vector with size $k_h k_v$ by G or by using the expression $C = (G_v)^T U G_h$, where U is a $k_v \times k_h$ q-ary matrix. Note that the codeword C is an $n_v \times n_h$ q-ary matrix.

In this work, we consider the product code $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$, where $\mathscr{C}_h = \mathscr{C}_v$ is a linear binary code with parameters $[n, n-1, 2]$, which is called a **single parity-check (SPC) code**. In this case, the parameters of the product code are $[n^2, (n-1)^2, 4]$. Here, $\mathscr{C}_h$ and $\mathscr{C}_v$ correct only one erasure. Since the minimum distance is 4, the code $\mathscr{C}$ corrects only 3 erasures, but in some special cases the code can correct more than 3 erasures.

Now, we introduce the definition of erasure pattern. An **erasure pattern of size** $m \times m$, with t erasures, where $0 \le t \le m^2$ and $1 \le m \le n$, is an array of size $m \times m$ where t of the entries correspond to the position of the erasures.

An erasure pattern of size $n \times n$ corresponds to a codeword of size $n \times n$, where the position of the erasures is the unique information we consider. An erasure pattern is said to be uncorrectable if and only if it contains a subpattern such that each row and each column have two or more erasures. Given a codeword with t erasures, the

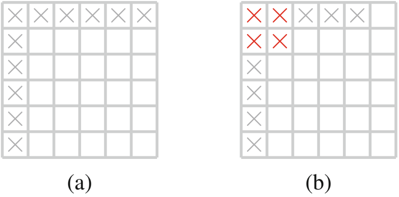


Fig. 1 Examples of erasure patterns of size 6×6 with 11 erasures. (**a**) Correctable erasure pattern of size 6×6 with 11 erasures. (**b**) Uncorrectable erasure pattern of size 6×6 with 11 erasures

decoder will perform iterative row-wise and column-wise decoding to recover the erased bits [1]. When a single bit is erased in a row or column, it can be recovered. If more than 1 bit is erased in a row (column), it is skipped. Decoding is performed until no further recovery is possible. Erasure patterns with more than three erasures may or may not be correctable.

Example 1 Consider the SPC code $\hat{\mathscr{C}}$ with parameters $[6, 5, 2]$. We can construct the binary product code $\mathscr{C} = \hat{\mathscr{C}} \otimes \hat{\mathscr{C}}$ with parameters $[36, 25, 4]$. In principle, we can only correct up to three erasures. Consider the erasure pattern in Fig. 1a. Since every column is a codeword of $\hat{\mathscr{C}}$, we can correct columns with one erasure, that is, erasures in every column but the first one. On the other hand, every row is a codeword of $\hat{\mathscr{C}}$ as well, so we can correct rows with one erasure and, then, we can correct completely this erasure pattern.

On the other hand, consider the erasure pattern in Fig. 1b. We can only correct seven erasures. The erasures in red cannot be corrected. □

For a codeword of size $n \times n$, erasure patterns with 3 erasures or less are always correctable. We would like to count the number of possible correctable and uncorrectable erasure patterns with t erasures, where $t \geq 4$. In this paper we consider the cases $t = 4, 5, 6, 7, 8$. A first approximation for the cases $t = 4, 5, 6$ can be found in [1, 3].

3 Counting Patterns

Assume we have a codeword of size $n \times n$ and that 4 erasures have occurred. The only uncorrectable erasure pattern of 4 erasures is formed by a square. We have $\binom{n}{2}^2$ possibilities, since we only have to choose two columns and two rows. As a consequence, the number of uncorrectable erasure patterns with 4 erasures is given by $\binom{n}{2}^2$.

In order to count the erasure patterns with 5 erasures, we just have to count the erasure patterns with four erasures and add one more. Therefore, the number of uncorrectable erasure patterns with 5 erasures is given by $\binom{n}{2}^2\binom{n^2-4}{1}$.

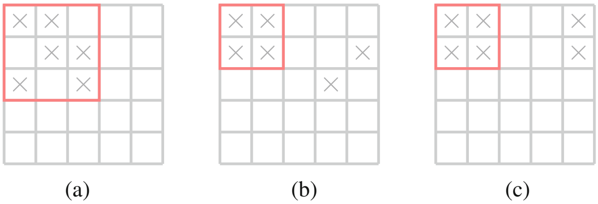


Fig. 2 Examples of uncorrectable erasure patterns with 6 erasures

It is easy to count the possible uncorrectable patterns with 6 erasures. We only have three possibilities (see Fig. 2). The case considered in Fig. 2a is easier. We select three columns and three rows and count the possible uncorrectable erasure patterns with 6 erasures of size 3 × 3, that is, $6\binom{n}{3}^2$. For the cases shown in Fig. 2b, c, one tends to count $\binom{n}{2}^2\binom{n^2-4}{2}$, that is, counting the uncorrectable erasure patterns of size 2 × 2 and adding two extra erasures. In this case, the erasure patterns with the form in Fig. 2c are considered three times. We are counting more erasure patterns than there exist. Then, what we do is counting how many of these erasure patterns there are and we subtract it twice from the total quantity (since we know they are counted three times): $\binom{n}{2}^2\binom{n^2-4}{2} - 2\left[2\binom{n}{2}\binom{n}{3}\right]$. Therefore, the total number of uncorrectable erasure patterns with 6 erasures is $\binom{n}{2}^2\binom{n^2-4}{2} - 4\binom{n}{2}\binom{n}{3} + 6\binom{n}{3}^2$.

Following the same counting method, we obtain the uncorrectable erasure patterns for 7 and 8 erasures.

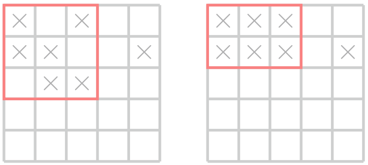
Theorem 2 *The number of uncorrectable erasure patterns with 7 erasures is given by $T_1 + T_2 + T_3 + T_4$, where*

$$T_1 = 6\binom{n}{3}^2\binom{n^2-9}{1}, \qquad T_2 = 2\binom{n}{2}\binom{n}{3}\binom{n^2-6}{1},$$

$$T_3 = 2\binom{n}{2}^2\left[8\binom{n-2}{3} + 4\binom{n-2}{2}\binom{(n-2)^2}{1} + \right.$$
$$\left. +2\binom{n-2}{1}\binom{(n-2)^2}{2} + 4\binom{n-2}{2}\binom{2(n-2)}{1}\right],$$

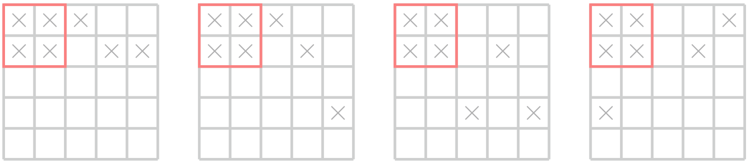
$$T_4 = \binom{n}{2}^2\left[2\binom{n-2}{1}^2 + 4\binom{n-2}{1}^2\binom{(n-2)^2-1}{1} + \binom{(n-2)^2}{3}\right].$$

Proof We consider all the possible uncorrectable patterns of size $n \times n$ with 7 erasures. To illustrate this proof, examples for all the possible uncorrectable patterns of size 5 × 5 with 7 erasures are given in Figs. 3–5.



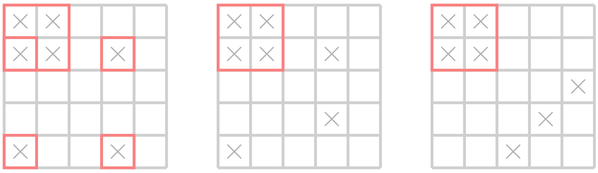
(a) (b)

Fig. 3 Examples of erasure patterns with 7 erasures corresponding to T_1 and T_2



(a) (b) (c) (d)

Fig. 4 Examples of uncorrectable erasure patterns with 7 erasures corresponding to T_3



(a) (b) (c)

Fig. 5 Examples of uncorrectable erasure patterns with 7 erasures corresponding to T_4

We start considering patterns with a complete uncorrectable subpattern of size 3×3 with 6 erasures and an additional erasure not included in this subpattern (see Fig. 3a). We take three columns and three rows; there are six ways to place the corresponding 6 erasures: $6\binom{n}{3}^2$. On the other hand, we have $n^2 - 9$ free places to put the remaining erasure: $\binom{n^2-9}{1}$. Then, the number of uncorrectable erasure patterns with this form is $T_1 = 6\binom{n}{3}^2\binom{n^2-9}{1}$.

We consider now patterns with an uncorrectable subpattern of size 2×3 with 6 erasures and an additional erasure (see Fig. 3b). We take two rows and three columns: $\binom{n}{2}\binom{n}{3}$. There are $n^2 - 6$ free places left to put the remaining erasure: $\binom{n^2-6}{1}$. Taking into account that we must also consider the case with subpatterns of size 3×2, the number of uncorrectable erasure patterns with this form is $T_2 = 2\binom{n}{2}\binom{n}{3}\binom{n^2-6}{1}$.

Consider now patterns shown in Figs. 4 and 5. These patterns are composed by a subpattern of size 2 × 2 with 4 erasures and 3 additional erasures. The only difference between these two groups is that patterns in Fig. 4 have to be considered twice (the patterns considered in this figure and the patterns obtained changing rows by columns).

For patterns in Fig. 4, we take first two columns and two rows: $\binom{n}{2}^2$ (for the subpattern). In Fig. 4a, from the remaining $n-2$ columns, we take three different columns, and we have to consider two different rows in each case: $2^3\binom{n-2}{3}$. In Fig. 4b, from the remaining $n-2$ columns, we take two different columns, and we have to consider two different rows in each case: $2^2\binom{n-2}{2}$. On the other hand, there are $(n-2)^2$ places to put the remaining erasure: $\binom{(n-2)^2}{1}$. In Fig. 4c, from the remaining $n-2$ columns, we take one, and we have to consider two different rows in each case: $2\binom{n-2}{1}$. Moreover, there are $(n-2)^2$ places to put the remaining two erasures, $\binom{(n-2)^2}{2}$. In Fig. 4d, from the remaining $n-2$ columns, we take two different columns, and we have to consider two different rows in each case: $2^2\binom{n-2}{2}$. Furthermore, there are $2(n-2)$ places to put the remaining erasure, so the total number is $2^2\binom{n-2}{2}\binom{2(n-2)}{1}$. Since we have to consider every case twice, we have that $T_3 = 2\binom{n}{2}^2\left[8\binom{n-2}{3} + 4\binom{n-2}{2}\binom{(n-2)^2}{1} + 2\binom{n-2}{1}\binom{(n-2)^2}{2} + 4\binom{n-2}{2}\binom{2(n-2)}{1}\right]$.

We consider now patterns in Fig. 5. In each case, we take two columns and two rows for the subpattern: $\binom{n}{2}^2$. In Fig. 5a, we take one column from the $n-2$ columns left and two rows: $2\binom{n-2}{1}$. The same happens for the other erasure that shares column with the subpattern: $2\binom{n-2}{1}$. There is one only possibility for the remaining erasure. Since, in this special case, we have two subpatterns of size 2 × 2 with 4 erasures, we are counting twice the number of subpatterns. Thus, we have to divide the total number by two: $2\binom{n-2}{1}^2$. In Fig. 5b, two of the erasures are considered in the same way as in the previous case: $2^2\binom{n-2}{1}^2$. For the third and last erasure, we have $(n-2)^2-1$ places where it can be considered: $\binom{(n-2)^2-1}{1}$. In Fig. 5c, none of the additional erasures shares column or row with the subpattern. Thus, there are $(n-2)^2$ possibilities to place these three erasures: $\binom{(n-2)^2}{3}$. As a consequence, we have that $T_4 = \binom{n}{2}^2\left[2\binom{n-2}{1}^2 + 4\binom{n-2}{1}^2\binom{(n-2)^2-1}{1} + \binom{(n-2)^2}{3}\right]$. □

Theorem 3 *The number of uncorrectable erasure patterns with* 8 *erasures is*

$$S_1 + S_2 + \left[\binom{n}{2}^2\binom{n^2-4}{4} - 5S_3 - 2S_4 - 4S_5 - S_6\right],$$

where

$$S_1 = 72\binom{n}{4}^2, \quad S_3 = 2\binom{n}{2}\binom{n}{4}, \quad S_5 = 9\binom{n}{3}^2,$$

$$\begin{aligned} S_2 = 6\binom{n}{3}^2 \Bigg[& \binom{(n-3)^2}{2} + \binom{3(n-3)}{1}^2 \\ & +2\binom{3(n-3)}{1}\binom{(n-3)^2}{1} + 2\binom{3}{1}^2\binom{n-3}{2} \Bigg], \end{aligned}$$

$$\begin{aligned} S_4 = 2\binom{n}{2}\binom{n}{3} \Bigg[& \binom{2(n-3)}{1}\binom{3(n-2)}{1} \\ & + \binom{2(n-3)}{1}\binom{(n-2)(n-3)}{1} + \binom{3(n-2)}{1}\binom{(n-2)(n-3)}{1} \\ & + \binom{(n-2)(n-3)}{2} + \binom{2}{1}^2\binom{n-3}{2} + \binom{3}{1}^2\binom{n-2}{2} \Bigg], \end{aligned}$$

$$S_6 = \frac{1}{2}\binom{n}{2}^2 \left[\binom{n-2}{2}^2 + 2\binom{2}{1}\binom{n-2}{2}\binom{n-2}{1} + \binom{2}{1}^2\binom{n-2}{1}^2\binom{n^2-9}{1} \right].$$

Due to the lack of space, this proof is not included. However, it follows the same idea used to prove Theorem 2. A complete counting process of patterns with 8 erasures has been performed. In Fig. 6, examples for the corresponding patterns considered for each group S_i, for $i = 1, 2, 3, 4, 5, 6$ are shown.

Finally, it is well-known that the number of erasure patterns of size $n \times n$ with t erasures is $\binom{n^2}{t}$. According to this number and the results given in Theorems 2 and 3 it is easy to check that the probability of finding an uncorrectable erasure pattern for $t = 7, 8$, is close to zero when n is large. Unfortunately, when the number of erasures grows, it becomes more difficult to count the number of possible uncorrectable patterns of size $n \times n$. We are working on an upper bound for t erasures, with $9 \leq t \leq 2n - 1$, that allows us to prove that the probability of finding an uncorrectable erasure pattern is close to 0 when n grows. This would prove that the probability of correcting a sent codeword when n is very large and $4 \leq t \leq 2n - 1$ is almost 1.

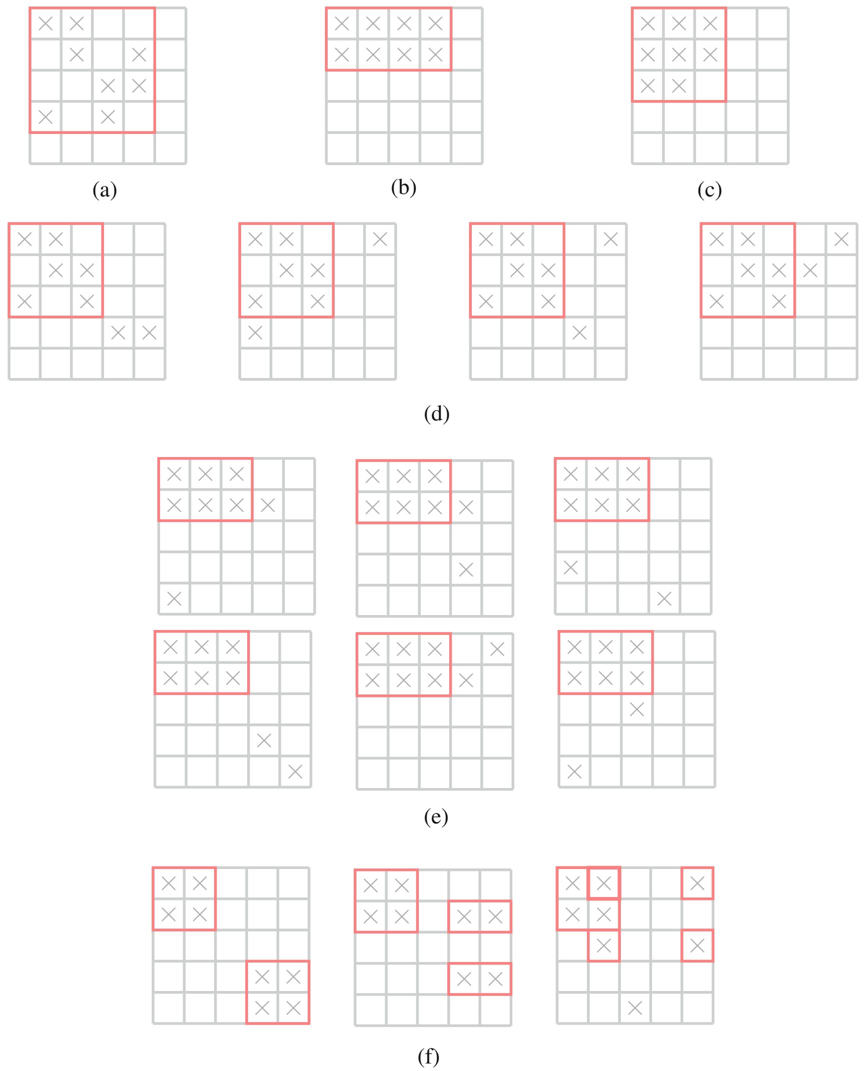


Fig. 6 Examples of uncorrectable erasure patterns with 8 erasures related to Theorem 3. (**a**) Uncorrectable erasure pattern corresponding to S_1. (**b**) Uncorrectable erasure pattern corresponding to S_3. (**c**) Uncorrectable erasure pattern corresponding to S_5. (**d**) Uncorrectable erasure patterns corresponding to S_2. (**e**) Uncorrectable erasure patterns corresponding to S_4. (**f**) Uncorrectable erasure patterns corresponding to S_6

Acknowledgements This work was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Ciencia e Innovación of the Gobierno de España. The work of the first author was also supported by a grant for postdoctoral students from the Generalitat Valenciana with reference APOSTD/2013/081.

References

1. Amutha, R., Verraraghavan, K., Srivatsa, S.K.: Recoverability study of SPC product codes under erasure decoding. Inf. Sci. **173**, 169–179 (2005). doi:10.1016/j.ins.2004.07.011
2. Elias, P.: Coding for noisy channels. In: IRE International Convention Record, part 4, pp. 37–46 (1955)
3. Kousa, M.A.: A novel approach for evaluating the performance of SPC product codes under erasure decoding. IEEE Trans. Commun. **50**(1), 7–11 (2002). doi:10.1109/26.975732
4. Kousa, M.A., Mugaibel, A.H.: Cell loss recovery using two-dimensional erasure correction for ATM networks. In: Proceedings of the Seventh International Conference on Telecommunication Systems, Nashville, pp. 85–89 (1999)
5. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes, 6th edn. North-Holland, Amsterdam (1988)
6. Rankin, D.M., Gulliver, T.A.: Single parity check product codes. IEEE Trans. Commun. **49**(8), 1354–1362 (2001). doi:10.1109/26.939851
7. Simmons, J.M., Gallager, R.G.: Design of error detection scheme for class C service in ATM. IEEE/ACM Trans. Netw. **2**(1), 80–88 (1994). doi:10.1109/90.282611

Complementary Dual Codes for Counter-Measures to Side-Channel Attacks

Claude Carlet and Sylvain Guilley

Abstract We recall why linear codes with complementary duals (LCD codes) play a role in counter-measures to passive and active side-channel analyses on embedded cryptosystems. The rate and the minimum distance of such LCD codes must be as large as possible. We investigate constructions.

Keywords Linear codes with complementary duals (LCD) • Cyclic codes • Bose, Ray-Chaudhuri and Hocquenghem (BCH) codes • Generalized residue codes

1 Introduction

Codes play a central role in digital communication. Recently, it has been shown that codes can also help improve the security of the information processed by sensitive devices, especially against so-called side-channel attacks (SCA) or fault non-invasive attacks. This paper recalls that linear codes with complementary duals (called LCD), which are linear codes with supplementary duals, play an important role in armoring implementations against these two kinds of non-invasive attacks.

LCD codes, introduced by Massey [10], provide an optimum linear coding solution for the two-user binary adder channel. Asymptotically good LCD codes exist [11]. Some constructions are known: [1, 6, 7]. As another example, maximum rank distance (MRD) codes generated by the trace-orthogonal-generator matrices are LCD codes [13].

C. Carlet (✉)
LAGA, UMR 7539, CNRS, Department of Mathematics, University of Paris XIII and University of Paris VIII, 2 rue de la liberté, 93 526 Saint-Denis Cedex, France
e-mail: claude.carlet@univ-paris8.fr; claude.carlet@gmail.com

S. Guilley (✉)
TELECOM-ParisTech, Crypto Group, 37/39 rue Dareau, 75 634 Paris Cedex 13, France

Secure-IC S.A.S., 80 avenue des Buttes de Coësmes, 35 700 Rennes, France
e-mail: sylvain.guilley@telecom-paristech.fr; sylvain.guilley@enst.fr

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_9

However, SCA sheds a new light on LCD codes and poses more accurately the question of their effective construction achieving good minimum distance, especially in the context of large rate.

2 Motivation

Implementations of cryptographic algorithms are prone to SCA and fault attacks that aim at extracting the secret key when the algorithm is running over some device. Non-invasive attacks observe some leakage (such as electromagnetic emanations) or perturb internal data (for example with electromagnetic impulses), without damaging the system. They are a special concern insofar as they leave no evidence that they have been perpetrated. Those attacks can be classified into two categories:

- Side-channel attacks (SCA), that consist in passively recording some leakage, that is the source of information to retrieve the key;
- Fault injection attacks (FIA), that consist in actively perturbing the computation so as to obtain exploitable differences at the output.

Few generic protections, demonstrably provable against both threats, have been proposed. The best understood and most studied protection against SCA is achieved with masking. Every sensitive data x, say a binary vector, employed in the cryptographic algorithm is exclusived-or with one uniformly distributed random vector of the same length, called mask. We are interested in this article in a *homomorphic* computation. This means that the computations are carried out on the masked data itself. Therefore, it must be possible, from a masked sensitive variable, denoted by z, to recover x (e.g., for the final demasking at the end of the computation). This is possible if the sensitive data and the masks belong to two supplementary subspaces of a larger space vector. Indeed, by definition of supplementary subspaces, any element of the large space vector decomposes itself in a unique way as the sum of two elements (in Boolean vector spaces, the sum is the exclusive-or, denoted by "+" in the sequel). It is thus decided to interpret those two elements as the sensitive data and the mask. This method is called *Orthogonal Direct Sum Masking* (ODSM), see [3].

We call n the dimension of this large vector space, which practically is $\mathbb{F}_2^n$. Now, we call C and D the two supplementary vector spaces:

$$\mathbb{F}_2^n = C \oplus D \ . \tag{1}$$

The masks are the codewords of code D. By the rank-nullity theorem, if the dimension of C is k, then the dimension of D is $n - k$. Let us consider generator matrices G and G' of C and D, respectively. Then every vector $z \in \mathbb{F}_2^n$ can be written in a unique way as $z = xG + yG'$, $x \in \mathbb{F}_2^k, y \in \mathbb{F}_2^{n-k}$. If C and D are furthermore orthogonal with respect to the usual inner product, i.e., $D = C^\perp$, then C is said complementary dual.[1]

Definition 1 A *linear* code C is called *complementary dual* (LCD) if C and $C^\perp$ are supplementary, that is (given their dimensions), $C \cap C^\perp = \{0\}$.

Note that $D = C^\perp$ if and only if G' is a parity-check matrix of C, that is, $GG'^\top = 0$, where $G'^\top$ is the transposed matrix of matrix G'; we denote then G' by H. We can use an orthogonal projection to recover x and y from z: the relation $z = xG + yH$ implies $zH^\top = yHH^\top$ and $zG^\top = xGG^\top$. The next characterization is due to Massey [10]:

Proposition 2 *Let C be a linear code. Let G be a generator matrix of C and H a parity-check matrix. Then the three following properties are equivalent:*

1. *C is LCD,*
2. *The matrix $HH^\top$ is invertible,*
3. *The matrix $GG^\top$ is invertible.*

We deduce from $zH^\top = yHH^\top$ and $zG^\top = xGG^\top$, and from Proposition 2 that if C is LCD, the matrices of the two projections $z = xG + yH \mapsto x$ and $z \mapsto y$ are respectively (see also [10, Proposition 1]):

$$G^\top (GG^\top)^{-1} \text{ so that } x = zG^\top (GG^\top)^{-1} \quad , \tag{2}$$

$$H^\top (HH^\top)^{-1} \text{ so that } y = zH^\top (HH^\top)^{-1} \quad . \tag{3}$$

The quality of the masking is an important factor. Let $\phi : \mathbb{F}_2^n \to \mathbb{R}$ be a leakage function, that describes how z is leaked outside of device. The masked word z conceals the information x at *first degree* if for all pseudo-Boolean function $\phi : \mathbb{F}_2^n \to \mathbb{R}$ of unitary numerical degree [4, Sec. 2.1], all the averages of $\phi(z)$ over the masks $d \in D$ for a given x are equal irrespective of x. This means that $\forall x \in \mathbb{F}_2^k, \sum_{y \in \mathbb{F}^{n-k}} \phi(xG + yH)$ are the same, i.e., equal to $\sum_{y \in \mathbb{F}^{n-k}} \phi(yH)$ (for $x = 0$). Now, this notion can be generalized (see [2, Def. 2]). A zero-offset masking countermeasure is of *degree at least d* if $\forall x \in \mathbb{F}_2^k, \sum_{y \in \mathbb{F}^{n-k}} \phi(xG + yH) = \sum_{y \in \mathbb{F}^{n-k}} \phi(yH)$ for all ϕ of numerical degree at most d. The greater the degree of the countermeasure, the harder to pass a successful SCA. Actually, it is known from [3, 5] that the countermeasure is $(d-1)$-th degree secure if D has dual distance d, i.e., if C has minimal distance d. This result has been independently validated in [8] for $d \in \{1, 2\}$.

Let us now consider a fault injection attack (FIA). The state z is modified into $z+\varepsilon$, for some random $\varepsilon \in \mathbb{F}_2^n$. By supplementarity of C and D, there exists a unique ordered pair $(e, f) \in \mathbb{F}_2^k \times \mathbb{F}_2^{n-k}$ such that $\varepsilon = eG + fH$. A detection strategy could consist in decoding z into (x, y), and checking that we recover the genuine values

[1] "supplementary" would seem more appropriate than "complementary", but the term is more than 10 year old.

unchanged. However, x is sensitive: the purpose of the protection is exactly to avoid representing x by replacing it by z. The random variable y, from its side, does not convey any (statistically) exploitable information. So, checking whether or not the mask has been altered, i.e., $zH^{\mathsf{T}}(HH^{\mathsf{T}})^{-1} \overset{?}{=} y$, is a harmless detection strategy. This happens if and only if $f = 0$, i.e., $\varepsilon \in C$. As $\varepsilon = 0$ is pointless (since without observable effect), harmful faults only happen if $\varepsilon \in C \setminus \{0\}$. In particular, the Hamming weight of ε must be greater or equal to the minimal distance d of code C for the fault not to be detected. Now, given that the minimal distance d of C is a design parameter, it is set as high as possible.

Therefore, have C be LCD of greatest possible minimal distance simultaneously improves the resistance against SCA and FIA.

There are two kinds of designs that can benefit from the described protection. The first one is the implementation of hardware accelerators for block ciphers, such as the AES. In this case, the data to protect are typically bytes, with $k = 8$. The second kind is a general-purpose processor executing software cryptography. Its registers can be protected individually (hence $k = 32$). For an improved security, it can be advantageous to mask all the registers seen as one unique resource, made up of a few hundreds to a few thousands bits. Therefore, we are interested in codes of various dimensions, ranging from $k = 8$ to $k \approx 4{,}096$.

The problem is thus the following: for a given dimension k (architecture parameter) and minimal distance d (security parameter), find a LCD code of length n as small as possible (and therefore, of *rate* k/n as large as possible).

3 Constructions

LCD cyclic codes, which have a minoration on their minimal distance via the BCH bound, have been characterized in [15]. A potentially stronger lower bound on the minimum distance exists for the sub-class of quadratic-residue (QR) codes [12, 14], which can also be LCD. A QR code has for length a prime number n and has a minimal distance d at least $\sqrt{n}$. A binary QR code has length congruent with ± 1 modulo 8 and is LCD if the length is congruent with 1 modulo 8 [9]. Asymptotically, $\sqrt{n}$ is a rather low value compared with the Gilbert Varshamov bound, but such value is not far from what we need in our framework. The main drawback of QR codes is that their dimension equals $\frac{n\pm 1}{2}$ (if we include 1 as possible zero of QR codes), while we need larger dimensions. This leads us to considering a generalization of QR codes whose lengths are not prime.

3.1 LCD Cyclic Codes

We shall always consider n co-prime with q. Let β be a primitive n-th root of unity. Let C be a cyclic code of zeros $\{\beta^j, j \in J \subseteq \mathbb{Z}/n\mathbb{Z}\}$. The BCH bound states that the minimum distance of C is bounded below by the length of any string of consecutive elements in J, plus 1. As observed in [15]:

Proposition 3 *A binary cyclic code is LCD if and only if its set of zeros is stable by the multiplicative inverse, i.e., if and only if its generator polynomial $g(X)$ is self-reciprocal.*

Example 4 The binary cyclic code of length 17 whose zeros are

$$\{\beta^j,\ j = 0, 1, 2, 4, 8, 9, 13, 15, 16\}$$

is LCD and has parameters $[17, 8, 6]$ and its generator polynomial is $X^9 + X^6 + X^5 + X^4 + X^3 + 1$. Note that the set of zeros is stable under the Frobenius $\gamma \mapsto \gamma^2$, which makes the code binary, and that the string 15, 16, 0, 1, 2 in $\mathbb{Z}/17\mathbb{Z}$ has length 5; the BCH bound is then tight for this code.

3.2 LCD Generalized Residue Codes

A background on quadratic residue codes can be found in [14], and on generalized residues codes in [12]. Let n be any integer co-prime with q and let t be any positive integer. Let Q be the set of t-th powers in $\mathbb{Z}/n\mathbb{Z}$:

$$Q = \{i^t,\ i \in \mathbb{Z}/n\mathbb{Z}\} \subseteq \mathbb{Z}/n\mathbb{Z}\ \ .$$

Then Q is stable under multiplication in the sense that, for any $s \in Q$, the mapping $r \in Q \mapsto sr$ is valued in Q (but, since n is not assumed to be a prime, the image set of this mapping may be strictly included in Q, since there exist divisors of zero[2] in $\mathbb{Z}/n\mathbb{Z}$): for every $i^t, j^t \in Q$, we have indeed $i^t j^t = (ij)^t$. Note that, since n is not assumed to be a prime, $\mathbb{Z}/n\mathbb{Z} \setminus Q$ may not be stable under this same mapping. Assume that q belongs to Q. Then Q is stable under multiplication by q, in the strong sense that the mapping $r \in Q \mapsto qr$ has image set Q, since q being co-prime with n, the multiplication by q is a permutation of $\mathbb{Z}/n\mathbb{Z}$. Note that, for the same reason, $Q^* = Q \setminus \{0\}$ is also stable under multiplication by q. Let C be the cyclic code of length n over $\mathbb{F}_{q^m}$ whose zeros are β^i, $i \in Q$ (resp. $i \in Q^*$, $i \in \mathbb{Z}/n\mathbb{Z} \setminus Q$, $i \in \mathbb{Z}/n\mathbb{Z} \setminus Q^*$). Then C is a code over $\mathbb{F}_q$ since its set of zeros is stable under the Frobenius automorphism. And if additionally $-1 \in Q$ (that is, $n - 1$, which is also co-prime with n), then Q is stable under multiplication by -1 in $\mathbb{Z}/n\mathbb{Z}$ and C is LCD. We deduce, since we are looking for binary codes:

Proposition 5 *Let n be an odd positive integer and t be any positive integer. Let Q be the set of t-th powers in $\mathbb{Z}/n\mathbb{Z}$. Assume that 2 and -1 both belong to Q. Then*

[2] For the same reason, we do not exclude $i = 0$ in the definition of Q above, contrary to the definition of Q when n is a prime, since even if $i \neq 0$ is imposed, 0 may belong to Q.

the cyclic code of length n whose zeros are β^i, $i \in Q$ (resp. $i \in Q^*$, $i \in \mathbb{Z}/n\mathbb{Z} \setminus Q$, $i \in \mathbb{Z}/n\mathbb{Z} \setminus Q^*$) where β is a primitive n-th root of unity in an extension field of $\mathbb{F}_q$, is a binary cyclic LCD code.

Note that it is easy to find integers n such that 2 and -1 are quadratic residues modulo n, as the common divisors of an integer of the form $r^2 - 2$ and of an integer of the form $s^2 + 1$. Since n is not assumed to be a prime, the size of Q may be strictly smaller than $\frac{n+1}{2}$ and the dimension $k = n - \mathsf{card}(Q)$ of the code may be larger than $\frac{n-1}{2}$.

Remark 6 *For classical QR codes, n is a prime number (and $\mathbb{Z}/n\mathbb{Z}$ is then a field) and $t = 2$. Given a nonzero codeword $f(X)$ of minimum weight d in the code C of zeros β^i, $i \in Q^*$, and j a non-residue, the polynomial $f(X^j)$ is a nonzero codeword in the code of zeros β^i, $i \in \mathbb{Z}/n\mathbb{Z} \setminus Q$, and $f(X)f(X^j)$ belongs then to the intersection of these two codes and is a multiple of $\sum_{i=0}^{n-1} x^i$ which has weight n. Then $d^2 \geq n$ (but since the size of Q^* equals $\frac{n-1}{2}$, the dimension of the code is $\frac{n\pm1}{2}$, which is too small for our purpose).*

3.3 Generating the Codes by the Use of Idempotents

The generator polynomial of a cyclic code C of length n may be complex to calculate, because this needs to calculate in the Galois extension of $\mathbb{F}_q$ containing a primitive n-th root of unity β. An alternative way is to use an idempotent as generator of the code (this method is well-known and specially simple for classical quadratic residue codes, see [9]). Let $g(X)$ be the generator polynomial of a cyclic code C. We have $X^n - 1 = g(X)h(X)$ where $h(X)$ is co-prime with $g(X)$ since n is odd (all zeros of $X^n - 1$ being then simple). Bezout's theorem implies then the existence of two polynomials $u(X), v(X)$ such that $g(X)u(X) + h(X)v(X) = 1$, which implies $(g(X)u(X))^2 = g(X)u(X)$ [mod $X^n - 1$]. Then $E(X) = g(X)u(X)$ is an idempotent in $\mathbb{F}_q[X]/(X^n - 1)$. If β^i is a zero of $u(X)$, then it is also a zero of $g(X)$ because it cannot be in the same time a zero of $u(X)$ and a zero of $h(X)$. We deduce that $E(\beta^i) = 0$ if and only if $g(\beta^i) = 0$ and $E(X)$ is also a generator of C. This holds, since E has no other zero, $E(X)$ is equal to the product modulo $X^n - 1$ of $g(X)$ by an invertible polynomial modulo $X^n - 1$. Using that $E(X)$ is an idempotent, we have that $f(X) \in C$ if and only if $f(X)E(X) = f(X)$. This implies that $E(X)$ is unique, since if another idempotent $F(X)$ exists in C, we have $F(X)E(X) = F(X) = E(X)$. Note that E applied to n-th roots of unity takes values in $\mathbb{F}_2$. The idempotent of $C^\perp$ equals the reciprocal of $1 + E(X)$. The characterization of cyclic LCD codes by Massey recalled above gives then the following characterization:

Proposition 7 *Let C be a cyclic code of length n over $\mathbb{F}_q$. Let $E(X)$ be the idempotent of C. Then C is LCD if and only if $E(X)$ is self-reciprocal, that is, if and only if the idempotent associated to $C^\perp$ is $1 + E(X)$.*

We consider now again the case of generalized residue codes. If $q = 2 \in Q$, where Q is the set of t-th powers in $\mathbb{Z}/n\mathbb{Z}$, then the polynomial $P(X) = \sum_{j \in Q} X^j$ satisfies $P^2(X) = \sum_{j \in Q} X^{2j} \equiv P(X)$ [mod $X^n - 1$] and is then an idempotent. For every t-th power residue r, we have $P(\beta^r) = \sum_{j \in Q} \beta^{rj}$. If r is co-prime with n then we deduce that $P(\beta^r) = \sum_{j \in Q} \beta^j = P(\beta) \in \mathbb{F}_2$. Hence:

Proposition 8 *Let n be an odd positive integer and t be any positive integer. Let Q be the set of t-th powers in $\mathbb{Z}/n\mathbb{Z}$. Assume that 2 belongs to Q. Let C be the binary cyclic code of length n over $\mathbb{F}_q$ whose zeros are β^i, $i \in Q^*$ where β is a primitive n-th root of unity in an extension field of $\mathbb{F}_q$. Let $P(X) = \sum_{j \in Q} X^j$. If every nonzero element in Q is co-prime with n, then $P(X)$ or $1 + P(X)$ is the idempotent of the code C.*

Note that adding $\beta^0 = 1$ to the zeros of the code corresponds to multiplying the idempotent by $(X - 1)$ to obtain a generator polynomial (which is not an idempotent).

We have now a simple way to practically generate LCD generalized residue codes. But we need to check that the conditions "$2 \in Q$", "$-1 \in Q$" and "every nonzero element in Q is co-prime with n" can be satisfied simultaneously, in particular for $t = 2$ which is the most interesting case for our applications. This is work in progress but we already made the following observation:

Proposition 9 *Let p be any odd prime number and $n = p^2$. Let $Q = \{i^2, i \in \mathbb{Z}/n\mathbb{Z}\}$. Then every nonzero element in Q is co-prime with n.*

Indeed, let $0 < i = kp + l < n$. We have $k, l < p$. Then $i^2 \equiv l^2$ [mod p] and if $l \neq 0$ then i^2 is co-prime with p and then with n.

4 Constructing LCD Codes from Other LCD Codes

The LCD property is invariant under permutation of the codeword coordinates. The only other transformation that we known which preserves the LCD property is the direct sum.

Proposition 10 *If C and C' are LCD codes of parameters $[n, k, d]$ and $[n', k', d']$, respectively, then their* direct sum *$C \oplus C' = \{(c, c'), c \in C, c' \in C'\}$ is LCD of parameters $[n + n', k + k', min(d, d')]$.*

Indeed, $(C \oplus C')^\perp = C^\perp \oplus C'^\perp$ and then $(C \oplus C') \cap (C \oplus C')^\perp = (C \cap C^\perp) \oplus (C' \cap C'^\perp)$. The name of *direct sum* comes from the fact that the indices of the codewords of C and of those of C' being distinct, the sum of C and C' as vector-spaces is direct.

5 Conclusion and Perspectives

Complementary dual codes have applications in information protection. An example is that of a cryptographic implementation, be it hardware or software, that must be simultaneously protected against information leakage and information corruption, since both threats enable successful attacks. We construct cyclic LCD codes, and find codes of large minimal distances within the class of generalized residue codes. In addition to these codes, we detail some secondary constructions, using direct sum.

As a perspective, we aim at defining bounds for the minimal distance of LCD codes, and at finding codes that approach those bounds. Besides, LCD codes of *sparse* generator matrices would help reduce the implementation complexity.

Acknowledgements The authors are grateful to Patrick Solé for pointing relevant previous art.

References

1. Augot, D., Sendrier, N.: Idempotents and the BCH bound. IEEE Trans. Inf. Theory **40**(1), 204–207 (1994)
2. Bhasin, S., Danger, J.-L., Guilley, S., Najm, Z.: A low-entropy first-degree secure provable masking scheme for resource-constrained devices. In: Proceedings of the Workshop on Embedded Systems Security, WESS'13, New York, 29 Sept 2013, pp. 7:1–7:10. ACM, Montreal. doi:10.1145/2527317.2527324
3. Bringer, J., Carlet, C., Chabanne, H., Guilley, S., Maghrebi, H.: Orthogonal direct sum masking – a smartcard friendly computation paradigm in a code, with builtin protection against side-channel and fault attacks. In: WISTP, Heraklion, June 2014. Volume 8501 of LNCS, pp. 40–56. Springer (2014)
4. Carlet, C.: Boolean functions for cryptography and error correcting codes. In: Crama, Y., Hammer, P. (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering, pp. 257–397. Cambridge University Press, Cambridge (2010). Preliminary version available at: http://www.math.univ-paris13.fr/~carlet/chap-fcts-Bool-corr.pdf
5. Carlet, C.: Correlation-immune boolean functions for leakage squeezing and rotating s-box masking against side channel attacks. In: Gierlichs, B., Guilley, S., Mukhopadhyay, D. (eds.) SPACE, Kharagpur, 19th - 23rd October 2013 Volume 8204 of Lecture Notes in Computer Science, pp. 70–74. Springer (2013)
6. Chen, B., Dinh, H.Q., Liu, H.: Repeated-root constacyclic codes of length $2\ell^m p^n$. Finite Fields and Their Applications Volume 33, May 2015, pp. 137–159
7. Etesami, J., Hu, F., Henkel, W.: LCD codes and iterative decoding by projections, a first step towards an intuitive description of iterative decoding. In: GLOBECOM, Houston, pp. 1–4. IEEE (2011)
8. Grosso, V., Standaert, F.-X., Prouff, E.: low entropy masking schemes, revisited. In: Francillon, A., Rohatgi, P. (eds.) CARDIS, Berlin. Volume 8419 of LNCS, pp. 33–43. Springer (2013)
9. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. Elsevier, Amsterdam (1977). ISBN:978-0-444-85193-2
10. Massey., J.L.: Linear codes with complementary duals. Discret. Math. **106–107**, 337–342 (1992)

11. Sendrier, N.: Linear codes with complementary duals meet the Gilbert-Varshamov bound. Discret. Math. **285**, 345–347 (2004)
12. van Lint, J.H., MacWilliams, F.J.: Generalized quadratic residue codes. IEEE Trans. Inf. Theory **24**(6), 730–737 (1978)
13. Vasantha Kandasamy, W.B., Smarandache, F., Sujatha, R., Raja Durai, R.S.: Erasure Techniques in MRD Codes. 28 Apr 2012. ISBN-10:1599731770, ISBN-13:978-1599731773
14. Ward, H.N.: Quadratic residue codes and divisibility. In: Pless, V.S., Huffman, W.C. (eds.) Handbook of Coding Theory, pp. 827–870. Elsevier Science, Amsterdam/New York (1998)
15. Yang, X., Massey, J.L.: The condition for a cyclic code to have a complementary dual. Discret. Math. **126**(1), 391–393 (1994)

Input-State-Output Representation of Convolutional Product Codes

Joan-Josep Climent, Victoria Herranz, and Carmen Perea

Abstract In this paper, we present an input-state-output representation of a convolutional product code; we show that this representation is non minimal. Moreover, we introduce a lower bound on the free distance of the convolutional product code in terms of the free distance of the constituent codes.

Keywords Convolutional code • Product code • ISO representation • Free distance • Kronecker product

1 Introduction

The class of convolutional codes generalizes the class of linear block codes in a natural way. In comparison to the literature on linear block codes, there are only relatively few algebraic constructions of convolutional codes which have a good designed distance. There are several methods for constructing convolutional codes, for example by extending the constructions known for block codes to convolutional codes, such as the ones based on cyclic or quasi-cyclic constructions on block codes [7, 8, 10, 19].

Combining known codes is a powerful method to obtain new codes with better error correction capability avoiding the exponential increase of decoding complexity. For convolutional codes, we can find in the literature some powerful combining methods as woven convolutional codes [21, 22] and turbo codes [18]. More recently, as a natural extension of the direct product codes introduced by Elias [3], Bossert, Medina and Sidorenko [1] introduce the product of convolutional codes

J.-J. Climent (✉)
Departament d'Estadística i Investigació Operativa, Universitat d'Alacant, Ap. Correus 99,
E-03080 Alacant, Spain
e-mail: jcliment@ua.es

V. Herranz • C. Perea
Departamento de Estadística, Matemáticas e Informática, Centro de Investigación Operativa,
Universidad Miguel Hernández de Elche, Avenida del Ferrocarril, s/n. E-03202 Elche, Spain
e-mail: mavi.herranz@umh.es; perea@umh.es

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_10

and they show that every convolutional product code can be represented as a woven convolutional code (see also [11]).

On the other hand, it is well-known that there exists a close connection between linear systems over finite fields and convolutional codes. Rosenthal [13] provides an excellent survey of the different points of view about convolutional codes. By using the input-state-output representation of convolutional codes introduced by Rosenthal and York [16], Climent, Herranz and Perea [2] and Herrnaz [4] introduce the input-state-output representation of different serial and parallel concatenated convolutional codes, and by using them, they also present a construction of new codes with prescribed distance.

The rest of the paper is structured as follows. In Sect. 2 we present the basic notions and previous results related to convolutional codes and convolutional product codes. Then, in Sect. 3, we introduce two input-state-output representations of a convolutional product code and prove that none of them is minimal. Moreover we introduce a lower bound on the free distance of the convolutional product code.

2 Preliminaries

Let $\mathbb{F}$ be a finite field and denote by $\mathbb{F}[z]$ the polynomial ring on the variable z with coefficients in $\mathbb{F}$. A *convolutional code* $\mathscr{C}$ of rate k/n is a submodule of $\mathbb{F}[z]^n$ that can be described as (see [17, 20])

$$\mathscr{C} = \operatorname{im}_{\mathbb{F}[z]}(G(z)) = \{\boldsymbol{v}(z) \in \mathbb{F}[z]^n \mid \boldsymbol{v}(z) = G(z)\boldsymbol{u}(z) \text{ with } \boldsymbol{u}(z) \in \mathbb{F}[z]^k\}$$

where $\boldsymbol{u}(z)$ is the **information vector**, $\boldsymbol{v}(z)$ is the corresponding **codeword** and $G(z)$ is an $n \times k$ polynomial matrix with rank k called **generator** or **encoder matrix** of $\mathscr{C}$. Two full column rank matrices $G_1(z), G_2(z) \in \mathbb{F}[z]^{n\times k}$ are said to be **equivalent encoders** if and only if there exists a unimodular matrix $P(z) \in \mathbb{F}[z]^{k\times k}$ such that $G_2(z) = G_1(z)P(z)$. The **complexity** of a convolutional code $\mathscr{C}$ is the highest degree of the full size minors of any encoder of $\mathscr{C}$. A generator matrix of a convolutional code is called **minimal** if and only if the complexity is equal to the sum of the column degrees.

A generator matrix is said to be **catastrophic** [6] if there exists some input sequence $\boldsymbol{u}(z)$ with infinite nonzero entries which generates a codeword $\boldsymbol{v}(z) = G(z)\boldsymbol{u}(z)$ with a finite nonzero entries. A convolutional code $\mathscr{C}$ is **observable** if one, and therefore any, generator matrix $G(z)$ is right prime (see [14]). Furthermore, if $G(z)$ is a generator matrix of an observable convolutional code, then $G(z)$ is a noncatastrophic generator matrix (see [14]).

Let $\boldsymbol{v}(z) \in \mathscr{C}$ and assume that $\boldsymbol{v}(z) = \boldsymbol{v}_0 z^{\gamma} + \boldsymbol{v}_1 z^{\gamma-1} + \cdots + \boldsymbol{v}_{\gamma-1} z + \boldsymbol{v}_{\gamma}$ with $\boldsymbol{v}_t \in \mathbb{F}^n$, for $t = 0, 1, \ldots, \gamma-1, \gamma$. If we consider $\boldsymbol{v}_t = \begin{pmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{pmatrix}$, where $\boldsymbol{y}_t \in \mathbb{F}^{n-k}$ and $\boldsymbol{u}_t \in \mathbb{F}^k$, then the convolutional code $\mathscr{C}$ is equivalently described by the (A, B, C, D)

representation (see [13, 16, 17, 20])

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t, \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t, \end{aligned}\right\}, \quad t = 0, 1, 2, \ldots, \quad \boldsymbol{x}_0 = \boldsymbol{0}.$$

For each instant t, we say that $\boldsymbol{x}_t$ is the **state vector**, $\boldsymbol{u}_t$ is the **information vector**, $\boldsymbol{y}_t$ is the **parity vector**, and $\boldsymbol{v}_t$ is the **codeword**. In the linear systems theory, this representation is known as the **input-state-output (ISO) representation**.

If $\mathscr{C}$ is a rate k/n convolutional code with complexity δ, we call $\mathscr{C}$ an (n, k, δ)-code, and in that case, it is possible (see [9]) to choose matrices A, B, C and D of sizes $\delta \times \delta$, $\delta \times k$, $(n-k) \times \delta$ and $(n-k) \times k$, respectively. In convolutional coding theory, an ISO representation (A, B, C, D) having the above sizes is called a **minimal representation** and it is characterized through the condition that the pair (A, B) is **controllable**, that is (see [16]),

$$\operatorname{rank}\begin{pmatrix} B & AB & \cdots & A^{\delta-1}B \end{pmatrix} = \delta.$$

Moreover, if (A, B) is controllable, then the convolutional code defined by the matrices (A, B, C, D) is an observable code if and only if (A, C) is an observable pair (see [12]). Recall that (A, C) is an **observable** pair if (A^T, C^T) is a controllable pair.

The **free distance** of a convolutional code $\mathscr{C}$ can be characterized (see [5]) as

$$d_{free}(\mathscr{C}) = \min\left(\sum_{t=0}^{\infty} \operatorname{wt}(\boldsymbol{u}_t) + \sum_{t=0}^{\infty} \operatorname{wt}(\boldsymbol{y}_t)\right)$$

where the minimum has to be taken over all possible nonzero codewords and where wt denotes the Hamming weight. The free distance of an (n, k, δ)-code $\mathscr{C}$ is always upper-bounded (see [15]) by the **generalized Singleton bound**

$$d_{free}(\mathscr{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1.$$

In addition, the convolutional code $\mathscr{C}$ is called **maximum-distance separable** (MDS) if its free distance is equal to the generalized Singleton bound.

To finish this section, we introduce the product of two convolutional codes called "horizontal" and "vertical" codes respectively. Assume that $\mathscr{C}_h$ and $\mathscr{C}_v$ are horizontal (n_h, k_h, δ_h) and vertical (n_v, k_v, δ_v) codes respectively. Then, the **product convolutional code** (see [1, 11]) $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$ is defined to be the convolutional code whose codewords consist of all $n_v \times n_h$ matrices in which columns belong to $\mathscr{C}_v$ and rows belongs to $\mathscr{C}_h$. It is an $(n_h n_v, k_h k_v, \delta_h k_v + k_h \delta_v)$.

Encoding of the product convolutional code $\mathscr{C}$ can be done as follows (see [1, 11]). Let $G_v(z)$ and $G_h(z)$ be generator matrices of the component convolutional codes $\mathscr{C}_v$ and $\mathscr{C}_h$, respectively. Denote by $U(z)$ a $k_v \times k_h$ information matrix. Now, we can apply **row-column** encoding; i.e., every column of $U(z)$ is encoded using $G_v(z)$, and then every row of the resulting matrix $G_v(z)U(z)$ is encoded using $G_h(z)$ as $(G_v(z)U(z))G_h(z)^T$. We can also apply **column-row** encoding; i.e., every row of $U(z)$ is encoded using $G_h(z)$, and then every column of the resulting matrix $U(z)G_h(z)^T$ is encoding using $G_v(z)$ as $G_v(z)(U(z)G_h(z)^T)$. As a consequence of the associativity of the product of matrices, we get the same matrix in both cases. So, the codeword matrix $V(z)$ is given by

$$V(z) = G_v(z)\, U(z)\, G_h(z)^T,$$

and by using properties of the Kronecker product, we have

$$\operatorname{vect}(V(z)) = (G_h(z) \otimes G_v(z)) \operatorname{vect}(U(z))$$

where vect $(\cdot)$ is the operator that transforms a matrix into a vector by stacking the column vectors of the matrix below one another. So, the generator matrix $G(z)$ of the product convolutional code $\mathscr{C}$ is the Kronecker product

$$G(z) = G_h(z) \otimes G_v(z)$$

of the generator matrices of the horizontal and vertical codes.

3 ISO Representation of a Product Convolutional Code

Assume that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are the ISO representations of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathscr{C}_h$ and $\mathscr{C}_v$, respectively. Assume also that the $k_v \times k_h$ matrix U_t is the information matrix of the product code $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$.

By using the ISO representation of the horizontal code $\mathscr{C}_h$ we can encode the information vector $\boldsymbol{u}_t = \operatorname{vect}(U_t)$ as

$$\left.\begin{aligned} \boldsymbol{x}^h_{t+1} &= (A_h \otimes I_{k_v})\boldsymbol{x}^h_t + (B_h \otimes I_{k_v})\boldsymbol{u}_t \\ \boldsymbol{y}^h_t &= (C_h \otimes I_{k_v})\boldsymbol{x}^h_t + (D_h \otimes I_{k_v})\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}^h_t = \begin{pmatrix} \boldsymbol{y}^h_t \\ \boldsymbol{u}_t \end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \boldsymbol{x}^h_0 = \mathbf{0}. \tag{1}$$

Analogously, by using the ISO representation of the vertical code $\mathscr{C}_v$ we can encode the same information vector $\boldsymbol{u}_t$ as

$$\left.\begin{aligned} \boldsymbol{x}^v_{t+1} &= (I_{k_h} \otimes A_v)\boldsymbol{x}^v_t + (I_{k_h} \otimes B_v)\boldsymbol{u}_t \\ \boldsymbol{y}^v_t &= (I_{k_h} \otimes C_v)\boldsymbol{x}^v_t + (I_{k_h} \otimes D_v)\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}^v_t = \begin{pmatrix} \boldsymbol{y}^v_t \\ \boldsymbol{u}_t \end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \boldsymbol{x}^{|}_0 = \mathbf{0}, \tag{2}$$

Then we encode the parity vector $\boldsymbol{y}_t^h$ (respectively, $\boldsymbol{y}_t^v$) by using the vertical code $\mathscr{C}_v$ (respectively, the horizontal code $\mathscr{C}_h$) as

$$\left.\begin{aligned} \mathfrak{x}_{t+1}^v &= (I_{n_h-k_h} \otimes A_v)\mathfrak{x}_t^v + (I_{n_h-k_h} \otimes B_v)\boldsymbol{y}_t^h \\ \mathfrak{y}_t^v &= (I_{n_h-k_h} \otimes C_v)\mathfrak{x}_t^v + (I_{n_h-k_h} \otimes D_v)\boldsymbol{y}_t^h \end{aligned}\right\}, \quad \mathfrak{v}_t^v = \begin{pmatrix} \mathfrak{y}_t^v \\ \boldsymbol{y}_t^h \end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \mathfrak{x}_0^v = \mathbf{0}, \tag{3}$$

$$\left.\begin{aligned} \mathfrak{x}_{t+1}^h &= (A_h \otimes I_{n_v-k_v})\mathfrak{x}_t^h + (B_h \otimes I_{n_v-k_v})\boldsymbol{y}_t^v \\ \mathfrak{y}_t^h &= (C_h \otimes I_{n_v-k_v})\mathfrak{x}_t^h + (D_h \otimes I_{n_v-k_v})\boldsymbol{y}_t^v \end{aligned}\right\}, \quad \mathfrak{y}_t^h = \begin{pmatrix} \mathfrak{y}_t^h \\ \boldsymbol{y}_t^v \end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \mathfrak{x}_0^h = \mathbf{0}, \tag{4}$$

Then, by using properties of the Kronecker product we obtain the following result.

Theorem 1 *For the vectors $\mathfrak{y}_t^v$ and $\mathfrak{y}_t^h$ defined by expressions (3) and (4) respectively, it follows that $\mathfrak{y}_t^v = \mathfrak{y}_t^h$, for $t = 0, 1, 2, \ldots$*

Proof By induction over t. □

Next result establishes that the ISO representations defined by matrices in expressions (1)–(4) are minimal ISO representations.

Theorem 2 *Let us assume that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal ISO representations of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathscr{C}_h$ and $\mathscr{C}_v$, respectively. Then*

1. *The matrices $(A_h \otimes I_{k_v}, B_h \otimes I_{k_v}, C_h \otimes I_{k_v}, D_h \otimes I_{k_v})$ in expression (1) define a minimal ISO representation of an $(n_h k_v, k_h k_v, \delta_h k_v)$ convolutional code $\mathscr{C}_h(k_v)$.*
2. *The matrices $(I_{k_h} \otimes A_v, I_{k_h} \otimes B_v, I_{k_h} \otimes C_v, I_{k_h} \otimes D_v)$ in expression (2) define a minimal ISO representation of an $(k_h n_v, k_h k_v, k_h \delta_v)$ convolutional code $\mathscr{C}_v(k_h)$.*
3. *The matrices $(I_{n_h-k_h} \otimes A_v, I_{n_h-k_h} \otimes B_v, I_{n_h-k_h} \otimes C_v, I_{n_h-k_h} \otimes D_v)$ in expression (3) define a minimal ISO representation of an $((n_h-k_h)n_v, (n_h-k_h)k_v, (n_h-k_h)\delta_v)$ convolutional code $\mathscr{C}_v(n_h - k_h)$.*
4. *The matrices $(A_h \otimes I_{n_v-k_v}, B_h \otimes I_{n_v-k_v}, C_h \otimes I_{n_v-k_v}, D_h \otimes I_{n_v-k_v})$ in expression (4) define a minimal ISO representation of an $(n_h(n_v-k_v), k_h(n_v-k_v), \delta_h(n_v-k_v))$ convolutional code $\mathscr{C}_h(n_v - k_v)$.*

Proof The result follows from the fact that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal ISO representations and the properties of the Kronecker product of matrices. □

It is not difficult to show that the codes $\mathscr{C}_h(k_v)$ and $\mathscr{C}_h(n_v - k_v)$ (respectively, $\mathscr{C}_v(k_h)$ and $\mathscr{C}_v(n_h - k_h)$) correspond to the block parallel concatenation of convolutional codes described in [4, Section 5.3], and therefore

$$\begin{aligned} d_{free}\left(\mathscr{C}_h(k_v)\right) = d_{free}\left(\mathscr{C}_h(n_v - k_v)\right) = d_{free}\left(\mathscr{C}_h\right), \\ d_{free}\left(\mathscr{C}_v(k_h)\right) = d_{free}\left(\mathscr{C}_v(n_h - k_h)\right) = d_{free}\left(\mathscr{C}_v\right). \end{aligned} \tag{5}$$

Now, by using the second model of serial concatenated convolutional codes introduced in [2, 4] we have the following result.

Theorem 3 *With the same notation as in Theorem 2.*

1. *If $\mathscr{S}_1$ is the rate $k_hk_v/((n_h - k_h)n_v + k_hk_v)$ convolutional code defined by the serial concatenation of $\mathscr{C}_h(k_v)$ and $\mathscr{C}_v(n_h - k_h)$, then $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$, with*

$$\mathbf{A}_1 = \begin{bmatrix} I_{n_h-k_h} \otimes A_v & C_h \otimes B_v \\ O & A_h \otimes I_{k_v} \end{bmatrix}, \qquad \mathbf{B}_1 = \begin{bmatrix} D_h \otimes B_v \\ B_h \otimes I_{k_v} \end{bmatrix},$$

$$\mathbf{C}_1 = \begin{bmatrix} I_{n_h-k_h} \otimes C_v & C_h \otimes D_v \\ O & C_h \otimes I_{k_v} \end{bmatrix}, \qquad \mathbf{D}_1 = \begin{bmatrix} D_h \otimes D_v \\ D_h \otimes I_{k_v} \end{bmatrix},$$

   is an ISO representation of $\mathscr{S}_1$.

2. *If $\mathscr{S}_2$ is the rate $k_hk_v/(n_h(n_v - k_v) + k_hk_v)$ convolutional code defined by the serial concatenation of $\mathscr{C}_v(k_h)$ and $\mathscr{C}_h(n_v - k_v)$, then $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{D}_2)$, with*

$$\mathbf{A}_2 = \begin{bmatrix} A_h \otimes I_{n_v-k_v} & B_h \otimes C_v \\ O & I_{k_h} \otimes A_v \end{bmatrix}, \qquad \mathbf{B}_2 = \begin{bmatrix} B_h \otimes D_v \\ I_{k_h} \otimes B_v \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} C_h \otimes I_{n_v-k_v} & D_h \otimes C_v \\ O & I_{k_h} \otimes C_v \end{bmatrix}, \qquad \mathbf{D}_2 = \begin{bmatrix} D_h \otimes D_v \\ I_{k_h} \otimes D_v \end{bmatrix},$$

   is an ISO representation of $\mathscr{S}_2$.

Proof The result follows from Theorem 9 of [2]. □

In general the ISO representations $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$ and $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{D}_2)$ introduced in the above theorem are not minimal (see [2, 4]). In [2, 4] we can find some sufficient conditions to ensure the minimality of the above ISO representations.

Now, by Theorem 15 of [2] we have that

$$d_{free}(\mathscr{S}_1) \geq d_{free}(\mathscr{C}_h) \quad \text{and} \quad d_{free}(\mathscr{S}_2) \geq d_{free}(\mathscr{C}_v). \tag{6}$$

As a consequence of Theorem 1, by using the second model of parallel concatenation (see [4, Section 5.2]) we have the following result.

Theorem 4 *With the same notation as in Theorems 2 and 3.*

1. *If $\mathscr{P}_1$ is the rate (k_hk_v/n_hn_v) convolutional code defined by the parallel concatenation of $\mathscr{S}_1$ and $\mathscr{C}_v(k_h)$, then $(\mathfrak{A}_1, \mathfrak{B}_1, \mathfrak{C}_1, \mathfrak{D}_1)$ with*

$$\mathfrak{A}_1 = \left[\begin{array}{cc:c} I_{n_h-k_h} \otimes A_v & C_h \otimes B_v & O \\ O & A_h \otimes I_{k_v} & O \\ \hdashline O & O & I_{k_h} \otimes A_v \end{array}\right], \quad \mathfrak{B}_1 = \left[\begin{array}{c} D_h \otimes B_v \\ B_h \otimes I_{k_v} \\ \hdashline I_{k_h} \otimes B_v \end{array}\right]$$

$$\mathfrak{C}_1 = \left[\begin{array}{cc:c} I_{n_h-k_h} \otimes C_v & C_h \otimes D_v & O \\ O & C_h \otimes I_{k_v} & O \\ \hdashline O & O & I_{k_h} \otimes C_v \end{array}\right], \quad \mathfrak{D}_1 = \left[\begin{array}{c} D_h \otimes D_v \\ D_h \otimes I_{k_v} \\ \hdashline I_{k_h} \otimes D_v \end{array}\right]$$

   is an ISO representation of $\mathscr{P}_1$.

2. *If $\mathscr{P}_2$ is the rate $(k_h k_v / n_h n_v)$ convolutional code defined by the parallel concatenation of $\mathscr{S}_2$ and $\mathscr{C}_h(k_v)$, then $(\mathfrak{A}_2, \mathfrak{B}_2, \mathfrak{C}_2, \mathfrak{D}_2)$ with*

$$\mathfrak{A}_2 = \left[\begin{array}{cc|c} A_h \otimes I_{n_v - k_v} & B_h \otimes C_v & O \\ O & I_{k_h} \otimes A_v & O \\ \hline O & O & A_h \otimes I_{k_v} \end{array}\right], \quad \mathfrak{B}_2 = \left[\begin{array}{c} B_h \otimes D_v \\ I_{k_h} \otimes B_v \\ \hline B_h \otimes I_{k_v} \end{array}\right]$$

$$\mathfrak{C}_2 = \left[\begin{array}{cc|c} C_h \otimes I_{n_v - k_v} & D_h \otimes C_v & O \\ O & I_{k_h} \otimes C_v & O \\ \hline O & O & C_h \otimes I_{k_v} \end{array}\right], \quad \mathfrak{D}_2 = \left[\begin{array}{c} D_h \otimes D_v \\ I_{k_h} \otimes D_v \\ \hline D_h \otimes I_{k_v} \end{array}\right]$$

is an ISO representation of $\mathscr{P}_2$.

Note that, according to expressions (1)–(4) and Theorem 1, $\mathscr{P}_1$ is the product convolutional code $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$. Moreover, since $\mathfrak{A}_1$ is a matrix of size $(n_h \delta_v + \delta_h k_v) \times (n_h \delta_v + \delta_h k_v)$ and the complexity of $\mathscr{C}$ is $k_h \delta_v + \delta_h k_v$, we can ensure that the ISO representation $(\mathfrak{A}_1, \mathfrak{B}_1, \mathfrak{C}_1, \mathfrak{D}_1)$ provided by part 1 of Theorem 4 is nonminimal. By an analogous argument $\mathscr{P}_2$ is the product convolutional code $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$ and the ISO representation $(\mathfrak{A}_2, \mathfrak{B}_2, \mathfrak{C}_2, \mathfrak{D}_2)$ provided by part 2 of Theorem 4 is nonminimal.

Next result introduces a lower bound on the free distance d_{free} of the convolutional product code in terms of the constituent convolutional codes.

Theorem 5 *If $\mathscr{C}_h$ and $\mathscr{C}_v$ are (n_h, k_h, δ_h) and (n_v, k_v, δ_v) codes, respectively, then,*

$$d_{free}(\mathscr{C}_h \otimes \mathscr{C}_v) \geq \max\left\{d_{free}(\mathscr{C}_v), d_{free}(\mathscr{C}_h)\right\}.$$

Proof With the same notation as in Theorem 4, as a consequence of Theorem 5.8 of [4] we have that

$$d_{free}(\mathscr{P}_1) \geq \max\left\{d_{free}(\mathscr{S}_1), d_{free}(\mathscr{C}_v)\right\},$$
$$d_{free}(\mathscr{P}_2) \geq \max\left\{d_{free}(\mathscr{S}_2), d_{free}(\mathscr{C}_h)\right\}.$$

The result follows now by expressions (5) and (6) an the fact that $\mathscr{P}_1 = \mathscr{P}_2 = \mathscr{C}_h \otimes \mathscr{C}_v$. □

Acknowledgements This work was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Ciencia e Innovación of the Gobierno de España.

References

1. Bossert, M., Medina, C., Sidorenko, V.: Encoding and distance estimation of product convolutional codes. In: Proceedings of the 2005 IEEE International Symposium on Information Theory (ISIT 2005), Adelaide, pp. 1063–1066. IEEE (2005). doi:10.1109/ISIT.2005.1523502

2. Climent, J.J., Herranz, V., Perea, C.: A first approximation of concatenated convolutional codes from linear systems theory viewpoint. Linear Algebr. Appl. **425**, 673–699 (2007). doi:10.1016/j.laa.2007.03.017
3. Elias, P.: Error free coding. Trans. IRE Prof. Group Inf. Theory **4**(4), 29–37 (1954)
4. Herranz, V.: Estudio y construcción de códigos convolucionales: Códigos perforados, códigos concatenados desde el punto de vista de sistemas. Ph.D. thesis, Departamento de Estadística, Matemáticas e Informática, Universidad Miguel Hernández, Elche (2007)
5. Hutchinson, R., Rosenthal, J., Smarandache, R.: Convolutional codes with maximum distance profile. Syst. Control Lett. **54**(1), 53–63 (2005). doi:10.1016/j.sysconle.2004.06.005
6. Johannesson, R., Wan, Z.X.: A linear algebra approach to minimal convolutional encoders. IEEE Trans. Inf. Theory **39**(4), 1219–1233 (1993). doi:10.1109/18.243440
7. Justesen, J.: New convolutional code constructions and a class of asymptotically good time-varying codes. IEEE Trans. Inf. Theory **19**(2), 220–225 (1973). doi:10.1109/TIT.1973.1054983
8. Justesen, J.: An algebraic construction of rate $1/\nu$ convolutional codes. IEEE Trans. Inf. Theory **21**(5), 577–580 (1975). doi:10.1109/TIT.1975.1055436
9. Kalman, R.E., Falb, P.L., Arbib, M.A.: Topics in Mathematical System Theory. McGraw-Hill, New York (1969)
10. Massey, J.L., Costello, D.J., Justesen, J.: Polynomial weights and code constructions. IEEE Trans. Inf. Theory **19**(1), 101–110 (1973). doi:10.1109/TIT.1973.1054936
11. Medina, C., Sidorenko, V.R., Zyablov, V.V.: Error exponents for product convolutional codes. Probl. Inf. Transm. **42**(3), 167–182 (2006). doi:10.1134/S003294600603001X
12. Rosenthal, J.: An algebraic decoding algorithm for convolutional codes. Prog. Syst. Control Theory **25**, 343–360 (1999). doi:10.1007/978-3-0348-8970-4_16
13. Rosenthal, J.: Connections between linear systems and convolutional codes. In: Marcus, B., Rosenthal, J. (eds.) Codes, Systems and Graphical Models. The IMA Volumes in Mathematics and its Applications, vol. 123, pp. 39–66. Springer, New York (2001). doi:10.1007/978-1-4613-0165-3_2
14. Rosenthal, J., Smarandache, R.: Construction of convolutional codes using methods from linear systems theory. In: Proccedings of the 35th Allerton Conference on Communications, Control and Computing, Urbana, pp. 953–960. Allerton House, Monticello (1997)
15. Rosenthal, J., Smarandache, R.: Maximum distance separable convolutional codes. Appl. Algebra Eng. Commun. Comput. **10**(1), 15–32 (1999). doi:10.1007/s002000050120
16. Rosenthal, J., York, E.V.: BCH convolutional codes. IEEE Trans. Inf. Theory **45**(6), 1833–1844 (1999). doi:10.1109/18.782104
17. Rosenthal, J., Schumacher, J.M., York, E.V.: On behaviors and convolutional codes. IEEE Trans. Inf. Theory **42**(6), 1881–1891 (1996). doi:10.1109/18.556682
18. Sripimanwat, K. (ed.): Turbo Code Applications. A Journey from a Paper to Realization. Springer, Dordrecht (2005). doi:10.1007/1-4020-3685-X
19. Tanner, R.M.: Convolutional codes from quasicyclic codes: a link between the theories of block and convolutional codes. Technical Report USC-CRL-87-21, University of California, Santa Cruz (1987)
20. York, E.V.: Algebraic description and construction of error correcting codes: a linear systems point of view. Ph.D. thesis, Department of Mathematics, University of Notre Dame, Notre Dame (1997)
21. Zyablov, V., Shavgulidze, S., Skopintsev, O., Höst, S., Johannesson, R.: On the error exponent for woven convolutional codes with outer warp. IEEE Trans. Inf. Theory **45**(5), 1649–1653 (1999). doi:10.1109/18.771237
22. Zyablov, V.V., Shavgulidze, S., Johannesson, R.: On the error exponent for woven convolutional codes with inner warp. IEEE Trans. Inf. Theory **47**(3), 1195–1199 (2001). doi:10.1109/18.915681

Burst Erasure Correction of 2D Convolutional Codes

Joan-Josep Climent, Diego Napp, Raquel Pinto, and Rita Simões

Abstract In this paper we address the problem of decoding 2D convolutional codes over the erasure channel. In particular, we present a procedure to recover bursts of erasures that are distributed in a diagonal line. To this end we introduce the notion of balls around a burst of erasures which can be considered an analogue of the notion of sliding window in the context of 1D convolutional codes. The main result reduces the decoding problem of 2D convolutional codes to a problem of decoding a set of associated 1D convolutional codes.

Keywords 2D convolutional codes • Erasure channel

1 Introduction

When transmitting over an erasure channel the symbol sent either arrive correctly or they are erased. Internet is an important instance of such a channel. One of the problems that arises in this channel is that some packets get lost and the receiver experience it as a delay on the received information. The solutions proposed to deal with this problem are commonly based on the use of block codes. However, in recent years, there has been an increased interest in the study of one-dimensional (1D) convolutional codes over the erasure channel [2, 8–10] as a possible alternative for the widely use of block codes. Due to their rich structure 1D convolutional codes have an interesting property called sliding window property that allows adaptation to the correction process to the distribution of the erasure pattern. In the recent paper [10] it has been shown how it is possible to exploit this property in order

J.-J. Climent
Departament d'Estadística i Investigació Operativa, Universitat d'Alacant, Ap. Correus 99, E-03080 Alacant, Spain
e-mail: jcliment@ua.es

D. Napp • R. Pinto (✉) • R. Simões
Department of Mathematics, CIDMA – Center for Research and Development in Mathematics and Applications, University of Aveiro, Campus Universitario de Santiago, 3810-193 Aveiro, Portugal
e-mail: diego@ua.pt; raquel@ua.pt; ritasimoes@ua.pt

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_11

to easily recover erasures which are uncorrectable by any other kind of (block) codes. The codes proposed in this paper are codes with strong distance properties, called Maximal Distance Profile (MDP), reverse-MDP and complete-MDP, and simulations results have shown that they can decode extremely efficiently when compared to MDS block codes.

In the 1D case, if the received codeword is viewed as a finite sequence $\boldsymbol{v} = (v_0, v_1, \ldots, v_\ell)$, then the sliding windows is given by selecting a subsequence of $\boldsymbol{v}$, $(v_i, \ldots, v_{i+N})$, where $i, N \in \mathbb{N}$ depend on the erasure burst pattern. However, when considering two-dimensional (2D) convolutional codes [4–7, 11] the information is distributed in two dimensions and therefore there is not an obvious way to extend the idea of sliding window to the 2D case. In this work we propose several solutions for dealing with this problem by introducing the notion of *balls* around an erasure. We show that when considering these particular balls one reduces the problem of decoding 2D convolutional codes over the erasure channel to a problem related to decoding of 1D convolutional codes.

2 2D Convolutional Codes

In this section we recall the basic background on 2D finite support convolutional codes. Denote by $\mathbb{F}[z_1, z_2]$ the ring of polynomials in the two variables, z_1 and z_2, with coefficients in the finite field $\mathbb{F}$.

Definition 1 A 2D finite support convolutional code $\mathscr{C}$ of rate k/n is a free $\mathbb{F}[z_1, z_2]$-submodule of $\mathbb{F}[z_1, z_2]^n$ with rank k.

A full column rank polynomial matrix $\hat{G}(z_1, z_2) \in \mathbb{F}[z_1, z_2]^{n \times k}$ whose columns constitute a basis for $\mathscr{C}$, i.e., such that

$$\begin{aligned}\mathscr{C} &= \operatorname{im}_{\mathbb{F}[z_1,z_2]} \hat{G}(z_1, z_2) \\ &= \{\hat{\boldsymbol{v}}(z_1, z_2) \in \mathbb{F}[z_1, z_2]^n \mid \hat{\boldsymbol{v}}(z_1, z_2) = \\ &\quad \hat{G}(z_1, z_2)\hat{\boldsymbol{u}}(z_1, z_2) \text{ with } \hat{\boldsymbol{u}}(z_1, z_2) \in \mathbb{F}[z_1, z_2]^k \},\end{aligned}$$

is called an *encoder* of $\mathscr{C}$. The elements of $\mathscr{C}$ are called *codewords*.

If the code $\mathscr{C}$ admits a right factor prime encoder [3], then it can be equivalently described using an $(n-k) \times n$ full rank polynomial matrix $\hat{H}(z_1, z_2)$, called *parity-check matrix* of $\mathscr{C}$, as

$$\mathscr{C} = \ker_{\mathbb{F}[z_1,z_2]} \hat{H}(z_1, z_2) = \left\{\hat{\boldsymbol{v}}(z_1, z_2) \in \mathbb{F}[z_1, z_2]^n \mid \hat{H}(z_1, z_2)\hat{\boldsymbol{v}}(z_1, z_2) = \mathbf{0}\right\}.$$

We denote by $\mathbb{N}_0$ the set of nonnegative integers, and define an ordering in $\mathbb{N}_0^2$ as

$$(a, b) \prec (c, d) \text{ if and only if } a + b < c + d, \text{ or } a + b = c + d \text{ and } b < d. \qquad (1)$$

For a polynomial vector $\hat{\boldsymbol{v}}(z_1, z_2) \in \mathbb{F}[z_1, z_2]^n$, we write

$$\hat{\boldsymbol{v}}(z_1, z_2) = \boldsymbol{v}(0,0) + \boldsymbol{v}(1,0)z_1 + \boldsymbol{v}(0,1)z_2 + \cdots + \boldsymbol{v}(0,\gamma)z_2^{\gamma} = \sum_{0 \le a+b \le \gamma} \boldsymbol{v}(a,b) z_1^a z_2^b,$$

(with $\gamma \geq 0$) and we define its support as the set

$$\mathrm{supp}(\hat{\boldsymbol{v}}(z_1, z_2)) = \{(a,b) \in \mathbb{N}_0^2 \mid \boldsymbol{v}(a,b) \neq 0\}.$$

Moreover, we represent a polynomial matrix $\hat{H}(z_1, z_2)$ as

$$\begin{aligned} \hat{H}(z_1, z_2) &= H(0,0) + H(1,0)z_1 + H(0,1)z_2 + \cdots + H(0,\delta)z_2^{\delta} \\ &= \sum_{0 \le i+j \le \delta} H(i,j) z_1^i z_2^j, \end{aligned} \tag{2}$$

where $H(i, j) \neq 0$ for some (i, j) with $i + j = \delta$. We call δ the degree of $\hat{H}(z_1, z_2)$.

The *weight* of $\hat{\boldsymbol{v}}(z_1, z_2)$ is defined as

$$\mathrm{wt}\,(\hat{\boldsymbol{v}}(z_1, z_2)) = \sum_{(a,b) \in \mathbb{N}_0^2} \mathrm{wt}\,(\boldsymbol{v}(a,b))$$

where wt $(\boldsymbol{v}(a,b))$ is the number of nonzero entries of $\boldsymbol{v}(a,b)$ and the *distance* of a code is

$$\mathrm{dist}\,(\mathscr{C}) = \min\left\{\mathrm{wt}\,(\hat{\boldsymbol{v}}(z_1, z_2)) \;\mid \hat{\boldsymbol{v}}(z_1, z_2) \in \mathscr{C}, \text{ with } \hat{\boldsymbol{v}}(z_1, z_2) \neq \mathbf{0}\right\}.$$

We can expand the kernel representation

$$\hat{H}(z_1, z_2)\hat{\boldsymbol{v}}(z_1, z_2) = \sum_{0 \le a+b \le \gamma} \left[\sum_{0 \le i+j \le \delta} H(i,j)\boldsymbol{v}(a-i, b-j) \right] z_1^a z_2^b = 0$$

as

$$\mathbf{H}\mathbf{v} = \mathbf{0} \tag{3}$$

where $\mathbf{H}$, for $\delta = 3$, and $\mathbf{v}$ are given in Fig. 1, where O denotes the $(n-k) \times n$ zero matrix. To understand the structure of matrix $\mathbf{H}$, note that for $t = 0, 1, 2, \ldots$ in the columns corresponding to the block indices $\frac{t(t+1)}{2} + 1, \frac{t(t+1)}{2} + 2, \ldots, \frac{t(t+1)}{2} + t + 1$ appear all the coefficient matrices of $\hat{H}(z_1, z_2)$ ordered according to $\prec$ with the particularity that the matrices $H(i, j)$, with $i + j = d$, for $d = 0, 1, 2, \ldots, \delta - 1$, are separated from the matrices $H(i, j)$, with $i + j = d + 1$, by t zero blocks.

Suppose now that the vector $\hat{\boldsymbol{v}}(z_1, z_2)$ is transmitted through an erasure channel. Each one of the components of $\mathbf{v}$ is either received correctly or is considered

$$
\begin{array}{ccccccccccc}
H(0,0) & & & & & & & & & & \\
H(1,0) & H(0,0) & & & & & & & & & \\
H(0,1) & O & H(0,0) & & & & & & & & \\
H(2,0) & H(1,0) & O & H(0,0) & & & & & & & \\
H(1,1) & H(0,1) & H(1,0) & O & H(0,0) & & & & & & \\
H(0,2) & O & H(0,1) & O & O & H(0,0) & & & & & \\
H(3,0) & H(2,0) & O & H(1,0) & O & O & H(0,0) & & & & \\
H(2,1) & H(1,1) & H(2,0) & H(0,1) & H(1,0) & O & O & H(0,0) & & & \\
H(1,2) & H(0,2) & H(1,1) & O & H(0,1) & H(1,0) & O & O & H(0,0) & & \\
H(0,3) & O & H(0,2) & O & O & H(0,1) & O & O & O & H(0,0) & \\
 & H(3,0) & O & H(2,0) & O & O & H(1,0) & O & O & O & \ddots \\
 & H(2,1) & H(3,0) & H(1,1) & H(2,0) & O & H(0,1) & H(1,0) & O & O & \ddots \\
 & H(1,2) & H(2,1) & H(0,2) & H(1,1) & H(2,0) & O & H(0,1) & H(1,0) & O & \ddots \\
 & H(0,3) & H(1,2) & O & H(0,2) & H(1,1) & O & O & H(0,1) & H(1,0) & \ddots \\
 & & H(0,3) & O & O & H(0,2) & O & O & O & H(0,1) & \ddots \\
 & & & H(3,0) & O & O & H(2,0) & O & O & O & \ddots \\
 & & & H(2,1) & H(3,0) & O & H(1,1) & H(2,0) & O & O & \ddots \\
 & & & H(1,2) & H(2,1) & H(3,0) & H(0,2) & H(1,1) & H(2,0) & O & \ddots \\
 & & & H(0,3) & H(1,2) & H(2,1) & O & H(0,2) & H(1,1) & H(2,0) & \ddots \\
 & & & & H(0,3) & H(1,2) & O & O & H(0,2) & H(1,1) & \ddots \\
 & & & & & H(0,3) & O & O & O & H(0,2) & \ddots \\
 & & & & & & H(3,0) & O & O & O & \ddots \\
 & & & & & & H(2,1) & H(3,0) & O & O & \ddots \\
 & & & & & & H(1,2) & H(2,1) & H(3,0) & O & \ddots \\
 & & & & & & H(0,3) & H(1,2) & H(2,1) & H(3,0) & \ddots \\
 & & & & & & & H(0,3) & H(1,2) & H(2,1) & \ddots \\
 & & & & & & & & H(0,3) & H(1,2) & \ddots \\
 & & & & & & & & & H(0,3) & \ddots \\
 & & & & & & & & & & \ddots
\end{array}
\qquad
\begin{array}{c}
v(0,0) \\ v(1,0) \\ v(0,1) \\ v(2,0) \\ v(1,1) \\ v(0,2) \\ v(3,0) \\ v(2,1) \\ v(1,2) \\ v(0,3) \\ \vdots
\end{array}
$$

(a) (b)

Fig. 1 Parity check matrix **H**, for $\delta = 3$, and codeword **v**. (**a**) Matrix **H**. (**b**) Vector **v**

erasure. Denote by $\mathscr{E}(\hat{\boldsymbol{v}}(z_1, z_2))$ and $\bar{\mathscr{E}}(\hat{\boldsymbol{v}}(z_1, z_2))$ the sets of indices in which there are erasures and there are not erasures, respectively, i.e.,

$$
\begin{aligned}
\mathscr{E}(\hat{\boldsymbol{v}}(z_1, z_2)) &= \{(a, b) \in \operatorname{supp}(\hat{\boldsymbol{v}}(z_1, z_2)) \mid \text{there is an erasure in } \boldsymbol{v}(a, b)\}, \\
\bar{\mathscr{E}}(\hat{\boldsymbol{v}}(z_1, z_2)) &= \operatorname{supp}(\hat{\boldsymbol{v}}(z_1, z_2)) \setminus \mathscr{E}(\hat{\boldsymbol{v}}(z_1, z_2)).
\end{aligned}
$$

One can select the columns of the matrix in (3) that correspond to the coefficient of the erased elements to be the indeterminates of a new system. The rest of the columns in (3) will help us to compute the independent terms. The terms erasure and indeterminate are often used interchangeably. Hence, we denote by $\mathbf{H}_{\mathscr{E}}$ and $\mathbf{H}_{\bar{\mathscr{E}}}$ the submatrices of $\mathbf{H}$ whose block columns are indexed by $\mathscr{E}$ and $\bar{\mathscr{E}}$, respectively. Analogously, we denote $\mathbf{v}_{\mathscr{E}}$ and $\mathbf{v}_{\bar{\mathscr{E}}}$ to obtain $\mathbf{H}_{\mathscr{E}}\mathbf{v}_{\mathscr{E}} + \mathbf{H}_{\bar{\mathscr{E}}}\mathbf{v}_{\bar{\mathscr{E}}} = \mathbf{0}$. Note that as the channel is an erasure channel, $\mathbf{v}_{\bar{\mathscr{E}}}$, and therefore $\mathbf{H}_{\bar{\mathscr{E}}}\mathbf{v}_{\bar{\mathscr{E}}}$, is known. Hence, we obtain a system of linear nonhomogeneous equations

$$
\mathbf{H}_{\mathscr{E}}\mathbf{v}_{\mathscr{E}} = -\mathbf{H}_{\bar{\mathscr{E}}}\mathbf{v}_{\bar{\mathscr{E}}}, \tag{4}
$$

where the components of the vector $\mathbf{v}_{\mathscr{E}}$ are the indeterminates to be determined. Thus, in order to decode $\mathbf{v}_{\mathscr{E}}$ we need to solve system (4).

The next lemma shows the importance of the distance of a code when transmitting over the erasure channel.

Lemma 2 *Let* $\mathscr{C} = \ker_{\mathbb{F}[z_1,z_2]} \hat{H}(z_1, z_2)$ *be given. The following are equivalent:*

1. $\operatorname{dist}(\mathscr{C}) \geq d$.
2. Any $d - 1$ *erasures can be recovered.*
3. Any $d - 1$ *columns of* $\mathbf{H}_{\mathscr{C}}$ *are linearly independent.*

In the context of 1D convolutional codes the analogous set of homogeneous equations of (3) is

$$\begin{bmatrix} H_0 & & & \\ \vdots & \ddots & & \\ H_\alpha & \cdots & H_0 & \\ & \ddots & \vdots & \ddots & \\ & & H_\alpha & \cdots & H_0 \\ & & & \ddots & \vdots \\ & & & & H_\alpha \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_\gamma \end{bmatrix} = 0, \tag{5}$$

where $\mathscr{C} = \ker \hat{H}(z)$ with $\hat{H}(z) = H_0 + H_1 z + \cdots + H_\alpha z^\alpha$.

In this case every component of the received codeword $\boldsymbol{v} = (v_0, v_2, \ldots, v_\gamma)$ depends on the previous α components. In order to find the values of a burst of erasures occurring in $\boldsymbol{v}$, we can use the so-called *sliding window*, that is, we can select a suitable interval of consecutive components of $\boldsymbol{v}$, say $(v_i, \ldots, v_{i+N})$, and solve the corresponding system of equations (see [10]).

In the 2D case each component of $\mathbf{v}$, say $\boldsymbol{v}(a, b)$, depends on components which support lie in the *triangle* $\{(a - i, b - j) \mid 0 \leq i + j \leq \delta\}$, where δ is the degree of $\hat{H}(z_1, z_2)$ of the given 2D code $\mathscr{C} = \ker_{\mathbb{F}[z_1,z_2]} \hat{H}(z_1, z_2)$. It is not straightforward to extend the notion of the sliding window in this context in order to correct burst of 2D erasures. A particular case is treated in the following section.

3 Decoding Burst of Erasures on Lines

It is well-known that a phenomena observed in many channels modeled via the erasure channel is that errors tend to occur in bursts. This point is important to keep in mind when designing codes which are capable of correcting many errors over the erasure channel. In this preliminary work we aim at decoding burst of erasures that are distributed in a diagonal. We present a notion that can be considered as the analogue of the notion of sliding window, called *ball around a burst of erasures*, that will reduce the problem of decoding a 2D convolutional code to the problem of decoding a set of associated 1D convolutional codes.

Let us first suppose that the set of erasures of $\hat{\boldsymbol{v}}(z_1, z_2)$ contains a burst of erasures which support lie in a diagonal, i.e., given by

$$\mathscr{E}'(\hat{\boldsymbol{v}}(z_1, z_2)) = \{(r+t, s), (r+t-1, s+1), \dots, (r, s+t)\} \subset \mathscr{E}(\hat{\boldsymbol{v}}(z_1, z_2)). \tag{6}$$

Hence, Eq. (3) can be divided as

$$\mathbf{H}_{\mathscr{E}'}\mathbf{v}_{\mathscr{E}'} = -\mathbf{H}_{\bar{\mathscr{E}}'}\mathbf{v}_{\bar{\mathscr{E}}'} \tag{7}$$

where $\mathbf{H}_{\mathscr{E}'}$ and $\mathbf{H}_{\bar{\mathscr{E}}'}$ are submatrices of $\mathbf{H}$ whose block columns are indexed by $\mathscr{E}'(\hat{\boldsymbol{v}}(z_1, z_2))$ and $\bar{\mathscr{E}}'(\hat{\boldsymbol{v}}(z_1, z_2)) = \operatorname{supp}(\hat{\boldsymbol{v}}(z_1, z_2)) \setminus \mathscr{E}'(\hat{\boldsymbol{v}}(z_1, z_2))$, respectively, and $\mathbf{v}_{\mathscr{E}'}$ and $\mathbf{v}_{\bar{\mathscr{E}}'}$ are defined accordingly. If no confusion arises we use $\mathscr{E}$ and $\mathscr{E}'$ for $\mathscr{E}(\hat{\boldsymbol{v}}(z_1, z_2))$ and $\mathscr{E}'(\hat{\boldsymbol{v}}(z_1, z_2))$, respectively.

Definition 3 Let $\mathscr{E}'$ be given with $(r^f, s^f) = (r+t, s)$ and $(r^\ell, s^\ell) = (r, s+t)$ being the first and last position (ordered by $\prec$) in this set. We define $\delta + 1$ different *balls* around $\mathscr{E}'$ as

$$\Omega_{j,\delta}(\mathscr{E}') = \Big\{(a,b) \mid a \le r^f + j,\ b \le s^\ell + j,\\ r^f + s^f + j - \delta \le a + b \le r^f + s^f + j\Big\}$$

for $j = 0, 1, 2, \dots, \delta$.

Example 4 Consider the burst of erasures given by

$$\mathscr{E}' = \{(8,5), (7,6), (6,7), (5,8), (4,9)\},$$

then, $(r^f, s^f) = (8, 5)$ and $(r^\ell, s^\ell) = (4, 9)$, and for $\delta = 3$ Figure 2 shows the balls $\Omega_{0,3}(\mathscr{E}')$ and $\Omega_{1,3}(\mathscr{E}')$.

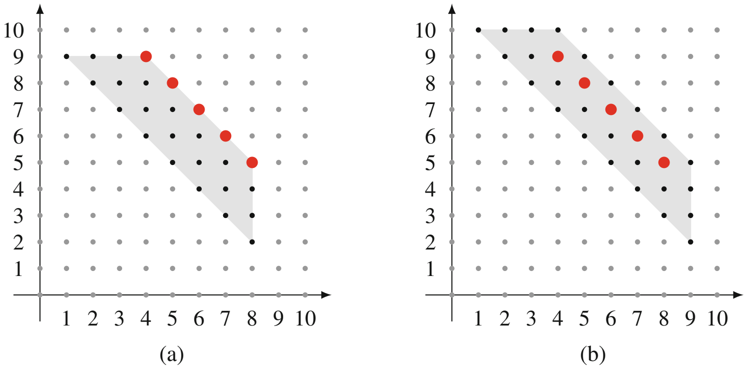


Fig. 2 Balls around the erasure given by $\mathscr{E}' = \{(8,5), (7,6), (6,7), (5,8), (4,9)\}$. (**a**) $\Omega_{0,3}(\mathscr{E}')$. (**b**) $\Omega_{1,3}(\mathscr{E}')$

By definition, the vector $\mathbf{v}_{\mathscr{E}'}$ contains a burst of erasures in a diagonal and $\mathbf{v}_{\bar{\mathscr{E}}'}$ may contain erasures as well. Depending on the structure of $\mathbf{H}_{\mathscr{E}'}$ and $\mathbf{H}_{\bar{\mathscr{E}}'}$ these errors may appear together in some of the equations of (7).

The following result gives a criterion to determine *some* sets of equations that involve *only* erasures in $\mathbf{v}_{\mathscr{E}'}$. The solution of such system would produce the desired decoding of $\mathbf{v}_{\mathscr{E}'}$.

Theorem 5 *Let* $\mathscr{C} = \ker_{\mathbb{F}[z_1,z_2]} \hat{H}(z_1,z_2)$, δ *the degree of* $\hat{H}(z_1,z_2)$ *and let* $\mathscr{E}$ *be the support of the erasures and* $\mathscr{E}'$ *be the support of a burst of erasures distributed on a diagonal line of a codeword* $\hat{\boldsymbol{v}}(z_1,z_2)$. *If* $\mathscr{E}'$ *are the only erasures in* $\Omega_{j,\delta}(\mathscr{E}')$, *i.e., if*

$$\mathscr{E} \cap \Omega_{j,\delta}(\mathscr{E}') = \mathscr{E}',$$

then, there exists a subsystem of (7) such that

$$\mathbf{H}^{j}_{\mathscr{E}'}\mathbf{v}_{\mathscr{E}'} = \boldsymbol{a}_j \tag{8}$$

where $\boldsymbol{a}_j$ *is a subvector of* $\mathbf{H}_{\bar{\mathscr{E}}'}\mathbf{v}_{\bar{\mathscr{E}}'}$ *that does not contain any erasures, and*

$$\mathbf{H}^{j}_{\mathscr{E}'} = \begin{bmatrix} H(j,0) & & & & & & \\ H(j-1,1) & H(j,0) & & & & & \\ H(j-2,2) & H(j-1,1) & H(j,0) & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ H(0,j) & H(1,j-1) & H(2,j-2) & & & & \\ & H(0,j) & H(1,j-1) & \cdots & H(j,0) & & \\ & & \ddots & \ddots & & \ddots & \\ & & & & H(0,j) & H(1,j-1) & \\ & & & & & H(0,j) & \end{bmatrix},$$

for $j = 0, 1, \dots, \delta$

is a $(n-k)(t+1+\delta-j) \times n(t+1)$ *submatrix of* $\mathbf{H}_{\mathscr{E}'}$, *with* $t+1$ *the cardinality of* $\mathscr{E}'$.

The structure of the matrices $\mathbf{H}^{j}_{\mathscr{E}'}$ have the same structure as the matrices in (5) which appear in the decoding problem of 1D convolutional codes, see [1, 8] for more details, and therefore the solution of (8) is analogous to the decoding problem of 1D convolutional codes.

It was shown in [10] that there exists a type of 1D convolutional codes, called (reverse or complete) MDP, that perform particularly well over the erasure channel. This together with Theorem 5 suggest that in order to construct a 2D convolutional code $\mathscr{C} = \ker_{\mathbb{F}[z_1,z_2]} \hat{H}(z_1,z_2)$ with good decoding properties one can construct a parity-check matrix $\hat{H}(z_1,z_2) = \sum_{0 \le a+b \le \delta} H(a,b) z_1^a z_2^b$ such that the associated 1D convolutional codes are given by $\mathscr{C}^{(j)} = \ker_{\mathbb{F}[z]} \hat{H}^{(j)}(z)$ with $\hat{H}^{(j)}(z) = H_0^{(j)} + H_1^{(j)}z + \cdots + H_\nu^{(j)}z^\nu$ and $H_k^{(j)} = H(j-k, k)$, for $k = 0, 1, \ldots, j$ are MDP.

4 Conclusions

In this paper we have proposed a method to recover erasures $\mathscr{E}'$ in a 2D (finite support) convolutional code that are distributed in a diagonal line in the 2D plane. We have shown that if $\mathscr{E}'$ does not have more erasures *close* (meaning in a ball centered around $\mathscr{E}'$) then it is possible to consider $\mathscr{E}'$ as a burst of erasures of a set of 1D convolutional codes. Decoding these 1D convolutional codes would immediately imply the recovery of the $\mathscr{E}'$.

This procedure is far from solving all the possible erasure patterns but it represents the first step toward the development of an effective approach to solve more general patterns of erasures.

Acknowledgements This work was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Ciencia e Innovación of the Gobierno de España. The work of D. Napp, R. Pinto, and R. Simões was partially supported by Portuguese funds through the CIDMA – Center for Research and Development in Mathematics and Applications, and the Portuguese Foundation for Science and Technology ("FCT-Fundação para a Ciência e a Tecnologia"), within project UID/MAT/04106/2013.

References

1. Almeida, P., Napp, D., Pinto, R.: A new class of superregular matrices and MDP convolutional codes. Linear Algebra Appl. **439**, 2145–2157 (2013). doi:10.1016/j.laa.2013.06.013
2. Arai, M., Yamamoto, A., Yamaguchi, A., Fukumoto, S., Iwasaki, K.: Analysis of using convolutional codes to recover packet losses over burst erasure channels. In: Proceedings of the 2001 Pacific Rim International Symposium on Dependable Computing, Seoul, pp. 258–265. IEEE (2001). doi:10.1109/PRDC.2001.992706
3. Charoenlarpnopparut, C.: Applications of Gröbner bases to the structural description and realization of multidimensional convolutional code. Sci. Asia **35**, 95–105 (2009). doi:10.2306/scienceasia1513-1874.2009.35.095
4. Climent, J.J., Napp, D., Perea, C., Pinto, R.: A construction of MDS 2D convolutional codes of rate $1/n$ based on superregular matrices. Linear Algebra Appl. **437**, 766–780 (2012). doi:10.1016/j.laa.2012.02.032
5. Fornasini, E., Valcher, M.E.: Algebraic aspects of two-dimensional convolutional codes. IEEE Trans. Inf. Theory **40**(4), 1068–1082 (1994). doi:10.1109/18.335967
6. Jangisarakul, P., Charoenlarpnopparut, C.: Algebraic decoder of multidimensional convolutional code: constructive algorithms for determining syndrome decoder and decoder matrix based on Gröbner basis. Multidimens. Syst. Signal Process. **22**(1–3), 67–81 (2011)
7. Napp, D., Perea, C., Pinto, R.: Input-state-output representations and constructions of finite support 2D convolutional codes. Adv. Math. Commun. **4**(4), 533–545 (2010). doi:10.3934/amc.2010.4.533

8. Tomás, V.: Complete-MDP convolutional codes over the erasure channel. Ph.D. thesis, Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, Alicante (2010)
9. Tomás, V., Rosenthal, J., Smarandache, R.: Reverse-maximum distance profile convolutional codes over the erasure channel. In: Edelmayer, A. (ed.) Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS2010), Budapest, pp. 2121–2127 (2010)
10. Tomás, V., Rosenthal, J., Smarandache, R.: Decoding of convolutional codes over the erasure channel. IEEE Trans. Inf. Theory **58**(1), 90–108 (2012). doi:10.1109/TIT.2011.2171530
11. Weiner, P.A.: Multidimensional convolutional codes. Ph.D. thesis, Department of Mathematics, University of Notre Dame, Indiana (1998)

Variations on Minimal Linear Codes

Gérard Cohen and Sihem Mesnager

Abstract Minimal linear codes are linear codes such that the support of every codeword does not contain the support of another linearly independent codeword. Such codes have applications in cryptography, e.g. to secret sharing. We pursue here their study and construct asymptotically good families of minimal linear codes. We also push further the study of quasi-minimal and almost-minimal linear codes, relaxations of the minimal linear codes.

Keywords Minimal codes • Quasi-minimal codes

1 Introduction

A *minimal codeword* [10, 11] c of a linear code C is a codeword such that its support (set of non-zero coordinates) does not contain the support of another linearly independent codeword. Minimal codewords are useful for defining access structures in secret sharing schemes using linear codes. Determining the set of minimal codewords is hard for general linear codes, although this has been studied for some classes of specific linear codes. This led to work on how to find codes where all codewords are minimal, in order to facilitate the choice of access structures. The problem of finding a code satisfying this condition, called a *minimal linear code* has first been envisioned in [7] and later studied in [3, 13].

Interestingly, in [3], the motivation for finding minimal linear codes is no longer secret sharing but in a new proposal for secure two-party computation, where it is required that minimal linear codes are used to ensure privacy.

G. Cohen (✉)
Institut Télécom, Télécom ParisTech, UMR 7539, CNRS, Paris Cedex 16, France
e-mail: cohen@telecom-paristech.fr; cohen@enst.fr

S. Mesnager
LAGA (Laboratoire Analyse, Géometrie et Applications), UMR 7539, CNRS, Department of Mathematics, University of Paris XIII and University of Paris VIII, Sorbonne Paris Cité, Telecom ParisTech, Paris, France
e-mail: smesnager@univ-paris8.fr

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_12

It is pointed out in [3] that minimal codes are close to the notions of intersecting and separating codes [4, 5]. Such codes have been suggested for applications to oblivious transfer [2], secret sharing [1, 7, 13] or digital fingerprinting [12].

We will focus here on the non-binary case, where the notion of minimal codes is more restrictive than that of separating codes. Secret-sharing and secure two-party computations both crucially hinge on a large alphabet; thus, one cannot rely on the well-understood binary case only. We thus pursue in Sect. 2 the study of [3] on bounds and criteria for minimal linear codes and exhibit families of minimal codes with better rates (asymptotically non-zero). In Sect. 3, we relax the notion of minimal codes and introduce *quasi-minimal* linear codes. Quasi-minimal linear codes are codes where two non-zero codewords have the same support if and only if they are linearly dependent. This slight relaxation enables to exhibit families with improved non-zero asymptotic rates. Finally, we consider yet another generalization to almost-minimal codes, where the property is allowed to fail for a small proportion of codewords.

2 Minimal Codes: Bounds and Constructions

2.1 Definitions: Notations

We denote by $|F|$ the cardinality of a set F. Let $q = p^h$, where p is a prime number and $h \in \mathbb{N}*$. An $[n, k, d, d_{max}]_q$ code is a vector subspace of $\mathbb{F}_q^n$ of dimension k with minimum distance d and maximum distance d_{max}. The last two parameters refer to the minimal (resp. maximal) Hamming distance between two codewords of $\mathscr{C}$, or, equivalently, the minimal (resp. maximal) Hamming weight of a codeword of $\mathscr{C}$; they will be omitted when irrelevant. Normalized parameters will be denoted by $R = k/n, \delta = d/n, \delta_{max} = d_{max}/n$.

The *support* of a codeword $c \in \mathscr{C}$ is $supp(c) = \{i \in \{1, \dots, n\} | c_i \neq 0\}$. The *Hamming weight* of a codeword $c \in \mathscr{C}$ denoted by $wt(c)$ is the cardinality of its support: $wt(c) = |supp(c)|$. A codeword c *covers* a codeword c' if $supp(c') \subset supp(c)$.

Definition 1 (Minimal codeword) A codeword c is *minimal* if it only covers $\mathbb{F}_q \cdot c$, i.e. if $\forall c' \in \mathscr{C}, (supp(c') \subset supp(c)) \implies (c, c')$ linearly dependent [10].

Definition 2 (Minimal linear code) A linear code $\mathscr{C}$ is *minimal* if every non-zero codeword $c \in \mathscr{C}$ is minimal [7].

For a complete treatment and general references in coding theory, we refer to the book of MacWilliams and Sloane [9].

2.2 Bounds

Two non-constructive bounds on the rates of minimal codes are exhibited in [3]. We recall them without proofs.

Theorem 3 (Maximal Bound) *Let $\mathscr{C}$ a minimal linear $[n, k, d]$ q-ary code, then $R \leq \log_q(2)$ [3].*

Theorem 4 (Minimal Bound) *For any R, $0 \leq R = k/n \leq \frac{1}{2}\log_q(\frac{q^2}{q^2-q+1})$, there exists an infinite sequence of $[n, k]$ minimal linear codes [3].*

2.3 A Sufficient Condition

There exists a sufficient condition on weights for a given linear code to be minimal. More precisely, if the weights of a linear code are close enough to each other, then each nonzero codeword of the code is a minimal vector as described by the following statement.

Proposition 5 ([1]) *Let $\mathscr{C}$ be an $[n, k, d, d_{max}]$ code. If $\frac{d}{d_{max}} > \frac{q-1}{q}$ then $\mathscr{C}$ is minimal.*

Remark 6 Note that the stronger sufficient condition $\frac{d}{n} > \frac{q-1}{q}$ fails to provide asymptotically good codes; indeed, by the Plotkin bound [9], for any code, not necessarily linear, of length n, size M and distance d, if $d > (q-1)n/q$, then $M \leq d/(d-(1-q^{-1}))$.

On the other hand, for $\delta < 1 - q^{-1}$, the classical Varshamov-Gilbert bound [8] guarantees the existence of asymptotic families of codes with non zero rate $R(\delta, q)$.

2.4 Infinite Constructions

The general idea is to concatenate a q-ary "seed" or inner code (e.g. a simplex) with an infinite family of algebraic-geometric (AG) codes (the outer codes) [14], in such a way as to obtain a high enough minimum distance and conclude by Proposition 5.

In practice, we can take the seed to be the simplex code $\mathscr{S}_{q,r}[n = (q^r - 1)/(q - 1), k = r, d = d_{max} = q^{r-1}]_q$ (with $\delta > (q-1)/q$), set $r = 2m$ and concatenate with $AG[N, K = NR, D = N\Delta, D_{max} = N\Delta_{max}]_{q^{2m}}$. These codes exist lying almost on the Singleton bound, namely satisfying $R + \Delta = 1 - (q^m - 1)^{-1} > (q-1)/q$.

This concatenation results in the family $C[nN, kK, dD]_q$ with maximum distance at most $d_{max}N$. If $dD/d_{max}N = \Delta > (q-1)/q$, this family is minimal by Proposition 5.

It is not hard to check that, for example, choosing q large and α small enough, $m \geq 2, \Delta = (q-1)/q + \alpha, R = 1/q - 1/(q^m - 1) - \alpha > 0$, this is the case.

To summarize, we construct infinite families of codes with $R = 2m(1/q - 1/(q^m - 1) - \alpha)(q-1)/(q^{2m} - 1) \approx 2m/q^{2m}$ satisfying $\delta/\delta_{max} > (q-1)/q$, thus minimal. Note that, by the Plotkin bound, they necessarily satisfy $\delta < (q-1)/q$, so the fact that $\delta_{max} < 1$ is crucial.

3 Quasi-minimal Codes

We now relax the notion of minimal codes to that of *quasi-minimal* codes. Minimality prevents a codeword from having its support included in the support of a linearly independent codeword, whereas quasi-minimality only prevents two linearly independent codewords from having the same support.

3.1 Definitions and Properties

Definition 7 (Quasi-minimal codeword) A codeword c is *quasi-minimal* if $\forall c' \in \mathscr{C}, (supp(c') = supp(c)) \implies (c, c')$ linearly dependent.

Definition 8 (Quasi-minimal linear code) A linear code $\mathscr{C}$ is *quasi-minimal* if every non-zero codeword $c \in \mathscr{C}$ is quasi-minimal.

Quasi-minimality is clearly a weaker requirement than minimality. For instance, every binary code is obviously quasi-minimal.

3.2 Constructions

We now give a construction based on the Kronecker (tensor) product of codes, which yields infinite families of quasi-minimal codes with relatively slowly decreasing rates.

Proposition 9 *The product $\mathscr{C}_1 \otimes \mathscr{C}_2$ of a quasi-minimal $[n_1, k_1, d_1, (d_{max})_1]_q$ code $\mathscr{C}_1$ and of a quasi-minimal $[n_2, k_2, d_2, (d_{max})_2]_q$ code $\mathscr{C}_2$ is a quasi-minimal $[n_1 \times n_2, k_1 \times k_2, d_1 \times d_2, d_{max} \geq (d_{max})_1 \times (d_{max})_2]_q$ code.*

Proof The parameters are easy to check. For the quasi-minimality, let $c \neq 0, c'$ be two codewords of $\mathscr{C}_1 \otimes \mathscr{C}_2$. By definition of the tensor product, they can both be written as $n_1 \times n_2$ matrices where rows are codewords of $\mathscr{C}_1$ and columns are codewords of $\mathscr{C}_2$. More generally, the square of the $[q+1, 2, q]_q$ simplex code is a $[(q+1)^2, 4, q^2]_q$ minimal code. Let us assume that $supp(c) = supp(c')$. For

$i = 1, \dots, n_1$, $j = 1, \dots, n_2$ let c_i^1 (resp. $c_i'^1$) be the ith row of c (resp. c') and c_j^2 (resp. $c_j'^2$) be the jth column of c (resp. c'). For every i, $supp(c_i^1) = supp(c_i'^1)$, so $\exists\lambda_i$ such that $c_i'^1 = \lambda_i c_i^1$. With the same reasoning on the columns, for every j, there exists λ_j such that $c_j'^2 = \lambda_j c_j^2$. Then, all the λ_i's and λ_j's are equal and there exists λ such that $c' = \lambda c$, so c and c' are linearly dependent. Thus, $\mathscr{C}_1 \otimes \mathscr{C}_2$ is quasi-minimal. □

3.3 A Sufficient Condition

We now prove a sufficient condition for quasi-minimality, weaker than the one for minimality. This will then allow us to construct improved infinite classes of asymptotically good quasi-minimal codes by concatenation.

Theorem 10 *Let C be a linear $[n, k, d, d_{max}]_q$ code; if $d/d_{max} \geq (q-2)/(q-1)$, then C is quasi-minimal.*

Proof Let C be a linear $[n, k, d]_q$ code and let c, c' be two linearly independent codewords of C such that $supp(c) = supp(c')$. Let α be a primitive element of $\mathbb{F}_q$. Then, w.l.o.g., one can write c and c' by blocks, in the following way: $c = \beta_0 || \dots || \beta_{q-2} || 0$ and $c' = \alpha^0 \beta_0 || \dots || \alpha^{q-2} \beta_{q-2} || 0$. Let A_i be the size of the (possibly empty) block β_i. Then $wt(c) = wt(c') = \sum_{i=0}^{q-2} A_i \geq d$. We also have, for $j = 0, \dots, q-2$, $d(\alpha^j c, c') = \sum_{i \neq j} A_i \geq d$. If we sum all these inequalities, we get $(q-2) \sum_{i=0}^{q-2} A_i \geq (q-1)d$, hence $wt(c) \geq \frac{q-1}{q-2} d > d_{max}$, a contradiction. Thus, c and c' cannot exist and C is quasi-minimal. □

Example 11 For $q = 3$, consider the code $G[11, 5, 6, 9]_3$ obtained by shortening the extended ternary Golay code [9]. It is quasi-minimal by the previous theorem. Its (Kronecker) square is G^2, a $[121, 25, 36, \geq 81]_3$ quasi-minimal code by the previous proposition, although is does not satisfy the sufficient condition of Theorem 10.

Now, the celebrated non-constructive Varshamov-Gilbert bound implies the existence of infinite families of semi-constructive quasi-minimal codes with rate $R = 1 - h_q(\frac{q-2}{q-1}) > 0$. This is still far from the upper bound, derived analogously to the minimal case:

Theorem 12 (Maximal Bound) *Let $\mathscr{C}$ be a quasi-minimal linear $[n, k, d]_q$ code, then $R \leq \log_q(2)$.*

3.4 Infinite Constructions of Quasi-minimal Codes

Again, we concatenate a q-ary inner code (e.g. a simplex) with an infinite family of algebraic-geometric (AG) codes to get a high enough minimum distance and conclude by Theorem 10.

Continue taking for seed $\mathscr{S}_{q,r}[n = (q^r - 1)/(q-1), k = r, d = d_{max} = q^{r-1}]_q$, set $r = 2m$ and concatenate with $AG[N, K = NR, D = N\Delta]_{q^{2m}}$, obtaining the family $C[nN, kK, dD]_q$. Analogously to the minimal case, If $dD/d_{max}N = \Delta > (q-2)/(q-1)$, this family is quasi-minimal by Theorem 10.

Example 13
- Take $q = 4, \mathscr{S}_{4,4}[85, 4, 64]_4, \Delta = 2/3 + \alpha, R = 4/15$, resulting in an infinite construction of $[n, 16n/1275]$ quaternary codes.
- For $q = 3$, we can improve on the simplex code seed: indeed, take the already considered $C[11, 5, 6, 9]_3$ as inner code and $AG[N, NR, N\Delta]_{3^5}$ with $R + \Delta = 191/208$. Choose $\Delta = 3/4, R = 35/208$; then concatenation results in an infinite construction of quasi-minimal $[n, \approx 0.076n]$ ternary codes.

4 Almost-Minimal Codes

Definition 14 (Almost-minimal linear code) A linear code $\mathscr{C}$ is said (ϵ)*almost-minimal* if at most $q^{2\epsilon k}$ pairs of codewords are bad, for some fixed ϵ with $0 \leq \epsilon < 1/2$.

We now extend some results of [6] to almost-minimal codes.

Theorem 15 (Maximal Bound) *Let $\mathscr{C}$ an almost- minimal linear $[n, k, d]$ q-ary code, then $R \leq \log_q(2)/(1-\epsilon) + o(1)$.*

Proof By definition, at most $q^{\epsilon k+1}$ codewords can share the same support. Thus, $|\mathscr{C}| = q^k \leq q^{\epsilon k+1}2^n$ and $R = k/n \leq \log_q(2)/(1-\epsilon) + o(1)$. □

Theorem 16 (Minimal Bound) *For any positive $R = k/n$ such that*

$$R \leq \frac{1}{2-2\epsilon} \log_q\left(\frac{q^2}{q^2 - q + 1}\right) + o(1),$$

there exists an infinite sequence of $[n, k]$ almost-minimal linear codes.

Proof Let us fix n and k. For $a \in \mathbb{F}_q^n$, such that $|supp(a)| = i$, there are $q^i - q$ linearly independent vectors b such that $supp(b) \subset supp(a)$. The pair (a, b) belongs to $\begin{bmatrix} n-2 \\ k-2 \end{bmatrix}$ linear $[n, k]$ codes, where $\begin{bmatrix} x \\ k \end{bmatrix}$ denotes the q-ary Gaussian binomial coefficient. There are less than

$\sum_{i=0}^{n} \binom{n}{i}(q-1)^i(q^i-q) = (1+(q-1)q)^n - q^{n+1} \leq (q^2-q+1)^n$ such ordered bad (a,b) pairs. As long as $q^{2\epsilon k}\begin{bmatrix} n \\ k \end{bmatrix} \geq \begin{bmatrix} n-2 \\ k-2 \end{bmatrix}(q^2-q+1)^n$, there are linear $[n,k]$ codes containing no more than $q^{2\epsilon k}$ bad pairs, i.e. almost-minimal codes. For $k/n \leq \frac{1}{2-2\epsilon}\log_q(\frac{q^2}{q^2-q+1}) + o(1)$, this quantity is positive. □

4.1 Open Problem

Is it true that the best achievable rate of (quasi, almost) minimal codes is a decreasing function of q? A weaker statement holds: if q divides q', then a q'- (quasi, almost) minimal code yields a q-ary (quasi, almost) minimal code with the same rate.

Acknowledgements We thank Alexander Barg, Alain Patey and Zachi Tamo for helpful discussions.

References

1. Ashikhmin, A.E., Barg, A.: Minimal vectors in linear codes. IEEE Trans. Inf. Theory **44**(5), 2010–2017 (1998)
2. Brassard, G., Crépeau, C., Santha, M.: Oblivious transfers and intersecting codes. IEEE Trans. Inf. Theory **42**(6), 1769–1780 (1996)
3. Chabanne, H., Cohen, G., Patey, A.: Towards secure two-party computation from the wire-tap channel (2013). arXiv:1306.6265
4. Cohen, G.D., Lempel, A.: Linear intersecting codes. Discret. Math. **56**(1), 35–43 (1985)
5. Cohen, G.D., Encheva, S.B., Litsyn, S., Schaathun, H.G.: Intersecting codes and separating codes. Discret. Appl. Math. **128**(1), 75–83 (2003)
6. Cohen, G., Mesnager, S., Patey, A.: On minimal and quasi-minimal linear codes. In: Proceedings of Fourteenth International Conference on Cryptography and Coding (IMACC 2013), Oxford. Lecture Notes in Computer Science, vol. 8308, pp. 85–98. Springer, Heidelberg (2013)
7. Ding, C., Yuan, J.: Covering and secret sharing with linear codes. In: Calude, C., Dinneen, M.J., Vajnovszki, V. (eds.) Discrete Mathematics & Theoretical Computer Science, DMTCS Dijon 2003. Lecture Notes in Computer Science, vol. 2731, pp. 11–25. Springer, Berlin (2003)
8. Gilbert, E.N.: A comparison of signalling alphabets. Bell Syst. Tech. J. **31**(3), 504–522 (1952)
9. MacWilliams, F.J., Sloane, N.J.: The theory of error-correcting codes. North Holland, Amsterdam (1977)
10. Massey, J.L.: Minimal codewords and secret sharing. In: Proceedings of 6th Joint Swedish-Russian International Workshop on Information Theory, Molle, pp. 276–279 (1993)
11. Massey, J.L.: Some applications of coding theory in cryptography. In: Farrell, P.G. (ed.) Codes and Cyphers: Cryptography and Coding IV, pp. 33–47. Formara Ltd, Southend-on-Sea (1995)
12. Schaathun, H.G.: The Boneh-Shaw fingerprinting scheme is better than we thought. IEEE Trans. Inf. Forensics Secur. **1**(2), 248–255 (2006)
13. Song, Y., Li, Z.: Secret sharing with a class of minimal linear codes (2012). arXiv:1202.4058
14. Tsfasman, M.A., Vladut, S.G.: Algebraic-Geometric Codes. Kluwer, Dordrecht (1991)

Cryptanalysis of Public-Key Cryptosystems That Use Subcodes of Algebraic Geometry Codes

Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan

Abstract We give a polynomial time attack on the McEliece public key cryptosystem based on subcodes of algebraic geometry (AG) codes. The proposed attack reposes on the distinguishability of such codes from random codes using the Schur product. Wieschebrink treated the genus zero case a few years ago but his approach cannot be extent straightforwardly to other genera. We address this problem by introducing and using a new notion, which we call the *t–closure* of a code.

Keywords Algebraic geometry codes • Code-based cryptography • Schur products of codes • Distinguishers

1 Introduction

After the original proposal of code based encryption scheme due to McEliece [15] which was based on binary Goppa codes, several alternative proposals aimed at reducing the key size by using codes with a higher correction capacity. Among many others, generalised Reed–Solomon (GRS) codes are proposed in 1986 by Niederreiter [17] but are subject to a key-recovery polynomial time attack discovered by Sidelnikov and Shestakov [21] in 1992. To avoid this attack, Berger and Loidreau [1] proposed to replace GRS codes by some random subcodes of small codimension. This proposal has been broken by Wieschebrink [24] using Schur products of codes.

Another proposal was to use algebraic geometry (AG) codes, concatenated AG codes or their subfield subcodes [9]. The case of AG codes of genus 1 and 2 has been broken by Faure and Minder [6]. Then, Marquez et al. proved that the structure of a curve can be recovered from the very knowledge of an AG code [13, 14]

A. Couvreur (✉) • I. Márquez-Corbella (✉)
INRIA, SACLAY & LIX, CNRS UMR 7161, École Polytechnique, 91128 Palaiseau Cedex, France
e-mail: alain.couvreur@lix.polytechnique.fr; irene.marquez-corbella@inria.fr

R. Pellikaan (✉)
Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands
e-mail: g.r.pellikaan@tue.nl

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_13

without leading to an efficient attack. Finally a polynomial time attack of the scheme based on AG codes has been obtained by the authors in [3]. This attack consists in using the particular behaviour of AG codes with respect to the Schur product to compute a filtration of the public key by AG subcodes, which leads to the design of a polynomial time decoding algorithm allowing encrypted message recovery.

The genus zero case and Berger Loidreau's proposal raises a natural question **what about using subcodes of AG codes?** In this article we propose an attack of this scheme. Compared to the genus zero case, Wieschebrink's attack cannot extend straightforwardly and we need to introduce and use a new notion which we call the *t–closure* of a code. By this manner, we prove subcodes of AG codes to be non secure when the subcode has a small codimension. It is worth noting that choosing a subcode of high codimension instead of the code itself represents a huge loss in terms of error correction capacity and hence is in general a bad choice. For this reason, an attack on the small codimension codes is of interest.

Finally, it hardly needs to be recalled that this result does not imply the end of code-based cryptography since Goppa codes, alternant codes and more generally subfield subcodes of AG codes still resist to any known efficient attack. Their resistance to the presented attack is discussed at the end of the article.

Due to space reasons, many proofs are omitted in this extended abstract.

2 Notation and Prerequisites

2.1 Curves and Algebraic Geometry Codes

The interested reader is referred to [22, 23] for further details on the notions introduced in the present subsection. In this article, $\mathscr{X}$ denotes a smooth projective geometrically connected curve of genus g over a finite field $\mathbb{F}_q$. We denote by $P = (P_1, \dots, P_n)$ an n-tuple of mutually distinct $\mathbb{F}_q$-rational points of $\mathscr{X}$, by D_P the divisor $D_P = P_1 + \cdots + P_n$ and by E an $\mathbb{F}_q$-divisor of degree $m \in \mathbb{Z}$ and support disjoint from that of D_P.

The function field of $\mathscr{X}$ is denoted by $\mathbb{F}_q(\mathscr{X})$. Given an $\mathbb{F}_q$-divisor E on $\mathscr{X}$, the corresponding Riemann-Roch space is denoted by $L(E)$. The *algebraic geometry (AG) code* $\mathscr{C}_L(\mathscr{X}, P, E)$ of length n over $\mathbb{F}_q$ is the image of the evaluation map

$$\mathrm{ev}_P : \begin{cases} L(E) \longrightarrow & \mathbb{F}_q^n \\ \quad f \quad \longmapsto & (f(P_1), \dots, f(P_n)) \end{cases}$$

If $2g - 2 < m < n$, then by Riemann-Roch Theorem, $\mathscr{C}_L(\mathscr{X}, P, E)$ has dimension $m + 1 - g$ and minimum distance at least $n - m$.

When the curve is the projective line ¶^{1}, the corresponding codes are the so-called *generalised Reed–Solomon* (GRS) codes defined as:

$$\mathrm{GRS}_k(\mathbf{a}, \mathbf{b}) := \{(b_1 f(a_1), \dots, b_n f(a_n)) \mid f \in \mathbb{F}_q[x]_{<k}\}.$$

where $\mathbf{a}, \mathbf{b}$ are two n–tuples in $\mathbb{F}_q^n$ such that the entries of $\mathbf{a}$ are pairwise distinct and those of $\mathbf{b}$ are all nonzero and $k < n$.

Remark 1 See [8, Example 3.3] for a description of GRS codes as AG codes.

2.2 Schur Product

Given two elements $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{F}_q^n$, the *Schur product* is the component wise multiplication: $\mathbf{a} * \mathbf{b} = (a_1 b_1, \dots, a_n b_n)$. Let $\mathbf{a} \in \mathbb{F}_q^n$, we set $\mathbf{a}^0 := (1, \dots, 1)$ and by induction we define $\mathbf{a}^{j+1} := \mathbf{a} * \mathbf{a}^j$ for any positive integer j. If all entries of $\mathbf{b}$ are nonzero, we define $\mathbf{b}^{-1} := (b_1^{-1}, \dots, b_n^{-1})$ and thus, $\mathbf{b}^{-j} = \left(\mathbf{b}^j\right)^{-1}$ for any positive integer j.

For two codes $A, B \subseteq \mathbb{F}_q^n$, the code $A * B$ is defined by

$$A * B := \mathrm{Span}_{\mathbb{F}_q} \{\mathbf{a} * \mathbf{b} \mid \mathbf{a} \in A \text{ and } \mathbf{b} \in B\}.$$

For $B = A$, then $A * A$ is denoted as $A^{(2)}$ and, we define $A^{(t)}$ by induction for any positive integer t.

2.2.1 Application to Decoding, Error Correcting Pairs and Arrays

The notion of *error-correcting pair* (ECP) for a linear code was introduced by Pellikaan [18, 19] and independently by Kötter [10]. Broadly speaking, given a positive integer t, a t–ECP for a linear code $\mathscr{C} \subseteq \mathbb{F}_q^n$ is a pair of linear codes (A, B) in $\mathbb{F}_q^n$ satisfying $A * B \subseteq \mathscr{C}^\perp$ together with several inequalities relating t and the dimensions and (dual) minimum distances of A, B, C. This data provides a decoding algorithm correcting up to t errors in $O(n^3)$ operations in $\mathbb{F}_q$. ECP's provide a unifying point of view for several classical bounded distance decoding for algebraic and AG codes. See [11] for further details.

For an AG code, there always exists a t–ECP with $t = \lfloor \frac{d^* - 1 - g}{2} \rfloor$, where d^* denotes the *Goppa designed distance* (see [22, Definition 2.2.4]). Thus, ECP's allow to correct up to half the designed distance minus $g/2$. Filling this gap and correct up to half the designed distance is possible thanks to more elaborate algorithms based on the so-called *error correcting arrays*. See [5, 7] for further details.

2.2.2 Distinguisher and Cryptanalysis

Another and more recent application of the Schur product concerns cryptanalysis of code-based public key cryptosystems. In this context, the Schur product is a very powerful operation which can help to distinguish some algebraic codes such as AG codes from random ones. The point is that evaluation codes do not behave like random codes with respect to the Schur product: the square of an AG code is very small compared to that of a random code of the same dimension. Thanks to this observation, Wieschebrink [24] gave an efficient attack of Berger Loidreau's proposal [1] based on subcodes of GRS codes.

Recent attacks consist in pushing this argument forward and take advantage to this distinguisher in order to compute a filtration of the public code by a family of very particular subcodes. This filtration method yields an alternative attack on GRS codes [2]. Next it leads to a key recovery attack on wild Goppa codes over quadratic extensions in [4]. Finally in the case of AG codes, this approach lead to an attack [3] which consists in the computation of an ECP for the public code without retrieving the structure of the curve, the points and the divisor.

3 The Attack

Our public key is a non structured generator matrix $\mathbf{G}$ of a subcode C of $\mathscr{C}_L(\mathscr{X}, P, E)^{\perp}$ of dimension l, together with the error correcting capacity t. The goal of our attack is to recover the code $\mathscr{C}_L(\mathscr{X}, P, E)^{\perp}$ from the knowledge of C and then use the attack of [3] which provides a t–ECP and hence a decoding algorithm for $\mathscr{C}_L(\mathscr{X}, P, E)$, which yields a fortiori a decoding algorithm for C.

The genus zero case (i.e. the case of GRS codes) proposed in [1] was broken by Wieschebrink [24] as follows:

- C is the public key contained in some secret $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})$.
- Compute $C^{(2)}$ which is, with a high probability, equal to $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})^{(2)}$, which is itself equal to $\mathrm{GRS}_{2k-1}(\mathbf{a}, \mathbf{b}^2)$.
- Apply Sidelnikov Shestakov attack [21] to recover $\mathbf{a}$ and $\mathbf{b}^2$, then find $\mathbf{b}$.

Compared to Wieschebrink's approach, our difficulty is that the attack [3] is not a key-recovery attack but a blind construction of a decoding algorithm. For this reason, even if $C^{(2)}$ provides probably the code $\mathscr{C}_L(\mathscr{X}, P, E)^{(2)}$, it is insufficient for our purpose: we need to find $\mathscr{C}_L(\mathscr{X}, P, E)$. This is the reason why we introduce the notion of t–closures.

3.1 The *t*-Closure Operation

Definition 2 (*t*–closure) Let $C \subset \mathbb{F}_q^n$ be a code and $t \geq 2$ be an integer. The *t-closure* of C is defined by

$$\overline{C}^t = \left\{ \mathbf{a} \in \mathbb{F}_q^n \mid \mathbf{a} * C^{(t-1)} \subseteq C^{(t)} \right\}.$$

The code C is said to be *t- closed* if $\overline{C}^t = C$.

Proposition 3 *Let $C \in \mathbb{F}_q^n$, then for all $t \geq 2$,*

$$\overline{C}^t = \left(C^{(t-1)} * \left(C^{(t)}\right)^\perp\right)^\perp.$$

Proposition 4 *Let E be a divisor satisfying* $\deg(E) \geq 2g + 1$. *Then:*

(i) $\mathscr{C}_L(\mathscr{X}, P, E)^{(t)} = \mathscr{C}_L(\mathscr{X}, P, tE)$.
(ii) $\overline{\mathscr{C}_L(\mathscr{X}, P, E)}^t = \mathscr{C}_L(\mathscr{X}, P, E)$ *if* $\deg(E) \leq \frac{n-2}{t}$.

Proof (i) is proved in [3] and is a consequence of [16]. For (ii), Proposition 3 shows that

$$\overline{\mathscr{C}_L(\mathscr{X}, P, E)}^t = \left(\mathscr{C}_L(\mathscr{X}, P, E)^{(t-1)} * \left(\mathscr{C}_L(\mathscr{X}, P, E)^{(t)}\right)^\perp\right)^\perp. \tag{1}$$

Moreover, $\mathscr{C}_L(\mathscr{X}, P, tE)^\perp = \mathscr{C}_L(\mathscr{X}, P, (tE)^\perp)$ where $(tE)^\perp = D_P - tE + K$ for some canonical divisor K on $\mathscr{X}$. Thus, $\deg\left((tE)^\perp\right) = n - \deg(tE) + 2g - 2$. Since, by assumption, $\deg(E) \leq \frac{n-2}{t}$ we have $\deg\left((tE)^\perp\right) \geq 2g$. Moreover, since $\deg E \geq 2g + 1$, then, thanks to (i), Eq. (1) yields

$$\begin{aligned}\mathscr{C}_L(\mathscr{X}, P, (t-1)E) * \mathscr{C}_L(\mathscr{X}, P, tE)^\perp &= \mathscr{C}_L(\mathscr{X}, P, D_P - E + K)\\ &= \mathscr{C}_L(\mathscr{X}, P, E)^\perp.\end{aligned}$$

□

Corollary 5 *Let E be a divisor and $2g + 1 \leq \deg(E) \leq \frac{n-2}{2}$. Then $\overline{\mathscr{C}_L(\mathscr{X}, P, E)}^2 = \mathscr{C}_L(\mathscr{X}, P, E)$.*

Conjecture 6 If $2g + 1 \leq \deg(E) \leq \frac{n-1}{2}$, let C be subcode of $\mathscr{C}_L(\mathscr{X}, P, E)$ of dimension l such that and $2k + 1 - g \leq \binom{l+1}{2}$, where $k = \deg(E) + 1 - g$ is the dimension of $\mathscr{C}_L(\mathscr{X}, P, E)$, then the probability that $C^{(2)}$ is different from $\mathscr{C}_L(\mathscr{X}, P, 2E)$ tends to 0 when k tends to infinity.

We give a proof along the lines of [12, Remark 5] for the special case of subcodes of GRS codes. Our experimental results are in good agreement with this conjecture (see Table 1). The following corollary is central to our attack.

Corollary 7 *If $2g + 1 \leq \deg(E) \leq \frac{n-2}{2}$ and $2k + 1 - g \leq \binom{l+1}{2}$ for $k = \deg(E) + 1 - g$, then the equality $\overline{C}^2 = \mathscr{C}_L(\mathscr{X}, P, E)$ holds for random l-dimensional subcodes C of $\mathscr{C}_L(\mathscr{X}, P, E)$ with a probability tending to 0 when k tends to infinity.*

Table 1 Running times of the attack over Hermitian codes

q	n	k	t	`Time`	Key size	**w**	l	L
7^2	343	193	54	80 s	83 ko	2^{30}	50	1000
					137 ko	2^{43}	100	1000
					163 ko	2^{62}	150	1000
9^2	729	521	19	30 min	216 ko	2^{32}	50	500
					670 ko	2^{121}	200	500
					835 ko	2^{178}	400	500

3.2 Principle of the Attack

The public key consists in $C \subseteq \mathscr{C}_L(\mathscr{X}, P, E)^\perp$ and $t = \left\lfloor \frac{d^* - g - 1}{2} \right\rfloor$. Set $l := \dim C$. First, let us assume moreover that

$$2g + 1 \leq \deg(E) \leq \tfrac{n-1}{2},\ k = \deg(E) + 1 - g \ \text{ and } \ 2k - 1 + g \leq \tbinom{l+1}{2}.$$

Step 1. With a high probability, we may assume that $C^{(2)} = \mathscr{C}_L(\mathscr{X}, P, 2E)$ and hence $\overline{C}^2 = \mathscr{C}_L(\mathscr{X}, P, E)$ by Corollary 7. Thus, compute $\overline{C}^2$ by solving a linear system or by applying Proposition 3.
Step 2. Apply the polynomial time attack presented in [3] to obtain an ECP, denoted by (A, B), for $\mathscr{C}_L(\mathscr{X}, P, E)$. Which yields a decoding algorithm for C.
Estimated complexity: The computation of a closure costs $O(n^4)$ operations in $\mathbb{F}_q$ and the rest of the attack is in $O((\log(t + g))n^4)$ (see [3] for further details).

In case $\deg(E) > \frac{n-1}{2}$, then the attack can be applied to several shortenings of C whose 2–closures are computed separately and are then summed up to provide $\mathscr{C}_L(\mathscr{X}, P, E)$. This method is described and applied in [3, 4].

This attack has been implemented with MAGMA. To this end L random subcodes of dimension l from Hermitian codes of parameters $[n, k]_q$ were created. It turned out that for all created subcodes a t-ECP could be reconstructed. `Time` represents the average time of the attack obtained with an Intel ® CoreTM 2 Duo 2.8 GHz. The work factor **w** of an ISD attack is given. These work factors have been computed thanks to Christiane Peter's Software [20].

3.3 Which Codes Are Subject to This Attack?

Basically, the subcode $C \subseteq \mathscr{C}_L(\mathscr{X}, P, E)$ should satisfy:

(i) $\binom{\dim C+1}{2} \geq \dim \mathscr{C}_L(\mathscr{X}, P, 2E)$;
(ii) $2g + 1 \leq \deg E \leq \frac{n-2}{2}$;

The left-hand inequality of (ii) is in general satisfied. On the other hand, as explained above, the right-hand inequality of (ii) can be relaxed by using a shortening trick.

Constraint (i) is more central since a subcode which does not satisfies it will probably behave like a random code and it can be checked that a random code is in general 2–closed. Thus, computing the 2–closure of such a subcode will not provide any significant result. On the other hand, for an AG code of dimension k, subcodes which do not satisfy (i) have dimension smaller than $\sqrt{2k}$ and choosing such very small subcodes and decode them as subcodes of $\mathscr{C}_L(\mathscr{X}, P, E)$ would represent a big loss of efficiency. In addition, if these codes are too small they can be subject to generic attacks like information set decoding.

3.3.1 Subfield Subcodes Still Resist

Another class of subcodes which resist to this attack are the subcodes C such that $\overline{C}^2 \subsetneq \mathscr{C}_L(\mathscr{X}, P, E)$. It is rather difficult to classify such subcodes but there is a very identifiable family: the subfield subcodes. Let $\mathbb{F}$ be a proper subfield of $\mathbb{F}_q$ (here we assume q to be non prime) and let $C := \mathscr{C}_L(\mathscr{X}, P, E) \cap \mathbb{F}^n$ (and then apply a base field extension if one wants to have an $\mathbb{F}_q$–subcode). The point is that $C^2 \subseteq (\mathscr{C}_L(\mathscr{X}, P, E)^{(2)}) \cap \mathbb{F}_q^n$ and the 2-closure of C will in general differ from $\mathscr{C}_L(\mathscr{X}, P, E)$. For this reason, subfield subcodes resist to this kind of attacks. Notice that even in genus zero: subfield subcodes of GRS codes still resist to filtration attacks unless for the cases presented in [4].

References

1. Berger, T., Loidreau, P.: How to mask the structure of codes for a cryptographic use. Des. Codes Cryptogr. **35**(2), 63–79 (2005)
2. Couvreur, A., Gaborit, P., Gauthier-Umaña, V., Otmani, A., Tillich, J.P.: Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. Des. Codes Cryptogr. **73**(2), 641–646 (2014)
3. Couvreur, A., Márquez-Corbella, I., Pellikaan, R.: A polynomial time attack against algebraic geometry code based public key cryptosystems. Proceedings 2014 IEEE International Symposium on Information Theory, pp. 1446–1450 (2014)
4. Couvreur, A., Otmani, A., Tillich, J.P.: Polynomial time attack on wild McEliece over quadratic extensions. In: Nguyen, P., Oswald, E. (eds.) Advances in Cryptology – EUROCRYPT 2014. Lecture Notes in Computational Science, vol. 8441, pp. 17–39. Springer, Berlin/Heidelberg (2014)
5. Duursma, I.M.: Majority coset decoding. IEEE Trans. Inf. Theory **39**(3), 1067–1070 (1993)
6. Faure, C., Minder, L.: Cryptanalysis of the McEliece cryptosystem over hyperelliptic codes. In: ACCT 2008, Pamporovo, pp. 99–107 (2008)
7. Feng, G.L., Rao, T.: Decoding algebraic-geometric codes up to the designed minimum distance. IEEE Trans. Inf. Theory **39**(1), 37–45 (1993)
8. Høholdt, T., Pellikaan, R.: On the decoding of algebraic-geometric codes. IEEE Trans. Inf. Theory **41**(6, part 1), 1589–1614 (1995)
9. Janwa, H., Moreno, O.: McEliece public cryptosystem using algebraic-geometric codes. Des. Codes Cryptogr. **8**, 293–307 (1996)
10. Kötter, R.: A unified description of an error locating procedure for linear codes. In: Proceedings of Algebraic and Combinatorial Coding Theory, Voneshta Voda, pp. 113–117 (1992)

11. Márquez-Corbella, I., Pellikaan, R.: Error-correcting pairs for a public-key cryptosystem. In: Code-Based Cryptography Workshop 2012, Lyngby (2012). arXiv:1205.3647.
12. Márquez-Corbella, I., Martínez-Moro, E., Pellikaan, R.: The non-gap sequence of a subcode of a generalized Reed-Solomon code. Des. Codes Cryptogr. **66**(1–3), 317–333 (2013)
13. Márquez-Corbella, I., Martínez-Moro, E., Pellikaan, R.: On the unique representation of very strong algebraic geometry codes. Des. Codes Cryptogr. **70**(1–2), 215–230 (2014)
14. Márquez-Corbella, I., Martínez-Moro, E., Pellikaan, R., Ruano, D.: Computational aspects of retrieving a representation of an algebraic geometry code. J. Symb. Comput. **64**(0), 67–87 (2014). Mathematical and computer algebra techniques in cryptology
15. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DSN Progress Report 42–44, pp. 114–116 (1978)
16. Mumford, D.: Varieties defined by quadratic equations. In: Questions on Algebraic Varieties, C.I.M.E., III Ciclo, Varenna, 1969, pp. 29–100. Edizioni Cremonese, Rome (1970)
17. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Probl. Control Inf. Theory **15**(2), 159–166 (1986)
18. Pellikaan, R.: On decoding linear codes by error correcting pairs (1988). Preprint Technical University Eindhoven
19. Pellikaan, R.: On decoding by error location and dependent sets of error positions. Discret. Math. **106–107**, 369–381 (1992)
20. Peters, C.: Information-set decoding for linear codes over F_q. In: Post-Quantum Cryptography: Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010, Proceedings, Lecture Notes in Computer Science **6061**, pp. 81–94 (2010)
21. Sidelnikov, V.M., Shestakov, S.O.: On the insecurity of cryptosystems based on generalized Reed-Solomon codes. Discret. Math. **2**, 439–444 (1992)
22. Stichtenoth, H.: Algebraic Function Fields and Codes. Graduate Texts in Mathematics, vol. 254, 2nd edn. Springer, Berlin (2009)
23. Tsfasman, M., Vlăduţ, S., Nogin, D.: Algebraic Geometric Codes: Basic Notions. Mathematical Surveys and Monographs, vol. 139. American Mathematical Society, Providence (2007)
24. Wieschebrink, C.: Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In: Post-Quantum Cryptography. Lecture Notes in Computational Science, vol. 6061, pp. 61–72. Springer, Berlin/Heidelberg (2010)

Extending Construction X for Quantum Error-Correcting Codes

Akshay Degwekar, Kenza Guenda, and T. Aaron Gulliver

Abstract In this paper we extend the work of Lisonek and Singh on construction X for quantum error-correcting codes to finite fields of order p^2 where p is prime. Further, we give some new results on the Hermitian dual of repeated root cyclic codes. These results are used to construct new quantum error-correcting codes.

Keywords Quantum codes • Construction X • Optimal codes • Cyclic codes

1 Introduction

Quantum error correcting codes have been introduced as an alternative to classical codes for use in quantum communication channels. Since the landmark papers of Shor [7] and Steane [8], this field of research has grown rapidly. Recently, Lisonek and Singh [5] gave a variant of construction X that produces binary stabilizer quantum codes from arbitrary linear codes. In their construction, the requirement on the duality of the linear codes was relaxed. In this paper, we extend their work on construction X to obtain quantum error-correcting codes over finite fields of order p^2 where p is a prime number. Further, new results are obtained on the Hermitian dual of repeated root cyclic codes. These results are used to construct new quantum error-correcting codes.

The remainder of the paper is organized as follows. In Sect. 2, we present our main result on the extension of the quantum construction X. Section 3 characterizes

A. Degwekar
Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India
e-mail: degwekarakshay@gmail.com

K. Guenda (✉)
Faculty of Mathematics, USTHB, University of Science and Technology of Algiers, Bab Ezzouar, Algeria
e-mail: ken.guenda@gmail.com

T.A. Gulliver (✉)
Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada
e-mail: agullive@ece.uvic.ca

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_14

the generator polynomial of the Hermitian dual of a repeated root cyclic code. We also give the structure of cyclic codes of length $3p^s$ over $\mathbb{F}_{p^2}$ as well as the structure of the dual codes. Our interest in this class of codes comes from the importance of relaxing the condition $(n, p) = 1$, which allows us to consider codes other than the simple root codes.

2 Extending Construction X for $\mathbb{F}_p$

Let $\mathbb{F}_p$ denote the finite field with p elements and $\mathbb{F}_p^* = \mathbb{F}_p\backslash\{0\}$. Further, let $\mathbb{F}_{p^2}^n$ denote the vector space of all n-tuples over $\mathbb{F}_{p^2}$. For, $x \in \mathbb{F}_{p^2}^n$ denote the *conjugate* of x by $\overline{x} = x^p$, and for $x, y \in \mathbb{F}_{p^2}^n$, let $\langle x, y\rangle = \sum_{i=1}^n x_i \overline{y_i}$ denote the Hermitian inner product. Then the *norm* of x is defined as $||x|| = \langle x, x\rangle = \sum_{i=1}^n {x_i}^{p+1}$, and the *trace* of x as $\mathrm{Tr}(x) = x + \overline{x}$ [6]. Both the trace and norm are mappings from $\mathbb{F}_{p^2}$ to $\mathbb{F}_p$.

Usually a dual contained condition is required to construct CSS quantum codes as given by the following result.

Proposition 1 ([4]) *If there exists an $\mathbb{F}_{p^2}$-linear $[n, k, d]_{p^2}$ code B such that $B^\perp \subset B$, then there exists an $[[n, 2k - n, d]]_p$ quantum code.*

In the remainder of this section, we give some important lemmas which will be useful in the proof of our main result.

Lemma 2 *Let S be a subspace of $\mathbb{F}_{p^2}^n$ such that there exist $x, y \in S$ with $\langle x, y\rangle \neq 0$. Then for all $k \in \mathbb{F}_p$, there exists $z \in S$ with $||z|| = k$.*

Proof This is a non-constructive proof of the existence of the required element z. With the assumption on x and y, let $c \in \mathbb{F}_{p^2}$ and $g(c) = ||cx + y|| = \sum_{i=1}^n (cx_i + y_i)^{p+1}$ be a polynomial of degree $p + 1$ in c. We claim that as c ranges over the elements of $\mathbb{F}_{p^2}$, the rhs will range over all elements of $\mathbb{F}_p$.

Assume now that there exists some $k \in \mathbb{F}_p$ such that $\forall c \in \mathbb{F}_{p^2}, g(c) \neq k$. For each $i \in \mathbb{F}_p\backslash\{k\}$, let $S_i = \{c \in \mathbb{F}_{p^2};\ g(c) = i\}$. Since the polynomial g has degree $p + 1$, g can have at most $p + 1$ roots in any field. Then $|S_i| \le p + 1$, as the polynomial $g(c) - i$ can have at most $p + 1$ roots, and the S_i partition the set $\mathbb{F}_{p^2}$. Then $|\mathbb{F}_{p^2}| = p^2 \le \sum_{i\in\mathbb{F}_p\backslash\{k\}} |S_i| \le (p + 1)(p - 1) = p^2 - 1$, which is a contradiction. Hence the result follows. □

Lemma 3 *Let D be a subspace of $\mathbb{F}_{p^2}^n$ and assume that M is a basis for $D \cap D^{\perp_h}$. Then there exists an orthonormal set B such that $M \cup B$ is a basis for D.*

Proof The proof given here is a generalization of the proof for the analogous case presented in [5, Theorem 2]. Let W be a subspace of $\mathbb{F}_{p^2}^n$ such that

$$D = (D \cap D^{\perp_h}) \oplus W, \tag{1}$$

and let $l = \dim(W)$. For each $0 \leq i \leq l$, we can construct an orthonormal set S_i that is a basis for an i-dimensional subspace T_i of W such that

$$W = T_i \oplus (T_i^{\perp_h} \cap W). \tag{2}$$

The process is iterative. Define $S_0 := \phi$ and suppose that for some $0 \leq i < l$, the set S_i is an orthonormal basis for T_i such that $\dim(T_i) = i$ and (2) holds. Let x be a non-zero vector in $T^{\perp_h} \cap W$. Then there exists $y \in T^{\perp_h} \cap W$ such that $\langle x, y \rangle \neq 0$. If no such y exists, then $x \in D^{\perp_h}$, which would contradict (1) because the intersection of D and $D^{\perp_h}$ is $\{0\}$. Hence by Lemma 2, there must exist a $z \in T_i^{\perp_h} \cap W$ such that $||z|| = 1$. Set $S_{i+1} = S_i \cup \{z\}$. Clearly all the elements in S_{i+1} are orthogonal to each other. In addition, $||s|| = 1$ for all $s \in S_{i+1}$.

Let T_{i+1} be the subspace spanned by S_{i+1}. As $z \notin T_i$, we have that $\dim(T_{i+1}) = i + 1$. To show that

$$W = T_{i+1} \oplus (T_{i+1}^{\perp_h} \cap W), \tag{3}$$

we must first show that $T_{i+1} \cap T_{i+1}^{\perp_h} \cap W = 0$. Let $v \in T_{i+1} \cap T_{i+1}^{\perp_h} \cap W$. As $v \in T_{i+1}$, we have $v = u + cz$ where $u \in T_i$ and $c \in \mathbb{F}_{p^2}$. Since $v \in T_{i+1}^{\perp_h}$, we have for each $w \in T_i$ and each $d \in \mathbb{F}_{p^2}$ that

$$0 = \langle u + cz, w + dz \rangle = \langle u, w \rangle + \overline{d}\langle u, z \rangle + c\langle z, w \rangle + c\overline{d}||z|| = \langle u, w \rangle + c\overline{d}.$$

We must have $c = 0$ or else $\langle u, w \rangle + cd$ would not remain constant as d runs over the elements of $\mathbb{F}_{p^2}$. Thus $\langle u, w \rangle = 0$ for all $w \in T_i$, and hence $u \in T_i^{\perp_h}$. As $u \in T_i$ and $T_i \cap T_i^{\perp_h} = 0$, we obtain that $u = 0$. Hence v is also 0 and $T_{i+1} \cap T_{i+1}^{\perp_h} \cap W = 0$.

Next we show that $W = T_{i+1} + (T_{i+1} \cap W)$. Let $w \in W$. By assumption $W = T_i + (T_i^{\perp_h} \cap W)$, so there exist vectors $x \in T_i$ and $y \in T_i^{\perp_h} \cap W$ such that $w = x + y$. Now it is shown that $W = T_{i+1} + (T_{i+1}^{\perp_h} \cap W)$. By assumption $W = T_i + (T_i^{\perp_h} \bigcap W)$, so there exist vectors $x \in T_i$ and $y \in T_i \cap W$. Clearly $x \in T_{i+1}$ and for any $u + dz \in T_{i+1}$ (where $u \in T_i$ and $d \in \mathbb{F}_{p^2}$), we have

$$\begin{aligned}\langle y - \langle y, z \rangle z, u + dz \rangle &= \langle y, u \rangle + \overline{d}\langle y, z \rangle - \langle y, z \rangle \langle z, u \rangle - \overline{d}\langle y, z \rangle ||z|| \\ &= \overline{d}\langle y, z \rangle - \overline{d}\langle y, z \rangle = 0. \end{aligned} \tag{4}$$

Thus $y \in T_{i+1} \cap W$, and hence $W = T_{i+1} + (T_{i+1} \cap W)$. This completes the proof that (2) implies (3) assuming that the vector z is chosen as described above. □

Theorem 4 *For an $[n, k]_{p^2}$ linear code C, let $e = n - k - \dim(C \cap C^{\perp_h})$. Then there exists a quantum code with parameters $[[n + e, 2k - n, d]]_p$ and $d \geq \min(\mathrm{wt}(C), \mathrm{wt}(C + C^{\perp_h}) + 1)$.*

Proof We start with the observation that the equation $x^2 + 1 = 0$ always has a solution in $\mathbb{F}_{p^2}$. This can be proven using the fact that $\mathbb{F}^*_{p^2} = \mathbb{F}_{p^2} \setminus \{0\}$ is a cyclic group. Let β be a generator of $\mathbb{F}^*_{p^2}$. Then $\beta^k = -1$ for some k, and since $(-1)^2 = 1$, $\beta^{2k} = 1$ and $(p^2 - 1)|2k$, so k is even. Thus, $\beta^{\frac{k}{2}}$ is the required solution.

As defined previously

$$e = \dim(C^{\perp_h}) - \dim(C \cap C^{\perp_h}) = \dim(C + C^{\perp_h}) - \dim(C).$$

Let $s = \dim(C \cap C^{\perp_h})$, and G be the matrix

$$G = \begin{pmatrix} M_{s\times n} & 0_{s\times e} \\ A_{(n-e-2s)\times n} & 0_{(n-e-2s)\times e} \\ B_{e\times n} & \beta^{k/2} I_{e\times e} \end{pmatrix}, \tag{5}$$

where the size of the matrix is indicated by the subscripts, and 0 and I denote a zero matrix and identity matrix, respectively.

For a matrix P, let $r(P)$ denote the set of rows of P. The matrix G is constructed such that $r(M)$ is a basis for $C \cap C^{\perp_h}$, $r(M) \cup r(A)$ is a basis for C, $r(M) \cup r(B)$ is a basis for $C^{\perp_h}$, and $r(B)$ is an orthonormal set. The existence of such a matrix B follows from Lemma 3. Note that $r(M) \cup r(A) \cup r(B)$ is a basis for $C + C^{\perp_h}$.

Let E be the linear code for which G is a generator matrix. Further, let S denote the union of the first s rows of G and the last e rows of G, i.e. S is the set of rows of the matrix

$$S = \begin{pmatrix} M_{s\times n} & 0_{s\times e} \\ B_{e\times n} & \beta^{k/2} I_{e\times e} \end{pmatrix}. \tag{6}$$

We observe that each row of S is orthogonal to each row of G because any row from the first s rows of S represents a vector in $C \cap C^{\perp_h}$, and hence is orthogonal with all codewords in $C + C^{\perp_h}$, the code represented by G.

Consider a row from the last e rows in S. This row is orthogonal to the first $n - e - s$ rows of G because they represent the code C while the matrix B represents codewords from $C^{\perp_h}$. These rows of the matrix are orthogonal to each other because the rows of B are orthogonal and $\beta^{k/2} I$ will contribute 0. Any row z is self-orthogonal since from the construction $||z|| = 1$ and the identity matrix will contribute -1, giving an inner product of 0. This completes the proof of the observation. Thus, each vector from S belongs to $E^{\perp_h}$, and the vectors in S are linearly independent because

$$\dim(E^{\perp_h}) = n + e - (n - s) = e + s = |S|.$$

Hence S is a basis for $E^{\perp_h}$. Since S is a subset of G by construction, it follows that $E^{\perp_h} \subseteq E$.

Let x be a non-zero vector in E. Then x is a linear combination of rows of G. Due to the vertical block structure of G, we can write $x = (x^1|x^2)$, where $x^1 \in \mathbb{F}_{p^2}^n$ and $x^2 \in \mathbb{F}_{p^2}^e$. If none of the last e rows of G are contained in this linear combination with a non-zero coefficient, then $x^1 \in C\backslash\{0\}$, and so $\mathrm{wt}(x) = \mathrm{wt}(x^1) \geq \mathrm{wt}(C)$. If some of the last e rows of G are in this linear combination with a non-zero coefficient, then $x^1 \in C + C^{\perp_h}$ and $\mathrm{wt}(x) = \mathrm{wt}(x^1) + \mathrm{wt}(x^2) \geq \mathrm{wt}(C+C^{\perp_h})+1$. Thus E is an $[n+e, k+e, d]_{p^2}$ code with $d \geq \min(\mathrm{wt}(C), \mathrm{wt}(C+C^{\perp_h}) + 1)$ and $E^{\perp_h} \subseteq E$. The code E is such that $E^{\perp h} \subset E$, and thus the result follows from Proposition 1. □

Many constructions of quantum codes use self-orthogonal codes [1, 2], which corresponds to the case when $e = 0$ in Theorem 4. The results in the next section are required to construct the quantum codes in subsequent sections. Note that many of the results in the next section can easily be generalized to constacyclic codes.

3 The Hermitian Dual of Repeated Root Cyclic Codes

Let p be a prime number and C a cyclic code of length n over the finite field $\mathbb{F}_{p^2}$. Then C is given by the principal ideal $g(x)$ in $\dfrac{\mathbb{F}_{p^2}[x]}{\langle x^n - 1\rangle}$, and so $g(x)$ is called the generator polynomial for C. When the length n divides p, C is called a repeated root cyclic code.

In this section, we obtain the generator polynomial of the Hermitian dual of a repeated root cyclic code. We also give the structure of the cyclic codes of length $3p^s$ over $\mathbb{F}_{p^2}$ as well as the structure of the dual codes. Our interest in this class of codes comes from the importance of relaxing the condition $(n, p) = 1$, which allows us to consider codes other than simple root codes.

Let $f(x) = a_0 + a_1x + \ldots + a_rx^r$ be a polynomial in $\mathbb{F}_{q^2}[x]$, and $\overline{f(x)} = \overline{a_0} + \overline{a_1}x + \ldots + \overline{a_r}x^r$. The polynomial inverse of f is denoted by $f^\star(x) = x^r f(x^{-1}) = a_r + a_{r-1}x + \ldots + a_0x^r$, so then $f^\perp(x) = \overline{a_r} + \overline{a_{r-1}}x + \ldots + \overline{a_0}x^r$ is the orthogonal polynomial of f.

The following properties can easily be verified.

Lemma 5 *Let $f(x)$ and $g(x)$ be polynomials over $\mathbb{F}_{p^m}$. Then*

1. *Conjugation is additive: $\overline{f(x) + g(x)} = \overline{f(x)} + \overline{g(x)}$;*
2. *Conjugation is multiplicative: $\overline{f(x)g(x)} = \overline{f(x)}\,\overline{g(x)}$;*
3. *Polynomial inversion is additive if the polynomials have the same degree: $(f(x) + g(x))^\star = f(x)^\star + g(x)^\star$;*
4. *Polynomial inversion is multiplicative: $(f(x)g(x))^\star = f(x)^\star\, g(x)^\star$;*
5. *Inversion and conjugation commute with each other: $\overline{(f(x)^\star)} = (\overline{f(x)})^\star$; and*
6. *Both operations are self-inverses: $(f(x)^\star)^\star = f(x)$ and $\overline{\overline{f(x)}} = f(x)$.*

Lemma 6 *Let* $a(x) = a_0 + a_1x + \ldots + a_{n-1}x^{n-1}$ *and* $b(x) = b_0 + b_1x + \ldots + b_{n-1}x^{n-1}$ *be polynomials in* $\dfrac{\mathbb{F}_{p^2}[x]}{\langle x^n - 1\rangle}$*. Then* $a(x)\overline{b(x)} = 0$ *in* $\dfrac{\mathbb{F}_{p^2}[x]}{\langle x^n - 1\rangle}$ *if and only if* $(a_0, a_1, \ldots, a_{n-1})$ *is orthogonal to* $(\overline{b_{n-1}}, \overline{b_{n-2}}, \ldots, \overline{b_0})$ *and all its cyclic shifts. That is* $\langle a, \overline{b^\star}\rangle = 0 \iff a(x)b(x)^\perp = 0$.

Proof It is well known (see for example [3]), that if $a(x) = a_0 + a_1x + \ldots + a_{n-1}x^{n-1}$ and $b(x) = b_0 + b_1x + \ldots + b_{n-1}x^{n-1}$ are polynomials in $\dfrac{\mathbb{F}_{p^2}[x]}{\langle x^n - 1\rangle}$, then $a(x)b(x) = 0$ in $\dfrac{\mathbb{F}[x]}{\langle x^n - 1\rangle}$ if and only if $(a_0, a_1, \ldots, a_{n-1})$ is orthogonal to $(b_{n-1}, b_{n-2}, \ldots, b_0)$ and all its cyclic shifts. Hence by applying this fact to $a(x)$ and $\overline{b(x)}$, and noting that $\overline{\overline{b(x)}} = b(x)$, the result follows. □

We now use Lemma 6 to derive an expression for the Hermitian dual of a cyclic code. Let $S \subseteq R$ and let the annihilator be $\text{ann}(S) = \{g \in R | fg = 0,\ \forall f \in S\}$. Then ann$(S)$ is also an ideal of the ring and hence is generated by a polynomial.

Lemma 7 *If* $g(x)$ *generates the code* C*, then* $C^{\perp_h} = \text{ann}(\overline{g(x)}^\star)$.

Proof Assume that $g(x)$ generates the code C. Then each codeword in C has the form $a(x) = g(x)c(x)$. Let a codeword $b(x)$ lie in the Hermitian dual $C^{\perp_h}$. Then by Lemma 6 we have that

$$a(x)b^\perp(x) = 0,$$

and by Lemma 5 this is equivalent to

$$b(x)(\overline{g(x)}^\star) = 0. \tag{7}$$

Then by (7) we have that for a codeword $b(x)$, $b(x) \in C^{\perp_h} \iff b(x) \in \text{ann}(\overline{g(x)}^\star)$, which completes the proof. □

Lemma 8 *Assume that* $C = \langle g(x)\rangle$ *is a cyclic code of length* n *over* $\mathbb{F}_{p^2}$ *with generator polynomial* $g(x)$*. Define* $h(x) = \dfrac{x^n - 1}{g(x)}$*. Then we have that* $C^{\perp_h} = \langle h^\perp(x)\rangle$.

Proof From Lemma 7 it is known that $C^{\perp_h} = \text{ann}(g(x)^\perp)$. Thus, we must show that $\text{ann}(g^\perp(x)) = \langle h^\perp(x)\rangle$. One way containment is easy since $\langle h^\perp(x)\rangle \subseteq \text{ann}(g^\perp(x))$, which is true because $h^\perp(x)g^\perp(x) = (h(x)g(x))^\perp = (x^n - 1)^\perp = 0$ by Lemma 5. For containment the other way, we observe that since $\text{ann}(g^\perp(x))$ is an ideal of the polynomial ring $\dfrac{\mathbb{F}_{p^2}[x]}{\langle x^n - 1\rangle}$, it is generated by a polynomial, say $b^\perp(x)$. Then $b^\perp(x)g^\perp(x) = x^n - 1 = \lambda(x^n - 1)^\perp$ (because $b(x)$ is the smallest degree polynomial, this is an equality). Hence $b(x)g(x) = x^n - 1$, so it must be that $b(x) = h(x)$ since both are unitary polynomials. This completes the proof. □

Theorem 9 *Let $p > 3$ be a prime. Then*

1. *There exists $\omega \in \mathbb{F}_{p^2}$ such that $\omega^3 = 1$ and the factorization of $x^{3p^s} - 1$ into irreducible factors over $\mathbb{F}_{p^2}[x]$ is*

$$x^{3p^s} - 1 = (x-1)^{p^s}(x-\omega)^{p^s}(x-\omega^2)^{p^s};$$

2. *The cyclic codes of length $3p^s$ are always of the form*

$$\langle (x-1)^i (x-\omega)^j (x-\omega^2)^k \rangle,$$

where $0 \le i, j, k \le p^s$, and the code has $p^{2(3p^s-i-j-k)}$ codewords; and

3. *The Hermitian dual of the codes have the form*

$$C^{\perp_h} = \begin{cases} \langle (x-1)^{p^s-i}(x-\omega)^{p^s-j}(x-\omega^2)^{p^s-k} \rangle & \text{if } p \equiv 1 \mod 3, \\ \langle (x-1)^{p^s-i}(x-\omega^2)^{p^s-j}(x-\omega)^{p^s-k} \rangle & \text{if } p \equiv 2 \mod 3. \end{cases} \tag{8}$$

Proof

1. Since p is a prime number, $p \not\equiv 0 \mod 3$, and $p^2 - 1 = (p+1)(p-1)$, so either $p + 1 = 0 \mod 3$ or $p - 1 = 0 \mod 3$. Therefore an element of order 3 exists in $\mathbb{F}_{p^2}$. Let this element be ω, so then $(x-1)(x-\omega)(x-\omega^2) = x^3 - 1$. In a field of characteristic p, it is known that $x^n - 1 = (x^m - 1)^p$ if $n = mp$. Therefore we have that $x^{3p^s} - 1 = (x^3 - 1)^{p^s} = ((x-1)(x-\omega)(x-\omega^2))^{p^s}$.
2. From part 1, we know that the irreducible factors are $(x-1)$, $(x-\omega)$ and $(x-\omega^2)$, each of multiplicity p^s. As the generator polynomial divides $x^{3p^s} - 1$, the statement follows.
3. We know from Lemma 8 that

$$C^{\perp_h} = \langle h^{\perp}(x) \rangle,$$

and hence

$$\begin{aligned} C^{\perp_h} &= \overline{\langle \frac{(x-1)^{p^s}(x-\omega)^{p^s}(x-\omega^2)^{p^s}}{(x-1)^i(x-\omega)^j(x-\omega^2)^k} \rangle^{\star}} \\ &= \overline{\langle (x-1)^{p^s-i}(x-\omega)^{p^s-j}(x-\omega^2)^{p^s-k} \rangle^{\star}} \\ &= \overline{\langle [(x-1)^{p^s-i}]^{\star}[(x-\omega)^{p^s-j}]^{\star}[(x-\omega^2)^{p^s-k}]^{\star} \rangle} \\ &= \overline{\langle [-(x-1)^{p^s-i}][-\omega(x-\omega^{-1})^{p^s-j}]^{\star}[-\omega^2(x-\omega^{-2})^{p^s-k}]^{\star} \rangle} \\ &= \overline{\langle [(x-1)^{p^s-i}][(x-\omega^2)^{p^s-j}][(x-\omega)^{p^s-k}] \rangle} \\ &= \langle [(\overline{x-1})^{p^s-i}][(\overline{x-\omega^2})^{p^s-j}][(\overline{x-\omega})^{p^s-k}] \rangle \end{aligned}$$

$$= \langle[(x-1)^{p^s-i}][(x-\omega^{2p})^{p^s-j}][(x-\omega^{p})^{p^s-k}]\rangle$$

$$= \begin{cases} \langle(x-1)^{p^s-i}(x-\omega^2)^{p^s-j}(x-\omega)^{p^s-k}\rangle & \text{if } p \equiv 1 \mod 3, \\ \langle(x-1)^{p^s-i}(x-\omega)^{p^s-j}(x-\omega^2)^{p^s-k}\rangle & \text{if } p \equiv 2 \mod 3, \end{cases} \tag{9}$$

since $(x-1)^\star = -x+1 = -(x-1)$, $(x-\omega)^\star = -\omega x + 1 = -\omega(x-\omega^2)$, and $\omega^p = \omega$ if $p \equiv 1 \mod 3$ and $\omega^p = \omega^2$ if $p \equiv 2 \mod 3$, which completes the proof.

□

4 Extension to Simple Root Cyclic Codes

This section considers cyclic codes of length n over $\mathbb{F}_{p^2}$ such that $(p, n) = 1$. In this case, a cyclic code can be represented by its defining set Z. If m has order p^2 modulo n, then $\mathbb{F}_{p^{2m}}$ is the splitting field of $x^n - 1$ containing a primitive nth root of unity. Consider a primitive root β. Then $\{k|g(\beta^k) = 0,\ 0 \leq k < n\}$ is a defining set of C. Note that this set depends on the choice of β. We can make a canonical choice for β by fixing a primitive element α of $\mathbb{F}_{p^{2m}}$ and letting $\beta = \alpha^{\frac{p^{2m}-1}{n}}$. Let α be defined by the PrimitiveElement function in Magma. This will be used in the code constructions in the next section.

For n and m as defined above and $a \in \{0, \ldots, n-1\}$, the set $\{aq^j \mod n|0 \leq j < m\}$ is called a *cyclotomic coset modulo n*. It is well known that a defining set of a cyclic code of length n is the union of cyclotomic cosets modulo n. Let $\mathbb{Z}_n$ denote the set of integers modulo n. Clearly defining sets can be considered as subsets of $\mathbb{Z}_n$. For $S \subset \mathbb{Z}_n$, denote $\overline{S} = \mathbb{Z}_n \backslash \{S\}$ and $-p^2 S = \{-p^2 s \mod n | s \in S\}$.

We now prove the following lemma.

Lemma 10 *If C is a linear cyclic code with defining set Z, then* $\dim(C^{\perp_h}) - \dim(C \cap C^{\perp_h}) = |Z \cap -pZ|$.

Proof Let C be a linear cyclic code of length n, and $\prod_{k \in Z}(x - \beta^k)$ be the generator polynomial for C. Then from Lemma 8 the generator polynomial for $C^{\perp_h}$ is $\prod_{k \in -p\overline{Z}}(x - \beta^k)$, and the generator polynomial for $C \cap C^{\perp_h}$ is $\prod_{k \in Z \cap -p\overline{Z}}(x - \beta^k)$, which gives that

$$\dim(C^{\perp_h}) - \dim(C \cap C^{\perp_h}) = n - |-p\overline{Z}| - (n - |Z \cup -p\overline{Z}|)$$
$$= |Z \cup -p\overline{Z}| - |-p\overline{Z}| = |Z \cap -pZ|.$$

□

Theorem 11 *Assume n is divisible by $p^2 - 1$ and let C be an $[n, k]_{p^2}$ cyclic code with defining set Z such that $(Z \cap -pZ) \subseteq T = \{\frac{nk}{p^2-1} | k \in \{1, \ldots, p^2 - 1\}\}$. If*

$e = |Z \cap -pZ|$, then there exists an $[[n+e, 2k-n+e, d]]_p$ quantum code with $d \geq \min\{\mathrm{wt}(C), \mathrm{wt}(C_u)+1, \mathrm{wt}(C+C^{\perp_h})+2\}$ where the minimum is taken over the cyclic codes C_u with defining set $Z\backslash\{u\}$ for each $u \in Z \cap -pZ$.

Proof The proof requires a modification to the proof of Theorem 4, in particular the set of orthonormal vectors used is changed. First, observe that each of the elements in T is a cyclotomic coset and contains only one element. Let $q = p^2 - 1$, $n = (p^2-1)l = ql$, and ω be a (p^2-1)-th root of unity. Consider the polynomials

$$b_t(x) = \frac{x^n - 1}{x - \omega^t} = \sum_{i=0}^{l-1} \left(x^{qi+q-1} + \omega^t x^{qi+q-2} + \ldots + \omega^{(q-1)t} x^{qi}\right).$$

For convenience, we let $\{b_i | i \in 0, 1, \ldots, l\}$ also denote the corresponding codewords. This is an orthonormal set because

$$\langle b_u, b_v \rangle = q \sum_{i=0}^{l-1} (\omega^{i(u+vp)}) = q \sum i = 0^{l-1} (\omega^{i(u-v)}) = \begin{cases} 0 & u \neq v \\ ql & u = v \end{cases}.$$

To remove the ql factor, we can multiply each element by a constant. Thus, to add the rows for B to the matrix, we add $U = \{b_t | \frac{tn}{q} \in Z \cap -pZ\}$.

To prove the claim regarding the distance, we have three cases: no row from B is a linear combination, exactly one row from U is a linear combination with a non-zero coefficient, and at least two rows are a combination. The proof of the first and the last cases is the same as in the proof of Theorem 4. For the second case, let b_t be the row with non-zero coefficient. Then the code generated would be $\mathrm{span}(C, b_t)$, which is precisely the cyclic code with defining set $Z\backslash\{\frac{tn}{q-1}\}$. This completes the proof. □

Appendix: Code Construction Examples

In this appendix, comprehensive tables of codes generated using the results in the paper are presented. Table 1 presents quantum codes obtained using Theorem 4. Many of these codes have parameters better than the best known binary quantum codes. Table 2 presents the parameters of quantum codes obtained from repeated root cyclic codes using Theorem 9. These are codes of length $3p^s$ over fields of size p^2.

Table 1 Codes obtained using Theorem 4 and the best known binary QECCs

New code	Generator polynomial	Best known binary QECC
$[[33, 31, 2]]_3$	$x^{13} + \alpha^5 x^{12} + \alpha^7 x^{11} + \alpha^2 x^{10} + 2x^9 + 2x^8 + \alpha^3 x^7 + \alpha^6 x^6 + 2x^5 + \alpha^3 x^4 + \alpha^3 x^3 + \alpha^6 x^2 + \alpha^2$	$[[33, 31, 1]]_2$
$[[35, 33, 2]]_3$	$x + 1$	$[[35, 33, 1]]_2$
$[[39, 37, 2]]_3$	$x + 2$	$[[39, 37, 1]]_2$
$[[40, 26, 5]]_3$	$x^7 + \alpha x^6 + \alpha x^5 + \alpha^6 x^4 + x^3 + \alpha^7 x^2 + \alpha^5 x + \alpha$	$[[40, 26, 4]]_2$
$[[40, 24, 6]]_3$	$x^8 + \alpha^3 x^7 + \alpha x^6 + \alpha^7 x^5 + 2x^4 + x^3 + \alpha^2 x^2 + \alpha x + \alpha^2$	$[[40, 24, 5]]_2$
$[[41, 39, 2]]_3$	$x + 1$	$[[41, 39, 1]]_2$
$[[41, 9, 11]]_3$	$x^{16} + \alpha^5 x^{15} + \alpha^5 x^{14} + \alpha x^{13} + \alpha^6 x^{12} + \alpha^2 x^{11} + 2x^9 + \alpha^7 x^7 + 2x^6 + \alpha^3 x^5 + 2x^4 + \alpha^6 x^3 + \alpha^7 x^2 + \alpha^7$	$[[41, 9, 8]]_2$
$[[41, 19, 8]]_3$	$x^{11} + \alpha x^{10} + \alpha x^9 + \alpha^5 x^8 + 2x^7 + \alpha^3 x^6 + \alpha^2 x^5 + \alpha^3 x^4 + x^2 + \alpha^2 x + 2$	$[[41, 19, 6]]_2$
$[[40, 20, 7]]_3$	$x^{10} + \alpha^6 x^9 + \alpha^7 x^6 + \alpha^3 x^5 + 2x^4 + \alpha^3 x^3 + \alpha x + \alpha^2$	$[[40, 20, 6]]_2$
$[[41, 25, 6]]_3$	$x^8 + \alpha^6 x^7 + \alpha^7 x^6 + x^4 + \alpha^5 x^3 + \alpha^3 x^2 + 2x + \alpha^7$	$[[41, 25, 4]]_2$
$[[40, 10, 10]]_3$	$x^{15} + \alpha^6 x^{14} + x^{13} + \alpha^5 x^{12} + \alpha^2 x^{11} + x^{10} + \alpha^3 x^9 + \alpha^5 x^8 + x^7 + 2x^6 + \alpha^7 x^5 + \alpha^7 x^4 + x^3 + \alpha x^2 + \alpha^5 x + \alpha^5$	$[[40, 10, 8]]_2$
$[[40, 16, 8]]_3$	$x^{12} + \alpha^3 x^{11} + \alpha^3 x^{10} + x^9 + \alpha^5 x^8 + \alpha^5 x^7 + \alpha^5 x^6 + \alpha^5 x^5 + 2x^4 + x^3 + \alpha^6 x + \alpha^2$	$[[40, 16, 6]]_2$
$[[41, 13, 9]]_3$	$x^{14} + 2x^{13} + \alpha x^{12} + 2x^{10} + \alpha^2 x^9 + x^8 + \alpha^5 x^7 + \alpha^5 x^6 + x^5 + \alpha^3 x^4 + \alpha^6 x^3 + 2x^2 + \alpha^3 x + \alpha^5$	$[[41, 13, 7]]_2$
$[[41, 21, 7]]_3$	$x^{10} + \alpha^7 x^9 + \alpha^6 x^7 + \alpha^6 x^6 + 2x^5 + \alpha^7 x^4 + x^3 + \alpha^3 x + \alpha^5$	$[[41, 21, 6]]_2$
$[[41, 27, 5]]_3$	$x^7 + 2x^6 + \alpha^3 x^5 + \alpha^7 x^4 + \alpha^7 x^3 + 2x^2 + \alpha^3 x + \alpha^2$	$[[41, 27, 4]]_2$
$[[33, 31, 2]]_3$	$x + 1$	$[[33, 31, 1]]_2$
$[[40, 26, 5]]_3$	$x^7 + \alpha x^6 + \alpha x^5 + \alpha^6 x^4 + x^3 + \alpha^7 x^2 + \alpha^5 x + \alpha$	$[[40, 26, 4]]_2$
$[[40, 24, 6]]_3$	$x^8 + \alpha^3 x^7 + \alpha x^6 + \alpha^7 x^5 + 2x^4 + x^3 + \alpha^2 x^2 + \alpha x + \alpha^2$	$[[40, 24, 5]]_2$
$[[41, 39, 2]]_3$	$x + 1$	$[[41, 39, 1]]_2$
$[[41, 9, 11]]_3$	$x^{16} + \alpha^5 x^{15} + \alpha^5 x^{14} + \alpha x^{13} + \alpha^6 x^{12} + \alpha^2 x^{11} + 2x^9 + \alpha^7 x^7 + 2x^6 + \alpha^3 x^5 + 2x^4 + \alpha^6 x^3 + \alpha^7 x^2 + \alpha^7$	$[[41, 9, 8]]_2$
$[[41, 19, 8]]_3$	$x^{11} + \alpha x^{10} + \alpha x^9 + \alpha^5 x^8 + 2x^7 + \alpha^3 x^6 + \alpha^2 x^5 + \alpha^3 x^4 + x^2 + \alpha^2 x + 2$	$[[41, 19, 6]]_2$
$[[40, 20, 7]]_3$	$x^{10} + \alpha^6 x^9 + \alpha^7 x^6 + \alpha^3 x^5 + 2x^4 + \alpha^3 x^3 + \alpha x + \alpha^2$	$[[40, 20, 6]]_2$
$[[41, 25, 6]]_3$	$x^8 + \alpha^6 x^7 + \alpha^7 x^6 + x^4 + \alpha^5 x^3 + \alpha^3 x^2 + 2x + \alpha^7$	$[[41, 25, 4]]_2$
$[[40, 10, 10]]_3$	$x^{15} + \alpha^6 x^{14} + x^{13} + \alpha^5 x^{12} + \alpha^2 x^{11} + x^{10} + \alpha^3 x^9 + \alpha^5 x^8 + x^7 + 2x^6 + \alpha^7 x^5 + \alpha^7 x^4 + x^3 + \alpha x^2 + \alpha^5 x + \alpha^5$	$[[40, 10, 8]]_2$
$[[40, 16, 8]]_3$	$x^{12} + \alpha^3 x^{11} + \alpha^3 x^{10} + x^9 + \alpha^5 x^8 + \alpha^5 x^7 + \alpha^5 x^6 + \alpha^5 x^5 + 2x^4 + x^3 + \alpha^6 x + \alpha^2$	$[[40, 16, 6]]_2$
$[[41, 13, 9]]_3$	$x^{14} + 2x^{13} + \alpha x^{12} + 2x^{10} + \alpha^2 x^9 + x^8 + \alpha^5 x^7 + \alpha^5 x^6 + x^5 + \alpha^3 x^4 + \alpha^6 x^3 + 2x^2 + \alpha^3 x + \alpha^5$	$[[41, 13, 7]]_2$
$[[41, 21, 7]]_3$	$x^{10} + \alpha^7 x^9 + \alpha^6 x^7 + \alpha^6 x^6 + 2x^5 + \alpha^7 x^4 + x^3 + \alpha^3 x + \alpha^5$	$[[41, 21, 6]]_2$
$[[41, 27, 5]]_3$	$x^7 + 2x^6 + \alpha^3 x^5 + \alpha^7 x^4 + \alpha^7 x^3 + 2x^2 + \alpha^3 x + \alpha^2$	$[[41, 27, 4]]_2$

(continued)

Table 1 (continued)

New code	Generator polynomial	Best known binary QECC
$[[41, 9, 11]]_5$	$x^{16} + \alpha x^{15} + \alpha^{23}x^{14} + \alpha^3x^{13} + 4x^{12} + \alpha^{15}x^{11} + 3x^{10} + \alpha^{10}x^8 + \alpha^3x^7 + \alpha^{14}x^5 + \alpha^8x^4 + \alpha^{19}x^3 + 4x^2 + \alpha^{17}x + \alpha^8$	$[[41, 9, 8]]_2$
$[[40, 2, 12]]_5$	$x^{19} + \alpha^{20}x^{18} + \alpha^{22}x^{17} + \alpha^{10}x^{16} + \alpha^3x^{15} + \alpha^{20}x^{14} + \alpha^{21}x^{13} + \alpha^{22}x^{12} + \alpha^{20}x^{11} + \alpha^8x^{10} + 3x^9 + \alpha^{22}x^8 + 4x^7 + \alpha^{11}x^6 + \alpha^{23}x^5 + \alpha^{22}x^4 + \alpha^8x^3 + \alpha^5x^2 + \alpha^9x + \alpha^4$	$[[40, 2, 10]]_2$
$[[41, 5, 12]]_5$	$x^{18} + \alpha^{15}x^{17} + \alpha^{19}x^{16} + \alpha^{23}x^{15} + \alpha^{13}x^{14} + \alpha^5x^{13} + \alpha^7x^{12} + x^{11} + \alpha^{17}x^{10} + \alpha^3x^9 + \alpha^{19}x^8 + \alpha^{19}x^7 + \alpha^5x^6 + \alpha^{11}x^5 + \alpha^8x^4 + \alpha x^3 + \alpha^{10}x^2 + \alpha^5x + 1$	$[[41, 5, 9]]_2$
$[[40, 6, 11]]_5$	$x^{17} + \alpha^{10}x^{16} + \alpha^4x^{15} + \alpha^{22}x^{14} + \alpha^9x^{13} + \alpha * x^{12} + 3x^{11} + 4x^{10} + \alpha^5x^9 + \alpha^{16}x^8 + \alpha^{19}x^7 + \alpha^{22}x^6 + \alpha^9x^5 + \alpha^4x^4 + 4x^3 + \alpha^{17}x^2 + \alpha^{16}x + 4$	$[[40, 6, 8]]_2$
$[[39, 15, 9]]_5$	$x^{12} + 2x^{11} + \alpha^5x^{10} + \alpha^{16}x^9 + \alpha^3x^8 + \alpha^3x^7 + \alpha^{13}x^6 + \alpha^{15}x^5 + \alpha^{22}x^4 + x^3 + \alpha^9x^2 + 2x + \alpha^{16}$	$[[39, 15, 7]]_2$
$[[39, 23, 5]]_5$	$x^8 + \alpha^{21}x^7 + 3x^6 + \alpha x^5 + \alpha^{16}x^4 + \alpha^{17}x^3 + \alpha^2x^2 + \alpha^{21}x + \alpha^{16}$	$[[39, 23, 4]]_2$
$[[40, 22, 6]]_5$	$x^9 + \alpha^7x^8 + \alpha^8x^7 + \alpha^2x^6 + \alpha^{21}x^5 + \alpha^9x^4 + \alpha^{14}x^3 + \alpha^{20}x^2 + \alpha^{19}x + 4$	$[[40, 22, 5]]_2$
$[[41, 21, 7]]_5$	$x^{10} + \alpha^3x^9 + x^8 + \alpha^{10}x^7 + \alpha^2x^6 + \alpha^{22}x^5 + \alpha^{23}x^4 + \alpha x^3 + \alpha^{22}x^2 + \alpha^{15}x + 1$	$[[41, 21, 6]]_2$
$[[39, 11, 10]]_5$	$x^{14} + \alpha^{15}x^{12} + \alpha^{21}x^{11} + \alpha^{16}x^{10} + \alpha^{16}x^9 + 4x^8 + \alpha^3x^7 + 4x^5 + 4x^4 + \alpha^{22}x^3 + \alpha^{19}x^2 + \alpha^9x + \alpha^8$	$[[39, 11, 8]]_2$
$[[39, 19, 7]]_5$	$x^{10} + \alpha^{14}x^8 + \alpha^{14}x^7 + \alpha^4x^6 + \alpha x^5 + 4x^4 + \alpha^8x^3 + \alpha^3x^2 + \alpha^{14}x + \alpha^8$	$[[39, 19, 5]]_2$
$[[40, 18, 8]]_5$	$x^{11} + x^{10} + \alpha^{13}x^9 + \alpha^{17}x^8 + 2x^7 + \alpha^{14}x^6 + \alpha^{17}x^5 + 3x^4 + \alpha^{15}x^3 + \alpha^{21}x^2 + \alpha^{23}x + 4$	$[[40, 18, 6]]_2$
$[[31, 13, 6]]_5$	$x^9 + 3x^8 + x^6 + x^5 + 4x^4 + x^3 + 3x^2 + x + 4$	$[[31, 13, 5]]_2$
$[[32, 0, 11]]_5$	$x^{16} + 3x^{15} + 2x^{14} + x^{13} + x^{11} + 2x^{10} + x^9 + x^8 + 4x^7 + x^6 + x^5 + 3x^4 + 2x^3 + x + 1$	$[[32, 0, 10]]_2$
$[[31, 7, 8]]_5$	$x^{12}+4x^{11}+4x^{10}+2x^9+4x^8+2x^7+x^6+3x^5+x^4+x^3+2x+1$	$[[31, 7, 7]]_2$
$[[32, 12, 7]]_5$	$x^{10} + 3x^7 + x^6 + x^5 + x^4 + 3x^2 + 4x + 1$	$[[32, 12, 6]]_2$
$[[31, 25, 3]]_5$	$x^3 + x^2 + 3x + 4$	$[[31, 25, 2]]_2$
$[[32, 18, 5]]_5$	$x^7 + 3x^5 + 3x^3 + 4x^2 + 4$	$[[32, 18, 4]]_2$
$[[32, 6, 9]]_5$	$x^{13} + 2x^{11} + x^{10} + x^9 + 4x^8 + 3x^6 + 2x^5 + 4x^3 + 4x^2 + 4x + 4$	$[[32, 6, 8]]_2$
$[[33, 31, 2]]_5$	$x + 4$	$[[33, 31, 1]]_2$
$[[35, 33, 2]]_5$	$x + 4$	$[[35, 33, 1]]_2$
$[[37, 35, 2]]_5$	$x + \alpha^{16}$	$[[37, 35, 1]]_2$
$[[37, 35, 2]]_5$	$x + \alpha^{16}$	$[[37, 35, 1]]_2$
$[[25, 23, 2]]_5$	$x + \alpha^{16}$	$[[25, 23, 1]]_2$
$[[24, 20, 3]]_5$	$x^2 + \alpha^8x + \alpha^{17}$	$[[24, 20, 2]]_2$
$[[25, 21, 3]]_5$	$x^2 + \alpha^{13}x + \alpha^{17}$	$[[25, 21, 2]]_2$
$[[24, 18, 4]]_5$	$x^3 + \alpha^{10}x^2 + \alpha^{16}x + 3$	$[[24, 18, 2]]_2$
$[[25, 19, 4]]_5$	$x^3 + \alpha^{19}x^2 + \alpha^{10}x + \alpha^{21}$	$[[25, 19, 2]]_2$
$[[25, 17, 5]]_5$	$x^4 + \alpha^7x^3 + 4x^2 + \alpha^{16}x + 3$	$[[25, 17, 3]]_2$

(continued)

Table 1 (continued)

New code	Generator polynomial	Best known binary QECC
$[[33, 31, 2]]_5$	$x + 4$	$[[33, 31, 1]]_2$
$[[32, 0, 13]]_7$	$x^{16} + 2x^{15} + 3x^{14} + 4x^{12} + x^{11} + 4x^{10} + x^9 + 5x^8 + 4x^7 + 6x^6 + 5x^5 + 2x^4 + 3x^3 + 3x^2 + 4x + 1$	$[[32, 0, 10]]_2$
$[[31, 1, 12]]_7$	$x^{15} + 3x^{14} + 6x^{13} + 6x^{12} + 3x^{11} + 4x^{10} + x^9 + 2x^8 + 4x^6 + 3x^5 + x^4 + 3x^3 + 6x^2 + 2x + 6$	$[[31, 1, 11]]_2$
$[[33, 21, 5]]_7$	$x^6 + \alpha^{42}x^5 + \alpha^{33}x^4 + \alpha^{20}x^3 + \alpha^{30}x^2 + \alpha^6 x + \alpha^{15}$	$[[33, 21, 4]]_2$
$[[33, 31, 2]]_7$	$x + 6$	$[[33, 31, 1]]_2$

Table 2 Parameters of the quantum codes obtained from repeated root cyclic codes using Theorem 9

Code	Code	Code
$[[15, 9, 2]]_{25}$	$[[15, 7, 3]]_{25}$	$[[16, 6, 4]]_{25}$
$[[75, 69, 2]]_{25}$	$[[75, 59, 3]]_{25}$	$[[75, 49, 4]]_{25}$
$[[82, 26, 5]]_{25}$		
$[[375, 369, 2]]_{25}$	$[[375, 319, 3]]_{25}$	$[[375, 269, 4]]_{25}$
$[[21, 15, 2]]_{49}$	$[[21, 13, 3]]_{49}$	$[[21, 11, 4]]_{49}$
$[[21, 7, 5]]_{49}$	$[[22, 8, 5]]_{49}$	$[[21, 5, 6]]_{49}$
$[[23, 1, 7]]_{49}$		
$[[147, 141, 2]]_{49}$	$[[147, 127, 3]]_{49}$	$[[147, 113, 4]]_{49}$
$[[147, 85, 5]]_{49}$	$[[147, 71, 6]]_{49}$	

References

1. Guenda, K.: Sur l'équivalence des codes. Ph.D. thesis, Faculty of Mathematics, USTHB, Algiers (2010)
2. Guenda, K., Gulliver, T.A.: Symmetric and asymmetric quantum codes. Int. J. Quantum Inf. **11**(5) 1350047 [10 pages] DOI:10.1142/S0219749913500470 (2013)
3. Huffman, W.C., Pless, V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, New York (2003)
4. Ketkar, A., Klappenecker, A., Kumar, S., Sarvepalli, P.K.: Nonbinary stabilizer codes over finite fields. IEEE Trans. Inf. Theory **52**(11), 4892–4914 (2006)
5. Lisonek, P., Singh, V.K.: Quantum codes from nearly self-orthogonal quaternary linear codes. Des. Codes Cryptogr. **73**(2), 417–424 (2014)
6. Rotman, J.J.: Advanced Modern Algebra, 2nd edn. Prentice Hall, Upper Saddle River (2003)
7. Shor, P.W.: Scheme for reducing decoherence in quantum memory. Phys. Rev. A **52**(4), 2493–2493 (1995)
8. Steane, A.: Multiple particle interference and quantum error correction. Proc. R. Soc. A **452**(1954), 2551–2577 (1996)

On the Fan Associated to a Linear Code

Natalia Dück, Irene Márquez-Corbella, and Edgar Martínez-Moro

Abstract We will show how one can compute all reduced Gröbner bases with respect to a degree compatible ordering for code ideals – even though these binomial ideals are not toric. To this end, the correspondence of linear codes and binomial ideals will be briefly described as well as their resemblance to toric ideals. Finally, we will hint at applications of the degree compatible Gröbner fan to the code equivalence problem.

Keywords Linear code • Gröbner basis • Gröbner fan

1 Introduction

The Gröbner fan of an ideal in the commutative polynomial ring consists of polyhedral cones indexing the different leading ideals and is thus the geometric collection of all reduced Gröbner bases for this ideal. One application of the Gröbner fan is the so-called Gröbner walk which is the conversion of Gröbner bases.

With the software system TiGERS in [5] (Toric Gröbner bases Enumeration by Reverse Search) an efficient alternative for computing the Gröbner fan has been provided for the special case of toric ideals. Indeed, by identifying a reverse search tree on the cones of the Gröbner fan, a memory-less combinatorial Gröbner walk can be established that furthermore, requires no cost weight vectors.

Linear codes, on the other hand, can be linked to this whole subject by associating to each linear code a binomial ideal that encodes the information about the code

N. Dück
Hamburg University of Technology, Hamburg, Germany
e-mail: natalia.dueck@tuhh.de

I. Márquez-Corbella
INRIA, SACLAY & LIX, CNRS UMR 7161, École Polytechnique, 91128 Palaiseau Cedex, France
e-mail: irene.marquez-corbella@inria.fr

E. Martínez-Moro (✉)
Institute of Mathematics, University of Valladolid, Valladolid, Spain
e-mail: edgar@maf.uva.es

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_15

in the exponents. This correspondence proved to be very beneficial as it provided new approaches to several well-known problems in coding theory. Almost all applications, however, require the computation of a degree compatible Gröbner basis.

In this work, it will be shown how methods from the software system TiGERS developed by Rekha R. Thomas (see [5]) can be modified in order to compute all reduced Gröbner bases with respect to a degree compatible ordering for code ideals – even though these binomial ideals are not toric. To this end, the correspondence of linear codes and binomial ideals will be briefly described as well as their resemblance to toric ideals. Finally, we will hint at applications of the degree compatible Gröbner fan to the code equivalence problem.

2 The Degree Compatible Gröbner Fan

In this work we shall use the notion of Gröbner basis and the ideal associated to a linear code. Due to the restriction of the space we will not define what a Gröbner basis is, the reader can find a good introductory text for example in [3]. Also for simplicity we will restrict ourselves to binary linear codes even if all the computation could be done in general (see [7] for the ideal associated to a q-ary linear code).

Let $\mathbb{K}[\mathbf{x}]$ be the polynomial ring with variables $\mathbf{x} = x_1, \dots, x_n$ and coefficients an arbitrary field $\mathbb{K}$. We will define the ideal associated to a binary linear code $\mathscr{C}$ of length n as

$$I = I(\mathscr{C}) = \langle \{\mathbf{x}^{\Delta \mathbf{a}} - \mathbf{x}^{\Delta \mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{C}\} \rangle \subseteq \mathbb{K}[\mathbf{x}],$$

where the operation Δ means substitute the $\bar{0}, \bar{1}$ elements in the binary field $\mathbb{F}_2$ by the corresponding $0, 1$ in the set of integers $\mathbb{Z}$. In this extended abstract the Δ will be omitted if no confusion arises to simplify the notation.

This binomial ideal has been proved valuable for several applications and captures the combinatorial properties of the code (see [6] and the references therein). Note that for those applications $\mathbb{K}$ can be the binary field, which is the usual election, and in this case we must explicitly mark which terms are the leading terms.

In this paper it shall be assumed that the leading term of a binomial is the one with coefficient 1 and the non leading term has coefficient -1. Abusing the notation if $\mathbb{K}$ is the binary field, since $1 \equiv -1$, this writing of the binomials will be assumed as a formal pointer (in [5] the leading terms were underlined).

Note also that the explicit knowledge of the underlying term order is not necessary. In fact, in all the following computations only the leading term of each binomial has to be known.

In the rest of the paper we will use the following notation and concepts from [5]:

- $\mathscr{G}_{\succ}(I)$ is the reduced Gröbner basis for the ideal I w.r.t. the monomial order $\succ$,
- $C_{\succ}(I)$ is the Gröbner cone corresponding to $\mathscr{G}_{\succ}(I)$.
- $T_{\succ}(I)$ is the reverse search tree for the ideal I as constructed in [5, Definition 2.5]

Note that in [5] the *complete* Gröbner fan is considered, i.e., the whole $\mathbb{R}^n$, since the considered toric ideals are homogeneous w.r.t. a certain grading. This is not the case for our code ideal and so here the Gröbner fan is considered only in $\mathbb{R}^n_+$.

Proposition 1 ([4]) *Let $\succ$ be a term order and $\mathbf{v} \in C_{\succ}(I)$. For any $\mathbf{u} \in \mathbb{R}^n$ holds*

$$\mathrm{lt}_{\mathbf{u}}(I) = \mathrm{lt}_{\mathbf{v}}(I) \quad \Longleftrightarrow \quad \mathrm{lt}_{\mathbf{u}}(g) = \mathrm{lt}_{\mathbf{v}}(g) \quad \forall\, g \in \mathscr{G}_{\succ}(I),$$

where $\mathrm{lt}_{\mathbf{u}}$ *stands for the leading (initial) term (ideal) induced by the order $\succ$ given by the weight vector* $\mathbf{u}$.

Note that it is a well known fact that a reduced Gröbner basis for an ideal I w.r.t. a certain monomial order is degree compatible if and only if the corresponding Gröbner cone contains the all-one vector $\mathbf{1}$. From a coding-theory point of view, degree compatible orderings are the ones one must analyze since the weight of a vector is translated on the degree of a monomial. In this sense degree compatible orderings provide us a test set for the code and therefore a gradient descent decoding algorithm, see [2]. The following proposition characterizes when there is a unique degree compatible Gröbner basis.

Proposition 2 *Let $\mathscr{G}$ be a reduced Gröbner basis for $I(\mathscr{C})$ w.r.t. a certain degree compatible ordering $\succ$. The Gröbner basis $\mathscr{G}$ is the only reduced degree compatible Gröbner basis for $I(\mathscr{C})$ if and only if*

$$\deg(\mathbf{x}^a) > \deg(\mathbf{x}^b) \quad \textit{for all} \quad \mathbf{x}^a - \mathbf{x}^b \in \mathscr{G}. \tag{1}$$

Proof Assume that (1) holds but there is another Gröbner basis $\mathscr{G}'$ for $I(\mathscr{C})$ w.r.t. another degree compatible order $\succ'$. Since $\succ'$ is degree compatible, $\mathrm{lt}_{\succ'}(g) = \mathrm{lt}_{\succ}(g)$ for all $g \in \mathscr{G}$. And by Proposition 1 we see that $\mathrm{lt}_{\succ'}(I(\mathscr{C})) = \mathrm{lt}_{\succ}(I(\mathscr{C}))$ and thus, $\mathscr{G} = \mathscr{G}'$.

Or equivalently, we can argue that the all-one vector is in the interior of the cone $C_{\succ}(I(\mathscr{C}))$ and so clearly it cannot be contained in another cone in the Gröbner fan.

In order to show the other direction assume that (1) does not hold, i.e., there is at least one binomial $\mathbf{x}^a - \mathbf{x}^b$ in $\mathscr{G}$ such that $\deg(\mathbf{x}^a) = \deg(\mathbf{x}^b)$. Then $\mathbf{1} \notin \mathrm{Int}(C_{\succ}(I(\mathscr{C})))$ and in particular, there must be a neighbouring cone that also contains $\mathbf{1}$ and thus corresponds to a degree compatible ordering. □

In terms of the Gröbner fan the above proposition can also be expressed as follows: A reduced Gröbner basis $\mathscr{G}$ w.r.t. a degree compatible ordering is the only degree compatible Gröbner basis if and only if the all-one vector $\mathbf{1}$ lies in the interior of the Gröbner cone of $\mathscr{G}$.

We say that two binary linear codes $\mathscr{C}_1$ and $\mathscr{C}_2$ are *permutation equivalent* provided there is a permutation of coordinates which sends $\mathscr{C}_1$ to $\mathscr{C}_2$. In the same fashion two binomial degree compatible Gröbner bases are *permutation equivalent* if there is a permutation of the variables that transforms one into the other. There is a close relationship between code equivalence and the equivalence of the degree compatible Gröbner bases associated to their code ideals stated as follows: If the two degree compatible Gröbner bases are permutation equivalent so are the codes, unfortunately the converse is not true, given two permutation equivalent codes not all the degree compatible Gröbner bases are permutation equivalent (only two of them should be). The reader can see [1] for a proof of this discussion.

Indeed if one has only a unique degree compatible Gröbner basis for a given code (Proposition 2) checking permutation equivalence is reduced to checking if the two unique bases are permutation equivalent using the techniques in [1]. If this is not the case one needs to compute the whole set of degree compatible Gröbner bases which we call the *degree compatible Gröbner fan*. We will tackle this task in the following section.

3 Adapting the TiGERS Strategy

We can adapt the TiGERS Algorithm in [5] for computing the degree compatible Gröbner fan for $I(\mathscr{C})$ as follows: We start with a degree compatible Gröbner basis (note that this basis can be computed by the algorithm stated in [1]). By Proposition 2 we can determine whether it is the only degree compatible Gröbner basis or not. If not, we flip only those facet binomials where both terms have the same degree and recompute the Gröbner basis. Unfortunately due the lack of space these steps can not be detailed in this extended abstract but they are showed in [5]. Lemma 3 below guarantees that we will always find at least one facet binomial where both terms have the same degree. Additionally, we can employ the *reverse search tree* defined in [5] for traversing the Gröbner cones that are degree compatible.

Lemma 3 *Let $\mathscr{G}$ be the reduced Gröbner basis for $I(\mathscr{C})$ w.r.t. a degree compatible ordering. If $\mathscr{G}$ is not the only degree compatible Gröbner basis, that is $\mathbf{1} \notin \mathrm{Int}(C(I(\mathscr{C})))$, then among all the facet binomials of $\mathscr{G}$ is at least one binomial $\mathbf{x}^{\alpha} - \mathbf{x}^{\beta}$ such that $\deg(\mathbf{x}^{\alpha}) = \deg(\mathbf{x}^{\beta})$.*

Proof Let $\mathscr{G} = \left\{\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \mid 1 \leq i \leq j+k\right\}$ and order the binomials such that $\deg(\mathbf{x}^{\alpha_i}) > \deg(\mathbf{x}^{\beta_i})$ for $1 \leq i \leq j$ and $\deg(\mathbf{x}^{\alpha_i}) = \deg(\mathbf{x}^{\beta_i})$ for $1+j \leq i \leq j+k$.

Assume that all facet binomials are such that the degree of the leading term is strictly greater than the degree of the other term. Then the cone

$$C' = \left\{\mathbf{u} \in \mathbb{R}^n_+ \mid \alpha_i \cdot \mathbf{u} \geq \beta_i \cdot \mathbf{u} \text{ for all } 1 \leq i \leq j\right\}$$

equals the Gröbner cone $C(I(\mathscr{C}))$ of the Gröbner basis $\mathscr{G}$. But then $\mathbf{1} \in \mathrm{Int}(C') = \mathrm{Int}(C(I(\mathscr{C})))$, which is a contradiction. □

Lemma 4 *Let $\mathscr{G}_{new}$ be the reduced Gröbner basis obtained from $\mathscr{G}_{old}$ by flipping the facet binomial $\mathbf{x}^\alpha - \mathbf{x}^\beta$. Any new leading terms in $\mathscr{G}_{new}$, i.e., leading terms of $\mathscr{G}_{new}$ that do not appear in $\mathscr{G}_{old}$, are divisible by $\mathbf{x}^\alpha$.*

Proof Any new leading terms arise from the Gröbner basis computation of the quasi-monomial ideal

$$T := \left\{\mathbf{x}^\beta - \mathbf{x}^\alpha\right\} \cup T', \quad T' := \left\{\mathbf{x}^{\alpha_i} \mid \mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \in \mathscr{G}_{old}\right\}$$

that consists of the designated flipping binomial with changed leading term and all the other leading terms in $\mathscr{G}_{old}$. To be more precise, a new leading term arises from an S-polynomial of the form

$$S\left(\mathbf{x}^\beta - \mathbf{x}^\alpha, \mathbf{x}^{\alpha_i}\right) = \mathbf{x}^\gamma \mathbf{x}^\alpha, \quad \text{where } \mathbf{x}^\gamma = \mathrm{lcm}(\mathbf{x}^\beta, \mathbf{x}^{\alpha_i})/\mathbf{x}^\beta,$$

which is not being reduced to zero by the elements in T. When computing a Gröbner basis, then this S-polynomial is either reduced to zero or its remainder on division by the set T is added to the Gröbner basis of T. We distinguish the following situations:

1. Neither $\mathbf{x}^\beta$ nor any monomial in T' divides $\mathbf{x}^\gamma\mathbf{x}^\alpha$: The monomial $\mathbf{x}^\gamma\mathbf{x}^\alpha$ cannot be further reduced and thus is being attached to the Gröbner basis of T.
2. A monomial in T' divides $\mathbf{x}^\gamma\mathbf{x}^\alpha$: The monomial $\mathbf{x}^\gamma\mathbf{x}^\alpha$ is being reduced to zero and thus, this S-polynomial results in no new term.
3. The monomial $\mathbf{x}^\beta$ divides $\mathbf{x}^\gamma\mathbf{x}^\alpha$: Since $\mathbf{x}^\alpha$ and $\mathbf{x}^\beta$ have disjoint support (see [2]), $\mathbf{x}^\beta$ has to divide $\mathbf{x}^\gamma$, the monomial $\mathbf{x}^\gamma\mathbf{x}^\alpha$ is reduced to

$$\mathbf{x}^\gamma\mathbf{x}^\alpha - \mathbf{x}^\alpha \frac{\mathbf{x}^\gamma}{\mathbf{x}^\beta}\left(\mathbf{x}^\beta - \mathbf{x}^\alpha\right) = \frac{\mathbf{x}^\gamma}{\mathbf{x}^\beta}\left(\mathbf{x}^\alpha\right)^2.$$

So, whenever the S-polynomial cannot be reduced to zero, we obtain a monomial which is divisible by $\mathbf{x}^\alpha$. □

Proposition 5 *$T_\succ(I(\mathscr{C}))$ is an acyclic directed graph with a unique sink that we will call the reverse search tree.*

Proof We prove that $T_\succ(I(\mathscr{C}))$ is a tree by showing that there is no cycle in this construction. We show this by contradiction. For the other claims see the proof of [5, Theorem 2.6].

Assume that there is a cycle in the reverse search tree, say $\mathscr{G}_1 \longrightarrow \mathscr{G}_2 \longrightarrow \ldots \longrightarrow \mathscr{G}_\ell \longrightarrow \mathscr{G}_1$, where $\mathscr{G}_{i+1}$ is obtained from $\mathscr{G}_i$ by flipping along $\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i}$. Then $\mathscr{G}_i$ contains this binomial with leading term $\mathbf{x}^{\alpha_i}$ and $\mathscr{G}_{i+1}$ with leading term $\mathbf{x}^{\beta_i}$. Inspecting the cycle we see that the binomial $\mathbf{x}^{\alpha_1} - \mathbf{x}^{\beta_1}$ lies in $\mathscr{G}_1$ with leading term $\mathbf{x}^{\alpha_1}$ and appears in $\mathscr{G}_2$ with leading term $\mathbf{x}^{\beta_1}$. Then no binomial in $\mathscr{G}_2$ has the leading term $\mathbf{x}^{\alpha_1}$. However, as we arrive at $\mathcal{G}_1$ after ℓ flipping steps, we conclude that $\mathbf{x}^{\alpha_1} - \mathbf{x}^{\beta_1}$ must be inserted at some successive flipping step. Assume that this happens in the i_1th flipping process, $1 < i_1 \leq \ell$. Then by Lemma 4, $\mathbf{x}^{\alpha_1}$ is divisible by $\mathbf{x}^{\alpha_{i_1}}$. And since $\mathcal{G}_1$ is a Gröbner basis this implies that $\mathbf{x}^{\alpha_{i_1}}$ cannot be the leading term of any element in $\mathcal{G}_1$; it must have been inserted as a new leading term during some preceding flipping step, say $i_2 < i_1$. By the same argument the monomial $\mathbf{x}^{\alpha_{i_1}}$ is divisible by $\mathbf{x}^{\alpha_{i_2}}$ and then $\mathbf{x}^{\alpha_{i_2}}$ cannot appear as the leading term of any element in $\mathcal{G}_1$. Continuing this process we get a decreasing sequence of indices $i_1 > i_2 > i_3 > \ldots$ which eventually must terminate, say after k steps, i.e., $i_k = 1$. Then $\mathbf{x}^{\alpha_{i_k}} = \mathbf{x}^{\alpha_1}$ and from the divisibility relations $\mathbf{x}^{\alpha_{i_k}} \mid \mathbf{x}^{\alpha_{i_{k-1}}} \mid \ldots \mid \mathbf{x}^{\alpha_{i_2}} \mid \mathbf{x}^{\alpha_{i_1}} \mid \mathbf{x}^{\alpha_1}$ we actually obtain equality of all leading terms of the flipping binomials. However, this is a contradiction. □

The following proposition states the discussion at the end of Sect. 2.

Proposition 6 *Two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are permutation-equivalent if and only if they have the same degree compatible Gröbner fan structure, i.e., there is permutation $\sigma \in S_n$ such that $\sigma\,(Gfan(\mathcal{C}_1)) = Gfan(\mathcal{C}_2)$, where $\sigma\,(Gfan(\mathcal{C}_1))$ means permuting the variables in each of the degree compatible Gröbner basis within the fan.*

Example 7 Consider two binary [6, 3] codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with respective parity check matrices

$$H_1 = \begin{pmatrix} 1\,1\,0\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1 \end{pmatrix},\; H_2 = \begin{pmatrix} 1\,1\,1\,1\,1\,1 \\ 0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,1\,0\,0 \end{pmatrix}$$

In [1, Example 2 and 5] it is shown that these codes are not permutation-equivalent. Here, we show how the degree compatible Gröbner fans of both codes can be employed to show their non-equivalence. The degree compatible Gröbner fan for $\mathcal{C}_1$ consists of 8 Gröbner basis which are all of cardinality 6 (see Example 8). The Gröbner basis for $\mathcal{C}_2$ w.r.t. the grevlex basis is given by

$$\{x_3 - x_5, x_1 - x_5, x_4x_5 - x_2x_6, x_2x_5 - x_4x_6, x_2x_4 - x_5x_6\} \cup \left\{x_i^2 - 1 \mid i = 2, 4, 5, 6\right\}$$

and consists of nine elements. Thus, we can already conclude that these two codes cannot be permutation-equivalent.

Example 8 The reverse search tree $T_{\succ}(I(\mathcal{C}))$ for the binary [6, 3] code $\mathcal{C}_1$ from the previous example with $\succ$ being pure lex is given in Fig. 1.
And the Gröbner bases are (the flipping binomials are underlined)

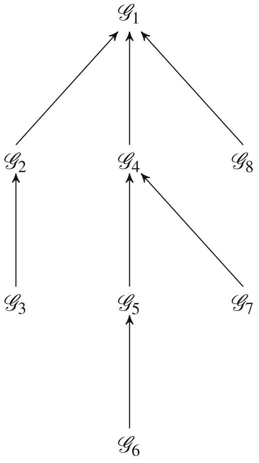


Fig. 1 The reverse search tree for $\mathscr{C}_1$

$$\begin{aligned}
\mathscr{G}_1 &= \left\{x_1 - x_2, x_3 - x_4, x_5 - x_6, x_2^2 - 1, x_4^2 - 1, x_6^2 - 1\right\}\\
\mathscr{G}_2 &= \left\{x_1 - x_2, \underline{x_4 - x_3}, x_5 - x_6, x_2^2 - 1, x_3^2 - 1, x_6^2 - 1\right\}\\
\mathscr{G}_3 &= \left\{\underline{x_2 - x_1}, x_4 - x_3, x_5 - x_6, x_1^2 - 1, x_3^2 - 1, x_6^2 - 1\right\}\\
\mathscr{G}_4 &= \left\{x_1 - x_2, x_3 - x_4, \underline{x_6 - x_5}, x_2^2 - 1, x_4^2 - 1, x_5^2 - 1\right\}\\
\mathscr{G}_5 &= \left\{x_1 - x_2, \underline{x_4 - x_3}, x_6 - x_5, x_2^2 - 1, x_3^2 - 1, x_5^2 - 1\right\}\\
\mathscr{G}_6 &= \left\{\underline{x_2 - x_1}, x_4 - x_3, x_6 - x_5, x_1^2 - 1, x_3^2 - 1, x_5^2 - 1\right\}\\
\mathscr{G}_7 &= \left\{\underline{x_2 - x_1}, x_3 - x_4, x_6 - x_5, x_1^2 - 1, x_4^2 - 1, x_5^2 - 1\right\}\\
\mathscr{G}_8 &= \left\{\underline{x_2 - x_1}, x_3 - x_4, x_5 - x_6, x_1^2 - 1, x_4^2 - 1, x_6^2 - 1\right\}.
\end{aligned}$$

4 Conclusions

We have shown how the computation of the degree compatible Gröbner fan of a code is useful for determining the code equivalence problem. Anyway one can not forget that this is an NP-problem and therefore the Gröbner basis computation comprises a hard step. Further research in the topic points toward analyzing heuristic techniques for eliminating the need of transverse the whole fan or at least for trying to deduce the answer from partial information about the initial Gröbner basis.

Acknowledgements N. Dück is partially supported by a grant of the Deutscher Akademischer Austauschdienst. I. Márquez-Corbella and E. Martínez-Moro are supported by the Spanish MINECO grant MTM2012-36917-C03-03.

References

1. Borges-Quintana, M., Borges-Trenard, M.A., Martínez-Moro, E.: On a Gröbner bases structure associated to linear codes. J. Discret. Math. Sci. Cryptogr. **10**(2), 151–191 (2007)
2. Borges-Quintana, M., Borges-Trenard, M.A., Fitzpatrick, P., Martínez-Moro, E.: Gröbner bases and combinatorics for binary codes. Appl. Algebra Eng. Commun. Comput. **19**(5), 393–411 (2008)
3. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Undergraduate Texts in Mathematics, 2nd edn. Springer, New York (1997)
4. Fukuda, K., Jensen, A.N., Thomas, R.R.: Computing Gröbner fans. Math. Comput. **76**(260), 2189–2212 (2007) (electronic)
5. Huber, B., Thomas, R.R.: Computing Gröbner fans of toric ideals. Exp. Math. **9**(3), 321–331 (2000)
6. Márquez-Corbella, I., Martínez-Moro, E.: Algebraic structure of the minimal support codewords set of some linear codes. Adv. Math. Commun. **5**(2), 233–244 (2011)
7. Márquez-Corbella, I., Martínez-Moro, E., Suárez Canedo, E.: On the ideal associated to any linear code. Adv. Math. Commun. **-**(-) (2014, submitted)

Lattice Encoding of Cyclic Codes from Skew-Polynomial Rings

Jérôme Ducoat and Frédérique Oggier

Abstract We propose a construction of lattices from cyclic codes from skew-polynomial rings. This construction may be seen as a variation of Construction A of lattices from linear codes, obtained from quotients of orders in cyclic division algebras. An application is coset encoding of wiretap space-time codes.

Keywords Lattices • Cyclic division algebras • Skew-polynomials • Cyclic codes

1 Introduction

Constructions of lattices from linear codes over finite fields (or rings) have been classically studied, starting from the so-called Construction A [4, 6] of lattices from binary linear codes. Let $\rho : \mathbb{Z}^N \mapsto \mathbb{F}_2^N$ be the map of reduction modulo 2 componentwise. Let $C \subset \mathbb{F}_2^N$ be an (N, k) linear binary code. Then $\rho^{-1}(C)$ is a lattice. One possible way of generalizing this construction is by considering cyclotomic fields [5]. Let $\mathbb{Q}(\zeta_p)$ be a cyclotomic field, with ring of integers $\mathbb{Z}[\zeta_p]$, where ζ_p is a primitive pth root of unity, p a prime. Let $\rho : \mathbb{Z}[\zeta_p]^N \mapsto \mathbb{F}_p^N$ be this time the reduction componentwise modulo the prime ideal $\mathfrak{p} = (1 - \zeta_p)$. Then $\rho^{-1}(C)$ is a lattice, when C is an (N, k) linear code over $\mathbb{F}_p$. In particular, $p = 2$ yields the binary Construction A. Similar constructions from number fields with a totally ramified prime and from totally real cyclic number fields with a completely split prime have been proposed respectively in [7] and [12]. Note that the latter construction has also been generalized to cyclic division algebras.

Let K/F be a cyclic extension of number fields, with respective maximal orders $\mathscr{O}_K$ and $\mathscr{O}_F$. We are proposing a variation of the above Constructions A, where lattices are obtained from quotients of the natural order Λ of a cyclic division algebra, as explained in Sect. 2, instead of quotients of the maximal order of number fields. The resulting quotient $\Lambda/\mathfrak{p}\Lambda$ of the natural order of a cyclic division algebra

J. Ducoat (✉) • F. Oggier (✉)
Division of Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, 637371 Singapore, Singapore
e-mail: jducoat@ntu.edu.sg; frederique@ntu.edu.sg

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_16

by a two-sided ideal $\mathfrak{p}\Lambda$, where $\mathfrak{p}$ is a prime ideal of $\mathcal{O}_F$ inert in K/F, turns out to be isomorphic to a ring of skew-polynomials. Denote this isomorphism by ψ. Let C be a cyclic code constructed over the ring of skew-polynomials (see Sect. 3) and let ρ denote the compositum of the canonical projection $\Lambda \to \Lambda/\mathfrak{p}\Lambda$ with ψ. Then $\rho^{-1}(C)$ is a lattice. Application of this construction to space-time coding, more specifically to coset encoding, is discussed in Sect. 4.

2 Quotients of Cyclic Division Algebras

Let K/F be a number field extension of degree n with cyclic Galois group $\langle\sigma\rangle$, and respective rings of integers $\mathcal{O}_K$ and $\mathcal{O}_F$. Consider the cyclic algebra

$$K \oplus Ke \oplus \cdots Ke^{n-1}$$

where $e^n = u \in F$, and $ek = \sigma(k)e$ for $k \in K$. We assume that u^i, $i = 0, \dots, n-1$, are not norms in K/F so that the algebra is division. Let Λ be its natural order

$$\Lambda = \mathcal{O}_K \oplus \mathcal{O}_K e \oplus \cdots \oplus \mathcal{O}_K e^{n-1}.$$

Let $\mathfrak{p}$ be a prime ideal of $\mathcal{O}_F$ so that $\mathfrak{p}\Lambda$ is a two-sided ideal of Λ. Assume that $\mathfrak{p}$ is inert in K/F, so that $\mathfrak{p}\mathcal{O}_K$ is a prime ideal of $\mathcal{O}_K$. Then $\Lambda/\mathfrak{p}\Lambda$ is an $\mathcal{O}_K/\mathfrak{p}\mathcal{O}_K$-algebra and from [9], we have the following isomorphism:

$$\Lambda/\mathfrak{p}\Lambda \simeq (\mathcal{O}_K/\mathfrak{p}\mathcal{O}_K) \oplus (\mathcal{O}_K/\mathfrak{p}\mathcal{O}_K)e \oplus \cdots \oplus (\mathcal{O}_K/\mathfrak{p}\mathcal{O}_K)e^{n-1}.$$

Note that since $\mathfrak{p}\mathcal{O}_K$ is a prime ideal of $\mathcal{O}_K$, the finite ring $\mathcal{O}_K/\mathfrak{p}\mathcal{O}_K$ is an integral domain, so is a finite field that we denote by $\mathbb{F}_q$. Here, $q = p^{nf}$, where p is the prime number lying below $\mathfrak{p}$ and f is the inertial degree of $\mathfrak{p}$ above p.

The algebra $\Lambda/\mathfrak{p}\Lambda$ can alternatively be described in terms of skew-polynomial with coefficients in $\mathcal{O}_K/\mathfrak{p}\mathcal{O}_K = \mathbb{F}_q$.

Definition 1 Given a ring R with a group $\langle\sigma\rangle$ acting on it, the skew-polynomial ring $S[x;\sigma]$ is the set of polynomials $s_0 + s_1 x + \ldots + x_n x^n$, $s_i \in S$ for $i = 0, \dots, n$, with $xs = \sigma(x)s$ for all $s \in S$.

Lemma 2 *There is an $\mathbb{F}_q$-algebra isomorphism between $\Lambda/\mathfrak{p}\Lambda$ and the quotient of $\mathbb{F}_q[x;\sigma]$ by the two-sided ideal generated by $x^n - u$.*

Proof We define the map

$$\varphi : \mathbb{F}_q[x;\sigma] \to \Lambda/\mathfrak{p}\Lambda$$
$$f(x) \mapsto f(e).$$

Using the isomorphism given above and in [9], it is easily seen that φ is a surjective $\mathbb{F}_q$-algebra homomorphism. Moreover, the kernel of φ is the two-sided ideal of

$\mathbb{F}_q[x;\sigma]$ generated by $x^n - u$. Indeed, it is easily seen that $x^n - u$ lies in $\ker(\varphi)$. Conversely, let $f(x) \in \ker(\varphi)$. We write

$$f(x) = \sum_{i=0}^{m} s_i x^i$$

for some $s_i \in \mathbb{F}_q$, $i = 0, \ldots, m$. Then $f(e) = 0$ in $\Lambda/\mathfrak{p}\Lambda$. Since the ring $\mathbb{F}_q[x;\sigma]$ is left Euclidean [10], there exist some polynomials $g(x)$ and $h(x)$ such that

$$f(x) = g(x)(x^n - u) + h(x)$$

where $h(x)$ has degree $\leq n - 1$. Hence, $f(e) = 0$ is equivalent to $h(e) = 0$. Yet, $0 = h(e) = r_0 + r_1 e + \cdots + r_{n-1}e^{n-1}$ in $\Lambda/\mathfrak{p}\Lambda \simeq (\mathscr{O}_K/\mathfrak{p}\mathscr{O}_K) \oplus (\mathscr{O}_K/\mathfrak{p}\mathscr{O}_K)e \oplus \cdots \oplus (\mathscr{O}_K/\mathfrak{p}\mathscr{O}_K)e^{n-1}$. Therefore, $r_0 = r_1 = \cdots = r_{n-1} = 0$ and $h(x) = 0$. We conclude that $f(x)$ is a (left) multiple of $x^n - u$. Consequently, $\ker(\varphi) = (x^n - u)$ and we get the desired isomorphism. □

Denote by ψ the inverse isomorphism of the one given in Lemma 2:

$$\psi : \Lambda/\mathfrak{p}\Lambda \cong \mathbb{F}_q[x;\sigma]/(x^n - u).$$

Note that since $u \in F$, $x^n - u$ belongs to the center of $\mathbb{F}_q[x;\sigma]$ and the ideal $(x^n - u)$ is two-sided.

Let $\mathscr{I}$ be a left ideal of Λ. Assume that $\mathscr{I} \cap \mathscr{O}_F \supset \mathfrak{p}$. Then $\mathscr{I}/\mathfrak{p}\Lambda$ is an ideal of $\Lambda/\mathfrak{p}\Lambda$. In the sequel, we will study the left ideal $\psi(\mathscr{I}/\mathfrak{p}\Lambda)$ of $\mathbb{F}_q[x;\sigma]/(x^n - u)$.

3 Polynomial Codes and a Variation of Construction A

Definition 3 ([3]) Let $f \in \mathbb{F}_q[x;\sigma]$ be a polynomial of degree n. If (f) is a two-sided ideal of $\mathbb{F}_q[x;\sigma]$, then a σ*-code* consists of codewords $a = (a_0, a_1, \ldots, a_{n-1})$ that are coefficient tuples of elements $a(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}$ of a left ideal of $\mathbb{F}_q[x;\sigma]/(f)$. The elements $a(x)$ are left multiples of a right divisor g of f. If f lies in the center of $\mathbb{F}_q[x;\sigma]$, then the σ-code corresponding to the left ideal $(g)/(f)$ is called *a central* σ*-code*.

Using the isomorphism ψ defined in Sect. 2, for every left ideal $\mathscr{I}$ of Λ, we consider the σ-code $C = \psi(\mathscr{I}/\mathfrak{p}\Lambda)$ over $\mathbb{F}_q$.

We set the map:

$$\rho : \Lambda \to \psi(\Lambda/\mathfrak{p}\Lambda) = \mathbb{F}_q[x;\sigma]/(x^n - u),$$

compositum of the canonical projection $\Lambda \to \Lambda/\mathfrak{p}\Lambda$ with ψ. We then set

$$L = \rho^{-1}(C) = \mathscr{I}.$$

Then L is a lattice, that is a $\mathbb{Z}$-module of rank $n^2[F:\mathbb{Q}]$ since $\mathcal{O}_K$ is a $\mathbb{Z}$-module of rank $n[F:\mathbb{Q}]$.

From this point of view, the above construction may be interpreted as a variation of Construction A [4], which consists of obtaining a lattice from a linear code over a finite field (ring), as shortly described in the introduction. This is also a generalization of the lattice construction of [8], defined over number fields.

Example 4 Let $K=\mathbb{Q}(i)$ and $F=\mathbb{Q}$. Then $\mathcal{O}_F=\mathbb{Z}$ and $\mathcal{O}_K=\mathbb{Z}[i]$. Set $p=3$, which remains inert in $\mathbb{Q}(i)$. Hence, $\mathbb{Z}[i]/3\mathbb{Z}[i]\simeq\mathbb{F}_9$. Let $\mathfrak{Q}$ be the quaternion division algebra defined by

$$\mathfrak{Q}=\mathbb{Q}(i)\oplus\mathbb{Q}(i)e,$$

with $e^2=-1$. Since $N_{K/F}(a+ib)=a^2+b^2$, $a,b\in\mathbb{Z}$, -1 cannot be a norm and $\mathfrak{Q}$ is indeed a quaternion division algebra. We set $\Lambda=\mathbb{Z}[i]\oplus\mathbb{Z}[i]e$ and $\mathcal{I}=(1+i+e)\Lambda$. Then $\mathcal{I}$ contains 3 since the norm of $1+i+e$ is 3. Let α denote a primitive root of $\mathbb{F}_9$ over $\mathbb{F}_3$, satisfying $\alpha^2+1=0$. We have

$$\psi((1+i+e)\mathrm{mod}3)=1+\alpha+x,$$

which is a right divisor of x^2+1 in $\mathbb{F}_9[x;\sigma]$:

$$x^2+1=(x-1+\alpha)(x+1+\alpha).$$

Therefore, the left ideal $(x+1+\alpha)\mathbb{F}_9[x;\sigma]/(x^2+1)$ consisting of the left multiples of $x+1+\alpha$ modulo x^2+1 is a central σ-code. Taking the pre-image by ψ, it corresponds to the left-ideal $\mathcal{I}/3\Lambda$, with $\mathcal{I}=\Lambda(1+i+e)$.

4 Application to Space-Time Codes

Cyclic division algebras are by now classically used to design space-time codes [2, 11]. Matrix codewords are obtained as follows. From now on, to make the notation easier, we assume that $u\in\mathcal{O}_F$. To any element $a=a_0+a_1e+\cdots+a_{n-1}e^{n-1}$ of Λ, we can associate a matrix in $\mathrm{Mat}_n(\mathcal{O}_K)$ (since $u\in\mathcal{O}_F$) by:

$$M(a)=\begin{bmatrix} a_0 & u\sigma(a_{n-1}) & u\sigma^2(a_{n-2}) & \cdots & u\sigma^{n-1}(a_1)\\ a_1 & u\sigma(a_0) & u\sigma^2(a_{n-1}) & \cdots & u\sigma^{n-1}(a_2)\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \vdots & \vdots & \vdots & & u\sigma^{n-1}(a_{n-1})\\ a_{n-1} & u\sigma(a_{n-2}) & u\sigma^2(a_{n-3}) & \cdots & u\sigma^{n-1}(a_0)\end{bmatrix}.$$

The map

$$\begin{aligned} \Lambda &\to \mathrm{Mat}_n(\mathcal{O}_K) \\ a &\mapsto M(a) \end{aligned}$$

is an $\mathcal{O}_K$-algebra injective homomorphism.

We apply this to our previous example.

Example 5 For $q = a + be$ in the natural order $\mathbb{Z}[i] \oplus \mathbb{Z}[i]e$ of the quaternion algebra $\mathfrak{Q}$, $a, b \in \mathbb{Z}[i]$

$$M(q) = \begin{bmatrix} a & -\bar{b} \\ b & \bar{a} \end{bmatrix}$$

where $\bar{\cdot}$ is the non-trivial Galois automorphism of $\mathbb{Q}(i)/\mathbb{Q}$. Let $t = (a + be)(1 + i + e)$ be an element of $\mathcal{I} = \Lambda(1 + i + e)$ (with $a, b \in \mathbb{Z}[i]$). Then

$$t = a(1 + i) - b + (a + b(1 - i))e.$$

Hence,

$$M(t) = \begin{bmatrix} a(1 + i) - b & -(\overline{a} + \overline{b}(1 + i)) \\ a + b(1 - i) & \overline{a}(1 - i) - \overline{b} \end{bmatrix}.$$

Note that $\mathcal{I} = \rho^{-1}(C)$ is a real lattice with rank 4 embedded in $\mathbb{R}^8$: by vectorizing the matrices $M(t)$ and separating real and imaginary parts, a generator matrix of this lattice is given by

$$\begin{bmatrix} 1 & 1 & 1 & 0 & -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & 1 & 0 & 1 & -1 & -1 \\ -1 & 0 & 1 & -1 & -1 & -1 & -1 & 0 \\ 0 & -1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Let now $v = (v_1, \ldots, v_n)$ be the information vector to be mapped to a lattice point in L, where L is used as a lattice code. The lattice $L = \rho^{-1}(C) = \mathcal{I}\Lambda$ may by construction be written as a union of cosets of $\mathfrak{p}\Lambda$, where each coset representative may be chosen to be a codeword in the code C. Namely, if g is a right divisor of $x^n - u$ and if a central σ-code $C = (g)/(x^n - u) \subset \mathbb{F}_q[x;\sigma]/(x^n - u)$ has dimension $k = n - \deg(g)$, since

$$\Lambda/\mathfrak{p}\Lambda \cong \mathbb{F}_q[x;\sigma]/(x^n - u)$$

there is an isomorphism

$$\mathcal{I}/\mathfrak{p}\Lambda \cong C.$$

This allows us to associate in a unique way a coset of $\mathfrak{p}\Lambda$ to a codeword. The mapping from v to a point in L may be done by attributing some information coefficients $v_1, \ldots, v_k$ to be encoded using the code C, and the rest of the information coefficients to be mapped to a point in the lattice $\mathfrak{p}\Lambda$. Coset encoding is necessary in the context of wiretap codes [1]: information symbols are mapped to a codeword in C, while random symbols are picked uniformly at random in the lattice $\mathfrak{p}\Lambda$ to confuse the eavesdropper. The construction of the lattice $L = \rho^{-1}(C) = \mathscr{I}$ thus enables coset encoding for wiretap space-time codes.

5 Future Work

In this paper, we presented a construction of lattices from cyclic codes from skew-polynomials, which can be seen as a variation of the well known Construction A of lattices from linear codes. Natural future research directions include:

- Linking the properties of the cyclic code C to that of the lattice $L = \rho^{-1}(C)$: there are standard duality results for the classical Construction A, relating the dual of the code with the dual of the lattice, as well as the weight enumerator of the code with the theta series of the lattice.
- Design of wiretap space-time codes: this consists of choosing the cyclic division algebras, the corresponding two-sided ideal and cyclic code, to optimize the confusion at the eavesdropper.

Acknowledgements The research of J. Ducoat and F. Oggier is supported by the Singapore National Research Foundation under Research Grant NRF-RF2009-07.

References

1. Belfiore, J.C., Oggier, F.: An error probability approach to MIMO wiretap channels. IEEE Trans. Commun. **61**(8), 3396–3403 (2013)
2. Berhuy, G., Oggier, F.: An Introduction to Central Simple Algebras and Their Applications to Wireless Communication. AMS, Providence (2013)
3. Boucher, D., Ulmer, F.: Coding with skew polynomial rings. J. Symb. Comput. **44**, 1644–1656 (2009)
4. Conway, J., Sloane, N.: Sphere Packings, Lattices and Groups. Springer, New York (1999)
5. Ebeling, W.: Lattices and Codes. A Course Partially Based on Lectures by Friedrich Hirzebruch. Advanced Lectures in Mathematics. Springer, Wiesbaden/New York (2013)
6. Forney, G.D.: Coset codes − part i: introduction and geometrical classification. IEEE Trans. Inf. Theory **34**(5), 1123–1151 (1988)
7. Kositwattanarerk, W., Ong, S., Oggier, F.: Wiretap encoding of lattices from number fields using codes over $\mathbb{F}_p$. In: Proceedings of the IEEE International Symposium on Information Theory (ISIT), Istanbul (2013)
8. Oggier, F., Belfiore, J.C.: Enabling multiplication in lattice codes via construction A. In: Proceedings of the IEEE International Workshop on Information Theory, Sevilla (2013)

9. Oggier, F., Sethuraman, B.A.: Quotients of orders in cyclic algebras and space-time codes. Adv. Math. Commun. **7**, 441–461 (2013)
10. Ore, O.: Theory of non-commutative polynomials. Ann. Math. **34**, 1644–1656 (1933)
11. Sethuraman, B.A., Rajan, B.S., Shashidhar, V.: Full-diversity, high-rate space-time block codes from division algebras. IEEE Trans. Inf. Theory **49**(10), 2596–2616 (2003)
12. Vehkalahti, R., Kositwattanarerk, W., Oggier, F.: Constructions a of lattices from number fields and division algebras. In: Proceedings of the IEEE International Symposium on Information Theory (ISIT), Honolulu (2014)

On Extendibility of Additive Code Isometries

Serhii Dyshko

Abstract For linear codes, the MacWilliams Extension Theorem states that each linear isometry of a code extends to a linear isometry of the whole space. But, in general, it is not the situation for nonlinear codes. In the paper it was proved, that if the length of an additive code is less than some threshold value, then an analogue of the MacWilliams Extension Theorem holds. One family of unextendible code isometries for the threshold value of code length is described.

Keywords Additive code • Code isometry • MacWilliams extension theorem

1 Introduction

The code is a subset of the space with the Hamming metric. A map that preserves the Hamming metric is called an isometry. The description of code isometries is fundamental because it helps to identify codes with equal metric parameters. Moreover, results, based on the properties of weight and distance enumerators, could be translated without any changes from a code to all its isometric codes.

Besides the metric, codes can have additional algebraic structures, for example the structure of a vector space or a group. The most developed are linear codes. A code is said to be linear if it is a vector space over the alphabet, where the alphabet is considered as a finite field. There is a full description of linear isometries of linear codes. The famous MacWilliams Extension Theorem states that every linear code isometry extends to a linear isometry of the whole space. The proof of the MacWilliams Extension Theorem firstly appeared in the works of MacWilliams and it was later refined by several authors. Namely, Ward and Wood greatly simplified it, using character theory approach (see [8]).

Unlike linear codes, there are nonlinear codes with isometries that do not extend to isometries of the whole space. In general, the problem of description of code isometries for nonlinear case is difficult. Nevertheless, in some classes of codes it can be solved. For example, in [1, 5] and [7] authors describe a lot of code families

S. Dyshko (✉)
IMATH, Université de Toulon, La Garde, France
e-mail: dyshko@univ-tln.fr

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_17

that satisfy extendibility property. There they also observe various classes that do not satisfy it. Among the studied families there are some subclasses of codes that achieve the Singleton bound (MDS codes, see [6, p. 20]), some subclasses of codes with equal distance between codewords (equidistant codes) and some perfect codes (see [6, Ch. § 11]).

In this paper we focus our attention on the class of additive codes and their additive isometries. A code is called additive if it is an additive abelian group. An isometry of an additive code is called additive if it preserves the group structure of the code. Nonlinear codes are not widely used in practice and are less developed, but it appears that additive codes with additional requirement of a special kind of self-orthogonality naturally describe quantum stabilizer codes that are used to protect quantum information (see [4]). The description of quantum code isometries greatly depends on the description of additive code isometries.

The main result of this paper is formulated in Theorem 10. We determine the length threshold for which an analogue of the MacWilliams Extension Theorems for additive codes holds. We also proved that this result cannot be improved by increasing the bound on the code length.

2 Additive Isometries of Space

Let L be a finite field, let m be a positive integer and let K be a subfield of L. A *code* is a subset of L^m. A code is called *K-linear* if it is a K-linear vector space in L^m. If the code is L-linear we call this code *linear*. *Additive* code is a code that is closed under addition. Any K-linear code is additive. In the other direction, any additive code in L^m is $\mathbb{F}_p$-linear, where p is the characteristic of L.

An *isometry* of a code $C \subseteq L^m$ is a map $f : C \to L^m$ that preserves the Hamming distance. If f is a K-linear map, then f is an isometry if and only if f preserves the Hamming weight.

Example 1 Consider two codes $C_1 = \{(0,0,0),(1,1,0),(\omega,0,1),(\omega^2,1,1)\}$ and $C_2 = \{(0,0,0),(0,\omega^2,\omega),(1,0,1),(1,\omega^2,\omega^2)\}$ in $\mathbb{F}_4^3$, where $\mathbb{F}_4 = \{0,1,\omega,\omega^2\}$ and $\omega + 1 = \omega^2$. All the codes are $\mathbb{F}_2$-linear. Define a map $f : C_1 \to C_2$ in the following way: $f\big((0,0,0)\big) = (0,0,0)$, $f\big((1,1,0)\big) = (0,\omega^2,\omega)$, $f\big((\omega,0,1)\big) = (1,0,1)$ and $f\big((\omega^2,1,1)\big) = (1,\omega^2,\omega^2)$. Evidently, the map f is $\mathbb{F}_2$-linear and preserves the Hamming weight. Therefore f is an $\mathbb{F}_2$-linear isometry of the $\mathbb{F}_2$-linear code C_1 in $\mathbb{F}_4^3$. Note that C_1 and C_2 are not $\mathbb{F}_4$-linear.

A map $f : L^m \to L^m$ is called *monomial*, if there exist a permutation $\pi \in S_m$ and $c_1, c_2 \ldots, c_m \in L \setminus \{0\}$ such that for all $u \in L^m$, $f(u) = f\big((u_1,u_2,\ldots,u_m)\big) = (c_1 u_{\pi(1)}, c_2 u_{\pi(2)}, \ldots, c_m u_{\pi(m)})$. It is easy to see that a monomial map is a linear isometry of L^m and each linear isometry of the whole space L^m is a monomial map.

Theorem 2 (MacWilliams Extension Theorem) *Let L be a finite field, let m be a positive integer and let $C \subseteq L^m$ be a linear code. Each linear isometry of C extends to a monomial map.*

Considering the arguments above, the MacWilliams Extension Theorem states that any linear isometry of a linear code extends to a linear isometry of the whole space.

We deal with K-linear isometries of a K-linear code and their extendibility to the whole space L^m. Hence, we describe all K-linear isometries of the space L^m. Let $\mathrm{Aut}_K(L)$ denotes the set of all K-linear invertible maps from L to itself.

Definition 3 A map $f : L^m \to L^m$ is called *general monomial* if there exist a permutation $\pi \in S_m$ and $g_1, \dots, g_m \in \mathrm{Aut}_K(L)$ such that for all $u \in L^m$ the following holds: $f(u) = f\big((u_1, u_2, \dots, u_m)\big) = \big(g_1(u_{\pi(1)}), g_2(u_{\pi(2)}), \dots, g_m(u_{\pi(m)})\big)$.

Proposition 4 *A general monomial map is a K-linear isometry of the space L^m. Moreover, any K-linear isometry of the space is a general monomial map.*

Proof From the definition it is easy to see that a general monomial map is a K-linear isometry. In [3], it was proved that any isometry of the space L^m is a composition of coordinate permutation and a tuple of permutations of the alphabet L, where the ith element in the tuple acts on the ith coordinate. Since a K-linear permutation of L is exactly an element of $\mathrm{Aut}_K(L)$ we get the statement of the proposition. □

A general theorem, analogue of the MacWilliams Extension Theorem, does not exist for nonlinear codes. This means that there is a nonlinear code and an isometry of the code that does not extend to an isometry of the whole space. Call such isometries *unextendible*. We have the same situation even if we look at additive codes. The counterexample is the following.

Example 5 Let $m = |K| + 1$. Consider two K-linear codes $C_1 = \langle v_1, v_2 \rangle_K$ and $C_2 = \langle u_1, u_2 \rangle_K$ with

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & x_1 & x_2 & \dots & x_{|K|} \end{pmatrix} \xrightarrow{f} \begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ 0 & \omega & \omega & \dots & \omega \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} ,$$

where $x_i \in K$ are all different and $\omega \in L \backslash K$. Define the K-linear map $f : C_1 \to C_2$ in the following way: $f(v_1) = u_1$ and $f(v_2) = u_2$. The map f is an isometry. But, there is no general monomial map that acts on C_1 in the same way as the map f. The first coordinates of all vectors in C_2 are always zero, but there is no such all-zero coordinate in C_1.

3 Extendibility of Additive Isometries

Let C be a K-linear code in L^m. Denote by $x_1, \dots, x_k \in L^m$ a K-linear basis of C. A matrix $A = (a_{ij})_{1 \le i \le k, 1 \le j \le m}$ with entries from L, formed by k rows that correspond to vectors $x_1, \dots, x_k$, is called a *generator matrix* of C. Let $\mathrm{M}_{a \times b}(F)$

denotes the set of all $a \times b$ matrices with the entries from a field F. Obviously, $A \in \mathrm{M}_{k\times m}(L)$.

Denote the degree of the extension $[L : K] = n$. Consider L as a n-dimensional vector space over K and fix its basis $b_1, \dots, b_n \in L$. This is equivalent to the establishment of an isomorphism $L \cong K^n$ of K-linear vector spaces. In the generator matrix A replace each entry $a_{ij} \in L$ by the corresponding vector-row $\big(a_{ij}^{(1)}, \dots, a_{ij}^{(n)}\big) \in K^n$, where $a_{ij} = \sum_{l=1}^{n} a_{ij}^{(l)} b_l$, for $i \in \{1, \dots, k\}$, $j \in \{1, \dots, m\}$. In result, we get a *K-generator matrix* $B \in \mathrm{M}_{k\times nm}(K)$ of C:

$$B = \left(\begin{array}{ccc|ccc|c|ccc} a_{11}^{(1)} & \dots & a_{11}^{(n)} & a_{12}^{(1)} & \dots & a_{12}^{(n)} & \dots & a_{1m}^{(1)} & \dots & a_{1m}^{(n)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ a_{k1}^{(1)} & \dots & a_{k1}^{(n)} & a_{k2}^{(1)} & \dots & a_{k2}^{(n)} & \dots & a_{km}^{(1)} & \dots & a_{km}^{(n)} \end{array}\right).$$

The K-generator matrix B can be observed as the concatenation of m smaller matrices, $B = (B_1|B_2|\dots|B_m)$, where B_i is the ith block of B, for $i \in \{1, \dots, m\}$.

We are interested in the subspace $V_i \subseteq K^k$, where $i \in \{1, \dots, m\}$, that is defined as a K-span of the columns of B_i. With the K-generator matrix B associate a *tuple of the subspaces* $V_1, \dots, V_m \subseteq K^k$.

Let $f : C \to L^m$ be a K-linear injective map. Let $A' \in \mathrm{M}_{k\times m}(L)$ be a matrix with the rows $f(x_1), \dots, f(x_m)$, where $x_1, \dots, x_k$ are the rows of A. The matrix A' is a generator matrix of $f(C)$. Define the K-generator matrix B' in the same way as we defined B. Let $V_1, \dots, V_m \subseteq K^k$ be the tuple of subspaces that correspond to B and let $U_1, \dots, U_m \subseteq K^k$ be the tuple of subspaces that correspond to B'.

Proposition 6 *The map f extends to a general monomial map if and only if there exists a permutation $\pi \in S_m$ such that $U_i = V_{\pi(i)}$, for $i \in \{1, \dots, m\}$.*

Proof If the map f extends to a general monomial transformation $h : L^m \to L^m$ (with the permutation $\pi \in S_n$), then the tuples $U_1, \dots, U_m$ and $V_{\pi(1)}, \dots, V_{\pi(m)}$ are equal. In the other direction, let B and B' be the K-generator matrices that correspond to the tuples of subspaces $V_1, \dots, V_m$ and $U_1, \dots, U_m$. Then there exist a permutation $\pi \in S_m$ and invertible matrices $G_i \in \mathrm{M}_{n\times n}(K)$, such that $B = (B_{\pi(1)}G_1|\dots|B_{\pi(m)}G_m) = (B'_1|\dots|B'_m) = B'$. This correspond to a general monomial transformation $h : L^m \to L^m$, such that $h = f$ on the code generated by B. □

We use the ideas presented in the proof of the MacWilliams Extension Theorem by Ward and Wood (see [8]) to get a description of K-linear isometries of K-linear codes in L^m. For a finite abelian group G let $\hat{G}$ be the set of all homomorphisms from $(G, +)$ to $(\mathbb{C} \setminus \{0\}, \cdot)$. There is defined a product of two homomorphisms: for $g, h \in \hat{G}$ define $(gh)(x) = g(x)h(x)$ for all $x \in G$. The set $\hat{G}$ with the defined product form a group and is called a *group of characters*.

Let X be a set and let A be a subset of X. An indicator function is a map $\mathbb{1}_A : X \to \{0, 1\}$, such that $\mathbb{1}_A(x) = 1$ if $x \in A$ and $\mathbb{1}_A(x) = 0$ – otherwise.

Proposition 7 *Let C be a K-linear code in L^m and $f : C \to L^m$ be a K-linear map. The map f is an isometry if and only if the following equality holds:*

$$\sum_{i=1}^{m} \frac{1}{|V_i|} \mathbb{1}_{V_i} = \sum_{i=1}^{m} \frac{1}{|U_i|} \mathbb{1}_{U_i} \ . \tag{1}$$

Proof Let U be a k-dimensional vector space over K with some fixed basis, where $k = \dim_K C$. Consider two K-linear maps $\lambda, \mu : U \to L^m$, defined as $\lambda(u) = u^T A$ and $\mu(u) = u^T A'$, for $u \in U$, where A is a generator matrix of C and A' is the corresponding generator matrix of $f(C)$. It appears that $\mu(u) = f(\lambda(u))$ for all $u \in U$, $\lambda(U) = C$ and $\mu(U) = f(C)$.

For the weight function $\mathrm{wt} : L \to \{0, 1\}$, which maps 0 to 0 and other elements to 1, the following holds: for all $a \in L : \frac{1}{|L|} \sum_{\chi \in \hat{L}} \chi(a) = 1 - \mathrm{wt}(a)$ (see [6, p. 143]). Using weight representation in terms of character sums, we have that for all $u \in U$:

$$m - \mathrm{wt}\big(\lambda(u)\big) = \frac{1}{|L|} \sum_{i=1}^{m} \sum_{\chi \in \hat{L}} \chi\big(\lambda_i(u)\big) \ . \tag{2}$$

For a K-linear map $\sigma : U \to L$ define a map $\hat{\sigma} : \hat{L} \to \hat{U}, \chi \mapsto \sigma \circ \chi$. Transforming the sum in Eq. (2), we get:

$$\sum_{i=1}^{m} \sum_{\chi \in \hat{L}} \hat{\lambda}_i(\chi) = |L| \sum_{\pi \in \hat{U}} \left(\sum_{i=1}^{m} \frac{1}{|\hat{\lambda}_i(\hat{L})|} \mathbb{1}_{\hat{\lambda}_i(\hat{L})}(\pi) \right) \pi \ .$$

By the definition, the map f is an isometry if for all $x \in C$, $\mathrm{wt}(x) = \mathrm{wt}(f(x))$, or the same, for all $u \in U$, $\mathrm{wt}(\lambda(u)) = \mathrm{wt}(\mu(u))$. Consequently, f is an isometry if and only if the following equality holds:

$$\sum_{\pi \in \hat{U}} \left(\sum_{i=1}^{m} \frac{1}{|\hat{\lambda}_i(\hat{L})|} \mathbb{1}_{\hat{\lambda}_i(\hat{L})}(\pi) \right) \pi = \sum_{\pi \in \hat{U}} \left(\sum_{i=1}^{m} \frac{1}{|\hat{\mu}_i(\hat{L})|} \mathbb{1}_{\hat{\mu}_i(\hat{L})}(\pi) \right) \pi \ .$$

Since different characters in $\hat{U}$ are linearly independent, the coefficients in the equation are equal for each $\pi \in \hat{U}$. This is equivalent to:

$$\sum_{i=1}^{m} \frac{1}{|\hat{\lambda}_i(\hat{L})|} \mathbb{1}_{\hat{\lambda}_i(\hat{L})} = \sum_{i=1}^{m} \frac{1}{|\hat{\mu}_i(\hat{L})|} \mathbb{1}_{\hat{\mu}_i(\hat{L})} \ .$$

It can be proved that this equality is equivalent to Eq. (1). □

To illustrate Proposition 7 we consider the following example observed in [9].

Example 8 Let C be a $\mathbb{F}_2$-linear code in $\mathbb{F}_4^3$, generated by three vectors: $C = \langle(1,1,0),(\omega,\omega,0),(1,0,1)\rangle_{\mathbb{F}_2}$, where $\mathbb{F}_4 = \{0,1,\omega,\omega^2\}$ with $\omega + 1 = \omega^2$. Define an $\mathbb{F}_2$-linear map $f : C \to \mathbb{F}_4^3$ on the generators in the following way: $f\big((1,1,0)\big) = (1,1,0)$, $f\big((\omega,\omega,0)\big) = (1,0,1)$ and $f\big((1,0,1)\big) = (\omega,\omega,0)$. Obviously, $f(C) = C$.

Consider the isomorphism of the $\mathbb{F}_2$-linear vector spaces $\mathbb{F}_4 \to \mathbb{F}_2^2$, $1 \mapsto (1,0)$ and $\omega \mapsto (0,1)$. We use the following generator matrix A and the corresponding $\mathbb{F}_2$-generator matrix B of the code C:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ \omega & \omega & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad B = \left(\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right).$$

Since f fixes the first generator vector of C and permute second and third, it is easy to construct the corresponding generator matrix A' and the $\mathbb{F}_2$-generator matrix B' of the code $f(C)$:

$$A' = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ \omega & \omega & 0 \end{pmatrix}, \quad B' = \left(\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right).$$

Now we calculate the tuples of subspaces $V_1, V_2, V_3 \subseteq \mathbb{F}_2^3$ and $U_1, U_2, U_3 \subseteq \mathbb{F}_2^3$. The subspaces are: $V_1 = \langle(1,0,1),(0,1,0)\rangle_{\mathbb{F}_2}$, $V_2 = \langle(1,0,0),(0,1,0)\rangle_{\mathbb{F}_2}$ and $V_3 = \langle(0,0,1)\rangle_{\mathbb{F}_2}$. In the same way, $U_1 = \langle(1,1,0),(0,0,1)\rangle_{\mathbb{F}_2}$, $U_2 = \langle(1,0,0),(0,0,1)\rangle_{\mathbb{F}_2}$ and $U_3 = \langle(0,1,0)\rangle_{\mathbb{F}_2}$. Equation (1) after multiplication by 4 from both sides becomes:

$$\mathbb{1}_{V_1} + \mathbb{1}_{V_2} + 2\mathbb{1}_{V_3} = \mathbb{1}_{U_1} + \mathbb{1}_{U_2} + 2\mathbb{1}_{U_3}\ .$$

One can verify that for the defined subspaces V_1, V_2, V_3 and U_1, U_2, U_3 the equality holds and thus, by Proposition 7, the map $f : C \to \mathbb{F}_4^3$ is an $\mathbb{F}_2$-linear isometry. Moreover, by Proposition 6, since the tuples of subspaces V_1, V_2, V_3 and U_1, U_2, U_3 do not coincide up to the order of terms, the isometry f is unextendible.

Regarding Eq. (1), we have the following statement.

Proposition 9 *Let W be a finite space over K and $U_1, \dots, U_r, V_1, \dots, V_s \subseteq W$ be different subspaces of W. Assume that $a_1, \dots, a_r, b_1, \dots, b_s > 0$ and*

$$\sum_{i=1}^{r} a_i \mathbb{1}_{U_i} = \sum_{i=1}^{s} b_i \mathbb{1}_{V_i}\ . \tag{3}$$

Then $\max\{r, s\}$ *is greater than the cardinality of K.*

Proof Among the subspaces $V_1, \dots, V_s, U_1, \dots, U_r$ choose one that is maximal by inclusion. It is either V_i for some $i \in \{1, \dots, s\}$, or U_j for some $j \in \{1, \dots, t\}$. In the first case $\dim_K V_i > 1$ and $V_i = \bigcup_{j=1}^{r}(V_i \cap U_j)$, where for all $j \in \{1, \dots, r\}$: $V_i \cap U_j \neq V_i$. Such coverings are discussed in [2] and there it was proved that $r > |K|$. Similarly, in the second case $s > |K|$. □

Theorem 10 *Let L be a finite field and let K be a proper subfield of L. Let $m \leq |K|$ and let C be a K-linear code in L^m. Each K-linear isometry of C extends to a K-linear isometry of the whole space.*

Proof From Proposition 4, to prove the theorem, it is enough to show that: if there exists such K-linear code $C \subseteq L^m$ and K-linear isometry $f : C \to L^m$ that does not extend to general monomial map, then $m > |K|$. Since f is an isometry, Proposition 7 implies that Eq. (1) holds. Let $V_1, \dots, V_m$ be a tuple of subspaces of C and $U_1, \dots, U_m$ be the corresponding tuple of subspaces of $f(C)$. There is an alternative: the tuples of the subspaces $V_1, \dots, V_m$ and $U_1, \dots, U_m$ or coincide up to a permutation of the elements, or not. In the first case, by Propositions 4 and 6, f extends to an isometry of the whole space L^m. In the second case in Eq. (1) group the equal terms from each side and eliminate the equal terms from the different sides. After canceling and elimination there exists $i \in \{1, \dots, m\}$ such that for all $j \in \{1, \dots, m\}$: $V_i \neq U_j$. So, we obtain an equation in form of Eq. (3), where conditions of Proposition 9 are satisfied. Therefore $m \geq \max\{r, s\} > |K|$. □

References

1. Avgustinovich, S.V., Solov'eva, F.I.: To the metrical rigidity of binary codes. Probl. Inf. Trans. **39**(2), 178–183 (2003). doi:10.1023/A:1025148221096
2. Clark, P.L.: Covering numbers in linear algebra. Am. Math. Mon. **119**(1), 65–67 (2012). doi:10.4169/amer.math.monthly.119.01.065
3. Constantinescu, I., Heise, W.: On the concept of code-isomorphy. J. Geom. **57**(1–2), 63–69 (1996). doi:10.1007/BF01229251
4. Gruska, J.: Quantum Computing. McGraw-Hill, London (1999)
5. Kovalevskaya, D.I.: On metric rigidity for some classes of codes. Probl. Inf. Trans. **47**(1), 15–27 (2011). doi:10.1134/S0032946011010029
6. MacWilliams, F., Sloane, N.: The Theory of Error-Correcting Codes: Vol.: 1. North-Holland Mathematical Library. North-Holland, Amsterdam/London/New York (1977)
7. Solov'eva, F., Honold, T., Avgustinovich, S., Heise, W.: On the extendability of code isometries. J. Geom. **61**(1–2), 2–16 (1998). doi:10.1007/BF01237489
8. Ward, H.N., Wood, J.A.: Characters and the equivalence of codes. J. Comb. Theory Ser. A **73**(2), 348–352 (1996). doi:10.1016/S0097-3165(96)80011-2
9. Wood, J.A.: Exotic automorphisms of additive codes. AMS Sectional Meeting, Louisville (2013)

The Extension Theorem with Respect to Symmetrized Weight Compositions

Noha ElGarem, Nefertiti Megahed, and Jay A. Wood

Abstract We will say that an alphabet A satisfies the extension property with respect to a weight w if every linear isomorphism between two linear codes in A^n that preserves w extends to a monomial transformation of A^n. In the 1960s MacWilliams proved that finite fields have the extension property with respect to Hamming weight. It is known that a module A has the extension property with respect to Hamming weight or a homogeneous weight if and only if A is pseudo-injective and embeds into $\hat{R}$. The main theorem presented in this paper gives a sufficient condition for an alphabet to have the extension property with respect to symmetrized weight compositions. It has already been proven that a Frobenius bimodule has the extension property with respect to symmetrized weight compositions. This result follows from the main theorem.

Keywords Linear codes over finite modules • Extension theorem • Symmetrized weight composition

1 Introduction

In the 1960s Florence Jessie MacWilliams proved in her doctoral dissertation [13] that two linear codes over a finite field are isometric if and only if they are monomially equivalent. Two linear codes of the same length are said to be isometric if there is a linear injective map from one to the other that preserves Hamming weight. In other words, two linear codes $C_1, C_2 \subset \mathbb{F}_q^n$ are isometric if there is a linear injective map $f : C_1 \to C_2$ such that $wt(f(c)) = wt(c)$ for every $c \in C_1$, where wt denotes the Hamming weight on $\mathbb{F}_q$. The codes are said to be monomially equivalent if there is a monomial transformation, or an $n \times n$ monomial matrix M, such that $C_2 = C_1 M$. Because monomial equivalence implies the existence of an

N. ElGarem (✉) • N. Megahed
Cairo University, Giza 12613, Egypt
e-mail: n_garem@aucegypt.edu; nefertiti@sci.cu.edu.eg

J.A. Wood
Western Michigan University, 1903 W Michigan Ave, Kalamazoo, MI 49008, USA
e-mail: jay.wood@wmich.edu

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_18

isometry, what MacWilliams proved for codes over finite fields is that any isometry can be extended to a monomial transformation. MacWilliams also proved a semi-linear version of this extension theorem. In 1996, a character theoretic proof of MacWilliams' result appeared in [14].

The publication of [12] rekindled the interest of researchers in codes over finite rings and the question arose, which types of rings satisfy MacWilliams' Extension Theorem? In [17], the character theoretic proof of [14] was generalized to prove that finite Frobenius rings satisfy the Extension Theorem with respect to Hamming weight. In [4] Dinh and López-Permouth proved some partial converses and provided a strategy to prove the full converse. The strategy led to a proof of the full converse in [18] for linear codes over finite rings with the Hamming weight.

In 1997, Constantinescu and Heise introduced a new weight on finite rings [2], namely, the homogeneous weight. The authors of [3] used combinatorial methods to prove the extension theorem for homogeneous weights over the ring $\mathbb{Z}_m$. Following their lead, Greferath and Schmidt proved that every Hamming weight isometry is a homogeneous weight isometry and vice versa, thereby translating all results on the Extension Theorem for Hamming weight to homogeneous weights and vice versa [11]. Greferath, Nechaev, and Wisbauer proved the Extension Theorem for Hamming and homogeneous weights over Frobenius bimodules in [10].

More general weight functions were considered next, specifically bi-invariant weight functions. A weight w on a ring R is said to be bi-invariant if $w(ux) = w(x) = w(xu)$ for every x in R and every unit u in R. The extension theorem was proved for bi-invariant weights in the case of finite chain rings in [6], in the case of $\mathbb{Z}_m$ in [7], in the case of finite direct products of finite chain rings in [9], in the case of matrix rings over finite fields in [19], and in the case of principal ideal rings, necessary and sufficient conditions were found for bi-invariant weights to satisfy the extension theorem in [8].

The present paper considers the Extension Theorem with respect to another type of weight, namely the symmetrized weight composition over certain module alphabets. The Extension Theorem for symmetrized weight compositions was proved for linear codes over finite fields in [5], over finite Frobenius rings in [15], and over Frobenius bimodule alphabets in [19]. In [1], Barra and Gluesing-Luerssen greatly simplified the proof in [15], and we apply their ideas to the case of certain module alphabets.

The following is a summary of the contents of this paper. Section 2 provides some basic definitions, as well as the Extension Theorems known for module alphabets equipped with Hamming weight. In Sect. 3, we apply some of the ideas of [1] to module alphabets. The main result of this paper (Theorem 13) states that a sufficient condition for an R-module A to satisfy the Extension Theorem with respect to symmetrized weight compositions is that A can be embedded into ${}_R\hat{R}$. This condition implies that a Frobenius bimodule satisfies the Extension Theorem with respect to symmetrized weight compositions.

The Extension Theorem for symmetrized weight compositions over finite Frobenius rings has been used in [15] and [16] to prove extension theorems for more general weight functions. We anticipate proving similar results in future work.

2 Background

Throughout this paper, let R be a finite ring with unity and let A be a finite left R-module; A will serve as the alphabet for linear codes. We will adopt the following convention: when dealing with maps on left R-modules, the input to the map will be written on the left. In other words, if we have a left R-module A and a map f on A, then for $a \in A$, we write af for $f(a)$.

Definition 1 A *linear code* of length n over the alphabet A is a left R-submodule $C \subset A^n$.

Definition 2 A *monomial transformation* of A^n is an R-linear automorphism T of A^n of the form

$$(a_1, \ldots, a_n)T = (a_{\sigma(1)}\tau_1, \ldots, a_{\sigma(n)}\tau_n),$$

where $(a_1, \ldots, a_n) \in A^n$, σ is a permutation of $\{1, 2, \ldots, n\}$ and $\tau_1, \ldots, \tau_n \in \mathrm{Aut}(A)$, the group of automorphisms of the left R-module A. If $\tau_1, \ldots, \tau_n$ all belong to some subgroup G of $\mathrm{Aut}(A)$, we say that T is a G-monomial transformation of A^n.

A weight on an alphabet A is defined to be a rational-valued function $w : A \to \mathbb{Q}$ with $w(0) = 0$. We define the extension property as follows.

Definition 3 Let A be an R-module. We say that the alphabet A *satisfies the extension property with respect to the Hamming weight* if every R-linear isomorphism between two R-linear codes in A^n that preserves Hamming weight extends to a monomial transformation of A^n.

The class of Frobenius bimodules stood out in coding theory as all Frobenius bimodules satisfy the extension property with respect to Hamming weight [10]. A Frobenius bimodule is defined as follows.

Definition 4 Let A be a bimodule over the ring R. We say that A is a *Frobenius bimodule* if ${}_RA \cong_R \hat{R}$ and $A_R \cong \hat{R}_R$, where $\hat{R} = \mathrm{Hom}_{\mathbb{Z}}(R, \mathbb{C}^{\times})$ is the character module of R.

The following theorem was proved in [17] and [18].

Theorem 5 *Let R be a finite ring and $A = R$. Then R satisfies the extension property with respect to Hamming weight if and only if R is Frobenius.*

Necessary and sufficient conditions for a module alphabet A to satisfy the extension property with respect to Hamming weight were established in [19]. The first condition is that the R-module alphabet A is pseudo-injective, in other words for every R-submodule B of A and every injective R-linear mapping $f : B \to A$, the mapping f extends to an R-linear mapping $\tilde{F} : A \to A$. The second condition that arises is that A have a cyclic socle. The socle of an R-module A is defined to be the sum of all its simple R-submodules. We note that a left R-module A has a cyclic socle if and only if A embeds into $\hat{R}$ ([19], Proposition 5.3).

Theorem 6 *Let R be a finite ring and A a finite R-module. Then A satisfies the extension property with respect to Hamming weight if and only if A is pseudo-injective and has a cyclic socle.*

3 The Extension Theorem for Symmetrized Weight Compositions

Given a weight w on an alphabet A, define the symmetry group of w as the set of all automorphisms of A that preserve w. Denote the symmetry group by

$$Sym(w) := \{\tau \in \mathrm{Aut}(A) | w(a\tau) = w(a) \text{ for every } a \in A\}.$$

Then for a general weight w, the extension property is defined as follows.

Definition 7 Let A be an alphabet and w a weight on A. Then A has the *extension property with respect to* w if for any two linear codes $C_1, C_2 \subset A^n$, and R-linear isomorphism $f : C_1 \to C_2$ that preserves w, f is extendable to a $Sym(w)$-monomial transformation of A^n.

The symmetry group of a weight w on an alphabet A acts on A on the right so that the orbit of an element a in A is $orb(a) = \{a\tau | \tau \in Sym(w)\}$. The symmetrized weight composition counts the number of entries of $x = (x_1, \ldots, x_n) \in A^n$ that belong to any given orbit of this action.

We now give the formal definition of the symmetrized weight composition.

Definition 8 Let G be a subgroup of the automorphism group of a finite R-module A. Define $\sim$ on A by $a \sim b$ if and only if $a = b\tau$ for some $\tau \in G$. Let A/G denote the orbit space of this action. The *symmetrized weight composition* is a function $\mathrm{swc} : A^n \times A/G \to \mathbb{Q}$ defined by,

$$\mathrm{swc}(x, a) = \mathrm{swc}_a(x) = |\{i : x_i \sim a\}|,$$

where $x = (x_1, \ldots, x_n) \in A^n$ and $a \in A/G$.

Note that if $a, b \in A$ are in the same orbit, then $\mathrm{swc}_a = \mathrm{swc}_b$ and so the symmetrized weight composition is well-defined.

Definition 9 The alphabet A has the *extension property with respect to* swc if for any two linear codes $C_1, C_2 \subset A^n$, and R-linear isomorphism $f : C_1 \to C_2$ that preserves swc, f is extendable to a G-monomial transformation of A^n.

We wish to find conditions on the module alphabet A equipped with swc to satisfy the extension property analogous to those found in Theorem 6 for Hamming weight. Theorem 13 gives a sufficient condition and its proof uses some of the ideas found in [1].

In order to prove the main theorem, we need a few results concerning admissible characters. More details can be found in [19] and [20] (where admissible characters were called generating characters).

Definition 10 Let A be a finite left R-module. We say a character $\rho \in \hat{A}$ is *(left) admissible* if ker ρ contains no nonzero left R-submodules. There is a corresponding notion of right admissible characters for right R-modules.

The proof of the main theorem requires a proposition from [20].

Proposition 11 ([20], Proposition 12) *Let A be a finite left R-module. Then A has an admissible character if and only if A can be embedded in ${}_R\hat{R}$.*

The reader will verify that Frobenius bimodules have admissible characters.

The condition that will appear in the main theorem (Theorem 13) is that the R-module alphabet A can be embedded into $\hat{R}$. As mentioned earlier, this condition is equivalent to the condition that the alphabet A has a cyclic socle due to the following result (Proposition 5.3 in [19]).

Proposition 12 *Let R be a ring and A a left R-module. Then $soc(A)$ is cyclic if and only if A can be embedded into ${}_R\hat{R}$.*

We now state and prove the main theorem.

Theorem 13 *Let A be a finite left R-module equipped with a symmetrized weight composition. If A can be embedded into $\hat{R}$, then A has the extension property with respect to the symmetrized weight composition. In particular, this theorem applies to Frobenius bimodules.*

Proof Suppose $C_1, C_2 \subset A^n$ are two R-linear codes, and $f : C_1 \to C_2$ is an R-linear isomorphism that preserves swc. Let M be the module underlying the two codes C_1, C_2 with $\lambda : M \to A^n$ and $\nu : M \to A^n$, the inclusion maps of C_1 and C_2 into A^n, respectively, and $\nu = \lambda \circ f$ (recall that inputs to functions are written on the left). Suppose $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $\nu = (\nu_1, \ldots, \nu_n)$, where $\lambda_i, \nu_i \in \mathrm{Hom}_R(M, A)$. Since f preserves swc, then $\mathrm{swc}_a(x\lambda) = \mathrm{swc}_a(x\nu)$ for every $a \in A/G$ and every $x \in M$. Following [1], if we fix $x \in M$ then there exists a permutation σ_x of $\{1, \ldots, n\}$ and elements $\phi_{j,x} \in G$ such that $x\lambda_j = x\nu_{\sigma_x(j)}\phi_{j,x}$ for each $j \in \{1, \ldots, n\}$. Let $\psi \in G$, noting that $G \subset \mathrm{Aut}(A)$, then for all j,

$$x\lambda_j\psi = x\nu_{\sigma_x(j)}\phi_{j,x}\psi. \tag{1}$$

Since A can be embedded into $\hat{R}$, it follows from Proposition 11 that A has an admissible character $\rho : A \to \mathbb{C}^\times$. Compose ρ with both sides of Eq. (1) to get

$$(x\lambda_j\psi)\rho = (x\nu_{\sigma_x(j)}\phi_{j,x}\psi)\rho.$$

We can now take the summation of the previous equation over all $j \in \{1, \ldots, n\}$

and all $\psi \in G$ yielding the following,

$$\begin{aligned}\sum_{j=1}^{n}\sum_{\psi\in G}(x\lambda_j\psi)\rho &= \sum_{j=1}^{n}\sum_{\psi\in G}(x\nu_{\sigma_x(j)}\phi_{j,x}\psi)\rho\\ &= \sum_{k=1}^{n}\sum_{\tau\in G}(x\nu_k\tau)\rho.\end{aligned}$$

Since the above equation is true for every $x \in M$, we have the following equation of characters of M,

$$\sum_{j=1}^{n}\sum_{\psi\in G}(\lambda_j\psi)\rho = \sum_{k=1}^{n}\sum_{\tau\in G}(\nu_k\tau)\rho. \tag{2}$$

We can now make use of the fact that characters of M are linearly independent, when considered as complex-valued functions on M. On the left hand side of Eq. (2), fix $j = 1$ and $\psi = id_A$. By the independence of characters it follows that there exists $k_1 \in \{1,\ldots,n\}$ and $\tau_1 \in G$ such that $\lambda_1 \circ \rho = \nu_{k_1}\tau_1 \circ \rho$. Then $im(\lambda_1 - \nu_{k_1}\tau_1) \subset \ker \varrho$. But ρ is an admissible character of A and therefore contains no non-zero submodules. It follows that $im(\lambda_1 - \nu_{k_1}\tau_1) = 0$ and so $\lambda_1 = \nu_{k_1}\tau_1$. Re-indexing (letting $\phi = \tau_1\psi$), shows that

$$\sum_{\psi\in G}(\lambda_1\psi)\rho = \sum_{\psi\in G}(\nu_{k_1}\tau_1\psi)\rho = \sum_{\phi\in G}(\nu_{k_1}\phi)\rho.$$

This allows us to reduce the outer summation in Eq. (2) by one. Proceeding by induction, we find a permutation σ and automorphisms $\tau_1, \ldots, \tau_n \in G$ with $\lambda_i = \nu_{\sigma(i)}\tau_i$. □

A natural question to ask is whether the converse of Theorem 13 is true. In other words, if the extension property holds for an R-module alphabet A equipped with a symmetrized weight composition, must A have a cyclic socle? Or equivalently must there be an embedding of A into $\hat{R}$? This remains an open question.

References

1. Barra, A., Gluesing-Luerssen, H.: MacWilliams extension theorems and the local-global property for codes over rings (2013). arXiv:1307.7159
2. Constantinescu, I., Heise, W.: A metric for codes over residue class rings of integers. Probl. Inf. Trans. **33**(3), 208–213 (1997)
3. Constantinescu, I., Heise, W., Honold, T.: Monomial extensions of isometries between codes over $\mathbb{Z}_m$. In: Proceedings of the Fifth International Workshop on Algebraic and Combinatorial Coding Theory, Sozopol, pp. 98–104. Unicorn, Shumen (1996)

4. Dinh, H.Q., López-Permouth, S.R.: On the equivalence of codes over rings and modules. Finite Fields Appl. **10**(4), 615–625 (2004). doi:10.1016/j.ffa.2004.01.001
5. Goldberg, D.: A generalized weight for linear codes and a Witt-MacWilliams theorem. J. Comb. Theory Ser. A **29**(3), 363–367 (1980)
6. Greferath, M., Honold, T.: On weights allowing for MacWilliams equivalence theorem. In: Proceedings of the Fourth International Workshop on Optimal Codes and Related Topics, Pamporovo, pp. 182–192 (2005)
7. Greferath, M., Honold, T.: Monomial extensions of isometries of linear codes ii: invariant weight functions on $\mathbb{Z}_m$. In: Proceedings of the Tenth International Workshop on Algebraic and Combinatorial Coding Theory, Zvenigorod, pp. 106–111 (2006)
8. Greferath, M., Honold, T., McFadden, C., Wood, J.A., Zumbrägel, J.: MacWilliams' extension theorem for bi-invariant weights over finite principal ideal rings. J. Comb. Theory Ser. A **125**, 177–193 (2014)
9. Greferath, M., McFadden, C., Zumbrägel, J.: Characteristics of invariant weights related to code equivalence over rings. Des. Codes Cryptogr. **66**(1–3), 145–156 (2013)
10. Greferath, M., Nechaev, R., Wisbauer, R.: Finite quasi-Frobenius modules and linear codes. J. Algebra Appl. **3**(3), 247–272 (2004)
11. Greferath, M., Schmidt, S.E.: Finite ring combinatorics and MacWilliams equivalence theorem. J. Comb. Theory Ser. A **92**(1), 17–28 (2000). doi:10.1006/jcta.1999.3033
12. Hammons, A., Kumar, P., Calderbank, A., Sloane, N., Solé, P.: The $\mathbb{Z}_4$ linearity of Kerdock, Preparata, Goethals, and related codes. IEEE Trans. Inf. Theory **40**(2), 301–319 (1994). doi:10.1109/18.312154
13. MacWilliams, F.J.: Combinatorial properties of elementary abelian groups. Ph.D. thesis, Harvard University, Cambridge (1962)
14. Ward, H.N., Wood, J.A.: Characters and the equivalence of codes. J. Comb. Theory Ser. A **73**(2), 348–352 (1996). doi:10.1016/S0097-3165(96)80011-2
15. Wood, J.A.: Extension theorems for linear codes over finite rings. In: Mora, T., Mattson, H. (eds.) Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Lecture Notes in Computer Science, vol. 1255, pp. 329–340. Springer, Berlin/Heidelberg (1997). doi:10.1007/3-540-63163-1_26
16. Wood, J.A.: Weight functions and the extension theorem for linear codes over finite rings. In: Proceedings of the Fourth International Conference on Finite Fields: Theory, Applications and Algorithms, University of Waterloo, Waterloo (1997)
17. Wood, J.A.: Duality for modules over finite rings and applications to coding theory. Am. J. Math. **121**, 555–575 (1999)
18. Wood, J.A.: Code equivalence characterizes finite Frobenius rings. Proc. Am. Math. Soc. **136**, 699–706 (2008). doi:10.1090/S0002-9939-07-09164-2
19. Wood, J.A.: Foundations of linear codes defined over finite modules: the extension theorem and MacWilliams identities. In: Solé, P. (ed.) CIMPA Summer School, Ankara, 18–29 Aug 2008. Series on Coding Theory and Cryptology, vol. 6, pp. 124–190. World Scientific (2009)
20. Wood, J.A.: Applications of finite Frobenius rings to the foundations of algebraic coding theory. In: Iyama, O. (ed.) 44th Symposium on Ring Theory and Representation Theory, Okayama University, Nagoya, pp. 235–245 (2012)

Minimal Realizations of Syndrome Formers of a Special Class of 2D Codes

Ettore Fornasini, Telma Pinho, Raquel Pinto, and Paula Rocha

Abstract In this paper we consider a special class of 2D convolutional codes (composition codes) with encoders $G(d_1, d_2)$ that can be decomposed as the product of two 1D encoders, i.e., $G(d_1, d_2) = G_2(d_2)G_1(d_1)$. In case that $G_1(d_1)$ and $G_2(d_2)$ are prime we provide constructions of syndrome formers of the code, directly from $G_1(d_1)$ and $G_2(d_2)$. Moreover we investigate the minimality of 2D state-space realization by means of a separable Roesser model of syndrome formers of composition codes, where $G_2(d_2)$ is a quasi-systematic encoder.

Keywords Encoders and syndrome forms • 2D composition codes • 2D state-space models

1 Introduction and Preliminary Concepts

Minimal state-space realization of convolutional codes play an important role in efficient code generation and verification. This question has been widely investigated in the literature for 1D codes [3, 6], however it is still open for the 2D case. Preliminary results concerning 2D encoder and code realizations have been presented in [10]. In this paper we study the syndrome former realization problem for a special class of 2D codes.

E. Fornasini
Department of Information Engineering, University of Padua, Padua, Italy
e-mail: ettore.fornasini@unipd.it

T. Pinho • R. Pinto (✉)
Center for Research and Development in Mathematics and Applications (CIDMA), Department of Mathematics, University of Aveiro, Aveiro, Portugal
e-mail: telma.pinho@ua.pt; raquel@ua.pt

P. Rocha
Research Center for Systems and Technologies, SYSTEC, Faculty of Engineering, University of Porto, Portugal
e-mail: mprocha@fe.up.pt

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_19

We consider 2D convolutional codes constituted by sequences indexed by $\mathbb{Z}^2$ and taking values in $\mathbb{F}^n$, where $\mathbb{F}$ is a field. Such sequences $\{w(i,j)\}_{(i,j)\in\mathbb{Z}^2}$ can be represented by bilateral formal power series

$$\hat{w}(d_1,d_2) = \sum_{(i,j)\in\mathbb{Z}^2} w(i,j)d_1^i d_2^j.$$

For $n \in \mathbb{N}$, the set of 2D bilateral formal power series over $\mathbb{F}^n$ is denoted by $\mathscr{F}_{2D}^n$. This set is a module over the ring $\mathbb{F}[d_1,d_2]$ of 2D polynomials over $\mathbb{F}$. The set of matrices of size $n \times k$ with elements in $\mathbb{F}[d_1,d_2]$ will be denoted by $\mathbb{F}^{n\times k}[d_1,d_2]$.

Given a subset $\mathscr{C}$ of sequences indexed by $\mathbb{Z}^2$, taking values in $\mathbb{F}^n$, we denote by $\hat{\mathscr{C}}$ the subset of $\mathscr{F}_{2D}^n$ defined by $\hat{\mathscr{C}} = \{\hat{w} \mid w \in \mathscr{C}\}$.

Definition 1 A 2D convolutional code is a subset $\mathscr{C}$ of sequences indexed by $\mathbb{Z}^2$ such that $\hat{\mathscr{C}}$ is a submodule of $\mathscr{F}_{2D}^n$ which coincides with the image of $\mathscr{F}_{2D}^k$ (for some $k \in \mathbb{N}$) by a polynomial matrix $G(d_1,d_2)$, i.e.,

$$\hat{\mathscr{C}} = \operatorname{im} G(d_1,d_2) = \{\hat{w}(d_1,d_2) \mid \hat{w}(d_1,d_2) = G(d_1,d_2)\hat{u}(d_1,d_2),\ \hat{u}(d_1,d_2) \in \mathscr{F}_{2D}^k\}.$$

It follows, as a consequence of [Theorem 2.2, [7]], that a 2D convolutional code can always be given as the image of a full column rank polynomial matrix $G(d_1,d_2) \in \mathbb{F}^{n\times k}[d_1,d_2]$. Such polynomial matrix is called an *encoder* of $\mathscr{C}$. A code with encoders of size $n \times k$ is said to have rate k/n.

A 2D convolutional code $\mathscr{C}$ of rate k/n can also be represented as the kernel of a $(n-k) \times n$ left-factor prime polynomial matrix (i.e. a matrix without left nonunimodular factors), as follows from [Theorem 1, [12]].

Definition 2 Let $\mathscr{C}$ be a 2D convolutional code of rate k/n. A left-factor prime matrix $H(d_1,d_2) \in \mathbb{F}^{(n-k)\times n}[d_1,d_2]$ such that

$$\hat{\mathscr{C}} = \ker H(d_1,d_2),$$

is called a syndrome former of $\mathscr{C}$.

Note that w is in $\mathscr{C}$ if and only if $H(d_1,d_2)\hat{w} = 0$.

Remark 3 This means that whereas codewords are output sequences of an encoder, they constitute the *output-nulling inputs* of a syndrome former of the code.

Given an encoder $G(d_1,d_2)$ of $\mathscr{C}$, a syndrome former of $\mathscr{C}$ can be obtained by constructing a $(n-k) \times n$ left-factor prime matrix $H(d_1,d_2)$ such that $H(d_1,d_2)G(d_1,d_2) = 0$. Moreover all syndrome formers of $\mathscr{C}$ are of the form $U(d_1,d_2)H(d_1,d_2)$, where $U(d_1,d_2) \in \mathbb{F}^{(n-k)\times(n-k)}[d_1,d_2]$ is unimodular.

2 Composition Codes and Their Syndrome Formers

In this section we consider a particular class of 2D convolutional codes generated by 2D polynomial encoders that are obtained from the composition of two 1D polynomial encoders. Such encoders/codes will be called *composition encoders/codes*. Our goal is to characterize the syndrome formers of such codes. The formal definition of composition encoders is as follows.

Definition 4 An encoder $G(d_1, d_2) \in \mathbb{F}^{n\times k}[d_1, d_2]$ such that

$$G(d_1, d_2) = G_2(d_2)G_1(d_1), \tag{1}$$

where $G_1(d_1) \in \mathbb{F}^{p\times k}[d_1]$ and $G_2(d_2) \in \mathbb{F}^{n\times p}[d_2]$ are 1D encoders, is said to be a composition encoder.

Note that the requirement that $G_i(d_i)$, for $i = 1, 2$, is a 1D encoder implies the condition that $G_i(d_i)$ is a full column rank matrix. Moreover this requirement clearly implies that $G_2(d_2)G_1(d_1)$ has full column rank, hence the composition $G_2(d_2)G_1(d_2)$ of two 1D encoders is indeed a 2D encoder.

The 2D composition code $\mathscr{C}$ associated with $G(d_1, d_2)$ is such that

$$\begin{aligned}\hat{\mathscr{C}} &= \operatorname{im} G(d_1, d_2) = G_2(d_2)(\operatorname{im} G_1(d_1))\\ &= \{\hat{w}(d_1, d_2) \mid \exists\, \hat{z}(d_1, d_2) \in \operatorname{im}(G_1(d_1)) \text{ such that } \hat{w}(d_1, d_2) = G_2(d_2)\hat{z}(d_1, d_2)\}.\end{aligned}$$

We shall concentrate on a particular class of composition codes, namely on those that admit a composition encoder $G(d_1, d_2)$ as in (1) with $G_2(d_2)$ and $G_1(d_1)$ both right-prime encoders (i.e., they admit a left polynomial inverse), and derive a procedure for constructing the corresponding syndrome formers based on 1D polynomial methods. This procedure will be useful later on for the study of state-space realizations.

It is important to observe that as $G_2(d_2)$ and $G_1(d_1)$ are both assumed to have polynomial inverses, then $G(d_1, d_2)$ also has a 2D polynomial left inverse (given by the product of the left inverses of $G_1(d_1)$ and $G_2(d_2)$) and therefore $G(d_1, d_2)$ is right-zero prime[1](*rZP*). Recall that if a 2D convolutional code admits a right-zero prime encoder then all its *rFP* encoders are *rZP*. Moreover, the corresponding syndrome formers are also *lZP* (see Prop. A.4 of [4]).

[1] A polynomial matrix $G(d_1, d_2)$ is right/left-zero prime (*rZP/lZP*) if the ideal generated by the maximal order minors of $G(d_1, d_2)$ is the ring $\mathbb{F}[d_1, d_2]$ itself, or equivalently if and only if admits a polynomial left/right inverse. Moreover right/left-zero primeness implies right/left-factor primeness(*rFP/lFP*).

Since $G_2(d_2) \in \mathbb{F}^{n\times p}[d_2]$ is right-prime there exists a unimodular matrix $U(d_2) \in \mathbb{F}^{n\times n}[d_2]$ such that

$$U(d_2)G_2(d_2) = \begin{bmatrix} I_p \\ 0 \end{bmatrix}.$$

We shall partition $U(d_2)$ as

$$U(d_2) = \begin{bmatrix} L_2(d_2) \\ H_2(d_2) \end{bmatrix}, \tag{2}$$

where $L_2(d_2)$ has p rows.

It is easy to check that, if $H_1(d_1) \in \mathbb{F}^{(p-k)\times p}[d_1]$ is a syndrome former of the 1D convolutional code $\operatorname{im} G_1(d_1)$ (i.e., $H_1(d_1)$ is left-prime and is such that $H_1(d_1)G_1(d_1) = 0$), then

$$\begin{bmatrix} H_1(d_1)L_2(d_2) \\ H_2(d_2) \end{bmatrix} G_2(d_2)G_1(d_1) = 0. \tag{3}$$

This reasoning leads to the following proposition.

Proposition 5 *Let $\mathscr{C}$, with $\hat{\mathscr{C}} = \operatorname{im} G(d_1, d_2)$, be a composition code with $G(d_1, d_2) \in \mathbb{F}^{n\times k}[d_1, d_2]$ such that $G(d_1, d_2) = G_2(d_2)G_1(d_1)$, where $G_2(d_2) \in \mathbb{F}^{n\times p}[d_2]$ and $G_1(d_1) \in \mathbb{F}^{p\times k}[d_1]$ are both right-prime 1D encoders. Let further $H_1(d_1)$ be a $(p-k)\times p$ 1D syndrome former of $\operatorname{im} G_1(d_1)$ and define $\begin{bmatrix} L_2(d_2) \\ H_2(d_2) \end{bmatrix}$ as in (2). Then*

$$H(d_1, d_2) = \begin{bmatrix} H_1(d_1)L_2(d_2) \\ H_2(d_2) \end{bmatrix}$$

is a syndrome former of $\mathscr{C}$.

Proof Since (3) is obviously satisfied and $H(d_1, d_2)$ has size $(n-k)\times n$, we only have to prove that $H(d_1, d_2)$ is left-factor prime. Note that as $H_1(d_1)$ is left-prime, there exists $R_1(d_1) \in \mathbb{F}^{p\times(p-k)}[d_1]$ such that $H_1(d_1)R_1(d_1) = I_{p-k}$. Now it is easy to see that

$$R(d_1, d_2) = U(d_2)^{-1} \begin{bmatrix} R_1(d_1) & 0 \\ 0 & I_{n-p} \end{bmatrix}.$$

constitutes a polynomial right inverse of $H(d_1, d_2)$. Consequently $H(d_1, d_2)$ is left-zero prime which implies that it is left-factor prime as we wish to prove. □

3 State-Space Realizations of Encoders and Syndrome Formers

In this section we recall some fundamental concepts concerning 1D and 2D state-space realizations of transfer functions, having in mind the realizations of encoders and syndrome formers.

A 1D state-space model

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ w(t) = Cx(t) + Du(t) \end{cases}$$

denoted by $\Sigma^{1D}(A, B, C, D)$ is a realization of dimension m of $M(d) \in \mathbb{F}^{s\times r}[d]$ if $M(d) = C(I_m - Ad)^{-1}Bd + D$. Moreover, it is a minimal realization if the size of the state x is minimal among all the realizations of $M(d)$. The dimension of a minimal realization of $M(d)$ is called the *McMillan degree* of $M(d)$ and is given by $\mu(M) = \operatorname{int deg} \begin{bmatrix} M(d) \\ I_r \end{bmatrix}$, where $\operatorname{int deg} M(d)$ is the maximum degree of its r-order minors [11].

As for the 2D case, there exist several types of state-space models [1, 2]. In our study we shall consider *separable Roesser models* [13]. These models have the following form:

$$\begin{cases} x_1(i+1, j) = A_{11}x_1(i, j) + A_{12}x_2(i, j) + B_1u(i, j) \\ x_2(i, j+1) = A_{21}x_1(i, j) + A_{22}x_2(i, j) + B_2u(i, j) \\ y(i, j) = C_1x_1(i, j) + C_2x_2(i, j) + Du(i, j) \end{cases} \tag{4}$$

where A_{11}, A_{12}, A_{21}, A_{22}, B_1, B_2, C_1, C_2 and D are matrices over $\mathbb{F}$, with suitable dimensions, u is the input-variable, y is the output-variable, and $x = (x_1, x_2)$ is the state variable where x_1 and x_2 are the horizontal and the vertical state-variables, respectively. The dimension of the system described by (4) is given by the size of x. Moreover either $A_{12} = 0$ or $A_{21} = 0$. The separable Roesser model corresponding to Eqs. (4) with $A_{12} = 0$ is denoted by $\Sigma_{12}^{2D}(A_{11}, A_{21}, A_{22}, B_1, B_2, C_1, C_2, D)$, whereas the one with $A_{21} = 0$ is denoted by $\Sigma_{21}^{2D}(A_{11}, A_{12}, A_{22}, B_1, B_2, C_1, C_2, D)$.

The remaining considerations of this section can be stated for both cases when $A_{12} = 0$ or $A_{21} = 0$, however we just consider $A_{12} = 0$; the case $A_{21} = 0$ is completely analogous, with the obvious adaptations.

Definition 6 $\Sigma_{12}^{2D}(A_{11}, A_{21}, A_{22}, B_1, B_2, C_1, C_2, D)$ is said to be a realization of the 2D polynomial matrix $M(d_1, d_2) \in \mathbb{F}^{s\times r}[d_1, d_2]$ if

$$M(d_1, d_2) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} I - A_{11}d_1 & 0 \\ -A_{21}d_2 & I - A_{22}d_2 \end{bmatrix}^{-1} \left(\begin{bmatrix} B_1 \\ 0 \end{bmatrix} d_1 + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} d_2 \right) + D.$$

As it is well known different realizations of $M(d_1, d_2)$ may not have the same dimension. For the sake of efficient implementation, we are interested in studying the realizations of $M(d_1, d_2)$ with minimal dimension. Such realizations are called *minimal*. The *Roesser McMillan degree* of $M(d_1, d_2)$, $\mu_R(M)$, is defined as the dimension of a minimal realization of $M(d_1, d_2)$.

Note that every polynomial matrix $M(d_1, d_2) \in \mathbb{F}^{s\times r}[d_1, d_2]$ can be factorized as follows:

$$M(d_1, d_2) = M_2(d_2)M_1(d_1), \tag{5}$$

where $M_2(d_2) = \left[I_n \mid \cdots \mid I_n d_2^{\ell_2}\right] N_2 \in \mathbb{F}^{s\times p}[d_2]$ and $M_1(d_1) = N_1 \left[I_k \; \ldots \; I_k d_1^{\ell_1}\right]^T \in \mathbb{F}^{p\times r}[d_1]$, with N_2 and N_1 constant matrices.

If N_2 has full column rank and N_1 has full row rank we say that (5) is an *optimal decomposition* of $M(d_1, d_2)$. As shown in [8, 9], if (5) is an optimal decomposition, given a minimal realization $\Sigma^{1D}(A_{11}, B_1, \bar{C}_1, \bar{D}_1)$ of $M_1(d_1)$ (of dimension $\mu(M_1)$) and a minimal realization $\Sigma^{1D}(A_{22}, \bar{B}_2, C_2, \bar{D}_2)$ of $M_2(d_2)$ (of dimension $\mu(M_2)$) then the 2D system $\Sigma_{12}^{2D}(A_{11}, A_{21}, A_{22}, B_1, B_2, C_1, C_2, D)$, where $A_{21} = \bar{B}_2\bar{C}_1$, $B_2 = \bar{B}_2\bar{D}_1$, $C_1 = \bar{D}_2\bar{C}_1$ and $D = \bar{D}_2\bar{D}_1$, is a minimal realization of $M(d_1, d_2)$ of dimension $\mu_R(M) = \mu(M_1) + \mu(M_2)$. A similar reasoning can be made if we factorize $M(d_1, d_2) = \bar{M}_1(d_1)\bar{M}_2(d_2)$, where $\bar{M}_1(d_1) \in \mathbb{F}^{s\times \bar{p}}[d_1]$ and $\bar{M}_2(d_2) \in \mathbb{F}^{\bar{p}\times r}[d_2]$, for some $p \in \mathbb{N}$, to obtain a minimal realization $\Sigma_{21}^{2D}(A_{11}, A_{12}, A_{22}, B_1, B_2, C_1, C_2, D)$ of $M(d_1, d_2)$.

Note that, since both encoders and syndrome formers are (2D) polynomial matrices, they both can be realized by means of (4). However, when considering realizations of an encoder $G(d_1, d_2) = G_2(d_2)G_1(d_1)$ we shall take $A_{12} = 0$ and $y = w$; on the other hand when considering realizations of a syndrome former $H(d_1, d_2) = H_1(d_1)H_2(d_2)$, we shall take $A_{21} = 0$, $u = w$ and $y = 0$, (cf. Remark 3).

4 Minimal Syndrome Former Realizations of a Special Class of Composition Codes

In the sequel the composition codes $\mathscr{C}$ to be considered are such that $\hat{\mathscr{C}} = \operatorname{im} G(d_1, d_2)$, where the encoder $G(d_1, d_2)$ is as in (1) and satisfies the following properties:

(P1) $G_1(d_1)$ is a minimal 1D polynomial encoder[2] (for instance, prime and column reduced[3]), with full row rank over $\mathbb{F}$;

(P2) $G_2(d_2)$ is a quasi-systematic 1D polynomial encoder, i.e., there exists an invertible matrix $T \in \mathbb{F}^{n\times n}$ such that $TG_2(d_2) = \begin{bmatrix} I_p \\ \bar{G}_2(d_2) \end{bmatrix}$, $\bar{G}_2(d_2) \in \mathbb{F}^{(n-p)\times p}[d_2]$.

Note that both $G_1(d_1)$ and $G_2(d_2)$ are minimal encoders of the corresponding 1D convolutional codes. Moreover, $G(d_1, d_2)$ is a *minimal encoder* of $\mathscr{C}$, i.e., it has minimal Roesser McMillan degree among all encoders of $\mathscr{C}$, [9, 10], in the sequel we denote this minimal degree by $\mu(\mathscr{C})$.

In what follows, we shall derive a syndrome former construction for the code $\mathscr{C}$, based on Proposition 5. Define

$$H_1(d_1) = \begin{bmatrix} L_1(d_1) & 0 \\ 0 & I \end{bmatrix} \in \mathbb{F}^{(n-k)\times n}[d_1] \text{ and } H_2(d_2) = \begin{bmatrix} I & 0 \\ -\bar{G}_2(d_2) & I \end{bmatrix} T \in \mathbb{F}^{n\times n}[d_2],$$

where $L_1(d_1) \in \mathbb{F}^{(p-k)\times p}[d_1]$ and $\begin{bmatrix} -\bar{G}_2(d_2) & I \end{bmatrix} \in \mathbb{F}^{(n-p)\times n}[d_2]$ are 1D syndrome formers of the 1D convolutional codes im $G_1(d_1)$ and im $G_2(d_2)$, respectively. Let

$$\begin{aligned} H(d_1, d_2) &= H_1(d_1)H_2(d_2) && (6) \\ &= \begin{bmatrix} L_1(d_1) & 0 \\ -\bar{G}_2(d_2) & I \end{bmatrix} T. && (7) \end{aligned}$$

It is easy to see that $H(d_1, d_2)$ is a syndrome former of $\mathscr{C}$. It can be shown that it is possible to assume, without loss of generality, that (6) is an optimal decomposition of $H(d_1, d_2)$. Then

$$\mu_R(H) = \mu(H_1) + \mu(H_2) = \mu(L_1) + \mu(-\bar{G}_2) = \mu(L_1) + \mu(G_2).$$

Note that since $L_1(d_1)$ is a syndrome former of the 1D convolutional code im $G_1(d_1)$ and $G_1(d_1)$ is a minimal encoder of im $G_1(d_1)$, it follows that $\mu(L_1) \geq \mu(G_1)$, [5, 6], and hence $\mu_R(H) \geq \mu_R(G)$. Moreover, $\mu(L_1) = \mu(G_1)$ if $L_1(d_1)$ has minimal McMillan degree among all syndrome formers of im $G_1(d_1)$, for instance, if $L_1(d_1)$ is row reduced, [5, 6], (which can always be assumed without loss of generality, since otherwise pre-multiplication of $H(d_1, d_2)$ by a suitable unimodular matrix $U(d_1)$ yields another syndrome former for $\mathscr{C}$, with $L_1(d_1)$ row reduced); in this case $\mu_R(H) = \mu_R(G)$.

[2] A minimal 1D encoder is an encoder with minimal McMillan degree among all the encoders of the same code.

[3] A full row (column) rank matrix $M(d) \in \mathbb{F}^{n\times k}[d]$ is said to be row (column) reduced if int deg $M(d)$ is equal to the sum of the row (column) degrees of $M(d)$; in that case $\mu(M) =$ int deg $M(d)$.

Thus given the encoder $G(d_1, d_2)$ we have constructed a syndrome former $H(d_1, d_2)$, as in Proposition 5. Moreover, based on the special properties of $G(d_1, d_2)$, we have shown that the minimal realizations of $H(d_1, d_2)$ have dimension $\mu_R(H) = \mu_R(G) = \mu(\mathscr{C})$ (recall that $G(d_1, d_2)$ is a minimal encoder).

We next show that $\mu_R(H)$ is minimal among the McMillan degree of all syndrome formers of $\mathscr{C}$ with similar structure as $H(d_1, d_2)$.

Theorem 7 *Let $\mathscr{C}$, with $\hat{\mathscr{C}} = \operatorname{im} G(d_1, d_2)$, be a 2D composition code, and assume that $G(d_1, d_2) = G_2(d_2)G_1(d_1)$, where $G_1(d_1)$ and $G_2(d_2)$ satisfy properties (P1) and (P2), respectively. Let further $\tilde{H}(d_1, d_2) = \begin{bmatrix} X_1(d_1) & 0 \\ X_{21}(d_2) & X_{22}(d_2) \end{bmatrix} T$ be a syndrome former of $\mathscr{C}$, where $X_1(d_1) \in \mathbb{F}^{(p-k)\times p}[d_1]$, $X_{21}(d_2) \in \mathbb{F}^{(n-p)\times p}[d_2]$, $X_{22}(d_2) \in \mathbb{F}^{(n-p)\times(n-p)}[d_2]$ and $T \in \mathbb{F}^{n\times n}$ as in (P2). Then $\mu_R(\tilde{H}) \geq \mu(\mathscr{C})$.*

Proof Note that $\tilde{H}(d_1, d_2)G(d_1, d_2) = 0$ if and only if

$$\begin{cases} X_1(d_1)G_1(d_1) = 0 \\ \left(X_{21}(d_2) + X_{22}(d_2)\bar{G}_2(d_2)\right) G_1(d_1) = 0. \end{cases} \tag{8}$$

Then $X_1(d_1)$ must be a syndrome former of the 1D convolutional code $\operatorname{im} G_1(d_1)$ and consequently $\mu(X_1) \geq \mu(G_1)$ [6]. On the other hand we have that $X_{21}(d_2) + X_{22}(d_2)\bar{G}_2(d_2) = 0$, that is equivalent to $\begin{bmatrix} X_{21}(d_2) & X_{22}(d_2) \end{bmatrix} \begin{bmatrix} I \\ \bar{G}_2(d_2) \end{bmatrix} = 0$, and therefore $\begin{bmatrix} X_{21}(d_2) & X_{22}(d_2) \end{bmatrix}$ is a syndrome former of the 1D convolutional code $\begin{bmatrix} I \\ \bar{G}_2(d_2) \end{bmatrix}$. Hence $\mu\left(\begin{bmatrix} X_{21} & X_{22} \end{bmatrix}\right) \geq \mu\left(\begin{bmatrix} I \\ \bar{G}_2 \end{bmatrix}\right)$, since $\begin{bmatrix} I \\ \bar{G}_2(d_2) \end{bmatrix}$ is a minimal encoder of $\operatorname{im} \begin{bmatrix} I \\ \bar{G}_2(d_2) \end{bmatrix}$. Now, since $\tilde{H}(d_1, d_2) = \begin{bmatrix} X_1(d_1) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ X_{21}(d_2) & X_{22}(d_2) \end{bmatrix} T$, it is not difficult to see that

$$\begin{aligned} \mu_R(\tilde{H}) = \mu(X_1) + \mu\left(\begin{bmatrix} X_{21} & X_{22} \end{bmatrix}\right) &\geq \mu(G_1) + \mu\left(\begin{bmatrix} I \\ \bar{G}_2 \end{bmatrix}\right) \\ &= \mu(G_1) + \mu\left(T^{-1} \begin{bmatrix} I \\ \bar{G}_2 \end{bmatrix}\right) = \mu_R(G) = \mu(\mathscr{C}). \end{aligned}$$

□

Corollary 8 *Using the notation and conditions of Theorem 7, the syndrome former of $\mathscr{C}$ given by (7) has minimal Roesser McMillan degree among all syndrome formers of the same structure.*

Acknowledgements This work was supported by Portuguese funds through the CIDMA – Center for Research and Development in Mathematics and Applications, and the Portuguese Foundation for Science and Technology ("FCT-Fundação para a Ciência e a Tecnologia"), within project UID/MAT/04106/2013.

References

1. Attasi, S.: Systèmes linèaires homogènes á deux indices. Technical report, Rapport Laboria (1973)
2. Fornasini, E., Marchesini, G.: Algebraic realization theory of two-dimensional filters. In: Ruberti, A., Mohler, R. (eds.) Variable Structure Systems with Application to Economics and Biology. Lecture Notes in Economics and Mathematical Systems, vol. 111, pp. 64–82. Springer-Verlag, Berlin-Heidelberg-New York (1975)
3. Fornasini, E., Pinto, R.: Matrix fraction descriptions in convolutional coding. Linear Algebra Appl. **392**, 119–158 (2004)
4. Fornasini, E., Valcher, M.E.: Algebraic aspects of two-dimensional convolutional codes. IEEE Trans. Inf. Theory **40**(4), 1068–1082 (1994)
5. Forney, G.: Convolutional codes I: algebraic structure. IEEE Trans. Inf. Theory **16**(6), 720–738 (1970)
6. Forney, G.: Structural analysis of convolutional codes via dual codes. IEEE Trans. Inf. Theory **19**, 512–518 (1973)
7. Lévy, B.: 2D polynomial and rational matrices, and their applications for the modeling of 2-D dynamical systems. Ph.d. dissertation, Stanford University, Stanford (1981)
8. Lin, T., Kawamata, M., Higuchi, T.: Decomposition of 2-D separable-denominator systems: existence, uniqueness, and applications. IEEE Trans. Circuits Syst. **34**(3), 292–296 (1987)
9. Pinho, T.: Minimal state-space realizations of 2D convolutional codes. Phd dissertation, Department of Mathematics, University of Aveiro (2014)
10. Pinho, T., Pinto, R., Rocha, P.: Minimal realizations of a special case of 2D codes. In: Proceedings of the MTNS, Groningen (2014)
11. Pinho, T., Pinto, R., Rocha, P.: Realization of 2D convolutional codes of rate $\frac{1}{n}$ by separable $r$oesser models. Des. Codes Cryptogr. **70**, 241–250 (2014)
12. Rocha, P., Willems, J.C.: Controllability of 2-D systems. IEEE Trans. Autom. Control **36**(4), 413–423 (1991)
13. Roesser, R.P.: A discrete state-space model for linear image processing. IEEE Trans. Autom. Control **20**(1), 1–10 (1975)

Shifted de Bruijn Graphs

Ragnar Freij

Abstract We are studying a generalization of the de Bruijn graphs, with applications to storage. We use spectral methods to enumerate the Euler circuits in this graph, which correspond to (very long) strings accessing every string of fixed length exactly once, with the reader reset at regular intervals. We prove that, when the alphabet is of size q, the subwords considered are of length n and a new reader is initiated every k letters, there are exactly $(q^k)!^{q^n}/q^{k+n}$ such exhaustive words. The enumeration generalizes classic results by Tutte, and relates crucially to subtree enumeration in large networks.

Keywords de Bruijn graphs • Euler cycles • Exact enumeration • Spanning trees • String networks

1 Introduction

De Bruijn graphs are an old class of graphs [14], which have recently earned a lot of attention from the network storage and bioinformatics communities [1, 2, 9] but still wait for some of the attention it deserves from coding theorists. They are used to encode large data strings in terms of their substrings carrying non-trivial information, and provide an example of fast-encodable, fast-searchable datastructures for strings. De Bruijn graphs have also found their way to pure mathematics, for example Cooper and Graham have constructed a higher-dimensional analogue (where strings are replaced by arrays) [4], and Ehrenborg et al. have studied a version that encodes permutation pattern containment [5]. The nodes of the graph are strings of length n, and there is a (directed) edge from $v_1 \cdots v_n$ to $u_1 \cdots u_n$ if $v_2 \cdots v_n = u_1 \cdots u_{n-1}$. Any such graph, where the strings are from an alphabet on q letters, embeds into the universal de Bruijn graph $D(q, n)$, consisting of all q^n such strings, with the same adjacency relations.

R. Freij (✉)
Department of Communications and Networking, Aalto University School of Electrical Engineering, P.O. Box 13000, 00076 Aalto, Finland
e-mail: ragnar.freij@aalto.fi

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_20

The original result of de Bruijn graphs was computing the number of Hamiltonian circuits in $D(2, n)$, which by construction equals the number of binary strings on $2^n + n - 1$ letters, that contain any binary n-letter string exactly once. Flye Saint-Marie proved, in response to a question in l'Intermédiaire des Mathématiciens, that there are exactly $2^{2^n - n - 1}$ such exhaustive words [6]. In subsequent work, using more graph theoretical techniques, van Aardenne-Ehrenfest and de Bruijn generalized this to the q-letter case [14]. Recently, Rosenfeld has neatly demonstrated how to solve such problems using the spectral theory of arc graphs [10, 11].

A prototypical application of de Bruijn graphs in bioinformatics is related to genome assembly [3]. Here, one has a circular genome, and in a laboratory one can *repeatedly* read short subsequences of this genome. After many enough such reads, one has read the entire genome, but it remains to put together the reads into the original circular string. Every way to do this corresponds to an Eulerian cycle in the de Bruijn graphs spanned by the reads. In this paper, we study a shifted version of this problem.

This means that the relevant graph that models the gathered data has vertices indexed by strings of length n, and there is a directed edge from $v_1 \cdots v_n$ to $u_1 \cdots u_n$ if $v_{k+1} \cdots v_n = u_1 \cdots u_{n-k}$. Here,$k$ is the length of a byte, and will in practical applications often, but not always, divide n. The universal such graph for n-letter strings from a q-letter alphabet is denoted by $D(q, n, k)$, and our main result in this paper will be studying its spectrum and enumerating its Eulerian circuits.

2 Preliminaries on Graph Spectra

To fix some notation, a graph is a pair $G = (V(G), E(G))$ of a vertex set and an edge set. All our graphs are directed and finite, with loops and multiple edges allowed. When $e = (u, v)$ is a directed edge, we write $t(e) = u$ and $h(e) = v$ (read "tail of e" and "head of e", respectively). If $h(e) = t(e)$, then we say that e is a *loop*.

A Hamiltonian circuit in a graph is a circuit that passes through every node exactly once, and an Eulerian circuit is a circuit that passes every edge exactly once. While it is in general NP-hard to show that a given graph has a Hamiltonian circuit [8], having an Eulerian circuit has a much easier criterion. Indeed, a directed graph contains an Eulerian circuit if and only if it is connected and has that the indegree $\delta_+(v)$ and the outdegree $\delta_-(v)$ agree on each node v in G. (This result dates back to Euler.) When this is the case, the following classic result by de Bruijn, Aardenne-Ehrenfest, Smith and Tutte [14] relate Eulerian cycles (with a given starting edge e_0) to directed spanning trees (directed away from a given root vertex v_0). For $v \in V(G)$, let $\tau(G, v)$ be the number of trees in G directed away from v, and for $e \in E(G)$, let $\epsilon(G, e)$ be the number of Eulerian cycles in G starting at e.

Theorem 1 (BEST Theorem) *Let $G = (V, E)$ be a connected digraph with $\delta_+(v) = \delta_-(v)$ for every $v \in V$. Fix $e_0 \in E$ and let $v_0 = t(e_0)$. Then*

$$\epsilon(G, e_0) = \tau(G, v_0) \prod_{u \in V} (\delta_+(u) - 1)!.$$

Besides the original proof in [14], a good, self-contained proof of Theorem 1 occurs in [12], section 5.6. As the number of Euler cycles $\epsilon(G, e_0)$ does not depend on e_0, it follows curiously that $\tau(G, v_0)$ does not depend on the choice of v_0. Moreover, the number of rooted trees has a following important interpretation as a determinant (or as a product of eigenvalues):

For a digraph G on n nodes, we define its adjacency matrix $A = A(G)$ to be the $n \times n$ matrix whose entries $A_{i,j}$ are the number of edges from node i to node j. We also define the Laplacian matrix $L = L(G)$ by

$$L_{i,j} = -A_{i,j} \text{ if } i \neq j, \text{ and } L_{i,i} = \delta_-(i) A_{i,i}.$$

In particular, if G has uniform outdegree d, then $L(G) = dI - A(G)$. As the row sums in $L(G)$ are zero by construction, L has 0 as an eigenvalue.

The following celebrated theorem by Tutte [13] is a directed version of the matrix-tree theorem, and relates these matrices to the numbers $\tau(G, v)$ and $\epsilon(G, e)$.

Theorem 2 *Let G be a digraph on n nodes with Laplacian matrix $L = L(G)$. Let $\mu_0 = 0, \mu_1, \ldots, \mu_{n-1}$ be the eigenvalues of $L(G)$. Then $\tau(G, v) = \frac{1}{n} \prod_{i \neq 0} \mu_i$.*

This allows us to compute $\tau(G, v)$, which we will henceforth denote by $\tau(G)$, as $1/n$ times the product of the zeroes of the (reduced) characteristic polynomial

$$\frac{\chi(L(G))(t)}{t} := \frac{\det(tI - L(G))}{t}.$$

This is in turn the coefficient of t in $\det(L(G) - tI)$, so it can be computed as

$$\tau(G) = \frac{1}{n} \cdot \frac{\mathrm{d}}{\mathrm{d}t} \det(L(G) - tI)|_{t=0}.$$

In the case where G is d-regular, which will interest us most, we have $L(G) = dI - A(G)$, so we immediately get the following theorem.

Theorem 3 *Let G be a digraph on N nodes with $\delta_-(v) = \delta_+(v) = d$ for every $v \in V(G)$. Fix a root vertex $u \in V(G)$. The number of rooted spanning trees in G is*

$$\tau(G, u) = \frac{1}{N} \cdot \frac{\mathrm{d}}{\mathrm{d}t} \chi(A(G))(t)|_{t=d},$$

and the number of Eulerian cycles in G is

$$\epsilon(G) = (d-1)!^N \cdot \tau(G,u).$$

The arc graph of G is the digraph $\Gamma(G)$ with $V(\Gamma(G)) = E(G)$, and an edge (e, f) whenever $h(e) = t(f)$. In particular, $\Gamma(G)$ has no multiple edges, and has a loop for every loop in G. Clearly an Eulerian cycle in G corresponds to a Hamiltonian cycle in $\Gamma(G)$. The following relationship between the spectra of G and $\Gamma(G)$ was demonstrated in [10], but will here be given a slightly different presentation.

Let D_+ and D_- be matrices with rows indexed by edges and columns indexed by vertices of G. The entries are given by $D_+(e,u) = 1_{u=h(e)}$ and $D_-(e,u) = 1_{u=t(e)}$, so D_+ keeps track of incoming edges and D_- keeps track of outgoing edges from every node. It is easy to verify that the adjacency matrices can be written

$$A(G) = D_+ D_-^T$$

and

$$A(\Gamma(G)) = D_+^T D_- = (D_-^T D_+)^T.$$

But this implies that, for every eigenvalue $\lambda \neq 0$, we have

$$A(G)\mathbf{x} = D_+ D_-^T(\mathbf{x}) = \lambda\mathbf{x} \iff A(\Gamma(G))^T D_-^T(\mathbf{x}) = D_-^T D_+ D_-^T(\mathbf{x}) = \lambda D_-^T(\mathbf{x}),$$

so the non-zero eigenvalues of $A(G)$ and $A(\Gamma(G))$ agree. This proves that the characteristic polynomials satisfy the equation

$$\chi(A(\Gamma(G)))(t) = t^{M-N}\chi(A(G))(t), \tag{1}$$

where M and N are the numbers of edges and vertices respectively in the graph G.

3 Shifted DeBruijn Graphs

The DeBruijn graphs were constructed to solve the following innocent looking problem: Is there a binary word $w_1 \cdots w_n$ that, when read cyclically, contains all binary words on n letters as a factor exactly once? If so, how many such words are there? The problem was first solved in [6], and the solution was later generalized to the q-letter case in [14]. The strikingly beautiful result is that there are exactly

$$\frac{(q!)^{q^{n-1}}}{q^n}$$

such words. As explained in the introduction, these words are usually interpreted as Hamiltonian cycles in the digraph $D(q,n)$. But when $n > 1$, the edge $u_1 \cdots u_{n-1} \leftarrow u_2 \cdots u_n$ of $D(q,n-1)$ can be considered labelled by the string $u_1 \cdots u_n \in [q]^n$, and under this correspondence we see that $D(q,n) \cong \Gamma(D(q,n-1))$, so we might as well regard the exhaustive strings as *Eulerian* cycles in $D(q,n-1)$.

In this paper, we introduce the three-parameter family of *shifted de Bruijn graphs* $D(q,n,k)$, for $q \geq 1$, $1 \leq k \leq n$. The set of nodes is the same as before, $V(D(q,n,k)) = V(D(q,n)) = [q]^n$, but this time we have an edge $(u,v) \in E(D(q,n,k))$ whenever $u_{k+1} \cdots u_n = u_1 \cdots u_{n-k}$. In particular, we get the specialization $D(q,n,1) = D(q,n)$. It is also easy to see that we have $D(q,n\ell,k\ell) \cong D(q^\ell,n,k)$ for every integer ℓ. So the shifted de Bruijn graphs are only novel when n and k are relatively prime.

As explained in the introduction, computer science applications will often have n divisible by k (as a string typically consists of an integer number of bytes). In applications from chemistry and bioinformatics, where the bytes for example correspond to observed DNA sequences read from a long string, such assumptions are much less natural. The notion of shifted de Bruijn graphs was suggested by Richard Ehrenborg (personal communication).

We extend our definition to the case $0 \leq r \leq k$, by letting $D(q,r,k)$ have q^r vertices (labelled by words in $[q]^r$) and q^{k-r} arcs between every ordered pair (u,v) of nodes. The following is our key lemma for understanding shifted de Bruijn graphs, and also explains why the definition is natural when $r > k$.

Lemma 4 *For any integers $q,k \geq 1$, $n \geq 0$, we have $\Gamma(D(q,n,k)) \cong D(q,n+k,k)$.*

Proof We can label the edges of $D(q,n,k)$ as e_w by $(n+k)$-letter words $w \in [q]^{n+k}$, where $t(e_{u_1 \cdots u_{n+k}}) = u_1 \cdots u_n$ and $h(e_{u_1 \cdots u_{n+k}}) = u_{k+1} \cdots u_{n+k}$. Note that when $n < k$, there are exactly q^{k-n} words with this property, for every $u_1 \cdots u_n$ and $u_{k+1} \cdots u_{n+k}$.

Now $\{e_w : w \in [q]^{n+k}\} = V(\Gamma(D(q,n,k)))$, and $[q]^{n+k} = V(D(q,n+k,k))$. There is an arc (e_u, e_v) in $\Gamma(D(q,n,k))$ if and only if $u_{k+1} \cdots u_{n+k} = v_1 \cdots v_n$, which is equivalent to (u,v) being an arc of $D(q,n+k,k)$. This proves the lemma. □

Lemma 5 *Let $0 \leq r < k$, and fix any $v \in V(G)$. Then $\tau(D(q,r,k),v) = q^{k(q^r-1)}q^{-r}$.*

Proof In the range $r < k$, $D(q,r,k)$ is a complete graph on $N = q^r$ nodes with duplicated edges (in both directions). A spanning tree of $D(q,r,k)$, cannot contain two edges between the same pair of points, so it must correspond to a spanning tree in K_N, together with $N-1$ independent choices of one out of q^{k-r} parallel edges. Indeed, once the tree is fixed, the direction of each edge is determined by the tree being rooted at v.

A complete graph has N^{N-2} spanning trees (this is folklore, see [12] for a proof), which gives

$$\tau(D(q,r,k),v) = N^{N-2}q^{(k-r)(q^r-1)} = q^{r(q^r-2)}q^{(k-r)(q^r-1)} = q^{k(q^r-1)}q^{-r}.$$

□

4 The Spectrum of Shifted de Bruijn Graphs

Since the shifted de Bruijn graphs are regular, we can apply Theorem 3 to count the number of Euler cycles via the spectrum of the adjacency matrix (rather than the laplacian). This is valuable, because by Lemma 4 the shifted de Bruijn graphs are arc graphs, so we can use Eq. (1) to understand the spectra of their adjacency matrices.

Theorem 6 *The number of Eulerian cycles in $D(q,n,k)$ satisfies*

$$\epsilon(D(q,n,k)) = (q^k)!^{q^n}q^{-k-n}.$$

Proof We first consider the number of rooted trees $\tau(D(q,n,k)){:=}\tau(D(q,n,k),u)$ (recall that this does not depend on u). By Theorem 3, we have

$$\tau(D(q,n,k)) = \frac{1}{N}\cdot\frac{\mathrm{d}}{\mathrm{d}t}\chi(A(D(q,n,k)))(t)|_{t=d}.$$

Note that $D(q,n,k)$ has $M = q^{n+k}$ edges and $N = q^n$ vertices. By Lemma 4, we have that $D(q,n+k,k) = \Gamma(D(q,n,k))$, and by Eq. 1, we thus get

$$\chi(A(D(q,n+k,k))) = t^{q^{n+k}-q^n}\chi(A(D(q,n,k))).$$

It follows by induction that

$$\chi(A(D(q,n,k))) = t^{q^n-q^r}\chi(A(D(q,r,k))),$$

where r is the remainder of n modulo k. The graph $D(q,n,k)$ is q^k-regular for every n by construction, and thus

$$\chi(A(D(q,n,k)))(q^k) = \chi(L(D(q,n,k)))(0) = 0$$

(as the Laplacian matrix is always singular). We can now use Theorem 3 (together with the observation that $D(q,n,k)$ has q^n vertices and is $d = q^k$-regular for every n) to obtain

$$\begin{aligned}\tau(D(q,n,k)) &= q^{-n}\frac{\mathrm{d}}{\mathrm{d}t}\chi(A(D(q,n,k)))(t)|_{t=q^k}\\ &= q^{-n}\frac{\mathrm{d}}{\mathrm{d}t}t^{q^n-q^r}\chi(A(D(q,r,k)))(t)|_{t=q^k}\\ &= q^{-n}q^{k(q^n-q^r)}\frac{\mathrm{d}}{\mathrm{d}t}\chi(A(D(q,r,k)))(t)|_{t=q^k}\\ &= q^{-n}q^{k(q^n-q^r)}q^r\tau(D(q,r,k))\\ &= q^{-n}q^{k(q^n-q^r)}q^{k(q^r-1)} = q^{k(q^n-1)}q^{-n},\end{aligned}$$

where the last line is Lemma 5.

Finally, Theorem 3 yields

$$\begin{aligned}\epsilon(D(q,n,k)) &= (d-1)!^N\cdot\tau(D(q,n,k))\\ &= (q^k-1)!^{q^n}q^{k(q^n-1)}q^{-n}\\ &= (q^k)!^{q^n}q^{-k-n}.\end{aligned}$$

□

Note that this formula is consistent with the "homogeneity" property $D(q,n,k) \cong D(q^\ell, \frac{n}{\ell}, \frac{k}{\ell})$, and reduces when $k = 1$ to the known formula $\epsilon(D(q,n)) = \frac{q!^{q^n}}{q^{n+1}}$.

As mentioned in the introduction, application of de Bruijn graphs often concern certain subgraphs of the universal de Bruijn graph. Indeed, we are often not interested in all n letter strings on a q letter alphabet, but only in some certain subset of them, for example those occurring in a certain text file or those observed when examining random sequences from a genome. If the subgraph in question is generic at least globally (meaning that there is no huge clustering of the relevant strings), the spectral structure of $D(q,n,k)$ is still of great relevance. For example, a natural way to sample from the data set is by performing random walk on $D(q,n,k)$ and testing for containment in the data set. For these purposes, it is relevant to know the mixing properties, or the convergence rate of random walk on $D(q,n,k)$.

For a beautiful treatment of how to bound useful invariants such as access time and cover time of random walk in terms of its spectrum, see [7]. Of course, the limiting distribution is uniform over all strings, and for now, let us only remark that the convergence rate of random walk on a regular graph G is given by λ/d, where d is the regular degree of the graph, and $|\lambda|$ is the second to largest modulus of an eigenvalue of G. (The largest eigenvalue will always be d, with eigenvector the uniform distribution.) In the case of de Bruijn graphs, we have $d = q^k$ and $\lambda = 0$, which only proves that the convergence will be faster than exponential, but a much finer result should be obtainable by a more detailed study.

These are only a few of the reasons why understanding the global spectral behaviour of de Bruijn graphs helps us use substructures for storage applications. It is our hope that many more such applications will appear in the near future, not least through this conference.

References

1. Bowe, A., Onodera, T., Sakadane, K., Shibuya, T.: Succinct de Bruijn graphs. In: Algorithms in Bioinformatics. Lecture Notes in Computer Science, pp. 225–235. Springer, Berlin (2012)
2. Chikhi, R., Rizk, G.: Space-efficient and exact de Bruijn graph representation based on a bloom filter. Algorithms Mol. Biol. **8**, 9 (2013)
3. Compeau, P., Pevzner, P., Tesler, G.: How to apply de Bruijn graphs to genome assembly. Nat. Biotechnol. **29**, 987–991 (2011)
4. Cooper, J., Graham, R.: Generalized de Bruijn cycles (2004). arXiv:0402324
5. Ehrenborg, R., Kitaev, S., Steingrímsson, E.: Number of cycles in the graph of 312-avoiding permutations (2013). arXiv:1310.1520
6. Flye Saint-Marie, C.: Solution to question 48. l'Intermédiaire des Math. **1**, 107–110 (1894)
7. Lovász, L.: Random walks on graphs: a survey. In: Combinatorics, Paul Erdös is Eighty, pp. 1–46. János Bolyai Mathematical Society, Budapest (1993)
8. Picoleau, C.: Complexity of the Hamiltonian cycle in regular graph problem. Theor. Comput. Sci. **131**(2), 463–473 (1994)
9. Rödland, E.: Compact representation of k-mer de Bruijn graphs for genome read assembly. BMC Bioinform. **14**, 19 (2013)
10. Rosenfeld, V.: Some spectral properties of the arc-graph. Commun. Math. Comput. Chem. **43**, 41–48 (2001)
11. Rosenfeld, V.: Enumerating de Bruijn sequences. Commun. Math. Comput. Chem. **45**, 71–83 (2002)
12. Stanley, R.: Enumerative Combinatorics, Vol. 2. Cambridge Studies in Advanced Mathematics, vol. 62. Cambridge University Press, New York (1999)
13. Tutte, W.: The dissection of equilateral triangles into equilateral triangles. Proc. Camb. Philos. Soc. **44**, 71–83 (1948)
14. van Aardenne-Ehrenfest, T., de Bruijn, N.: Circuits and trees in oriented linear graphs. Simon Stevin **28**, 143–173 (1951)

New Examples of Non-Abelian Group Codes

Cristina García Pillado, Santos González, Victor Markov, Consuelo Martínez, and Alexandr Nechaev

Abstract It has been known some time ago that there are one-sided group codes that are not abelian codes, however the similar question for group codes was not known until we constructed an example of a non-abelian group code using the group ring F_5S_4. The proof needs some computational help, since we need to know the weight distribution of all abelian codes of length 24 over the prime field of 5 elements. It is natural to ask, is it really relevant that the group ring is semisimple? What happens in the case of characteristic 2 and 3? Our interest to these questions is connected also with the following open question: does the property of all group codes for the given group to be abelian depend on the choice of the base field (the similar property for left group codes does)? We have addressed this question, again with computer help, proving that there are also examples of non-abelian group codes in the non-semisimple case. The results show some interesting differences between the cases of characteristic 2 and 3. Moreover, using the group $SL(2, F_3)$ instead of the symmetric group we can prove, without using a computer for it, that there is a code over F_2 of length 24, dimension 6 and minimal weight 10. It has greater minimum distance than any abelian group code having the same length and dimension over F_2, and moreover this code has the greatest minimum distance among all binary linear codes with the same length and dimension. The existence of such code gives a good reason to study non-abelian group codes.

Keywords Group code • Abelian group code • Semisimplicity

1 Introduction

Let $G = \{g_0 = e, g_1, \ldots, g_{n-1}\}$ be a finite group and F a field. Any (left) ideal L of the group ring FG defines a (left) group code $K(L)$ of length n over F by the

C.G. Pillado (✉) • S. González • C. Martínez (✉)
Department of Mathematics, University of Oviedo, Calvo Sotelo, s/n, 33007 Oviedo, Spain
e-mail: cpillado@orion.ciencias.uniovi.es; cmartinez@uniovi.es

V. Markov • A. Nechaev
Department of Mechanics and Mathematics, Moscow State University, Moscow, Russia
e-mail: vtmarkov@yandex.ru

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_21

rule

$$(a_0, a_1, \ldots, a_{n-1}) \in K(L) \Leftrightarrow a_0 g_0 + a_1 g_1 + \ldots + a_{n-1} g_{n-1} \in L.$$

In what follows e denotes the identity element of any group G and all groups we consider will be finite.

We can consider the natural action of the symmetric group S_n on the n-dimensional space F^n defined as permutation of coordinates:

$$\sigma(a_1, \ldots, a_n) = (a_{\sigma^{-1}(1)}, \ldots, a_{\sigma^{-1}(n)}) \quad \forall (a_1, \ldots, a_n) \in F^n.$$

Two codes $C_1,\ C_2 \subseteq F^n$ are permutation equivalent if there exists a permutation $\sigma \in S_n$ such that $C_2 = \sigma(C_1)$. For a given code $C \subseteq F^n$ the group of all permutations $\sigma \in S_n$ such that $\sigma(C) = C$ is denoted by $PAut$(C).

Any code which is permutation equivalent to $K(L)$ for some (left) ideal L of the ring FG is called a *(left) G-code*. Notice that a given G-code can be realized also as H group code for a different H. In particular it is possible that G is not abelian, but H is abelian. A code is called *abelian* if it is an A-code for some abelian group A [1]. It is known that all G-codes are abelian if $|G| < 24$, but there exist non-abelian group codes when $|G| = 24$. In [4–6] we provide an example which involves the group $G = S_4$ and the finite field $F = F_5$. It was already presented in 2011 in the 3rd ICMCTA. The semisimple group ring $R = FG$ contains 5 two-sided minimal ideals generated by 5 central elements (see [3]). Three of them define abelian codes and the other two define [24, 9, 8]-non-abelian group codes.

Now we study how dependent this result is on the semisimplicity of the group ring $R = FG$. What happens in the cases $F = F_2$ and $F = F_3$?

2 Case $F = F_3$

Following the same lines that were used in the case of $F = F_5$ it can be proved:

Theorem 1 *Consider $F = F_3$ and the ideal $I = c_2(e + c_1)R$, where*

$$c_2 = (1, 2)(3, 4) + (1, 3)(2, 4) + (1, 4)(2, 3)$$

and

$$c_1 = (1, 2) + (1, 3) + (1, 4) + (2, 3) + (2, 4) + (3, 4)$$

are central elements of $R = FS_4$. The code $K(I)$ is not abelian.

The proof that this ideal is non-abelian is based on the computation of the weight distribution for this ideal. Then a search through all ideals of dimension 9 in rings FA, where A is an abelian group of order 24, is needed. To make the search faster, the group A is expressed as a direct sum of the cyclic group B of order 3 and an abelian group D of order 8, what leads to an isomorphism $FA \cong FB[D] \cong FB \otimes_F FD$ and FD is well known for every group of the kind.

No weight distribution obtained coincides with that of I.

3 Case $F = F_2$

The central element in F_5S_4 that generates an ideal that defines a non-abelian code, becomes, in the case of characteristic 2, the element

$$\alpha = e + \sum \tau + \sum_{v \in V} v + \sum c$$

where τ, v and c run respectively over the set of transpositions, the Klein subgroup V and the set of 4-cycles.

The element α generates an ideal of dimension 5 containing a basis which consists of elements of weight 16 that defines a $[24, 5, 8]$-code. The weight distribution of such an ideal is:

- 1 element of weight 0
- 15 elements of weight 8
- 15 elements of weight 16
- 1 element of weight 24

Is the corresponding $[24, 5, 8]$-code C abelian?

Considering the abelian group $C_2 \oplus C_{12} =< b > \oplus < a >$, the element

$$x = 1 + a + a^6 + a^7 + b + ab + a^6b + a^7b$$

generates an ideal which defines a $[24, 5, 8]$-code with the same weight distribution as C. This is not sufficient to assure that both codes are permutation equivalent, however it is easy to find a linear weight preserving transformation between these codes. So both codes are permutation equivalent and this code does not work as a possible counterexample to our question.

An exhaustive computer search proves the following statements:

- Given an arbitrary code in $R = F_2S_4$ there is always an abelian code with the same weight distribution and parameters.
- Every ideal of dimension less than or equal to 8, 12 or greater than or equal to 16 is permutation equivalent to an abelian code.

So we only need to check ideals of dimensions 9, 10 and 11 (the cases of dimensions 15, 14 and 13 follow by duality).

Theorem 2 *Consider the ideal Y generated by the element*

$$y = (3,4) + (2,4) + (1,2)(3,4) + (1,3,2,4) + (1,2,4) + (1,4,3) + (1,2,3,4) + (1,4,3,2)$$

The code $K(Ry)$ is not abelian.

The ideal Y has dimension 9 and its weight distribution is:

- 1 element of weight 0
- 87 elements of weight 8
- 336 elements of weight 12
- 87 elements of weight 16
- 1 element of weight 24

As it has just been mentioned, we know that there are ideals in FA for $A = C_4 \oplus C_6$ and $A = C_2 \oplus C_2 \oplus C_6$ that have the same parameters and the same weight distribution as the ideal Y. The result by Bernal et al. [1] characterizing abelian group codes as those codes C such that $PAut(C)$ contains a regular abelian subgroup does not help to prove that this code is non-abelian because the group $PAut(C)$ is too large. The third way is to consider the linear weight preserving transformations from Y to the abelian codes mentioned above. Initially, the exhaustive search of such transformations seemed impossible even using a computer. But if one takes a basis of the ideal Y consisting of 9 elements of weight 8 and divides it into blocks of 3 elements each, one can check the linear weight preserving transformations defined only on the corresponding 3-dimensional spaces. But *by pure chance* the first block happened to have no such transformations into the abelian codes we were interested in.

4 A "Good" Non-abelian Code

The examples we have provided have rather poor parameters when considered to abelian codes of the same length and dimension. The following example is better from this point of view.

The following result can be proved.

Theorem 3 *There are no abelian codes over F_2 having length* 24*, dimension* 6 *and minimal weight greater than* 8.

Now we can describe the main construction.

We start with the following example given in [2]. If the base field is $E = F_4$ then there exist left group codes over the quaternion group $Q = \{\pm 1, \pm i, \pm j, \pm k\}$ that are non-abelian. One of these codes is linked to a left ideal L of the group ring EQ generated by the element $i + j + k + \vartheta \cdot 1 + \vartheta^2 \cdot (-1)$, where ϑ generates E over $F = F_2$. We can prove that L has dimension 3 over E and minimal weight 5.

Inside of the group $G = SL(2, F_3)$ we can identify a subgroup isomorphic to the quaternion group. Denote this subgroup also by Q and select an element $t \in G$ having order 3. So the group $T =< t >$ contains three elements and G decomposes into a semi-direct product $G = Q \leftthreetimes T$, which implies that FG is a free right module over FT with basis Q. Now there are two orthogonal idempotents in the ring $R = FT$, namely $e_1 = e + t + t^2$ and $e_2 = t + t^2$. Obviously this leads to the decomposition of the ring FT into the direct sum of two subrings: $R = Re_1 \oplus Re_2$. The first summand is isomorphic to F. A direct check shows that the element $te_2 = e + t^2$ satisfies in Re_2 the equation $x^2 + x + 1 = 0$, and ϑ is a root of the same equation. So there exists a ring isomorphism between E and Re_2 such that 1 corresponds to $e_2 = t + t^2$ and ϑ corresponds to $te_2 = e + t^2$. This isomorphism can be naturally extended to an F-linear mapping $f : EQ \to FG$ acting identically on Q. The image $f(L)$ happens to be a two-sided ideal in FG and since f doubles the dimension of any subspace in EQ and the weight of each element in EQ, the ideal $f(L)$ has dimension 6 and minimal weight 10. The previous theorem assures that the corresponding code is non-abelian.

Now, we can use the following result:

Theorem 4 (Theorem 5.1 [6]) *Let F be a subfield of a field E and G a group. If all G-codes over E are abelian then all G-codes over F are abelian.*

As an immediate corollary we obtain

Theorem 5 *For any finite field K of characteristic* 2 *there exists a non-abelian group code of length* 24 *and dimension* 6*.*

It is known that the greatest minimum distance of a linear binary code of dimension 6 and length 24 is 10 [7]. So the code constructed above is "better" than any abelian group code and achieves the upper bound of minimum distance in the class of all binary linear codes (naturally with the same length and dimension).

Acknowledgements This work was partially supported by project MTM2010-18370-C04-01* and by RFBR grant 11-01-00794-a.

References

1. Bernal, J., del Río, A., Simón, J.: An intrinsical description of group codes. Des. Codes Cryptogr. **51**(3), 289–300 (2009)
2. Couselo, E., González, S., Markov, V., Nechaev, A.: Loop codes. Discret. Math. Appl. **14**(2), 163–172 (2004)
3. Curtis, R., Reiner, I.: Representation Theory of Finite Groups and Associative Algebras. Wiley, New York (1962)
4. García Pillado, C., González, S., Martínez, C., Markov, V., Nechaev, A.: Group codes which are not abelian group codes. In: 3rd International Castle Meeting on Coding Theory and Applications, Cardona, pp. 123–127. Universitat Autònoma de Barcelona, Servei de Publicacions (2011)

5. García Pillado, C., González, S., Martínez, C., Markov, V., Nechaev, A.: When are all group codes of a noncommutative group abelian (a computational approach)? J. Math. Sci. **186**(5), 578–585 (2012)
6. García Pillado, C., González, S., Martínez, C., Markov, V., Nechaev, A.: Group codes over non-abelian groups. J. Algebra Appl. **12**(7) (2013). doi:10.1142/S0219498813500370
7. Grassl, M.: Bounds on the minimum distance of linear codes and quantum codes. Online available at http://www.codetables.de (2007). Accessed on 29 April 2014

Cyclic Convolutional Codes over Separable Extensions

José Gómez-Torrecillas, F.J. Lobillo, and Gabriel Navarro

Abstract We show that, under mild conditions of separability, an ideal code, as defined in Lopez-Permouth and Szabo (J Pure Appl Algebra 217(5):958–972, 2013), is a direct summand of an Ore extension and, consequently, it is generated by an idempotent element. We also design an algorithm for computing one of these idempotents.

Keywords Separable extension • Convolutional code • Ideal code

1 Introduction

Most of the codes used in engineering support a vector space structure (linear block codes) or become a direct summand of a free module over a polynomial ring (convolutional codes). In the linear case, the benefits are increased if we also consider the notion of cyclicity, since the vector space is also endowed with an algebra structure and cyclic codes come to be ideals. Over convolutional codes, this notion requires something more sophisticated than a simple extension of the block one [1, 4, 5], and the underlying working algebra is no longer a polynomial ring but a skew polynomial ring $A[z;\sigma]$ over a finite commutative semisimple algebra A. Very recently, in [3], these codes are called ideal codes and they are defined over Ore polynomial rings $A[z;\sigma,\delta]$, where A is a finite (possibly non–commutative) semisimple algebra. Nevertheless, effective procedures and results to compute generator matrices, parity check matrices and dual codes are only provided whenever A is a separable group algebra of a finite group over a finite field, see [3, Sections 4 and 5]. In this work we aim to cover more examples, see Example 3, than the aforementioned papers by assuming only certain mild conditions of separability in A, see Theorem 1. In particular, we prove that every ideal code is generated by

J. Gómez-Torrecillas (✉) • F.J. Lobillo (✉)
Department of Algebra and CITIC, University of Granada, Granada, Spain
e-mail: gomezj@ugr.es; jlobillo@ugr.es

G. Navarro (✉)
Department of Computer Sciences and AI, and CITIC, University of Granada, Granada, Spain
e-mail: gnavarro@ugr.es

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_22

an idempotent element (Theorem 4). We also provide an algorithm for computing a generating idempotent (Algorithm 3) which, in particular, is applicable to the ideal codes from [1], see Example 2. For brevity, we only shall consider Ore polynomial rings $A[z;\sigma,\delta]$, where the σ-derivation $\delta = 0$, albeit, under suitable conditions, our results remain true for a more general δ. In our examples, except for 0 and 1, we shall write the elements of a finite field $\mathbb{F}$ as powers of a primitive element and not as polynomials.

2 Separable Extensions and Ideal Codes

A ring extension $S \subseteq R$ of non–commutative rings is called separable if there exists $p = \sum_i a_i \otimes b_i \in R \otimes_S R$ such that $rp = pr$ for all $r \in R$ and $\sum_i a_i b_i = 1$. This element is called a *separability element*. In a separable extension, R–submodules of a left R–module which are S–direct summands are also R–direct summands, as proved in [2]. This is the key feature for us. Concretely, let I be a left ideal of R which is a direct summand of R as an S-submodule, and let ι denote the section of the projection $\pi : R \to R/I$ viewed as a morphism of left S-modules. Hence I is also a direct summand of R as R-module. In fact, if $\sum_i a_i \otimes b_i$ is a separability element, then $\beta : R/I \to R$, defined by $\beta(r + I) = \sum_i a_i \iota(b_i r) = \sum_i r a_i \iota(b_i)$, is the section of π viewed as a morphism of left R-modules.

Our first goal is to extend separability to Ore extensions. By $\mathrm{Aut}(R)$ we denote the set of ring automorphisms of a ring R. Let us fix $\sigma \in \mathrm{Aut}(R)$ with $\sigma(S) \subseteq S$. Even though that σ needs not to be an S–bimodule map, it is possible to extend it to a map $\sigma^{\otimes} : R \otimes_S R \to R \otimes_S R$ given by $\sigma^{\otimes}(a \otimes b) = \sigma(a) \otimes \sigma(b)$. We recall that the Ore extension $R[z;\sigma]$, where $\sigma \in \mathrm{Aut}(R)$, is the free right R–module with basis the powers of z and multiplication defined by the rule

$$az = z\sigma(a) \text{ for all } a \in R.$$

Hence the elements in $R[z;\sigma]$ are polynomials in z with coefficients on the right, and $R \subseteq R[z;\sigma]$ as polynomials of degree 0.

Theorem 1 *Let $S \subseteq R$ be a separable extension with separability element $p = \sum_i a_i \otimes_S b_i \in R \otimes_S R$ and $\sigma \in \mathrm{Aut}(R)$ with $\sigma(S) \subseteq S$. If $\sigma^{\otimes}(p) = p$, then $S[z;\sigma_{|S}] \subseteq R[z;\sigma]$ is separable and a separability element of the extension is given by $\overline{p} = \sum_i a_i \otimes_{S[z;\sigma_{|S}]} b_i \in R[z;\sigma] \otimes_{S[z;\sigma_{|S}]} R[z;\sigma]$.*

Example 2 When dealing with σ-cyclic convolutional codes (σ-CCC's) [1], Theorem 1 can always be applied. We recall that a σ-CCC is a left ideal I of $A[z;\sigma]$, where $A = \mathbb{F}[x]/\langle x^n - 1\rangle$ with $(n, \mathrm{char}(\mathbb{F})) = 1$, and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$. Since A is a finite product of finite field extensions of $\mathbb{F}$, it is enough to compute a separability element under the conditions of Theorem 1 for each block field. It is easy to see that the sum of all of them is a separability element of the whole algebra A. To illustrate the method, we shall detail the following example.

Let $\mathbb{F}_4 = \mathbb{F}_2[\alpha]/(\alpha^2 + \alpha + 1)$ be the field with four elements and $A = \frac{\mathbb{F}_4[x]}{(x^5-1)}$. Hence $A \cong K_0 \times K_1 \times K_2$, where

$$K_0 = \frac{\mathbb{F}_4[x]}{(x+1)}, K_1 = \frac{\mathbb{F}_4[x]}{(x^2+\alpha x+1)} \text{ and } K_2 = \frac{\mathbb{F}_4[x]}{(x^2+\alpha^2 x+1)}.$$

Following [1], consider the isomorphisms $\psi_{12} : K_1 \to K_2$ and $\psi_{21} : K_2 \to K_1$ given by $\psi_{12}(x) = \alpha^2 x + 1$ and $\psi_{21}(x) = \alpha x + \alpha$. Let $\sigma : A \to A$ be the automorphism defined by $\sigma(x) \equiv \sigma(1, x, x) = (1, \psi_{21}(x), \psi_{12}(x)^4) = x^3$, by using Chinese Remainder Theorem (CRT). A separability element for each block field extension can be obtained from dual bases. Concretely, for dual basis $\{a_i\}_i$ and $\{b_i\}_i$, simply set $p = \sum_i a_i \otimes b_i$. In this case, $\{1\}$ is a self-dual normal basis of K_0. For K_1, a normal basis is $\{x, x^4\}$, and its dual basis is given by $\{\alpha x, (\alpha x)^4\}$. Apply ψ_{12} to obtain that $\{\alpha^2 x+1, (\alpha^2 x+1)^4\}$ and $\{x+\alpha, (x+\alpha)^4\}$ are dual basis for K_2. By using CRT, it is straightforward to calculate all these elements in A and compute a separability element p:

$$\begin{aligned} p = {} & (x^4 + x^3 + x^2 + x + 1) \otimes (x^4 + x^3 + x^2 + x + 1) \\ & + (\alpha^2 x^4 + \alpha^2 x^3 + \alpha x^2 + \alpha) \otimes (x^4 + x^3 + \alpha^2 x^2 + \alpha^2) \\ & + (\alpha x^3 + \alpha^2 x^2 + \alpha^2 x + \alpha) \otimes (\alpha^2 x^3 + x^2 + x + \alpha^2) \\ & + (\alpha^2 x^4 + \alpha^2 x^2 + \alpha x + \alpha) \otimes (x^4 + x^2 + \alpha^2 x + \alpha^2) \\ & + (\alpha x^4 + \alpha^2 x^3 + \alpha^2 x + \alpha) \otimes (\alpha^2 x^4 + x^3 + x + \alpha^2) \end{aligned}$$

Observe that, by construction, $\sigma^{\otimes}(p) = p$, so $\overline{p}$ is a separability element for the extension $\mathbb{F}_4[z] \subseteq A[z; \sigma]$.

Example 3 Let $A = \mathscr{M}_2(\mathbb{F}_8)$ be the ring of 2×2 matrices over the field with 8 elements, where $\mathbb{F}_8 = \mathbb{F}_2[\alpha]/(\alpha^3 + \alpha + 1)$. Let $\sigma : A \to A$ be the inner automorphism given by $\sigma(X) = UXU^{-1}$, where

$$U = \begin{pmatrix} 0 & \alpha^2 \\ \alpha^3 & \alpha^6 \end{pmatrix}.$$

The reader may check that the order of σ is 3. It is well-known that $\mathbb{F}_8 \subset A$ is a separable extension and

$$e = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

is a separability element. Hence, since $(|\sigma|, \mathrm{char}(\mathbb{F}_8)) = 1$, it is not difficult to check that $p = |\sigma|^{-1}(e + \sigma^{\otimes}(e) + (\sigma^2)^{\otimes}(e))$ is a separability element of the extension such that $\sigma^{\otimes}(p) = p$. Concretely

$$
\begin{aligned}
p = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \alpha^4 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ \alpha^4 & 1 \end{pmatrix} \\
+ \begin{pmatrix} \alpha^3 & \alpha^6 \\ 1 & \alpha^3 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ \alpha & 0 \end{pmatrix} + \begin{pmatrix} 0 & \alpha^3 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & \alpha^3 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & \alpha^6 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} \alpha^4 & 1 \\ \alpha & \alpha^4 \end{pmatrix}.
\end{aligned}
$$

Note that this last trick can be used for the group algebras discussed in [3].

From now on $\mathbb{F}$ denotes a finite field and A a finite semisimple $\mathbb{F}$–algebra. By Artin–Wedderburn's theorem and Wedderburn's little theorem, A is isomorphic to a finite direct product of matrix rings over finite field extensions of $\mathbb{F}$. Since matrix rings over a field are separable extension of their base field and finite fields are separable, see Examples 2 and 3, it follows from [2, Proposition 2.5] that $\mathbb{F} \subseteq A$ is a separable extension. Denote by p a separability element. Let also $\mathscr{B} = \{v_0, \dots, v_{n-1}\}$ be a basis of A over $\mathbb{F}$. Under the usual identification $\mathbb{F}^n[z] \cong \mathbb{F}[z]^n$, $\mathscr{B}$ induces an isomorphism of $\mathbb{F}[z]$–modules $\mathfrak{v} : A[z;\sigma] \to \mathbb{F}[z]^n$ given by $\mathfrak{v}(\sum_i z^i f_i) = \left(\sum_i z^i f_{i,0}, \dots, \sum_i z^i f_{i,n-1}\right)$ where, for all i, $f_i = f_{i,0}v_0 + \dots + f_{i,n-1}v_{n-1}$ and the $\mathbb{F}[z]$ action on $A[z;\sigma]$ is given by left multiplication. We denote $\mathfrak{p} = \mathfrak{v}^{-1}$.

By [3], an *ideal code* is a left ideal $I \leq A[z;\sigma]$ such that $\mathfrak{v}(I)$ is a direct summand of $\mathbb{F}[z]^n$. Ideal codes generalize the notion of cyclicity in convolutional codes given in [1]. Nevertheless, under the conditions of Theorem 1, I is also a direct summand as left ideal of $A[z;\sigma]$.

Theorem 4 *Let A be a finite dimensional semisimple algebra over a finite field $\mathbb{F}$, p a separability element of the extension and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ with $\sigma^{\otimes}(p) = p$. Then each ideal code is a direct summand of $A[z;\sigma]$ as a left ideal. In particular, it is generated by an idempotent.*

The following problem still remains open [3].

Conjecture 5 Let A be a finite dimensional semisimple algebra over a finite field and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$. Then each ideal code is a direct summand of $A[z;\sigma]$.

3 Computing a Generating Idempotent

We follow the notation of Sect. 2. Let $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ with $\sigma^{\otimes}(p) = p$. Let I be the left ideal of $R = A[z;\sigma]$ generated by $G = \{g_0, \dots, g_{t-1}\}$. Observe that R is generated as an $\mathbb{F}[z]$–module by $\mathscr{B}$, so a generator matrix $M(G)$ for $\mathfrak{v}(I)$ has as rows the vectors $\{\mathfrak{v}(v_i g_j) \mid 0 \leq i \leq n-1, 0 \leq j \leq t-1\}$. Let us consider the presentation of R/I

$$\begin{tikzcd} R^t \arrow[r, "\cdot G"] & R_R \arrow[r, "\pi"] & R/I \arrow[r] & 0 \end{tikzcd} .$$

Hence, by means of the isomorphisms of $\mathbb{F}[z]$-modules $\mathfrak{v}$ and $\mathfrak{p}$, R/I is identified with the cokernel of the right multiplication by the matrix $M(G)$. The left ideal I is an ideal code if and only if it is a $\mathbb{F}[z]$–direct summand of R, equivalently, if and only if the Smith canonical form H of $M(G)$ is basic, that is, H is the matrix of size $tn \times n$ given by $H = \left(\begin{smallmatrix} I_k & 0 \\ 0 & 0 \end{smallmatrix}\right)$, where k is the dimension of the code, and I_k is the identity matrix of order k. Let P and Q invertible matrices with coefficients in $\mathbb{F}[z]$ with suitable sizes such that $PM(G)Q = H$. Let also $V = \left(\begin{smallmatrix} 0 \\ I_{n-k} \end{smallmatrix}\right)$, and V^T the transpose of V. Since the morphism h, given by right multiplication by QV, is also a cokernel map for $M(G)$ with splitting morphism s, given by right multiplication by $V^T Q^{-1}$, there exists an isomorphism of $\mathbb{F}[z]$–modules $\gamma : R/I \to \mathbb{F}[z]^{n-k}$ uniquely determined by h and π. Define $\iota : R/I \to R$ by $\iota = \mathfrak{p}s\gamma$. It is straightforward to check that ι is an splitting morphism for π as morphism of $\mathbb{F}[z]$–modules. According to Sect. 2, the homomorphism of left R–modules $\beta : R/I \to R$ defined as

$$\beta(r + I) = \sum_i a_i \iota(b_i r + I)$$

for all $r + I \in R/I$ splits π. In particular, $e = \beta(1 + I)$ is an idempotent in R which generates a complement of I in R and, since $\pi(1 - e) = 0$, then $f = 1 - e$ generates the left ideal I. Now, e can be explicitly computed:

$$e = \beta(1 + I) = \sum_i a_i \mathfrak{p}(\mathfrak{v}(b_i) M_h M_s).$$

The above reasoning proves the correctness of the following algorithm.

Example 6 (continuation of Example 2) Let I be the left ideal of $A[x; \sigma]$ generated by

$$g = z\left(\alpha x^4 + \alpha^2 x^3 + \alpha^2 x + \alpha\right) + \left(\alpha x^3 + \alpha^2 x^2 + \alpha^2 x + \alpha\right).$$

With the aid of some mathematical software, one may compute $M(g)$, which is called the σ-circulant matrix of g in [1], whose Smith normal form decomposition is $H = PM(g)Q$, where $H = \left(\begin{smallmatrix} I_2 & 0 \\ 0 & 0 \end{smallmatrix}\right)$. Therefore, I is a σ-cyclic convolutional code of dimension 2 and length 5. Following Algorithm 3, a generating idempotent is given by

$$\begin{aligned} f = {} & z^3\left(\alpha^2 x^4 + \alpha x^3 + \alpha x^2 + \alpha^2 x\right) + z^2\left(x^4 + x^3 + x^2 + x\right) \\ & + z\left(x^4 + x^3 + x^2 + x\right) + \left(\alpha x^4 + \alpha^2 x^3 + \alpha^2 x^2 + \alpha x\right). \end{aligned}$$

A basic encoder may be calculated by considering the first two rows of the matrix $PM(g)$, and the reader may check that the degree of I is $\delta = 2$. Then, this is a $(5, 2, 2)_4$ convolutional code and its free distance is bounded by 9, the singleton bound. Actually, we may calculate the first terms of the column and row distances

Algorithm 3 Computation of a generating idempotent

```
Input: $G = \{g_0, \dots, g_{t-1}\} \subseteq R = A[z;\sigma]$ non-zero. Assumption. $\mathbb{F} \subseteq A$ is finite semisimple
    with separability element $p = \sum_i a_i \otimes b_i$ in the conditions of Theorem 4.
Output: An idempotent $f \in R$ such that $Rf = Rg_0 + \cdots + Rg_{t-1}$, or zero if it does not exist.
Compute the matrix $M(G)$
Compute the Smith normal form decomposition $H = PM(G)Q$
if $H$ is not basic then
    return 0
end if
$V \leftarrow \left(\begin{smallmatrix} 0 \\ I_{n-k} \end{smallmatrix}\right)$, where $k = \operatorname{rank}(H)$ and $n = \dim_{\mathbb{F}}(A)$
$M_h \leftarrow QV$, $M_s \leftarrow V^T Q^{-1}$, $M \leftarrow M_h M_s$
Compute $e_i = \mathfrak{p}(\mathfrak{v}(b_i) \cdot M)$ for all $i$
$e \leftarrow \sum_i a_i e_i$
return $f = 1 - e$
```

of I [6]. Concretely, $d_0^c = 6$, $d_0^r = 8$, $d_1^c = 8$. Hence, the free distance of I is $d_{\text{free}}(I) = 8$.

Example 7 (continuation of Example 3) Let $\mathscr{B} = \left\{\left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right)\right\}$ be the chosen basis of $\mathscr{M}_2(\mathbb{F}_8)[z;\sigma]$ as $\mathbb{F}_8[z]$–module. Let I be the left ideal of $\mathscr{M}_2(\mathbb{F}_8)[z;\sigma]$ generated by the polynomial $g = z^2 \left(\begin{smallmatrix} \alpha & \alpha^2 \\ \alpha^3 & \alpha^4 \end{smallmatrix}\right) + z \left(\begin{smallmatrix} \alpha^4 & 0 \\ \alpha^3 & 1 \end{smallmatrix}\right) + \left(\begin{smallmatrix} \alpha^2 & 1 \\ \alpha & \alpha^6 \end{smallmatrix}\right)$. Hence,

$$M(g) = \begin{pmatrix} \alpha^6 z^2 + \alpha^2 & z^2 + 1 & \alpha^3 z^2 + z & \alpha^4 z^2 + z \\ \alpha^2 z^2 + \alpha & \alpha^3 z^2 + \alpha^6 & \alpha^6 z^2 + \alpha^5 z & z^2 \\ \alpha^2 z^2 + \alpha^6 z & \alpha^3 z^2 + \alpha^6 z & \alpha^3 z + \alpha^2 & \alpha^3 z + 1 \\ \alpha^5 z^2 + \alpha^4 z & \alpha^6 z^2 & \alpha z + \alpha & \alpha^6 \end{pmatrix}$$

whose Smith normal form decomposition is given by $H = PM(g)Q$, where

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & \alpha^6 z + \alpha^4 & \alpha^5 z^2 + \alpha^3 z + \alpha^4 & \alpha^4 z^3 + \alpha^3 z^2 + \alpha^4 z + \alpha \\ 0 & \alpha^5 z + \alpha^6 & \alpha^4 z^2 + \alpha^5 z + \alpha^6 & \alpha^3 z^3 + \alpha^3 z + \alpha^3 \\ 0 & \alpha & z & \alpha^6 z^2 + z + \alpha^5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{and } P = \begin{pmatrix} \alpha^4 & \alpha & 0 & 0 \\ z + \alpha^2 & \alpha^4 z + \alpha^3 & \alpha^4 & 0 \\ \alpha^4 z^2 + \alpha^2 z + \alpha^4 & \alpha z^2 + \alpha^3 z + \alpha^5 & \alpha^4 z & 0 \\ \alpha z^2 + \alpha^3 z + \alpha^4 & \alpha^5 z^2 + \alpha^3 z + \alpha^5 & \alpha z + \alpha^6 & 1 \end{pmatrix}.$$

Therefore, I is an ideal code of dimension 2 and length 4. Following Algorithm 3, the morphism h and its section s are given by the matrices

$$M_h = \begin{pmatrix} \alpha^5 z^2 + \alpha^3 z + \alpha^4 & \alpha^4 z^3 + \alpha^3 z^2 + \alpha^4 z + \alpha \\ \alpha^4 z^2 + \alpha^5 z + \alpha^6 & \alpha^3 z^3 + \alpha^3 z + \alpha^3 \\ z & \alpha^6 z^2 + z + \alpha^5 \\ 0 & 1 \end{pmatrix} \text{ and } M_s = \begin{pmatrix} 0 & \alpha & \alpha^5 z + \alpha^6 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Hence, $\iota(a+I) = \mathfrak{p}(\mathfrak{v}(a)\cdot M)$ for any $a+I \in R/I$, where

$$M = \begin{pmatrix} 0 & \alpha^6 z^2 + \alpha^4 z + \alpha^5 & \alpha^3 z^3 + \alpha^2 z^2 + \alpha^3 & \alpha^4 z^3 + \alpha^3 z^2 + \alpha^4 z + \alpha \\ 0 & \alpha^5 z^2 + \alpha^6 z + 1 & \alpha^2 z^3 + \alpha^5 & \alpha^3 z^3 + \alpha^3 z + \alpha^3 \\ 0 & \alpha z & \alpha^5 z^2 + \alpha^6 z & \alpha^6 z^2 + z + \alpha^5 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

So, the parity check idempotent polynomial is $e = \mu(id \otimes \iota)(p)$. Concretely,

$$e = z^3 \begin{pmatrix} \alpha^6 & 1 \\ \alpha^5 & \alpha^6 \end{pmatrix} + z^2 \begin{pmatrix} \alpha^5 & \alpha \\ \alpha^2 & 0 \end{pmatrix} + z \begin{pmatrix} 0 & \alpha^2 \\ \alpha^3 & 1 \end{pmatrix} + \begin{pmatrix} \alpha^6 & 1 \\ \alpha & \alpha^2 \end{pmatrix}$$

and the generating idempotent of I is

$$f = 1 - e = z^3 \begin{pmatrix} \alpha^6 & 1 \\ \alpha^5 & \alpha^6 \end{pmatrix} + z^2 \begin{pmatrix} \alpha^5 & \alpha \\ \alpha^2 & 0 \end{pmatrix} + z \begin{pmatrix} 0 & \alpha^2 \\ \alpha^3 & 1 \end{pmatrix} + \begin{pmatrix} \alpha^2 & 1 \\ \alpha & \alpha^6 \end{pmatrix}.$$

Again, we may calculate the first terms of the column and row distances of the ideal code I. Concretely, $d_0^c = 3$, $d_1^c = 4$ and $d_0^r = 4$. Hence, by [6], the free distance of I is $d_{\text{free}}(I) = 4$. The degree is 2, therefore it is a $(4,2,2)_8$ convolutional code. The singleton bound is 7.

Acknowledgements Research partially supported by grant MTM2010-20940-C02-01 from the Ministerio de Ciencia e Innovación of the Spanish Government and FEDER, and by grant mP-TIC-14 (2014) from CEI-BioTic Granada.

References

1. Gluesing-Luerssen, H., Schmale, W.: On cyclic convolutional codes. Acta Appl. Math. **82**(2), 183–237 (2004)
2. Hirata, K., Sugano, K.: On semisimple extensions and separable extensions over non commutative rings. J. Math. Soc. Jpn. **18**(4), 360–373 (1966)
3. López-Permouth, S.R., Szabo, S.: Convolutional codes with additional algebraic structure. J. Pure Appl. Algebra **217**(5), 958–972 (2013)
4. Piret, P.: Structure and constructions of cyclic convolutional codes. IEEE Trans. Inf. Theory **22**(2), 147–155 (1976)
5. Roos, C.: On the structure of convolutional and cyclic convolutional codes. IEEE Trans. Inf. Theory **25**(6), 676–683 (1979)
6. Smarandache, R., Gluesing-Luerssen, H., Rosenthal, J.: Constructions of MDS-convolutional codes. IEEE Trans. Inf. Theory **47**(5), 2045–2049 (2001)

Reachability of Random Linear Systems over Finite Fields

Uwe Helmke, Jens Jordan, and Julia Lieb

Abstract This paper deals with the probability of classical system-theoretic properties of random linear systems defined over a finite field. Explicit formulas are derived for the probability that the reachability matrix of a linear system has rank r. We also calculate the probability that the parallel connection of two single-input systems is reachable. Our results should be viewed as a first step to calculate the probability that a network of linear systems is reachable.

Keywords Linear systems • Finite fields • Reachability • Grassmann manifolds • Parallel connection

1 Introduction

In network control of multi-agent systems a natural question of interest regards the characterization of reachability and observability properties of classes of interconnected linear systems. Networks of linear systems defined over finite fields are of interest in many application areas, as they provide efficient tools to investigate quantization effects, for studying convolutional codes and Boolean networks and promise interesting applications to network coding; see e.g. [6, 10, 11] and [8]. In such applications it becomes important to estimate the probability that randomly chosen interconnection parameters entail reachability. It is for such reasons that we became interested in estimating the probability that a linear control system is reachable.

In this paper we determine the probability that a discrete-time linear control system

$$x(\tau + 1) = Ax(\tau) + Bu(\tau), \quad x(0) = 0, \quad \tau = 0, 1, 2, \ldots$$

U. Helmke (✉) • J. Jordan (✉) • J. Lieb (✉)
Institute of Mathematics, University of Würzburg, Würzburg, Germany
e-mail: helmke@mathematik.uni-wuerzburg.de; jordan@mathematik.uni-wuerzburg.de; julia.lieb@mathematik.uni-wuerzburg.de

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_23

with system matrices $A \in \mathbf{F}^{n\times n}$, $B \in \mathbf{F}^{n\times m}$ defined over a finite field $\mathbf{F}$ is reachable. We do this in Sect. 3 by calculating the number of state space pairs (A, B) with a reachability space of fixed dimension. One of our main results is Theorem 1, which establishes an explicit formula for the number of reachable pairs. This extends earlier results by [7], derived for $m \leq 2$ inputs. Our proof of Theorem 1 rests on the Hermite cell decomposition of the quotient space of reachable pairs, introduced in [4] and [5]. We also determine the probability for a parallel connection of two single-input systems to be reachable. This is based on an explicit formula for the number of pairs of coprime polynomials over $\mathbf{F}$.

2 Counting Points of the Grassmannian

We begin with a brief summary of well-known properties of Grassmannians over a finite field $\mathbf{F}$ with cardinality $|\mathbf{F}|$. Throughout this paper we assume that $\mathbf{F}$ is endowed with the uniform probability distribution that assigns to each field element the same probability

$$t = \frac{1}{|\mathbf{F}|}.$$

The **Grassmannian** over a finite field $\mathbf{F}$ is the set $G_k(\mathbf{F}^n)$ of all k-dimensional linear subspaces $V \subset \mathbf{F}^n$ (more precisely, it is the set of $\mathbf{F}$-rational points of the Grassmann variety, but this distinction does not play a significant role in this paper). To count the number of elements in $G_k(\mathbf{F}^n)$ one can proceed in at least two different ways. The first approach proceeds by identifying the Grassmann manifold $G_k(\mathbf{F}^n)$ of k-dimensional linear subspaces of $\mathbf{F}^n$ with the homogeneous space $GL_n(\mathbf{F})/\mathscr{P}$ by the parabolic subgroup

$$\mathscr{P} = \begin{pmatrix} GL_k(\mathbf{F}) & \mathbf{F}^{k\times(n-k)} \\ 0 & GL_{n-k}(\mathbf{F}) \end{pmatrix}$$

of block upper triangular matrices. It is well-known and easily established that the general linear group $GL_n(\mathbf{F})$ of invertible $n \times n$-matrices has exactly

$$|GL_n(\mathbf{F})| = t^{-n^2} \prod_{j=1}^{n} (1 - t^j) \tag{1}$$

elements. Therefore the Grassmannian $G_k(\mathbf{F}^n)$ has exactly

$$|G_k(\mathbf{F}^n)| = \frac{|GL_n(\mathbf{F})|}{|GL_k(\mathbf{F})||GL_{n-k}(\mathbf{F})||\mathbf{F}^{k\times(n-k)}|} = t^{-k(n-k)} \prod_{j=1}^{n-k} \frac{(1 - t^{k+j})}{(1 - t^j)} \tag{2}$$

many points. In particular, we conclude the well-known formula that the projective space $\mathbf{P}^{n-1}(\mathbf{F})$ has exactly $1 + t^{-1} + \cdots + t^{1-n}$ many elements.

Alternatively, one constructs a cell decomposition of the Grassmannian $G_k(\mathbf{F}^n)$

$$G_k(\mathbf{F}^n) = \bigsqcup_{a \in \mathscr{A}_{k,n}} S(a)$$

into finitely many disjoint Euclidean spaces $S(a)$; see e.g. [9]. Recall that a Schubert-cell $S(a) \subset G_k(\mathbf{F}^n)$ is defined for each sequence $a = (a_1, \ldots, a_k)$ of strictly increasing integers $1 \leq a_1 < \ldots < a_k \leq n$. Let $\mathscr{A}_{k,n}$ denote the set of such sequences a. Thus, $S(a)$ can be identified with the set of all full row rank $k \times n$ matrices X that are in a-row echelon canonical form, i.e. $X = (x_{ij}) \in \mathbf{F}^{k \times n}$ has the standard basis vectors $e_1, \ldots, e_k$ at columns $a_1, \ldots, a_k$ and satisfies $x_{ij} = 0$ for $j < a_i$. A simple counting argument shows that each Schubert cell $S(a)$ is uniquely characterized by exactly

$$\dim S(a) = d(a) := k(n-k) - \sum_{i=1}^{k}(a_i - i) \tag{3}$$

free parameters and therefore consists of $t^{-d(a)}$ elements. This implies that the number of elements of the Grassmannian is given as

$$|G_k(\mathbf{F}^n)| = t^{-k(n-k)} \sum_{a \in \mathscr{A}_{k,n}} t^{\sum_{i=1}^{k}(a_i - i)}. \tag{4}$$

In particular, we deduce from (2) the identity of power series

$$\sum_{a \in \mathscr{A}_{k,n}} t^{\sum_{i=1}^{k}(a_i - i)} = \prod_{j=1}^{k} \frac{(1 - t^{n-k+j})}{(1 - t^j)}. \tag{5}$$

3 Probability of Reachable Systems

We consider linear control systems of the form

$$x(\tau + 1) = Ax(\tau) + Bu(\tau), \quad x(0) = 0, \quad \tau = 0, 1, 2, \ldots \tag{6}$$

and system matrices $A \in \mathbf{F}^{n \times n}$, $B \in \mathbf{F}^{n \times m}$; see [2] for a summary of linear systems theory, developed over an arbitrary field. A linear system (6) is called reachable, if for any element $\xi \in \mathbf{F}^n$ there exists a finite sequence of inputs $u_0, \ldots, u_{\tau_*} \in \mathbf{F}^m$ such that the sequence of states $x(0), x(1), \ldots, x(\tau_* + 1)$ generated by (6) satisfies $x(\tau_* + 1) = \xi$. We then say that ξ is reached from the initial condition $x(0) = 0$

in $\tau_* + 1$ steps. A simple characterization of reachable linear systems is available using the so-called Kalman test. Explicitly, (6) is reachable if and only if the $n \times nm$-reachability matrix satisfies

$$rank(B, AB, \cdots, A^{n-1}B) = n.$$

Let $\Sigma^{cr}_{n,m}(\mathbf{F})$ denote the set of all such reachable pairs and let $|\Sigma^{cr}_{n,m}(\mathbf{F})|$ denote its cardinality. We are interested in calculating the number of reachable pairs $(A, B) \in \mathbf{F}^{n\times n} \times \mathbf{F}^{n\times m}$, i.e., the cardinality of $|\Sigma^{cr}_{n,m}(\mathbf{F})|$. Equivalently, for the equidistribution on $\mathbf{F}^{n\times(n+m)}$, we want to compute the probability

$$P_{n,m}(t) := \frac{|\Sigma^{cr}_{n,m}(\mathbf{F})|}{|\mathbf{F}^{n\times n} \times \mathbf{F}^{n\times m}|}$$

of a pair $(A, B) \in \mathbf{F}^{n\times n} \times \mathbf{F}^{n\times m}$ to be reachable. Our first theorem generalizes an earlier result by [7] that has been restricted to the case $m \leq 2$.

Theorem 1 *The probability that a pair* $(A, B) \in \mathbf{F}^{n\times n} \times \mathbf{F}^{n\times m}$*,* $n, m \geq 1$*, is reachable is equal to*

$$P_{n,m}(t) = \prod_{j=m}^{n+m-1} (1 - t^j) = 1 - t^m + O(t^{m+1}). \tag{7}$$

In particular, we obtain for $n \geq 2$

$$(1 - t^m)(1 - (n-1)t^{m+1}) \leq P_{n,m}(t) \leq (1 - t^m)(1 - t^{m+1}). \tag{8}$$

Proof Clearly, reachability is invariant under the state space similarity transformations $(A, B) \mapsto (TAT^{-1}, TB)$ with $T \in GL_n(\mathbf{F})$. Thus, $GL_n(\mathbf{F})$ acts on $\Sigma^{cr}_{n,m}(\mathbf{F})$ via similarity. We denote the corresponding orbit space by $\Sigma_{n,m}(\mathbf{F})$. Since (A, B) is reachable, the similarity action is a free action and therefore the map

$$GL_n(\mathbf{F}) \to GL_n(\mathbf{F}) \cdot (A, B), \quad T \mapsto (TAT^{-1}, TB)$$

from the group to the group orbit is injective. This implies that the cardinalities of $\Sigma^{cr}_{n,m}(\mathbf{F})$ and $\Sigma_{n,m}(\mathbf{F})$ are related as

$$|\Sigma^{cr}_{n,m}(\mathbf{F})| = |GL_n(\mathbf{F})| \cdot |\Sigma_{n,m}(\mathbf{F})|.$$

Thus, it amounts to determine the number of $\mathbf{F}$-rational points in the quasi-affine algebraic variety $\Sigma_{n,m}(\mathbf{F})$. This we do following [4], using a cell decomposition of $\Sigma_{n,m}(\mathbf{F})$ that is obtained by fixing the so-called Hermite indices of reachable pairs. This is the main point where our analysis departs from [7], who use the more complicated Kronecker invariants rather the Hermite indices. We refer to [4] and [5] for further details and proofs of the subsequent statements on cell decompositions of $\Sigma_{n,m}(\mathbf{F})$. Specifically, $\Sigma_{n,m}(\mathbf{F})$ admits a disjoint decomposition into finitely many affine spaces

$$\Sigma_{n,m}(\mathbf{F}) = \bigsqcup_{K \in K_{n,m}} Her_K,$$

parameterized by the combinations $K = (K_1, \ldots, K_m)$ of n into m parts. Here, the Hermite cells

$$Her_K = \mathbf{F}^{n(K)}$$

are affine spaces of dimension $n(K) = \sum_{i=1}^{m}(m - i + 1)K_i$. In contrast, the Grassmannian $G_{m-1}(\mathbf{F}^{n+m-1})$ has the cell decomposition

$$G_{m-1}(\mathbf{F}^{n+m-1}) = \bigsqcup_{a \in \mathscr{A}_{m-1,n+m-1}} S(a)$$

into finitely many Schubert cells $S(a) = \mathbf{F}^{d(a)}$, where $d(a)$ is given by (3). Clearly, the map $f : K_{n,m} \longrightarrow \mathscr{A}_{m-1,n+m-1}$

$$(K_1, \ldots, K_m) \mapsto a_K := (K_m + 1, K_m + K_{m-1} + 2, \ldots, K_m + \cdots + K_2 + m - 1)$$

is bijective and therefore defines a bijection $Her_K \mapsto S(a_K)$ of Hermite cells and Schubert cells, respectively. Since

$$nm - \sum_{i=1}^{m}(m - i + 1)K_i = \sum_{i=1}^{m}\left(n - \sum_{j=1}^{i} K_j\right) = \sum_{i=1}^{m-1}\left(\sum_{j=i+1}^{m} K_j\right),$$

we obtain

$$\begin{aligned}\dim S(a_K) &= n(m-1) + \frac{(m-1)m}{2} - \sum_{i=1}^{m-1}\left(\sum_{j=i+1}^{m} K_j + m - j + 1\right)\\ &= n(m-1) - \sum_{i=1}^{m-1}\left(\sum_{j=i+1}^{m} K_j\right) = \dim Her_K - n,\end{aligned}$$

for all $K \in K_{n,m}$. In particular, both spaces $\Sigma_{n,m}(\mathbf{F})$ and $G_{m-1}(\mathbf{F}^{n+m-1})$ admit cell decompositions that are indexed by the combinations of n into m nonnegative parts. Moreover, the mutual dimensions of the Schubert and Hermite cells differ by n, respectively. Thus, although no direct relation between these very different spaces is known, the cardinalities of their $\mathbf{F}$-rational points can be easily compared. Explicitly, we obtain

$$|\Sigma_{n,m}(\mathbf{F})| = \sum_{K \in K_{n,m}} t^{-n(K)} = t^{-n} \sum_{a \in \mathscr{A}_{m-1,n+m-1}} t^{-d(a)} = t^{-n} |G_{m-1}(\mathbf{F}^{n+m-1})|. \tag{9}$$

By (2), the Grassmannian $G_{m-1}(\mathbf{F}^{n+m-1})$ has exactly

$$t^{-n(m-1)} \prod_{j=1}^{n} \frac{(1-t^{m+j-1})}{(1-t^j)}$$

many elements. Thus, the cardinality of $\Sigma^{cr}_{n,m}(\mathbf{F})$ is equal to

$$|\Sigma^{cr}_{n,m}(\mathbf{F})| = |GL_n(\mathbf{F})||\Sigma_{n,m}(\mathbf{F})| = t^{-n(m+n)} \cdot \prod_{j=m}^{n+m-1} (1-t^j),$$

which completes the proof of (7). The bounds (8) are easily deduced from (7). □

We emphasize that the preceding theorem implies that the probability $P_{n,m}(t)$ increases monotonically to 1, whenever the cardinality of the field $\mathbf{F}$ grows unbounded. Moreover, the approximation error is asymptotically given as t^m and therefore decreases exponentially in the number of inputs m.

3.1 Systems with *r*-Dimensional Reachability Subspace

One can extend Theorem 1 in a rather straightforward way to determine the number of pairs (A, B) with r-dimensional reachability subspace. Consider, for any $r = 0, \ldots, n$, the set

$$S^r_{n,m}(\mathbf{F}) := \{(A, B) \in \mathbf{F}^{n\times n} \times \mathbf{F}^{n\times m} \mid rank(B, AB, \cdots, A^{n-1}B) = r\}.$$

In particular, $\Sigma^{cr}_{n,m}(\mathbf{F}) = S^n_{n,m}(\mathbf{F})$. To compute the cardinality of $S^r_{n,m}(\mathbf{F})$ we consider the set S_r of all systems of the form

$$A = \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix}, \quad B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix},$$

where $(A_1, B_1) \in \Sigma^{cr}_{r,m}(\mathbf{F})$ and $A_2 \in \mathbf{F}^{(r\times(n-r))}$ and $A_3 \in \mathbf{F}^{((n-r)\times(n-r))}$ are arbitrary. This space has cardinality

$$|S_r| = t^{-n(n-r)}|\Sigma^{cr}_{r,m}(\mathbf{F})| = t^{-n^2+r(n-m)} \prod_{j=m}^{r+m-1}(1-t^j).$$

Theorem 2 *The cardinality of $S^r_{n,m}(\mathbf{F})$ is equal to*

$$|S^r_{n,m}(\mathbf{F})| = t^{-n^2-rm} \frac{\prod_{j=r+1}^{n}(1-t^j)\prod_{j=m}^{r+m-1}(1-t^j)}{\prod_{j=1}^{n-r}(1-t^j)}. \tag{10}$$

Proof Let $\mathscr{P} \subset GL_n(\mathbf{F})$ denote the parabolic subgroup of all block upper triangular matrices of the form

$$p \in \mathscr{P} = \begin{pmatrix} GL_r(\mathbf{F}) & \mathbf{F}^{r\times(n-r)} \\ 0 & GL_{n-r}(\mathbf{F}) \end{pmatrix},$$

which acts on the product space $GL_n(\mathbf{F}) \times S_r$ via

$$(g, (A, B)) \mapsto (gp^{-1}, (pAp^{-1}, pB)). \tag{11}$$

Let $GL_n(\mathbf{F}) \times_{\mathscr{P}} S_r$ denote the quotient space with respect to this free group action. We then have the well-defined map $\phi : GL_n(\mathbf{F}) \times_{\mathscr{P}} S_r \longrightarrow S^r_{n,m}(\mathbf{F})$, which sends each orbit $[g, (A, B)]$ of (11) to the element (gAg^{-1}, gB). This map is a bijection and induces a $G_r(\mathbf{F}^n)$-bundle on S_r. Therefore, we obtain the equality of cardinalities $|GL_n(\mathbf{F}) \times_{\mathscr{P}} S_r| = |S^r_{n,m}(\mathbf{F})|$. Moreover, the cardinality of the orbit space $GL_n(\mathbf{F}) \times_{\mathscr{P}} S_r$ is equal to

$$\frac{|GL_n(\mathbf{F})||S_r|}{|\mathscr{P}|} = |G_r(\mathbf{F}^n)||S_r|.$$

Using Theorem 1, this implies

$$|S^r_{n,m}(\mathbf{F})| = t^{r^2-n^2} \frac{\prod_{j=r+1}^{n}(1-t^j)}{\prod_{j=1}^{n-r}(1-t^j)} |\Sigma^{cr}_{r,m}(\mathbf{F})| = t^{-n^2-rm}$$

$$\frac{\prod_{j=r+1}^{n}(1-t^j)\prod_{j=m}^{r+m-1}(1-t^j)}{\prod_{j=1}^{n-r}(1-t^j)}.$$

This completes the proof. □

3.2 Parallel Connection of Single-Input Systems

We next compute the probability that a parallel connection of two linear systems is reachable.

Theorem 3 *The probability that the parallel connected system*

$$\begin{aligned} x_1(\tau+1) &= A_1 x_1(\tau) + b_1 u(\tau) \\ x_2(\tau+1) &= A_2 x_2(\tau) + b_2 u(\tau) \end{aligned} \tag{12}$$

with state vectors $x_1 \in \mathbf{F}^{n_1}$ and $x_2 \in \mathbf{F}^{n_2}$ is reachable is equal to

$$(1-t)\prod_{j=1}^{n_1}(1-t^j)\prod_{j=1}^{n_2}(1-t^j). \tag{13}$$

Proof Consider the right coprime factorizations $(zI - A_i)^{-1}b_i = N_i(z)d_i(z)^{-1}$, where $N_i(z) \in \mathbf{F}^{n_i \times 1}[z]$ and $d_i(z) \in \mathbf{F}[z]$ monic with $\deg(N_i) < \deg(d_i) = n_i$. Since (A_i, b_i) is reachable if and only if the entries of N_i are linearly independent over $\mathbf{F}$, there are $|Gl_{n_i}(\mathbf{F})| = t^{-n_i^2}\prod_{j=1}^{n_i}(1-t^j)$ possibilities for N_i. In [1] it has been shown that reachability of (12) is equivalent to reachability of (A_i, b_i) and coprimeness of the scalar polynomials $d_1(z)$ and $d_2(z)$. The number of coprime pairs of monic polynomials $(d_1(z), d_2(z))$ is equal to $t^{-n_1-n_2}(1-t)$; see [3]. Thus, the number of pairs (A_i, b_i) such that (12) is reachable is

$$t^{-n_1-n_2}(1-t)t^{-n_1^2-n_2^2}\prod_{j=1}^{n_1}(1-t^j)\prod_{j=1}^{n_2}(1-t^j),$$

which proves the theorem. □

4 Conclusions

We compare cell decompositions of the moduli space of reachable linear systems with the Grassmannian to derive an explicit formula for the probability that a linear system is reachable. The formula has been extended to count the number of reachable linear systems with r-dimensional reachability subspace, as well as to compute the probability that the parallel connection of two linear single-input systems is reachable. Future research will concern the extension to the parallel connection of finitely many multivariable systems and to general networks of systems.

Acknowledgements This research has been partially supported by the grant DFG 1858/13-1 from the German Research Foundation.

References

1. Fuhrmann, P.A.: On controllability and observability of systems connected in parallel. IEEE Trans. Circuits Syst. **22**, 57 (1975)
2. Fuhrmann, P.A., Helmke, U.: The Mathematics of Networks of Linear Systems. Springer, New York (2015)
3. Garcia-Armas, M., Ghorpade, S.R., Ram, S.: Relatively prime polynomials and nonsingular Hankel matrices over finite fields. J. Comb. Theory Ser. A **118**(3), 819–828 (2011)
4. Helmke, U.: Topology of the moduli space for reachable linear dynamical systems: the complex case. Math. Syst. Theory **19**, 155–187 (1986)
5. Helmke, U.: The Cohomology of Moduli Spaces for Linear Dynamical Systems. Regensburger Mathematische Schriften, vol. 24. Fak. für Mathematik der Univ., Regensburg (1993)
6. Hutchinson, R., Rosenthal, J., Smarandache, R.: Convolutional codes with maximum distance profile. Syst. Control Lett. **54**(1), 53–63 (2005)
7. Kociecky, M., Przyluski, K.M.: On the number of controllable linear systems over a finite field. Linear Algebra Appl. **122–124**, 115–122 (1989)
8. Li, S., Yeung, R., Cai, N.: Linear Network Coding. IEEE Trans. Inf. Theory **49**(2), 371–381 (2003)
9. Milnor, J., Stasheff, J.: Characteristic Classes. Princeton University Press, Princeton (1974)
10. Rosenthal, J., York, E.V.: BCH convolutional codes. IEEE Trans. Inf. Theory **45**, 1833–1844 (1999)
11. Sundaram, S., Hadjicostis, Ch.: Structural controllability and observability of linear systems over finite fields with applications to multi–agent systems. IEEE Trans. Autom. Control **58**(1), 60–73 (2013)

Classification of MDS Codes over Small Alphabets

Janne I. Kokkala, Denis S. Krotov, and Patric R.J. Östergård

Abstract A q-ary code of length n, size M, and minimum distance d is called an $(n, M, d)_q$ code. An $(n, q^k, d)_q$ code with $d = n - k + 1$ is said to be maximum distance separable (MDS). Here we show that every code with parameters $(k + d - 1, q^k, d)_q$ where $k, d \geq 3$ and $q = 5, 7$, is equivalent to a linear code, which implies that the $(6, 5^4, 3)_5$ code and the $(n, 7^{n-2}, 3)_7$ codes for $n = 6, 7, 8$ are unique. We also show that there are 14, 8, 4, and 4 equivalence classes of $(n, 8^{n-2}, 3)_8$ codes for $n = 6, 7, 8, 9$, respectively. This work is continuation of a previous article classifying $(5, q^3, 3)_q$ codes for $q = 5, 7, 8$.

Keywords Classification • MDS codes • Perfect codes • Latin hypercubes

1 Introduction

A q-ary *code* of length n consists of a set of *codewords* that form a subset of the *words* in $\mathbb{Z}_q^n$. The *Hamming distance* between two words is the number of coordinates in which they differ. The *minimum distance* of a code is the smallest distance between any two distinct codewords. A code with minimum distance d is able to detect errors in up to $d-1$ coordinates and correct errors in up to $\lfloor (d-1)/2 \rfloor$ coordinates. A q-ary code of length n, size M, and minimum distance d is called an $(n, M, d)_q$ code.

Two codes, C and C', are said to be *equivalent*, denoted by $C \cong C'$, if there is a permutation of the coordinates and permutations of the coordinate values, separately for each coordinate, that map the codewords of C onto the codewords of C'. These operations form a group G of order $n!(q!)^n$. An element $g \in G$ mapping a code C onto itself is called an automorphism of C.

J.I. Kokkala (✉) • P.R.J. Östergård
Department of Communications and Networking, Aalto University School of Electrical Engineering, P.O. Box 13000, 00076 Aalto, Finland
e-mail: janne.kokkala@aalto.fi; patric.ostergard@aalto.fi

D.S. Krotov
Sobolev Institute of Mathematics, Novosibirsk State University, 630090 Novosibirsk, Russia
e-mail: krotov@math.nsc.ru

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_24

The Singleton bound states for the size of an $(n, M, d)_q$ code that $M \leq q^{n-d+1}$. Codes meeting the Singleton bound are called *maximum distance separable* (MDS); see [11] and [7, Chapter 11]. For codes with $d = 3$, the Hamming bound states that $M \leq q^n/[1 + n(q-1)]$. Codes attaining this bound are called *perfect*.

A common question in coding theory is whether codes with given parameters exist. Further, the corresponding codes can be classified up to equivalence. In this work, we focus mainly on $(n, q^{n-2}, 3)_q$ codes, that is, one-error-correcting MDS codes. The $(q+1, q^{q-1}, 3)_q$ codes are perfect, and they can further be shortened to get a family of $(n, q^{n-2}, 3)_q$ codes for $3 \leq n \leq q+1$. Linear $\big((q^m-1)/(q-1), q^{(q^m-1)/(q-1)-m}, 3\big)_q$ codes, called *Hamming codes*, are perfect and exist for all $m \geq 2$ and prime powers q. For $m = 2$, we get perfect MDS codes.

Unrestricted (that is, either linear or nonlinear) MDS codes are discussed for example in [1]. In particular, it is known that $d \leq q$ for even q and that $d \leq q - 1$ for odd q. The Hamming bound implies that $(n, q^{n-2}, 3)_q$ codes do not exist for $n > q + 1$. For short codes with these parameters, the following is known. The $(3, q, 3)_q$ codes are trivially unique. The $(4, q^2, 3)_q$ codes (for $q \geq 3$) correspond to graeco-latin squares. The nonexistence of graeco-latin squares for $q = 6$ is well-known, and for $q \leq 8$, they have been classified by McKay [8]. The $(5, q^3, 3)_q$ codes (for $q \geq 4$) correspond to graeco-latin cubes. The $(5, 4^3, 3)_4$ code is unique, which was proved by Alderson [2]. The nonexistence of $(5, 6^3, 3)_6$ codes follows from the nonexistence of $(4, 6^2, 3)_6$ codes. The $(5, q^3, 3)_q$ codes for $q = 5, 7, 8$ were recently classified in [5]; there are 1, 1, and 12484 equivalence classes for $q = 5, 7, 8$, respectively. As for the perfect $(q+1, q^{q-1}, 3)_q$ codes, the cases $q = 5$ and $q \geq 7$ have remained open. Quite early, Lindström [6] showed that there are more than one equivalence class of $(q+1, q^{q-1}, 3)_q$ codes when $q > 8$ is a proper prime power. Heden [4] showed that for prime q, $(q+1, q^{q-1}, 3)_q$ codes of certain type are equivalent to Hamming codes.

This work consists of two parts. We prove that every code with parameters $(k + d - 1, 7^k, d)_7$ or $(k + d - 1, 5^k, d)_5$, where $k, d \geq 3$, is equivalent to a linear code, using the fact that the unique graeco-latin cubes of orders 5 and 7 correspond to linear codes. This implies that the $(6, 5^4, 3)_5$ MDS code and the $(n, 7^{n-2}, 3)_7$ MDS codes for $n = 6, 7, 8$ are unique.

In the second part, we present an approach for generating the $(n, q^{n-2}, 3)_q$ codes starting from the $(n', q^{n'-2}, 3)_q$ codes for $n' < n$. We apply this approach to the case $q = 8$ by starting from the $(4, 8^2, 3)_8$ and $(5, 8^3, 3)_8$ codes, and classify the $(n, 8^{n-2}, 3)_8$ codes for $6 \leq n \leq 9$. There are 14, 8, 4, and 4 equivalence classes of $(n, 8^{n-2}, 3)_q$ codes for $n = 6, 7, 8, 9$, respectively. For one-error-correcting codes, the previously known cases along with the cases considered in this work are shown in Table 1.

Table 1 Kown numbers of equivalence classes of $(n, q^{n-2}, 3)_q$ codes. The cases marked with $*$ have been settled in the present work

$n \setminus q$	2	3	4	5	6	7	8	≥9 proper prime power	≥11 prime
3	1	1	1	1	1	1	1	1	1
4		1	1	1	0	7	2165		
5			1	1	0	1	12484		
6				*1	0	*1	*14		
7					0	*1	*8		
8						*1	*4		
9							*4		
$q+1$ (≥10)								≥2	≥1

2 Preliminaries

In this section, we consider basic properties of MDS codes and give definitions of latin hypercubes which are equivalent to certain MDS codes.

2.1 MDS Codes and Latin Hypercubes

Definition 1 A *latin hypercube* of order q and dimension n is an $q \times q \times \cdots \times q$ (n times) array of elements from $\mathbb{Z}_q$ such that if $n-1$ coordinates are fixed, each symbol occurs exactly once. For $n = 1, 2, 3$, they are *permutations*, *latin squares* and *latin cubes*, respectively.

Definition 2 Two latin squares are *orthogonal* if each pair in $\mathbb{Z}_q \times \mathbb{Z}_q$ occurs exactly once when the squares are superimposed. Two latin hypercubes are orthogonal if when the hypercubes are superimposed, each $q \times q$ subarray is a superimposed pair of orthogonal latin squares.

An n-dimensional latin hypercube corresponds to an $(n+1, q^n, 2)_q$ MDS code: for example, let the word $(x_1, \ldots, x_{n+1})$ be a codeword iff x_{n+1} occurs at position $(x_1, \ldots, x_n)$ in the latin hypercube. Similarly, a pair of orthogonal n-dimensional latin hypercubes corresponds to an $(n+2, q^n, 3)_q$ MDS code: let the word $(x_1, \ldots, x_{n+2})$ be a codeword iff x_{n+1} and x_{n+2} occur at position $(x_1, \ldots, x_n)$ in the first and second latin hypercube, respectively.

Definition 3 A latin hypercube f of prime order q is called *linear* if

$$\alpha_0(f(x_1, \ldots, x_n)) = \alpha_1(x_1) + \cdots + \alpha_n(x_n) \tag{1}$$

for some permutations $\alpha_0, \alpha_1, \ldots, \alpha_n$ of $\mathbb{Z}_q$.

Definition 4 A k-tuple $(f_1, \dots, f_k)$ of latin hypercubes of prime order q is called *linear* if there are permutations $\alpha_1, \dots, \alpha_n, \beta_1, \dots \beta_k$ of $\mathbb{Z}_q$ such that

$$\beta_i(f_i(x_1, \dots, x_n)) = a_{i,1}\alpha_1(x_1) + \cdots + a_{i,n}\alpha_n(x_n) + a_{i,0} \qquad i = 1, \dots, k$$

for some coefficients $a_{i,j}$, $j \in \{0, 1, \dots, n\}, i \in \{1, \dots, k\}$ (we may always assume w.l.o.g. that $a_{1,1} = \cdots = a_{1,n} = 1$ and $a_{1,0} = 0$).

A latin hypercube is linear iff the corresponding MDS code is equivalent to a linear code, and an orthogonal pair of latin hypercubes is linear iff the corresponding MDS code is equivalent to a linear code.

2.2 Properties of MDS Codes and Some Definitions

Definition 5 For an $(n, M, d)_q$ code C, let $s(C, \alpha, v)$ be the $(n-1, M', d)_q$ code obtained by removing a coordinate α and maintaining the codewords in C that have the symbol v at that coordinate. This operation is called *shortening*.

Definition 6 For an $(n, M, d)_q$ code C, let $e(C, \alpha, v)$ be the $(n+1, M, d)_q$ code obtained by adding a new coordinate with symbol v at that coordinate to the codewords in C, such that $s(e(C, \alpha, v), \alpha, v) = C$. This operation is called *extending*.

Consider an $(n, q^{n-2}, 3)_q$ MDS code C. For a set of $k = n-2$ coordinates, each k-tuple of elements from $\mathbb{Z}_q$ occurs exactly once in the given coordinates in C. We observe that $s(C, \alpha, v)$ is an $(n-1, q^{n-3}, 3)_q$ MDS code. Further, C can be expressed as a union of extended $(n-1, q^{n-3}, 3)_q$ codes, as $C = \bigcup_{v \in \mathbb{Z}_q} e(s(C, \alpha, v), \alpha, v)$ for any α.

3 Theoretical Results

Definition 7 A *rectangle* of directions i, j $(i \neq j)$, is a quadruple $(a = (a_1, \dots, a_n), b = (b_1, \dots, b_n), c = (c_1, \dots, c_n), d = (d_1, \dots, d_n))$ of elements of $\mathbb{Z}_q^n$ such that $a_i = b_i, c_i = d_i, b_j = c_j, d_j = a_j$ and $a_k = b_k = c_k = d_k$ for all $k \in \{1, \dots, n\} \setminus \{i, j\}$.

Lemma 8 *For every linear latin hypercube f of prime order q and dimension $n \geq 2$, there is a unique function* $\mathrm{Rect}_f : \mathbb{Z}_q^3 \to \mathbb{Z}_q$ *such that for every rectangle (a, b, c, d) it holds* $f(a) = \mathrm{Rect}_f(f(b), f(c), f(d))$.

Proof Using the notation in (1), we see that $f(a) = \alpha_0^{-1}(\alpha_0(f(b)) - \alpha_0(f(c)) + \alpha_0(f(d)))$. □

Lemma 9 *A linear hypercube f of order q can be uniquely reconstructed from the function* Rect_f *and the values $f(x_1,\dots,x_n)$ where only one of x_i is nonzero.*

Proof The value $f(x_1,\dots,x_n)$ can be determined recursively by reducing the number of nonzero values among $x_1,\dots,x_n$ using

$$\begin{aligned} f(x_1,\dots,x_i,\dots,x_j,\dots,x_n) = \mathrm{Rect}_f(\ & f(x_1,\dots,0,\dots,x_j,\dots,x_n),\\ & f(x_1,\dots,0,\dots,0,\dots,x_n),\\ & f(x_1,\dots,x_i,\dots,0,\dots,x_n)). \end{aligned}$$

□

Lemma 10 *Let f be a latin hypercube of dimension $n \geq 4$, for which any latin hypercube obtained from f by fixing one argument is linear. Then f is linear.*

Proof The latin hypercube t defined as $t(x_1,\dots,x_{n-1}) = f(x_1,\dots,x_{n-1},0)$ is linear. We assume w.l.o.g. that $t(x_1,\dots,x_{n-1}) = x_1 + \dots + x_{n-1}$ and $f(0,\dots,0,i) = i$ for all i. Then $\mathrm{Rect}_t(a,b,c) = a - b + c$.

For $j \in \{1,\dots,n-1\}$, let r_j be the linear $(n-1)$-dimensional hypercube obtained from f by fixing the jth argument to be zero. Now r_j and t share a common $(n-2)$-dimensional hypercube l, which implies that $\mathrm{Rect}_t = \mathrm{Rect}_l = \mathrm{Rect}_{r_j}$. Further, because $r_j(i,0,\dots,0) = r_j(0,i,0,\dots,0) = \cdots = r_j(0,\dots,0,i) = i$ for all $i \in \mathbb{Z}_q$, Lemma 9 implies that $r_j(x_1,\dots,x_{n-1}) = x_1 + \cdots + x_{n-1}$. In other words, $f(x_1,\dots,x_n) = x_1 + \cdots + x_n$ if at least one of x_j is zero.

Finally, for $i \in \mathbb{Z}_q$, consider $s_i(x_2,\dots,x_n) = f(i,x_2,\dots,x_n)$. Again, s_i is linear and shares a common $(n-2)$-dimensional hypercube with t, so $\mathrm{Rect}_t = \mathrm{Rect}_{s_i}$. On the other hand, $s_i(j,0,\dots,0) = s_i(0,j,0,\dots,0) = \cdots = s_i(0,\dots,0,j) = i + j$. Thus, s_i is uniquely determined, and $s_i(x_2,\dots,x_n) = i + x_2 + \cdots + x_n$. Thus $f(x_1,\dots,x_n) = x_1 + \cdots + x_n$. □

Lemma 11 *Let q be a prime, let $c \in \mathbb{Z}_q$, let $\alpha,\beta \in \mathbb{Z}_q \setminus \{0\}$, and let γ_1, γ_2 and γ_3 be permutations of $\mathbb{Z}_q$ such that $\gamma_1(x) + \gamma_2(y) + \gamma_3(z) = c$ iff $x + \alpha^{-1}y + \beta^{-1}z = 0$. Then γ_i is an affine transformation of $\mathbb{Z}_q$, that is, $\gamma_i(x) = a_i x + b_i$ for some $a_i, b_i \in \mathbb{Z}_q$, for all $i = 1,2,3$.*

Proof For all x, we find that $\gamma_1(x) - \gamma_1(x-1) = [c - \gamma_2(-\alpha x) - \gamma_3(0)] - [c - \gamma_2(-\alpha x) - \gamma_3(\beta)] = \gamma_3(\beta) - \gamma_3(0)$. Thus γ_1 is affine. Similarly, so are γ_2 and γ_3. □

Lemma 12 *Let $(f_1,\dots,f_k)$ be a k-tuple of latin hypercubes of prime order q and dimension $n \geq 3$ such that fixing the values of any $n-3$ variables in any two of $f_1,\dots,f_k$ always results in a linear pair of latin cubes. Then $(f_1,\dots,f_k)$ is linear.*

Proof By induction and Lemma 10, $f_1, \dots, f_k$ are linear latin hypercubes. W.l.o.g. we may assume that

$$f_1(x_1, \dots, x_n) = x_1 + \cdots + x_n,$$
$$f_i(x_1, \dots, x_n) = \gamma_{i,1}(x_1) + \cdots + \gamma_{i,n}(x_n),$$

for all $i = 2, \dots, k$.

Consider some $i \in \{2, \dots, k\}$. Fixing the last $n-3$ arguments of f_1 and f_i by zeros, we get a linear pair of latin cubes (g, h). For some permutations α_1, α_2, α_3, β_1, β_2,

$$\begin{aligned} g(x_1, x_2, x_3) &= x_1 + x_2 + x_3 = \beta_1^{-1}(\alpha_1(x_1) + \alpha_2(x_2) + \alpha_3(x_3)), \\ h(x_1, x_2, x_3) &= \gamma_{i,1}(x_1) + \gamma_{i,2}(x_2) + \gamma_{i,3}(x_3) \\ &= \beta_2^{-1}(b_1\alpha_1(x_1) + b_2\alpha_2(x_2) + b_3\alpha_3(x_3)). \end{aligned}$$

By Lemma 11 the permutations α_1, α_2, α_3, are affine transformations of $\mathbb{Z}_q$, and thus also $\gamma_{i,1}$, $\gamma_{i,2}$, and $\gamma_{i,3}$ are affine transformations of $\mathbb{Z}_q$. Similarly, we establish that $\gamma_{i,j}$ is affine for every i from 1 to k and every j from 1 to n. So, $(f_1, \dots, f_k)$ is a linear k-tuple of latin hypercubes by the definition. □

Theorem 13 *Every code with parameters* $(k+d-1, 7^k, d)_7$ *or* $(k+d-1, 5^k, d)_5$*, where* $k, d \geq 3$*, is equivalent to a linear code.*

Proof Every code with the considered parameters is the set of solutions of a system

$$f_i(x_1, \dots, x_k) = x_{k+i}, \qquad \text{for } i = 1, \dots, d-1,$$

where $f_1, \dots, f_{d-1}$ are latin hypercubes, every two being orthogonal. In particular, fixing any $k-3$ arguments of both f_i and f_j, $i \neq j$, always results in a pair of orthogonal latin cubes. By the previous computational results [5], every such pair of order 5 or 7 is linear. By Lemma 12, the $(d-1)$-tuple of latin hypercubes $(f_1, \dots, f_{d-1})$ is linear, which means that the code is equivalent to a linear code.

□

Corollary 14 (MDS conjecture, $q = 7$**)** *Every* $(n, 7^k, d)_7$ *MDS code,* $k \geq 2$*,* $d \geq 3$*, satisfies* $n \leq 8$.

Proof For $k \geq 3$, such codes are linear, and the MDS conjecture is true for linear codes when q is prime [3]. For $k = 2$, the nonexistence of an $(n, q^k, n-k+1)_q$ MDS code for $n > q+1$ is well-known [11]. □

Corollary 15 *The* $(6, 5^4, 3)_5$ *MDS code and the* $(n, 7^{n-2}, 3)_7$ *MDS codes for* $n = 6, 7, 8$ *are unique.*

Proof The linear perfect codes are the Hamming codes. Consider a linear $(n, 7^{n-2}, 3)_7$ code for $n = 6, 7$. After multiplication of columns, the parity check matrix is of the form $\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & a_1 & \cdots & a_{n-1} \end{pmatrix}$, where a_i are distinct. The set $\{a_i\}_i$ can be mapped to $\{0, 1, \ldots, n-2\}$ by an affine transformation $x \mapsto bx+c$. Multiplying the second row by b, adding the first row multiplied by c to the second row, multiplying the first column by b^{-1}, and permuting the columns yields $\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & n-2 \end{pmatrix}$. Thus all such linear codes are equivalent. □

4 Algorithm for Exhaustive Generation

We observe that the codes $s(C, \alpha_1, v_1)$ and $s(C, \alpha_2, v_2)$ with $\alpha_1 \neq \alpha_2$ share a common shortened $(n-2, q^{n-4}, 3)_q$ code: for $\alpha_1 > \alpha_2$, the codes $s(s(C, \alpha_1, v_1), \alpha_2, v_2)$ and $s(s(C, \alpha_2, v_2), \alpha_1 - 1, v_1)$ are equal. This fact is central to our algorithm, as we construct the $(n, q^{n-2}, 3)_q$ codes by considering their shortened codes.

To reduce the number of partial and full codes obtained during the search, we generate only codes that are of certain form. Given an ordered set $\hat{S}^{n-1} = (\hat{C}_i^{n-1})_i$ of equivalence class representatives of $(n-1, q^{n-3}, 3)_q$ codes, we construct only $(n, q^{n-2}, 3)_q$ codes C for which the following conditions apply:

- $s(C, 1, 0) = \hat{C}_i^{n-1}$ for some i
- If $s(C, \alpha, v) \cong \hat{C}_j^{n-1}$, then $i \leq j$.

Each $(n, q^{n-2}, 3)_q$ code C is equivalent to a code that satisfies these properties. For ease of notation, we define a function ϕ that maps each $(n-1, q^{n-3}, 3)_q$ code C to a number such that $C \cong \hat{C}_{\phi(C)}^{n-1}$.

The algorithm makes repeated use of the following procedure which, given an index i and a coordinate-value pair (α, v), finds all possible codes C for which $e(\hat{C}_i^{n-1}, 1, 0) \cup e(C, \alpha + 1, v)$ has minimum distance 3 and $\phi(C) \geq i$, up to a permutation of values $1, 2, \ldots, q-1$ in the first coordinate. We loop over all $j = i, i+1, \ldots, |\hat{S}^{n-1}|$ and all coordinate-value pairs (α', v') such that $s(\hat{C}_j^{n-1}, \alpha', v') \cong s(\hat{C}_i^{n-1}, \alpha, v)$. At each step, we find any isomorphism h that maps the coordinate-value pair (α', v') to $(1, 0)$ such that $s(h\hat{C}_j^{n-1}, 1, 0) = s(\hat{C}_i^{n-1}, \alpha, v)$, and loop over all automorphisms g of $s(\hat{C}_i^{n-1}, \alpha, v)$. We define $\tilde{g}$ to be the isomorphism that keeps the first coordinate intact and applies g to the last $n-2$ coordinates. Finally, we consider $C = \tilde{g}h\hat{C}_j^{n-1}$ and report it if the minimum distance of $e(\hat{C}_i^{n-1}, 1, 0) \cup e(C, \alpha + 1, v)$ is 3.

We generate the $(n, q^{n-2}, 3)_q$ codes in two phases. In the first phase, we find codes of the form $e(\hat{C}_i^{n-1}, 1, 0) \cup e(C', 2, 0)$ that have minimum distance 3 and $\phi(C') \geq i$. These codes are possible subsets of the $(n, q^{n-2}, 3)_q$ codes, and form the seeds for the next phase.

In the second phase, we start from a seed $C = e(\hat{C}_i^{n-1}, 1, 0) \cup e(C', 2, 0)$ and augment it to full codes in all possible ways. We do this by finding sets of

Table 2 The number of inequivalent seeds obtained during the search, the number of inequivalent $(n, 8^{n-2}, 3)_8$ codes, and the CPU time used

n	# of seeds	# of inequivalent codes	CPU time in hours
6	107	14	15
7	9	8	49
8	6	4	340
9	4	4	1,516

$(n-1, q^{n-3}, 3)_q$ codes $\{C''_v\}_{v\in\mathbb{Z}_q}$ for which $\phi(C''_v) \geq i$ and $\bigcup_{v\in\mathbb{Z}_q} e(C''_v, 3, v)$ is an $(n, q^{n-2}, 3)_q$ code that contains all the codewords in the seed C. For fixed v, finding all possible choices of C''_v such that $e(\hat{C}_i^{n-1}, 1, 0) \cup e(C''_v, 3, v)$ has minimum distance 3 can be done with the procedure presented above. The additional restriction that $s(C''_v, 2, 0) = s(C', 2, v)$ either rejects the code immediately or yields a unique permutation of the values in the first coordinate of C''_v. The restriction that $e(C', 2, 0) \cup e(C''_v, 3, v)$ has minimum distance 3 can also be used to reject some choices of C''_v. Finally, we loop over all possible sets $\{C''_v\}_{v\in\mathbb{Z}_q}$ and report $\bigcup_{v\in\mathbb{Z}_q} e(C''_v, 3, v)$ if it has minimum distance 3 and all of its subcodes D have $\phi(D) \geq i$.

We perform isomorph rejection on the seeds and the full codes. To determine whether two codes are equivalent, we transfer the codes to graphs as in, for example, [5], and use *nauty* [9] to solve the graph isomorphism problem. The software also produces the automorphism group of a graph, so it can be used to find all automorphisms of a code. For large graphs, we use the sparse mode of *nauty* with the random Schreier method enabled.

The search was carried out for $q = 8$ for $n = 6, 7, 8, 9$, starting from $(5, 8^3, 3)_8$ codes. The numbers of inequivalent seeds and the numbers of inequivalent $(n, 8^{n-2}, 3)_8$ codes obtained during the search are shown in Table 2 along with the CPU time used for the search. The times refer to a single core of a Intel Xeon E5-2665 CPU. The numbers of seeds obtained depend on the exact representatives $\hat{C}_i^{n-1}$ used, so they may differ in repeated studies. Of the four perfect codes, one is equivalent to the Hamming code and three are nonlinear. We find that the nonlinear codes cannot be obtained by any known constructions, including those in [10].

References

1. Alderson, T.L.: Extending MDS codes. Ann. Comb. **9**, 125–135 (2005)
2. Alderson, T.L.: (6, 3)-MDS codes over an alphabet of size 4. Des. Codes Cryptogr. **38**, 31–40 (2006)
3. Ball, S.: On sets of vectors of a finite vector space in which every subset of basis size is a basis. J. Eur. Math. Soc. **14**, 733–748 (2012)
4. Heden, O.: On perfect p-ary codes of length $p + 1$. Des. Codes Cryptogr. **46**, 45–56 (2008)

5. Kokkala, J.I., Östergård, P.R.J.: Classification of Graeco-Latin cubes. J. Comb. Des. (2014). doi:10.1002/jcd.21400
6. Lindström, B.: On group and nongroup perfect codes in q symbols. Math. Scand. **25**, 149–158 (1969)
7. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
8. McKay, B.: Latin squares. http://cs.anu.edu.au/~bdm/data/latin.html (2011). Accessed 2 May 2014
9. McKay, B.D., Piperno, A.: Practical graph isomorphism, II. J. Symb. Comput. **60**, 94–112 (2014)
10. Phelps, K., Rifà, J., Villanueva, M.: Kernels and p-kernels of p^r-ary 1-perfect codes. Des. Codes Cryptogr. **37**, 243–261 (2005)
11. Singleton, R.C.: Maximum distance q-nary codes. IEEE Trans. Inf. Theory **10**, 116–118 (1964)

On the Automorphism Groups of the $\mathbb{Z}_2\mathbb{Z}_4$-Linear Hadamard Codes and Their Classification

Denis S. Krotov and Mercè Villanueva

Abstract It is known that there are exactly $\lfloor \frac{t-1}{2} \rfloor$ and $\lfloor \frac{t}{2} \rfloor$ nonequivalent $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes of length 2^t, with $\alpha = 0$ and $\alpha \neq 0$, respectively, for all $t \geq 3$. In this paper, it is shown that each $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code with $\alpha = 0$ is equivalent to a $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code with $\alpha \neq 0$, so there are only $\lfloor \frac{t}{2} \rfloor$ nonequivalent $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes of length 2^t. Moreover, the orders of the permutation automorphism groups of the $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes are given.

Keywords $\mathbb{Z}_2\mathbb{Z}_4$-linear codes • Additive codes • Hadamard codes • Automorphism group

1 Introduction

Let $\mathbb{Z}_2$ and $\mathbb{Z}_4$ be the rings of integers modulo 2 and modulo 4, respectively. Let $\mathbb{Z}_2^n$ be the set of all binary vectors of length n and let $\mathbb{Z}_4^n$ be the set of all quaternary vectors of length n. For a vector $x = (x_1, \ldots, x_n) \in \mathbb{Z}_2^n$ and a set $I \subseteq \{1, \ldots, n\}$, we denote by $x|_I$ the vector x restricted to the coordinates in I.

Any nonempty subset C of $\mathbb{Z}_2^n$ is a binary code and a subgroup of $\mathbb{Z}_2^n$ is called a *binary linear code*. Similarly, any nonempty subset $\mathscr{C}$ of $\mathbb{Z}_4^n$ is a quaternary code and a subgroup of $\mathbb{Z}_4^n$ is called a *quaternary linear code*. Let $\mathscr{C}$ be a quaternary linear code. Since $\mathscr{C}$ is a subgroup of $\mathbb{Z}_4^n$, it is isomorphic to an Abelian group $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$, and we say that $\mathscr{C}$ is of type $2^\gamma 4^\delta$ as a group. Quaternary codes can be seen as binary codes under the usual Gray map defined as $\varphi(0) = (0,0)$, $\varphi(1) = (0,1)$, $\varphi(2) = (1,1)$, $\varphi(3) = (1,0)$ in each coordinate. If $\mathscr{C}$ is a quaternary linear code, then the binary code $C = \varphi(\mathscr{C})$ is called a $\mathbb{Z}_4$*-linear code*.

D.S. Krotov (✉)
Sobolev Institute of Mathematics, Novosibirsk State University, 630090 Novosibirsk, Russia
e-mail: krotov@math.nsc.ru

M. Villanueva
Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Barcelona, Spain
e-mail: merce.villanueva@uab.cat

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_25

Additive codes were first defined by Delsarte in 1973 as subgroups of the underlying Abelian group in a translation association scheme [7, 8]. In the special case of a binary Hamming scheme, that is, when the underlying Abelian group is of order 2^n, the additive codes coincide with the codes that are subgroups of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$. In order to distinguish them from additive codes over finite fields [3], they are called $\mathbb{Z}_2\mathbb{Z}_4$-*additive codes* [4]. Since $\mathbb{Z}_2\mathbb{Z}_4$-additive codes are subgroups of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$, they can be seen as a generalization of binary (when $\beta = 0$) and quaternary (when $\alpha = 0$) linear codes. As for quaternary linear codes, $\mathbb{Z}_2\mathbb{Z}_4$-additive codes can also be seen as binary codes by considering the extension of the usual Gray map: $\Phi : \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta \longrightarrow \mathbb{Z}_2^n$, where $n = \alpha + 2\beta$, given by

$$\Phi(x, y) = (x, \varphi(y_1), \ldots, \varphi(y_\beta)) \\ \forall x \in \mathbb{Z}_2^\alpha,\ \forall y = (y_1, \ldots, y_\beta) \in \mathbb{Z}_4^\beta.$$

If $\mathscr{C}$ is a $\mathbb{Z}_2\mathbb{Z}_4$-additive code, $C = \Phi(\mathscr{C})$ is called a $\mathbb{Z}_2\mathbb{Z}_4$-*linear code*. Moreover, a $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathscr{C}$ is also isomorphic to an Abelian group $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$, and we say that $\mathscr{C}$ (or equivalently the corresponding $\mathbb{Z}_2\mathbb{Z}_4$-linear code $C = \Phi(\mathscr{C})$) is of type $(\alpha, \beta; \gamma, \delta)$.

Let S_n be the symmetric group of permutations on the set $\{1, \ldots, n\}$, and let $\mathrm{id} \in S_n$ be the identity permutation. The group operation in S_n is the function composition, denoted by $\circ$. The composition $\sigma_1 \circ \sigma_2$ maps any element x to $\sigma_1(\sigma_2(x))$. A $\sigma \in S_n$ acts linearly on words of $\mathbb{Z}_2^n$ or $\mathbb{Z}_4^n$ by permuting the coordinates, $\sigma((c_1, \ldots, c_n)) = (c_{\sigma^{-1}(1)}, \ldots, c_{\sigma^{-1}(n)})$.

Let $\mathscr{C}$ be a $\mathbb{Z}_2\mathbb{Z}_4$-additive code of type $(\alpha, \beta; \gamma, \delta)$. We can assign a permutation $\pi_x \in S_n$ to each codeword $x = (x'_1, \ldots, x'_\alpha, x_1, \ldots, x_{2\beta}) \in C = \Phi(\mathscr{C})$, such that $\pi_x = \pi_{12} \circ \pi_{34} \circ \cdots \circ \pi_{2\beta-1\,2\beta}$, where

$$\pi_{ij} = \begin{cases} \mathrm{id} & \text{if } (x_i, x_j) = (0, 0) \text{ or } (1, 1) \\ (i, j) & \text{otherwise.} \end{cases}$$

Given two codewords of C, $x = (x', x_1, \ldots, x_{2\beta})$ and $y = (y', y_1, \ldots, y_{2\beta})$, define $x \star y = x + \pi_x(y)$. Then, we have that $(C, \star)$ is an Abelian group [22], which is isomorphic to $(\mathscr{C}, +)$ since

$$\begin{aligned} x \star y &= (x' + y',\ \varphi(\varphi^{-1}(x_1, x_2) + \varphi^{-1}(y_1, y_2)), \\ &\qquad \ldots,\ \varphi(\varphi^{-1}(x_{2\beta-1}, x_{2\beta}) + \varphi^{-1}(y_{2\beta-1}, y_{2\beta}))) \\ &= \Phi(\Phi^{-1}(x) + \Phi^{-1}(y)). \end{aligned}$$

There are $\mathbb{Z}_2\mathbb{Z}_4$-linear codes in several important classes of binary codes. For example, $\mathbb{Z}_2\mathbb{Z}_4$-linear perfect single error-correcting codes (or 1-perfect codes) are found in [22] and fully characterized in [6]. Also, in subsequent papers [5, 13, 14, 19, 20], $\mathbb{Z}_2\mathbb{Z}_4$-linear extended perfect and Hadamard codes are studied and classified independently for $\alpha = 0$ and $\alpha \neq 0$. Finally, in [17, 21, 23], $\mathbb{Z}_2\mathbb{Z}_4$-linear

Reed-Muller codes are also studied. Note that $\mathbb{Z}_2\mathbb{Z}_4$-linear codes have allowed to classify more binary nonlinear codes, giving them a structure as $\mathbb{Z}_2\mathbb{Z}_4$-additive codes.

A *(binary) Hadamard code* of length n is a binary code with $2n$ codewords and minimum distance $n/2$ [16]. The $\mathbb{Z}_2\mathbb{Z}_4$-additive codes such that, under the Gray map, give a Hadamard code are called $\mathbb{Z}_2\mathbb{Z}_4$*-additive Hadamard codes* and the corresponding $\mathbb{Z}_2\mathbb{Z}_4$-linear codes are called $\mathbb{Z}_2\mathbb{Z}_4$*-linear Hadamard codes*, or just $\mathbb{Z}_4$*-linear Hadamard codes* when $\alpha = 0$. The classification of $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes is given by the following results. For any integer $t \geq 3$ and each $\delta \in \{1, \dots, \lfloor (t+1)/2 \rfloor\}$, there is a unique (up to equivalence) $\mathbb{Z}_4$-linear Hadamard code of type $(0, 2^{t-1}; t+1-2\delta, \delta)$, and all these codes are pairwise nonequivalent, except for $\delta = 1$ and $\delta = 2$, where the codes are equivalent to the linear Hadamard code, that is, the dual of the extended Hamming code [14]. Therefore, the number of nonequivalent $\mathbb{Z}_4$-linear Hadamard codes of length 2^t is $\lfloor \frac{t-1}{2} \rfloor$ for all $t \geq 3$. On the other hand, for any integer $t \geq 3$ and each $\delta \in \{0, \dots, \lfloor t/2 \rfloor\}$, there is a unique (up to equivalence) $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code of type $(2^{t-\delta}, 2^{t-1} - 2^{t-\delta-1}; t+1-2\delta, \delta)$. All these codes are pairwise nonequivalent, except for $\delta = 0$ and $\delta = 1$, where the codes are equivalent to the linear Hadamard code [5]. Therefore, the number of nonequivalent $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes of length 2^t with $\alpha \neq 0$ is $\lfloor t/2 \rfloor$ for all $t \geq 3$.

Two structural properties of binary codes are the rank and the dimension of the kernel. The *rank* of a code C, denoted by r, is simply the dimension of the linear span, $\langle C \rangle$, of C. The *kernel* of a code C is defined as $\mathrm{Ker}(C) = \{x \in \mathbb{Z}_2^n : x + C = C\}$ [2]. If the all-zero vector belongs to C, $\mathrm{Ker}(C)$ is a linear subcode of C. We denote by k the dimension of $\mathrm{Ker}(C)$. In general, C can be written as the union of cosets of $\mathrm{Ker}(C)$, and $\mathrm{Ker}(C)$ is the largest linear code for which this is true [2]. The $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes can also be classified using either the rank or the dimension of the kernel, as it is proven in [14, 20], where these parameters are computed.

Two $\mathbb{Z}_2\mathbb{Z}_4$-additive codes $\mathscr{C}_1$ and $\mathscr{C}_2$ both of type $(\alpha, \beta; \gamma, \delta)$ are said to be *monomially equivalent*, if one can be obtained from the other by permuting the coordinates and (if necessary) changing the signs of certain $\mathbb{Z}_4$ coordinates. Two $\mathbb{Z}_2\mathbb{Z}_4$-additive or $\mathbb{Z}_2\mathbb{Z}_4$-linear codes are said to be *permutation equivalent* if they differ only by a permutation of coordinates. The *monomial automorphism group* of a $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathscr{C}$, denoted by $\mathrm{MAut}(\mathscr{C})$, is the group generated by all permutations and sign-changes of the $\mathbb{Z}_4$ coordinates that preserves the set of codewords of $\mathscr{C}$, while the *permutation automorphism group* of $\mathscr{C}$ or $C = \Phi(\mathscr{C})$, denoted by $\mathrm{PAut}(\mathscr{C})$ or $\mathrm{PAut}(C)$, respectively, is the group generated by all permutations that preserves the set of codewords [12].

The permutation automorphism group of a code is also an invariant, so it can help in the classification of some families of codes. Moreover, the automorphism group can also be used in decoding algorithms and to describe some other properties like the weight distribution. The permutation automorphism group of $\mathbb{Z}_2\mathbb{Z}_4$-linear (extended) 1-perfect codes has been studied in [15, 19]. The permutation automorphism group of (nonlinear) binary 1-perfect codes has also been studied before, obtaining some partial results [1, 9–11]. Finally, the permutation automorphism group of $\mathbb{Z}_2\mathbb{Z}_4$-additive Hadamard codes has been studied in [18].

2 Classification of $\mathbb{Z}_2\mathbb{Z}_4$-Linear Hadamard Codes

In [14] and [5], $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes are classified independently for $\alpha = 0$ and $\alpha \neq 0$. In this section, we show that each $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code with $\alpha = 0$ is equivalent to a $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code with $\alpha \neq 0$, so there are only $\lfloor \frac{t}{2} \rfloor$ nonequivalent $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes of length 2^t.

We say that a function f from $\mathbb{Z}_2^i \times \mathbb{Z}_4^j$ to $\mathbb{Z}_2^s \times \mathbb{Z}_4^t$ is *affine* if $f(\bar{0}) - f(x) - f(y) + f(x+y) = \bar{0}$ for every x and y from $\mathbb{Z}_2^i \times \mathbb{Z}_4^j$ (here and in what follows, $\bar{0}$ denotes the all-zero vector). Equivalently, $f(\cdot) - f(\bar{0})$ is a linear function, i.e., a group homomorphism. Let $\mathscr{B}$ be the set of all affine functions from $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^{\ddot{\delta}}$ to $\mathbb{Z}_4$. These $\mathbb{Z}_4$-valued functions on $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^{\ddot{\delta}}$ can be considered as words of length $2^{\gamma+2\ddot{\delta}}$ over $\mathbb{Z}_4$. Denote $D_{\gamma,\ddot{\delta}} = \{x : \mathbb{Z}_2^\gamma \times \mathbb{Z}_4^{\ddot{\delta}} \to \mathbb{Z}_2^2 \,:\, x(\cdot) = \varphi(g(\cdot))$ for some $g \in \mathscr{B}\}$.

Lemma 1 *$D_{\gamma,\ddot{\delta}}$ is a $\mathbb{Z}_4$-linear Hadamard code of length $n = 2^{\gamma+2\ddot{\delta}+1}$ and type $(0, n/2; \gamma, \delta)$, where $\delta = \ddot{\delta} + 1$.*

Proof There are $4 \cdot 2^\gamma \cdot 4^{\ddot{\delta}} = 2n$ affine functions in $\mathscr{B}$. The set $\mathscr{B}$ is closed under the addition over $\mathbb{Z}_4$; so after applying the Gray map, $D_{\gamma,\ddot{\delta}}$ is a $\mathbb{Z}_4$-linear code of length $2^\gamma \cdot 4^{\ddot{\delta}} \cdot 2 = n$. Clearly, the minimum Hamming distance is $n/2$. □

Define the function $\varphi^+ : \mathbb{Z}_4 \to \{0,1\}$ by $\varphi^+(0) = \varphi^+(3) = 0$, $\varphi^+(1) = \varphi^+(2) = 1$. Again, the $\mathbb{Z}_2$-valued or $\mathbb{Z}_4$-valued functions on $\mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^\delta$ can be considered as words of length $2^{\ddot{\gamma}+2\delta}$ over $\mathbb{Z}_2$ or $\mathbb{Z}_4$, respectively. Let $\mathscr{A}$ be the set of all affine functions f from $\mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^\delta$ to $\mathbb{Z}_4$ that map the all-zero vector to 0 or 2: $f(\bar{0}) \in \{0,2\}$. Denote $C_{\ddot{\gamma},\delta} = \{h : \mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^\delta \to \mathbb{Z}_2 \,:\, h(\cdot) = \varphi^+(f(\cdot))$ for some $f \in \mathscr{A}\}$.

Lemma 2 *$C_{\ddot{\gamma},\delta}$ is a $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code of length $n = 2^{\ddot{\gamma}+2\delta}$ and type $(\alpha, \beta; \gamma, \delta)$, where $\gamma = \ddot{\gamma} + 1$, $\alpha = 2^{\ddot{\gamma}+\delta}$ corresponding to the elements of order at most 2 of $\mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^\delta$, and $\beta = 2^{\ddot{\gamma}+\delta-1}(2^\delta - 1)$ corresponding to the pairs of opposite elements of order* 4.

Proof There are $2 \cdot 2^{\ddot{\gamma}} \cdot 4^\delta = 2n$ affine functions in $\mathscr{A}$. The set $\mathscr{A}$ is closed under the addition over $\mathbb{Z}_4$; so the Gray map image $A = \Phi(\mathscr{A})$ can also be considered as a $\mathbb{Z}_2\mathbb{Z}_4$-linear code with $2^{\ddot{\gamma}+\delta+1}$ coordinates over $\mathbb{Z}_2$, which correspond to the elements of order at most 2 of $\mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^\delta$.

Now, we will see that the code A can be obtained from $C_{\ddot{\gamma},\delta}$ by repeating twice every coordinate. That is, strictly speaking, A is permutation equivalent to $\{(h,h) : h \in C_{\ddot{\gamma},\delta}\}$. Indeed, given $v \in \mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^\delta$ of order 4 and an affine function $f \in \mathscr{A}$, the values $\varphi^+(f(v))$ and $\varphi^+(f(-v))$ of the corresponding codeword of $C_{\ddot{\gamma},\delta}$ each occurs both in $\varphi(f(v))$ and $\varphi(f(-v))$. If the order of $v \in \mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^{\delta}$ is 2 or less, then $\varphi^+(f(v))$ is duplicated in $\varphi(f(v))$.

Finally, it is easy to check that the minimum Lee distance for the set of affine functions $\mathscr{A}$ is $n = 2^{\ddot{\gamma}+2\delta}$; so the minimum Hamming distance of $C_{\ddot{\gamma},\delta}$ is the half of this value, that is, $n/2$. □

Lemma 3 *Let* $f : \mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^{\delta} \to \mathbb{Z}_4$ *be an affine function. Then* $h(\cdot) = \varphi^+(f(\cdot))$ *belongs to* $C_{\ddot{\gamma},\delta}$*.*

Proof In the case that $f(\bar{0}) \in \{0, 2\}$, $C_{\ddot{\gamma},\delta}$ contains h by definition. On the other hand, if $f(\bar{0}) \in \{1, 3\}$, we will use that $\varphi^+(l) = \varphi^+(3-l)$ for $l \in \mathbb{Z}_4$. Then, $h(\cdot) = \varphi^+(f(\cdot)) = \varphi^+(3-f(\cdot))$. Since $3-f(\cdot)$ is an affine function and $3-f(\bar{0}) \in \{0, 2\}$, we obtain that $h \in C_{\ddot{\gamma},\delta}$. □

Theorem 4 *The* $\mathbb{Z}_4$*-linear Hadamard code* $D_{\gamma,\ddot{\delta}}$ *of length n and type* $(0, n/2; \gamma, \delta)$ *is permutation equivalent to the* $\mathbb{Z}_2\mathbb{Z}_4$*-linear Hadamard code* $C_{\gamma+1,\ddot{\delta}}$ *of type* $(\alpha, \beta; \gamma+2, \delta-1)$ *with* $\alpha \neq 0$.

Proof Consider a function f in $\mathscr{B}$ and the related function $g(v, e) = f(v)+2ef(\bar{0})$, where $v \in \mathbb{Z}_2^{\gamma} \times \mathbb{Z}_4^{\ddot{\delta}}$ and $e \in \mathbb{Z}_2$. We can see that

$$\varphi(f(v)) = \left(\varphi^+(g(-v, 1)), \varphi^+(g(v, 0))\right),$$

$$\varphi(f(-v)) = \left(\varphi^+(g(v, 1)), \varphi^+(g(-v, 0))\right).$$

In order to check these equalities, it is convenient to represent $f(v)$ as $f_0(v) + f(\bar{0})$, where $f_0 : \mathbb{Z}_2^{\gamma} \times \mathbb{Z}_4^{\ddot{\delta}} \to \mathbb{Z}_4$ is a group homomorphism (in particular, $f_0(-v) = -f_0(v)$).

Since g is an affine function from $v \in \mathbb{Z}_2^{\gamma+1} \times \mathbb{Z}_4^{\ddot{\delta}}$ to $\mathbb{Z}_4$, we can deduce from Lemma 3 that there is a fixed coordinate permutation that sends every codeword of $D_{\gamma,\ddot{\delta}}$ to a codeword of $C_{\gamma+1,\ddot{\delta}}$. □

Corollary 5 *There are exactly* $\lfloor \frac{t}{2} \rfloor$ *nonequivalent* $\mathbb{Z}_2\mathbb{Z}_4$*-linear Hadamard codes of length* 2^t*.*

3 The Permutation Automorphism Group

Considering the representation of a code as the union of cosets of its kernel, it is possible to prove the following fact.

Proposition 6 *If $\delta \geq 2$, then the order of the automorphism group of C satisfies*

$$|\mathrm{Aut}(C)| \leq p \cdot 2^{\frac{1}{2}\ddot{\gamma}(\ddot{\gamma}+1)+2\ddot{\gamma}\delta+\frac{3}{2}\delta(\delta+1)} \prod_{i=1}^{\ddot{\gamma}}(2^i-1) \prod_{j=1}^{\delta}(2^j-1),$$

where $p = 6$ if $\delta = 2$ and $p = 1$ if $\delta \geq 3$.

By Lemma 3, any nonsingular affine transformation of $\mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^{\delta}$ belongs to $\mathrm{Aut}(C)$. Therefore, since for $\delta \geq 3$, the number of nonsingular affine transformations coincides with the upper bound given in Proposition 6, we obtain the following result:

Theorem 7 *The automorphism group of the $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code C of type $(\alpha, \beta; \ddot{\gamma}+1, \delta)$, with $\delta \geq 3$, is the group of nonsingular affine transformations of $\mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^{\delta}$. Therefore, its order is*

$$|\mathrm{Aut}(C)| = 2^{\frac{1}{2}\ddot{\gamma}(\ddot{\gamma}+1)+2\ddot{\gamma}\delta+\frac{3}{2}\delta(\delta+1)} \prod_{i=1}^{\ddot{\gamma}}(2^i-1) \prod_{j=1}^{\delta}(2^j-1).$$

In the case $\delta = 2$, there are non-affine permutations in the automorphism group, and the resulting formula again coincides with the upper bound of Proposition 6.

Theorem 8 *The automorphism group of the $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code C of type $(\alpha, \beta; \ddot{\gamma}+1, 2)$ consists of all permutations expressed as $\psi\alpha$, where α is a nonsingular affine transformation of $\mathbb{Z}_2^{\ddot{\gamma}} \times \mathbb{Z}_4^{2}$ and ψ is the identity permutation or one of five non-affine permutations. The order of the automorphism group is*

$$|\mathrm{Aut}(C)| = 6 \cdot 2^{\frac{1}{2}\ddot{\gamma}(\ddot{\gamma}+1)+4\ddot{\gamma}+9} \cdot 3 \prod_{i=1}^{\ddot{\gamma}}(2^i-1).$$

Acknowledgements The work of the first author has been partially supported by the Russian Foundation for Basic Research under Grant 13-01-00463-a and by the Target Program of SB RAS for 2012-2014 (integration project no. 14).

The work of the second author has been partially supported by the Spanish MICINN under Grants TIN2010-17358 and TIN2013-40524-P and by the Catalan AGAUR under Grant 2014SGR-691.

References

1. Avgustinovich, S.V., Solov'eva, F.I., Heden, O.: On the structure of symmetry groups of Vasil'ev codes. Probl. Inf. Transm. **41**(2), 105–112 (2005). doi:10.1007/s11122-005-0015-5
2. Bauer, H., Ganter, B., Hergert, F.: Algebraic techniques for nonlinear codes. Combinatorica **3**(1), 21–33 (1983). doi:10.1007/BF02579339

3. Bierbrauer, J.: Introduction to Coding Theory. Discrete and Combinatorial Mathematics Series. Chapman & Hall/CRC, Boca Raton (2005)
4. Borges, J., Fernández-Córdoba, C., Pujol, J., Rifà, J., Villanueva, M.: Z_2Z_4-linear codes: generator matrices and duality. Des. Codes Cryptogr. **54**(2), 167–179 (2010). doi:10.1007/s10623-009-9316-9
5. Borges, J., Phelps, K.T., Rifà, J.: The rank and kernel of extended 1-perfect Z_4-linear and additive non-Z_4-linear codes. IEEE Trans. Inf. Theory **49**(8), 2028–2034 (2003). doi:10.1109/TIT.2003.814490
6. Borges, J., Rifà, J.: A characterization of 1-perfect additive codes. IEEE Trans. Inf. Theory **45**(5), 1688–1697 (1999). doi:10.1109/18.771247
7. Delsarte, P.: An Algebraic Approach to Association Schemes of Coding Theory. Philips Research Reports Supplements, vol. 10. N.V. Philips' Gloeilampenfabrieken, Eindhoven (1973)
8. Delsarte, P., Levenshtein, V.I.: Association schemes and coding theory. IEEE Trans. Inf. Theory **44**(6), 2477–2504 (1998). doi:10.1109/18.720545
9. Fernández-Córdoba, C., Phelps, K.T., Villanueva, M.: Involutions in binary perfect codes. IEEE Trans. Inf. Theory **57**(9), 5926–5932 (2011). doi:10.1109/TIT.2011.2162185
10. Heden, O.: On the symmetry group of perfect 1-error correcting binary codes. J. Comb. Math. Comb. Comput. **52**, 109–115 (2005)
11. Heden, O., Pasticci, F., Westerbäck, T.: On the existence of extended perfect binary codes with trivial symmetry group. Adv. Math. Commun. **3**(3), 295–309 (2009). doi:10.3934/amc.2009.3.295
12. Huffman, W.C., Pless, V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, Cambridge (2003)
13. Krotov, D.S.: Z_4-Linear perfect codes. Diskretn. Anal. Issled. Oper. Ser.1 **7**(4), 78–90 (2000). In Russian http://mi.mathnet.ru/eng/da281. Translated to English at http://arxiv.org/abs/0710.0198
14. Krotov, D.S.: Z_4-linear Hadamard and extended perfect codes. In: Augot, D., Carlet, C. (eds.) WCC2001, International Workshop on Coding and Cryptography, Paris. Electronic Notes in Discrete Mathematics, vol. 6, pp. 107–112. Elsevier B.V. (2001). doi:10.1016/S1571-0653(04)00161-1
15. Krotov, D.S.: On the automorphism groups of the additive 1-perfect binary codes. In: Borges, J., Villanueva, M. (eds.) 3rd International Castle Meeting on Coding Theory and Application, Cardona, pp. 171–176. Universitat Autònoma de Barcelona. Servei de Publicacions, Bellaterra (2011)
16. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North Holland, Amsterdam (1977)
17. Pernas, J., Pujol, J., Villanueva, M.: Classification of some families of quaternary Reed–Muller codes. IEEE Trans. Inf. Theory **57**(9), 6043–6051 (2011). doi:10.1109/TIT.2011.2119465
18. Pernas, J., Pujol, J., Villanueva, M.: Characterization of the automorphism group of quaternary linear Hadamard codes. Des. Codes Cryptogr. **70**(1–2), 105–115 (2014). doi:10.1007/s10623-012-9678-2
19. Phelps, K.T., Rifà, J.: On binary 1-perfect additive codes: some structural properties. IEEE Trans. Inf. Theory **48**(9), 2587–2592 (2002). doi:10.1109/TIT.2002.801474
20. Phelps, K.T., Rifà, J., Villanueva, M.: On the additive (Z_4-linear and non-Z_4-linear) Hadamard codes: rank and kernel. IEEE Trans. Inf. Theory **52**(1), 316–319 (2006). doi:10.1109/TIT.2005.860464
21. Pujol, J., Rifà, J., Solov'eva, F.I.: Construction of Z_4-linear Reed–Muller codes. IEEE Trans. Inf. Theory **55**(1), 99–104 (2009). doi:10.1109/TIT.2008.2008143
22. Rifà, J., Pujol, J.: Translation-invariant propelinear codes. IEEE Trans. Inf. Theory **43**(2), 590–598 (1997). doi:10.1109/18.556115
23. Solov'eva, F.I.: On Z_4-linear codes with the parameters of Reed-Muller codes. Probl. Inf. Transm. **43**(1), 26–32 (2007). Translated from Probl. Peredachi Inf. 43(1), 32–38 (2007). doi:10.1134/S0032946007010048

Linear Batch Codes

Helger Lipmaa and Vitaly Skachek

Abstract In an application, where a client wants to obtain many symbols from a large database, it is often desirable to balance the load. Batch codes (introduced by Ishai et al. in STOC 2004) do exactly that: the large database is divided between many servers, so that the client has to only make a small number of queries to every server to obtain sufficient information to reconstruct all desired symbols.

In this work, we formalize the study of *linear* batch codes. These codes, in particular, are of potential use in distributed storage systems. We show that a generator matrix of a binary linear batch code is also a generator matrix of classical binary linear error-correcting code. This immediately yields that a variety of upper bounds, which were developed for error-correcting codes, are applicable also to binary linear batch codes. We also propose new methods for constructing large linear batch codes from the smaller ones.

Keywords Batch codes • Error-correcting codes • Computationally-private information retrieval • Load balancing • Distributed storage

1 Introduction

Consider the scenario where a client wants to retrieve m symbols from an n symbol database. Accessing a single server by all clients simultaneously can create serious performance problems. A simple solution is to replicate the whole database between M servers, so that the client can query approximately m/M symbols from every server. However, that solution require to store $N = Mn$ database symbols.

In an m-out-of-n CPIR (computationally-private information retrieval [5, 8]), the client wants to retrieve m symbols from an n symbol database without the storage provider getting to know which symbols were retrieved. An additional problem in this case is the storage provider's computational complexity that is $\Theta(n)$ per query

H. Lipmaa • V. Skachek (✉)
Institute of Computer Science, University of Tartu, J. Liivi 2, 50409 Tartu , Estonia
e-mail: helger.lipmaa@ut.ee; vitaly.skachek@ut.ee

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_26

in almost all known 1-out-of-n CPIR protocols. (The only exception is [9], where the per-query computational complexity is $O(n/\log n)$.) Just performing m instances of an 1-out-of-n CPIR protocol would result in a highly prohibitive computational complexity.

To tackle both mentioned problems, Ishai et al. [7] proposed to use *batch codes*. More precisely, let Σ be a finite alphabet. In an $(n, N, m, M, T)_\Sigma$ batch code, a database $\boldsymbol{f}$ of n strings in Σ is divided into M buckets where each bucket contains N/M strings in Σ. If a client is to obtain m symbols of the original database, he query (no more than) T symbols from each of the M buckets.

Batch codes have been recently studied very actively in the combinatorial setting. Namely, a *combinatorial batch code* (CBC) satisfies the additional requirement that every symbol of every bucket is equal to some symbol of the original database. (See for example [2–4, 16].) New constructions of combinatorial batch codes, based on affine planes and transversal designs, were recently presented in [15].

We stress that linear batch codes are also well suitable for the use in the *distributed data storage* [6]. The buckets can be viewed as servers. The reading of the requested data can be done "locally" from a small number of servers. If a small number of buckets stopped functioning, the data can be reproduced by reading data from (a small number) of other buckets.

In this paper, we formalize a framework for analysis of linear batch codes, which resembles that of the classical error-correcting codes (ECCs). As we show, generator matrices of good binary linear batch codes are also generator matrices of good classical ECCs. This immediately gives us a set of tools and bounds from the classical coding theory for analyzing binary linear batch codes. The converse, however, is not true: not every good binary linear ECC is a good linear batch code. Finally, we present a number of simple constructions of larger linear batch codes from the smaller ones. The preliminary version of this paper is available as [10].

2 Preliminaries

Let $[n] \triangleq \{1, 2, \ldots, n\}$. We denote by $\langle \boldsymbol{v}_i \rangle_{i\in[n]}$ the linear span of the vectors $\boldsymbol{v}_i$, $i \in [n]$, over some finite field $\mathbb{F}_q$. We use notation $\mathsf{d}_H(\boldsymbol{x}, \boldsymbol{y})$ to denote the Hamming distance between the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, and notation $\mathsf{w}_H(\boldsymbol{x})$ to denote the Hamming weight of $\boldsymbol{x}$. We also denote by $\mathbf{0}$ the row vector consisting of all zeros, and by $\boldsymbol{e}_i$ the row vector having one at position i and zeros elsewhere (the length of vectors will be clear from the context).

We start with the definition of a batch code. In this work, we focus on so-called *multiset* batch codes, as they were defined in [7].

Definition 1 ([7]) Let Σ be a finite alphabet. We say that $\mathscr{C}$ is an $(n, N, m, M, t)_\Sigma$ batch code over a finite alphabet Σ if it encodes any string $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Sigma^n$ into M strings (buckets) of total length N over Σ, namely $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M$, such that for each m-tuple (batch) of (not necessarily distinct) indices $i_1, i_2, \ldots, i_m \in [n]$, the symbols $x_{i_1}, x_{i_2}, \ldots, x_{i_m}$ can be retrieved by m users, respectively, by reading at most t symbols from each bucket, such that each symbol x_{i_ℓ} is recovered from the symbols read by the ℓ-th user alone.

The ratio $R \triangleq n/N$ is called the rate of the code.

If for the code $\mathscr{C}$ it holds that $t = 1$, then we use notation $(n, N, m, M)_\Sigma$ for it. This corresponds to an important special case when only one symbol is read from each bucket. Note that the buckets in this definition correspond to the devices in the above example, the encoding length N to the total storage, and the parameter t to the maximal load. If $\Sigma = \mathbb{F}_q$ is a finite field, we also use notation $(n, N, m, M, t)_q$ (or $(n, N, m, M)_q$) to denote $(n, N, m, M, t)_\Sigma$ (or $(n, N, m, M)_\Sigma$, respectively).

Definition 2 We say that an $(n, N, m, M, t)_q$ batch code is *linear*, if every symbol in every bucket is a linear combination of original database symbols.

3 Linear Batch Codes

In what follows, we consider the case of a linear batch code $\mathscr{C}$ with $t = 1$. Moreover, we limit ourselves to the case when $N = M$, which means that each encoded bucket contains just one symbol in $\mathbb{F}_q$.

Definition 3 For simplicity we refer to a linear $(n, N = M, m, M)_q$ batch code as $[M, n, m]_q$ batch code.

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ be an information string, and let $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)$ be an encoding of $\boldsymbol{x}$. Due to linearity of the code, each encoded symbol y_i, $i \in [M]$, can be written as $y_i = \sum_{j=1}^n g_{j,i} x_j$ for some symbols $g_{j,i} \in \mathbb{F}_q$, $j \in [n]$, $i \in [M]$. Then we can form the matrix $\boldsymbol{G}$ as follows:

$$\boldsymbol{G} = \left(g_{j,i}\right)_{j \in [n], i \in [M]} ,$$

and thus the encoding is $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G}$.

The $n \times M$ binary matrix $\boldsymbol{G}$ play a role similar to a generator matrix for a classical linear ECC. In the sequel, we call $\boldsymbol{G}$ a *generator matrix* of the batch code $\mathscr{C}$. We denote by $\boldsymbol{G}_i$ the i-th row of $\boldsymbol{G}$ and by $\boldsymbol{G}^{[\ell]}$ the ℓ-th column of $\boldsymbol{G}$.

Theorem 4 *Let $\mathscr{C}$ be an $[M, n, m]_q$ batch code. It is possible to retrieve $x_{i_1}, x_{i_2}, \ldots, x_{i_m}$ simultaneously if and only if there exist m non-intersecting sets $T_1, T_2, \ldots, T_m$ of indices of columns in $\boldsymbol{G}$, and for T_r there exists a linear combination of columns of $\boldsymbol{G}$ indexed by that set, which equals to the column vector $\boldsymbol{e}_{i_r}^T$, for all $r \in [m]$.*

Proof 1. For each $r \in [m]$

$$\boldsymbol{e}_{i_r}^T = \sum_{\ell \in T_r} \alpha_\ell \cdot \boldsymbol{G}^{[\ell]} ,$$

where all $\alpha_\ell \in \mathbb{F}_q$. Due to linearity, the encoding of $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ can be written as $\boldsymbol{y} = (y_1, y_2, \ldots, y_M) = \boldsymbol{x} \cdot \boldsymbol{G}$. Then,

$$x_{i_r} = \boldsymbol{x} \cdot \boldsymbol{e}_{i_r}^T = \boldsymbol{x} \cdot \left(\sum_{\ell \in T_r} \alpha_\ell \cdot \boldsymbol{G}^{[\ell]} \right) = \sum_{\ell \in T_r} \alpha_\ell (\boldsymbol{x} \cdot \boldsymbol{G}^{[\ell]}) = \sum_{\ell \in T_r} \alpha_\ell \cdot y_\ell ,$$

and therefore the value of x_{i_r} can be obtained by querying only the values of y_ℓ for $\ell \in T_r$. The conclusion follows from the fact that all T_r do not intersect.

2. To show the opposite direction of Theorem 4, we follow the idea of the proof of Theorem 1 in [1]. Let T_r, for $r \in [m]$, be a set of indices of entries in $\boldsymbol{y}$, which are used to retrieve x_{i_r}. We show that $\boldsymbol{e}_{i_r}^T \in \langle \boldsymbol{G}^{[\ell]} \rangle_{\ell \in T_r}$.

   Denote the vector space $W_r \triangleq \langle \boldsymbol{G}^{[\ell]} \rangle_{\ell \in T_r}$. Assume by contradiction that $\boldsymbol{e}_{i_r} \notin W_r$.

   Recall that the dual space of W_r, denoted by $W_r^\perp$, consists of all the vectors orthogonal to any vector in W_r. Since $\boldsymbol{e}_{i_r} \notin W_r$, there exists a vector $\boldsymbol{z} \in W_r^\perp$, which is not orthogonal to $\boldsymbol{e}_{i_r}$, i.e. $\boldsymbol{z} \cdot \boldsymbol{e}_{i_r} \neq 0$, and so $z_{i_r} \neq 0$. On the other hand, this vector $\boldsymbol{z}$ is orthogonal to any vector $\boldsymbol{G}^{[\ell]}$ for $\ell \in T_r$.

   Consider the encoding of the vectors $\boldsymbol{z}$ and $\boldsymbol{0}$, $\boldsymbol{z} \cdot \boldsymbol{G}$ and $\boldsymbol{0} \cdot \boldsymbol{G}$, respectively. In both cases, all the coordinates of $\boldsymbol{y}$ indexed by T_r are all zeros. Therefore, the result of the retrieval of the i_r-th encoded symbol in both cases is the same, yet $z_{i_r} \neq 0$. We obtain a contradiction.

Example 5 Consider the following linear binary batch code $\mathscr{C}$ whose 4×9 generator matrix is given by

$$\boldsymbol{G} = \begin{pmatrix} 1\,0\,1\,0\,0\,0\,1\,0\,1 \\ 0\,1\,1\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,1\,0\,1\,1\,0\,1 \\ 0\,0\,0\,0\,1\,1\,0\,1\,1 \end{pmatrix} .$$

Let $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$, $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G}$.

Assume that we want to retrieve the values of (x_1, x_1, x_2, x_2). We can retrieve (x_1, x_1, x_2, x_2) from the following set of equations:

$$\begin{cases} x_1 = y_1 \\ x_1 = y_2 + y_3 \\ x_2 = y_5 + y_8 \\ x_2 = y_4 + y_6 + y_7 + y_9 \end{cases} .$$

Moreover, it is straightforward to verify that any 4-tuple $(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$, where $i_1, i_2, i_3, i_4 \in [4]$, can be retrieved by using columns indexed by some four non-intersecting sets of indices in [9]. Therefore, the code $\mathscr{C}$ is a $[9, 4, 4]_2$ batch code. As a matter of fact, this code is the two-layer construction of "subcube code" in [7, Section 3.2].

Lemma 6 *Let $\mathscr{C}$ be an $[M, n, m]_q$ batch code. Then, each row of $\boldsymbol{G}$ has Hamming weight at least m.*

Proof Consider row j, for an arbitrary $j \in [n]$. We can retrieve the combination $(x_j, x_j, \ldots, x_j)$ if there are m non-intersecting sets of columns, such that sum of the symbols in each set is equal $\boldsymbol{e}_j^T$. Therefore, there are at least m columns in $\boldsymbol{G}$ with a nonzero symbol in position j. □

Lemma 7 *Let $\mathscr{C}$ be an $[M, n, m]_q$ batch code. Then, the matrix $\boldsymbol{G}$ is full rank.*

Proof We are able to recover any combination of size m of $\{x_1, x_2, \ldots, x_n\}$. Then, the column vectors

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \cdots \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

are all in the column space of $\boldsymbol{G}$. Therefore, the column space of $\boldsymbol{G}$ has dimension n, and so the matrix is full rank. □

The following theorem is the main result of this section. The presented proof of this theorem works only for *binary* batch codes.

Theorem 8 *Let $\mathscr{C}$ be an $[M, n, m]_2$ batch code $\mathscr{C}$ over $\mathbb{F}_2$. Then, $\boldsymbol{G}$ is a generator matrix of the classical error-correcting $[M, n, \geq m]_2$ code.*

Proof Let $\mathbb{C}$ be a classical ECC, whose generating matrix is $\boldsymbol{G}$. It is obvious that the length of $\mathbb{C}$ is M. Moreover, since the matrix $\boldsymbol{G}$ is a full rank matrix due to Lemma 7, we obtain that the dimension of $\mathbb{C}$ is n. Thus, the only parameter in question is the minimum distance of $\mathbb{C}$.

In order to show that the minimum distance of $\mathbb{C}$ is at least m, it will be sufficient to show that any non-zero linear combination of the rows of $\boldsymbol{G}$ has Hamming weight at least m. Consider an arbitrary linear combination of the rows of $\boldsymbol{G}$, whose indices are given by a set $T \neq \varnothing$,

$$\boldsymbol{z} = \sum_{i \in T} \boldsymbol{G}_i \ .$$

Take an arbitrary index $i_0 \in T$. Due to the properties of the batch codes we should be able to recover $(x_{i_0}, x_{i_0}, \ldots, x_{i_0})$ from $\boldsymbol{y}$. Therefore, there exist m disjoint sets of indices $S_1, S_2, \dots, S_m$, $S_i \subseteq [M]$, such that for all $i \in [m]$:

$$\sum_{j \in S_i} \boldsymbol{G}^{[j]} = \boldsymbol{e}_{i_0}^T \ . \tag{1}$$

Now, consider the sub-matrix $\boldsymbol{M}_i$ of $\boldsymbol{G}$ which is formed by the rows of $\boldsymbol{G}$ indexed by T and the columns of $\boldsymbol{G}$ indexed by S_i. Due to (1), the row of $\boldsymbol{M}_i$ that corresponds to the row i_0 in $\boldsymbol{G}$, has an odd number of ones in it. All other rows of $\boldsymbol{M}_i$ contain an even number of ones. Therefore, the matrix $\boldsymbol{M}_i$ contains an odd number of ones. This means that the vector of $\boldsymbol{z}$ will also contain an odd number of ones in the positions given by the set S_i. This odd number is at least one.

We conclude that $\boldsymbol{z}$ contains at least one '1' in positions given by the set S_i, for all $i \in [m]$. The sets S_i are disjoint, and therefore the Hamming weight of $\boldsymbol{z}$ is at least m. □

Example 9 The converse of Theorem 8 is generally not true. In other words, if $\boldsymbol{G}$ is a generator matrix of a classical error-correcting $[M, n, m]_2$ code, then the corresponding code $\mathscr{C}$ is not necessarily an $[M, n, m]_2$ batch code. For example, take $\boldsymbol{G}$ to be a generator matrix of the classical $[4, 3, 2]_2$ ECC as follows:

$$\boldsymbol{G} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} .$$

Let $\boldsymbol{x} = (x_1, x_2, x_3)$, $\boldsymbol{y} = (y_1, y_2, y_3, y_4) = \boldsymbol{x}\boldsymbol{G}$.

It is impossible to retrieve (x_2, x_3). This can be verified by the fact that

$$x_2 = y_1 + y_2 = y_3 + y_4 \quad \text{and} \quad x_3 = y_1 + y_3 = y_2 + y_4 \ ,$$

and so one of the y_i's is always needed to compute each of x_2 and x_3.

Remark 10 The topic of linear ECCs was very intensively studied over the years. Various well-studied properties of linear ECCs, such as MacWilliams identities [11], apply also to linear batch codes due to Theorem 8 (for $t = 1$, $M = N$ and $q = 2$). A variety of bounds on the parameters of ECCs, such as sphere-packing bound, Plotkin bound, Griesmer bound, Elias-Bassalygo bound, McEliece-Rodemich-Rumsey-Welch bound [13] (see also [14, Chapter 4], [12]) apply to the parameters of linear binary $[M, n, m]$ batch codes.

4 Constructions of New Codes

In this section we present several simple methods to construct new linear batch codes from the existing ones.

Theorem 11 *Let $\mathscr{C}_1$ be an $[M_1, n, m_1]_q$ batch code and $\mathscr{C}_2$ be an $[M_2, n, m_2]_q$ batch code. Then, there exists an $[M_1 + M_2, n, m_1 + m_2]_q$ batch code.*

Proof Let $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ be $n \times M_1$ and $n \times M_2$ generator matrices corresponding to $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively. Consider the following $n \times (M_1 + M_2)$ matrix

$$\hat{\boldsymbol{G}} = [\, \boldsymbol{G}_1 \mid \boldsymbol{G}_2 \,] \ .$$

This matrix corresponds to a batch code of length $M_1 + M_2$ with n variables. It is sufficient to show that any combination of $m_1 + m_2$ variables can be retrieved. By the assumption, the first (any) m_1 variables can be retrieved from the first M_1 coordinates of $\boldsymbol{y}$ and the last m_2 variables can be retrieved from the last M_2 coordinates of $\boldsymbol{y}$. This completes the proof. □

Theorem 12 *Let $\mathscr{C}_1$ be an $[M_1, n_1, m_1]_q$ batch code and $\mathscr{C}_2$ be an $[M_2, n_2, m_2]_q$ batch code. Then, there exists an $[M_1 + M_2, n_1 + n_2, \min\{m_1, m_2\}]_q$ batch code.*

Proof As before, denote by $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ the $n_1 \times M_1$ and $n_2 \times M_2$ generator matrices corresponding to $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively. Consider the following $(n_1 + n_2) \times (M_1 + M_2)$ matrix

$$\hat{\boldsymbol{G}} = \left[\begin{array}{c|c} \boldsymbol{G}_1 & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{G}_2 \end{array} \right] \ .$$

The matrix $\hat{\boldsymbol{G}}$ corresponds to a batch code of length $M_1 + M_2$ with $n_1 + n_2$ variables. Moreover, any combination of $\min\{m_1, m_2\}$ variables can be retrieved. If all unknowns are from $\{x_1, x_2, \ldots, x_{n_1}\}$, then they can be retrieved by using only the first M_1 columns of $\hat{\boldsymbol{G}}$. If all unknowns are from $\{x_{n_1+1}, x_{n_1+2}, \ldots, x_{n_1+n_2}\}$, then they can be retrieved by using only the last M_2 columns of $\hat{\boldsymbol{G}}$. Generally, some unknowns can be retrieved by using combinations of the first M_1 columns, while the other unknowns are retrieved using combinations of the last M_2 columns. Since the number of unknowns is at most $\min\{m_1, m_2\}$, we can always retrieve all of them simultaneously. □

Theorem 13 *Let $\mathscr{C}$ be an $[M, n, m]_q$ batch code, and let $\boldsymbol{G}$ be the corresponding $n \times M$ matrix. Then, the code $\hat{\mathscr{C}}$, defined by the $(n + 1) \times (M + m)$ matrix*

$$\hat{\boldsymbol{G}} = \left(\begin{array}{c|c} & 0\ 0 \cdots 0 \\ \boldsymbol{G} & \vdots\ \vdots\ \ddots\ \vdots \\ & 0\ 0 \cdots 0 \\ \hline \bullet\ \bullet\ \bullet\ \cdots \bullet & 1\ 1 \cdots 1 \end{array} \right)$$

$$\underbrace{\qquad\qquad\quad}_{M}\underbrace{\qquad\qquad\quad}_{m}$$

is an $[M+m, n+1, m]$ *batch code, where* $\bullet$ *stands for an arbitrary symbol in* $\mathbb{F}_q$.

Proof As before, let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n, x_{n+1})$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_{M+m}) = \boldsymbol{x}\hat{\boldsymbol{G}}$. Assume that we want to retrieve the vector $\boldsymbol{z} = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})$.

Take a particular x_{i_j} in $\boldsymbol{z}$, $j \in [m]$. Consider two cases. If $i_j \neq n+1$ then, since $\mathscr{C}$ is a batch code, we have

$$x_{i_j} = \sum_{\ell \in T_{i_j}} y_\ell \; + \; \xi \cdot x_{n+1} \; ,$$

where $T_{i_j} \subseteq [M]$ and $\xi \in \mathbb{F}_q$. In that case, if $\xi = 0$, then $x_{i_j} = \sum_{\ell \in T_{i_j}} y_\ell$. If $\xi \neq 0$, then $x_{i_j} = \sum_{\ell \in T_{i_j}} y_\ell + \xi \cdot y_{M+j}$. Observe that all T_{i_j} are disjoint due to the properties of a batch code.

In the second case, $i_j = n+1$, and we simply set $x_{i_j} = x_{n+1} = y_{M+j}$.

In both cases, we used sets $\{y_\ell \; : \; \ell \in T_{i_j} \cup \{M+j\}\}$ to retrieve x_{i_j}. These sets are all disjoint for $j \in [m]$.

We conclude that all m unknowns x_{i_j}, $j \in [m]$, can be retrieved simultaneously. □

Acknowledgements We thank Dominique Unruh for helpful discussions. The work of the authors is supported in part by the research grants PUT405 and IUT2-1 from the Estonian Research Council and by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS. The work of V. Skachek is also supported in part by the EU COST Action IC1104.

References

1. Bar-Yossef, Z., Birk, Y., Jayram, T.S., Kol, T.: Index coding with side information. IEEE Trans. Inf. Theory **57**(3), 1479–1494 (2011)
2. Bhattacharya, S., Ruj, S., Roy, B.: Combinatorial batch codes: a lower bound and optimal constructions. Adv. Math. Commun. **6**(2), 165–174 (2012)
3. Brualdi, R.A., Kiernan, K., Meyer, S.A., Schroeder, M.W.: Combinatorial batch codes and transversal matroids. Adv. Math. Commun. **4**(3), 419–431 (2010)
4. Bujtás, C., Tuza, Z.: Batch codes and their applications. Electron. Notes Discret. Math. **38**, 201–206 (2011)
5. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: Proceedings of the 36th Symposium on Foundations of Computer Science (FOCS), Milwaukee, Wisconsin, USA pp. 41–50 (1995)
6. Dimakis, A.G., Godfrey, P.B., Wu, Y., Wainwright, M.J., Ramchandran, K.: Network coding for distributed storage systems. IEEE Trans. Inf. Theory **59**(9), 4539–4551 (2010)
7. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Batch codes and their applications. In: Proceedings of the 36th ACM Symposium on Theory of Computing (STOC), Chicago (2004)
8. Kushilevitz, E., Ostrovsky, R.: Replication is NOT needed: SINGLE database, computationally-private information retrieval. In: Proceedings of the 38th Symposium on Foundations of Computer Science (FOCS), Miami Beach, Florida, USA pp. 364–373 (1997)

9. Lipmaa, H.: First CPIR protocol with data-dependent computation. In: Proceedings of the International Conference on Information Security and Cryptology (ICISC), Seoul, South Korea pp. 193–210 (2009)
10. Lipmaa, H., Skachek, V.: Linear batch codes. Available online http://arxiv.org/abs/1404.2796
11. MacWilliams, F.J.: A theorem on the distribution of weights in a systematic code. Bell Syst. Tech. J. **42**, 79–94 (1963)
12. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1978)
13. McEliece, R.J., Rodemich, E.R., Rumsey, H., Welch, L.R.: New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. IEEE Trans. Inf. Theory **IT-23**, 157–166 (1997)
14. Roth, R.M.: Introduction to Coding Theory. Cambridge University Press, Cambridge (2006)
15. Silberstein, N., Gál, A.: Optimal combinatorial batch codes based on block designs. Available online http://arxiv.org/abs/1312.5505
16. Stinson, D., Wei, R., Paterson, M.: Combinatorial batch codes. Adv. Math. Commun. **3**(1), 13–17 (2009)

An Extension of the Brouwer-Zimmermann Minimum Weight Algorithm

Petr Lisoněk and Layla Trummer

Abstract We study the algorithm for computing the minimum weight of a linear code that was invented by A. Brouwer and later extended by K.-H. Zimmermann. We show that matroid partitioning algorithms can be used to efficiently find a favourable (and sometimes best possible) sequence of information sets on which the Brouwer-Zimmermann minimum weight algorithm operates.

Keywords Linear code • Minimum weight • Brouwer-Zimmermann algorithm

1 Introduction

For a prime power q let $\mathbb{F}_q$ denote the field with q elements. We assume that q is small, hence all arithmetic operations in $\mathbb{F}_q$ are performed at unit cost. For $x \in \mathbb{F}_q^n$ let $\mathrm{wt}(x)$ denote the Hamming weight of x. By an $[n, k]_q$ linear code we mean a k-dimensional subspace of $\mathbb{F}_q^n$. Let d denote the minimum Hamming distance of distinct codewords in C, then d is also the minimum Hamming weight of non-zero codewords in C. Under the standard definitions of coding theory [2, 9] the code C can detect up to $d - 1$ errors and it can correct up to $\lfloor (d - 1)/2 \rfloor$ errors. Thus determining the value of d is critical for understanding of the error detection/correction capability of C.

Vardy [11] showed that for general linear binary codes, computing the minimum weight is an NP-hard problem, and the corresponding decision problem is NP-complete. Hence any algorithm for computing the minimum weight will run in superpolynomial time, unless P = NP.

This paper is concerned with the algorithm for computing the minimum weight of a linear code that was invented by A. Brouwer and later extended by K.-H. Zimmermann. The algorithm is outlined in Sect. 3. Before that, in Sect. 2 we review background from matroid theory. In Sect. 4 we propose our extension to the Brouwer-Zimmermann algorithm which consists in an efficient construction of a good (sometimes best possible) sequence of information sets for the given code. In

P. Lisoněk (✉) • L. Trummer
Department of Mathematics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada
e-mail: plisonek@sfu.ca; ltrummer@sfu.ca

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_27

Sect. 5 we compare our approach to the previous literature. Throughout the paper we also make references to the implementation of the Brouwer-Zimmermann algorithm that is available in Magma [3].

2 Matroid Partitioning

A *matroid* is a pair $M = (E, I)$ where E is the set of elements of M and I is a collection of subsets of E, called the *independent sets*, that satisfies certain axioms. Matroids are an axiomatic abstraction of the theory of linear dependence in vector spaces. We recommend [10] for a current and comprehensive survey of matroid theory.

A *matroid partitioning algorithm* takes as input matroids $M_i = (E, I_i)$ where $1 \leq i \leq r$. Note that all M_i have the same ground set E but in general their sets of independent sets may be different. The algorithm decides whether there exist sets $S_1, \ldots, S_r$ such that $S_i \in I_i$ for $1 \leq i \leq r$ (that is, S_i is an independent set with respect to the i-th matroid) and $\bigcup_{i=1}^{r} S_i = E$ and $S_i \cap S_j = \emptyset$ whenever $i \neq j$. If such a partition does exist, then the algorithm finds one such partition.

The first matroid partitioning algorithm was invented by Edmonds [5]. A very accessible description of Edmonds' algorithm, including a pseudocode for it, can be found in Section 8.7 of [8]. Assuming that matroid partition(s) do exist, the version of Edmonds' algorithm given in [8] finds a partition $S_1, \ldots, S_r$ such that the sequence $(|S_1|, |S_2|, \ldots, |S_r|)$ is *lexicographically maximal* among all sequences $(|U_1|, |U_2|, \ldots, |U_r|)$ where $U_1, \ldots, U_r$ is a matroid partition. Throughout the paper $|X|$ denotes the cardinality of set X.

The complexity analysis of Edmonds' algorithm in [8] shows that the algorithm runs in time $O(m^3 t)$ where $m = |E|$ and t is the maximum time required for testing whether $F \in I_i$, where F is some subset of E and i is some number between 1 and r.

For the type of matroids that we use in this paper, specialized matroid partitioning algorithms exist, see [4] and later references, and their time complexity is lower. However we choose to not go into more detail here, as all matroid partitioning algorithms run in polynomial time whereas the algorithms for computing the minimum weight of a linear code run overall in superpolynomial time (unless P = NP). Hence the matroid partitioning step will not be the bottleneck, not only asymptotically but also in practical computations, as we verified by a Magma implementation.

3 Brouwer-Zimmermann Minimum Weight Algorithm

An algorithm for computing the minimum weight of a linear code over a finite field was designed by A. Brouwer and subsequently extended by K.-H. Zimmermann. Henceforth we will refer to it as *Brouwer-Zimmermann algorithm,* abbreviated

BZ algorithm. A detailed description of the BZ algorithm can be found in [2, Section 1.8], in [1] and in [6]. A thorough implementation of the algorithm is available in Magma [3], and a description of the Magma implementation is given in [6]. In [7] the algorithm was adapted to finding minimum weights of $\mathbb{Z}_4$-linear quadratic residue codes.

Let C be a linear code whose minimum weight d we wish to determine. Let C^* be the set of non-zero codewords of C. Upon completion of each step, the BZ algorithm considers C^* as a disjoint union

$$C^* = C' \sqcup C''.$$

At this point, all elements of C' have been listed explicitly (but none of them needs to be stored permanently), thus yielding an upper bound $d \leq \overline{d}$ where $\overline{d}$ is the minimum weight of elements of C'. At the same time, the algorithm establishes a lower bound $\mathrm{wt}(c) \geq \underline{d}$, where c denotes an arbitrary element of C'', without listing any elements of C'' explicitly. If $\underline{d} \geq \overline{d}$, then the algorithm terminates with the message that the minimum weight of C equals $\overline{d}$. Otherwise, in the next step the algorithm augments the set C' so that it remains easy to lower bound the weights of elements of the new set C'', and the same process is repeated. It is desired that upon termination the set C' is as small as possible, since almost all effort of the algorithm is spent on listing the elements of C' and computing their weights.

Let G be a generator matrix for an $[n, k]_q$ code C. A subset $T \subseteq \{1, 2, \dots, n\}$ of size $|T| = k$ is called an *information set* for C if the corresponding columns in G are linearly independent. Then there also exists a generator matrix G_T for C such that the columns of G_T specified by T form an identity matrix. Each codeword of C is of the form uG_T for some $u \in \mathbb{F}_q^k$. We have

$$\mathrm{wt}(uG_T) \geq \mathrm{wt}(u) \qquad \text{for all } u \in \mathbb{F}_q^k. \tag{1}$$

3.1 Outline of the BZ Algorithm

Let C be an $[n, k]_q$ linear code with a generator matrix G. The BZ algorithm will determine the minimum weight of C as follows.

The algorithm will find subsets $T_1, \dots, T_\ell$ of $\{1, 2, \dots, n\}$ such that each T_i is an information set for C and $\bigcup_{i=1}^{\ell} T_i = \{1, 2, \dots, n\}$. Then Gauss-Jordan elimination is applied to G to construct matrices $G_1, \dots, G_\ell$ so that G_i is a generator matrix for C that has the identity matrix in the columns specified by T_i.

The sequence of non-negative integers $r_1, \dots, r_\ell$ is determined by

$$r_i := \left| T_i \setminus \bigcup_{j=1}^{i-1} T_j \right| \qquad \text{for } 1 \leq i \leq \ell. \tag{2}$$

The integers r_i are called *relative ranks* in [6], and we will follow this terminology.

We will assume that the sequence of relative ranks is *non-increasing,* that is,

$$r_1 \geq r_2 \geq \ldots \geq r_\ell.$$

The methods used in [2, 6] for constructing sets T_i and matrices G_i produce them in such a way that the sequence of relative ranks is non-increasing. Also, Eq. (4) shows that it is more beneficial for the BZ algorithm to operate on matrices with larger relative ranks prior to operating on matrices with smaller relative ranks.

While the algorithm is operating, it may decide to discard some matrices G_i because their relative ranks are so small that they will not get an opportunity to contribute to the lower bound on weights of codewords in C'' before the termination of the algorithm. Such decisions may be made on the fly, whenever the upper bound $\overline{d}$ decreases as a consequence of discovering a codeword of weight less than the previous value of $\overline{d}$. A numerical example of this phenomenon is given in [6]. From the complexity analysis point of view, predicting when such events will occur is not possible, yet the impact of these events on the running time of the algorithm is significant. In our analysis, we will assume that the algorithm operates throughout the entire computation on the sequence of matrices $G_1, \ldots, G_D$ where D is some integer such that $1 \leq D \leq \ell$. (The only exception may be the last iteration of the algorithm which operates on $G_1, \ldots, G_z$ for some $1 \leq z \leq D$.) We say that the algorithm operates up to *depth* D. We will show below that this assumption is consistent with one of the modes in which the BZ algorithm is used in practice. The value of D is determined once the sets $T_1, \ldots, T_\ell$ have been constructed. In [2] $D = \ell$ is always used. In [6] the choices for D are discussed: In the mode where the algorithm computes the minimum weight, the value of D is adjusted dynamically according to the changes of the value of $\overline{d}$. In the mode where the algorithm verifies a lower bound on the minimum weight, the optimal value of D can be determined in advance and it stays constant throughout the execution of the algorithm.

Each step of the algorithm is characterized by a pair of integers (w, j) where $1 \leq w \leq k$ and $1 \leq j \leq D$. The initial values of the main variables are $(w, j) := (1, 1)$ and $\overline{d} := n - k + 1$ (Singleton bound).

In step (w, j) the algorithm enumerates all codewords uG_j such that $\mathrm{wt}(u) = w$. During this process, whenever a codeword x is generated such that $\mathrm{wt}(x) < \overline{d}$, then we set $\overline{d} := \mathrm{wt}(x)$. In other words, the algorithm updates $\overline{d}$ according to

$$\overline{d} := \min\Big(\overline{d}\ ,\ \min\{\mathrm{wt}(uG_j)\ :\ u \in \mathbb{F}_q^k,\ \mathrm{wt}(u) = w\}\Big) \tag{3}$$

without storing the set of codewords in memory. The new value of $\underline{d}$ is determined as

$$\underline{d} := \sum_{i=1}^{j} \max(0, w + 1 - k + r_i) + \sum_{i=j+1}^{D} \max(0, w - k + r_i). \tag{4}$$

Using (1) it is easy to see [6] that any codeword $x \in C^*$ that has not been generated by the algorithm up to this point satisfies $\mathrm{wt}(x) \geq \underline{d}$. The algorithm now tests whether $\underline{d} \geq \overline{d}$. If this is the case, then the algorithm terminates with the message that the minimum weight of C equals $\overline{d}$. Otherwise, if $j < D$, then the algorithm proceeds to step $(w, j + 1)$, otherwise it proceeds to step $(w + 1, 1)$.

It is noted in [6] that the overall work of the algorithm can be reduced by the factor of $q - 1$ by using only left-normalized vectors u in (3), since codewords that are non-zero scalar multiples of each other have the same Hamming weight. The overall running time of the BZ algorithm is essentially determined by the total number of codewords of C that the algorithm generates during its execution. This value is called the *work factor* in [6]. Assuming that the algorithm terminates upon completing step (w, j), the work factor is

$$W(D, w, j) \;=\; j \sum_{z=1}^{w} \binom{k}{z} (q-1)^{z-1} + (D-j) \sum_{z=1}^{w-1} \binom{k}{z} (q-1)^{z-1}. \qquad (5)$$

3.2 Proving a Lower Bound on the Minimum Weight

The BZ algorithm can operate in different modes, and this is reflected for example by the fact that the Magma implementation of it [6] offers several different commands through which the algorithm can be invoked. The version of the algorithm that we outlined in Sect. 3.1 computes the minimum weight of C. It is also possible to use the algorithm to solve the following decision problem:

Given a linear code C and a positive integer L, is it true that the minimum weight of C is greater than or equal to L?

In order to solve this problem, the only modification required to the algorithm outlined in Sect. 3.1 is that the algorithm will terminate with output "yes" as soon as the inequalities $\overline{d} \geq L$ and $\underline{d} \geq L$ are both satisfied. If the algorithm ever comes across a codeword of weight less than L, then it will terminate with output "no." This mode of operation of BZ algorithm is available in Magma via the command `VerifyMinimumDistanceLowerBound`.

If the algorithm outputs "no," then it is in general impossible to predict the time at which the algorithm comes across a codeword of weight less than L. We will analyze the work factor in the case when the algorithm outputs "yes." Hence we are analyzing the complexity of using the BZ algorithm to *prove a lower bound on the minimum weight of a linear code.* In this case, the inequality $\overline{d} \geq L$ holds true throughout the execution of the algorithm. Hence we only need to analyze the work needed to obtain the inequality $\underline{d} \geq L$.

Given C and L as above, the algorithm starts by determining the information sets $T_1, \ldots, T_\ell$ and the corresponding relative ranks $r_1, \ldots, r_\ell$. Afterwards, the algorithm will consider in turn all possible values $D = 1, 2, \ldots, \ell$. For each such D the algorithm will determine the earliest (in the order of execution) pair (w, j) such that the right-hand side of (4) is greater than or equal to L, and it will compute the corresponding work factor $W(D, w, j)$ using (5). The value $D = D_0$ that minimizes the work factor required will be found, and the BZ algorithm will be invoked at this optimal depth D_0 to deliver the proof that L is a lower bound on the minimum weight of C.

4 Construction of Information Sets

The issue of finding information sets $T_1, \dots, T_\ell$ that yield a favourable sequence of relative ranks $r_1, \dots, r_\ell$ is a problem of its own. We address it in this section, and we will make our final comments about it in the next section.

It is intuitively clear from (4) that larger values of r_i should make the lower bound $\underline{d}$ grow faster, which means a faster completion of the algorithm. (Note that $r_i \le k$ for all i.) This motivates the following definition.

Definition 1 An *α-partition* of an $[n, k]_q$ linear code C is a partition of its set of coordinates into $\lfloor n/k \rfloor$ linearly independent sets of size k and, in case that k does not divide n, one linearly independent set of size $n \bmod k$.

Note that C has an α-partition if and only if there exists a sequence of information sets for C such that the corresponding sequence of relative ranks is

$$(k,\ \dots,\ k,\ n \bmod k) \tag{6}$$

where the last term is omitted if k divides n. It is easy to convert one object into the other one.

Proposition 2 *Let C be an $[n, k]_q$ linear code. There exists an algorithm with time complexity $O(n^3k^3)$ that decides whether C has an α-partition, and it outputs an α-partition of C if it exists.*

Proof Let G be a generator matrix for C. Let $E = \{1, \dots, n\}$ and let $r := \lceil n/k \rceil$. For $1 \le i \le r$ consider matroids $M_i = (E, I_i)$ defined as follows. Each set I_i consists of precisely those subsets $F \subseteq E$ such that the columns of G indexed by F are a linearly independent set in $\mathbb{F}_q^k$. Apply Edmonds' algorithm (Sect. 2) to $M_1, \dots, M_r$. The existence of an α-partition of C is equivalent to the existence of a matroid partition $S_1, \dots, S_r$ such that $(|S_1|, \dots, |S_r|) = (k, \dots, k)$ if k divides n, or $(|S_1|, \dots, |S_r|) = (k, \dots, k, n \bmod k)$ if k does not divide n. In either case this is the lexicographically maximal matroid partition possible, thus it will be found by Edmonds' algorithm in case that it exists.

For the conclusion about the running time, recall from Sect. 2 that Edmonds' algorithm runs in time $O(m^3t)$ where $m = |E|$ and t is the maximum time required for testing whether $F \in I_i$, where F is some subset of E and i is some number between 1 and r. In our case $m = |E| = n$. Let Z be an arbitrary subset of $E = \{1, \dots, n\}$ and suppose that we want to test whether $Z \in I_i$. (Recall that all sets

I_i are equal, hence the value of i is of no consequence.) If $|Z| > k$, then $Z \notin I_i$. If $|Z| \leq k$, then let Q denote the submatrix of G consisting of those columns of G indexed by Z. Then $Z \in I_i$ if and only if $\operatorname{rank}(Q) = |Z|$. This can be decided by Gauss-Jordan elimination in time $O(k^3)$. Overall testing whether $Z \in I_i$ can be done in time $O(k^3)$. Hence Edmonds' algorithm will run in time $O(n^3k^3)$. □

In [2, Section 1.8] it is noted that the BZ algorithm works efficiently if the code under consideration has many information sets which are pairwise disjoint. We now show that this objective can be achieved deterministically in polynomial time.

Proposition 3 *Let C be an $[n, k]_q$ linear code. There exists an algorithm with time complexity $O(n^3k^3)$ that determines N, the maximum number of pairwise disjoint information sets for C, and it finds a set of N pairwise disjoint information sets for C.*

Proof As in the proof of Proposition 2 we form the matroids M_i from the code C, except that we now take $r = n$, and we apply Edmonds' algorithm to them. Since Edmonds' algorithm delivers a matroid partition $S_1, \dots, S_r$ such that the sequence $(|S_1|, \dots, |S_r|)$ is lexicographically maximal among all matroid partitions, in particular the sequence $S_1, \dots, S_r$ will contain the maximum possible number of pairwise disjoint information sets for C. These will be the sets $S_1, \dots, S_N$ where N is the largest number i such that $|S_i| = k$. □

Our next statement shows that if an α-partition of C exists, then it is the optimal choice for the mode of the BZ algorithm that we study. The proof is skipped in this version due to the page limit; it will be included in the journal version of the paper.

Proposition 4 *If R is a sequence of relative ranks corresponding to an α-partition of C, then the work factor of the BZ algorithm for proving a lower bound on the minimum weight of C using R is less than or equal to the work factor when using any other sequence of relative ranks for C.*

5 Conclusion

The BZ algorithm starts its execution by finding information sets $T_1, \dots, T_\ell$ that yield the sequence of relative ranks $r_1, \dots, r_\ell$. It is clear from (4) that the sequence (r_i) has a serious impact on the operation of the algorithm, hence spending some effort on making a choice among available sequences (r_i) appears to be well justified.

In [2] the issue of choosing among different sequences (r_i) is not considered. The information sets are produced by one sweep of the generator matrix from left to right, by a sequence of Gaussian eliminations performed on rectangular matrices of decreasing size. This method guarantees $r_1 = k$ but not much can be inferred about the sequence (r_i) as a whole.

In [6] the choice among different sequences (r_i) is considered. The generator matrix is swept from left to right as in [2]. If the first pass fails to produce the sequence of relative ranks (6), then a random permutation is applied to the columns of the generator matrix, and the process is repeated over and over. The Magma implementation of the BZ algorithm uses this heuristic [6].

In this paper we present a deterministic algorithm running in time polynomial in n and k that finds information sets yielding the relative rank sequence (6), which has been deemed to be the most favourable situation for the BZ algorithm [6, p. 293], or it proves that this relative rank sequence can not be achieved for the code under investigation. We make some assumptions about the mode of operation of the BZ algorithm that allow us to assert that the sequence (6) is optimal in cases when it is achievable. We note that the same algorithm also always finds the maximum number of pairwise disjoint information sets for the given code, which is another objective that can be pursued [2]. Our timings show that the extra computation time required is negligible, hence we believe that we have proposed a useful extension to the Brouwer-Zimmermann algorithm.

Acknowledgements Research of both authors was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors thank Luis Goddyn for helpful comments and discussions.

References

1. Betten, A., Fripertinger, H., Kerber, A., Wassermann, A., Zimmermann, K.-H.: Codierungstheorie–Konstruktion und Anwendung Linearer Codes. Springer, Berlin (1998)
2. Betten, A., Braun, M., Fripertinger, H., Kerber, A., Kohnert, A., Wassermann, A.: Error-Correcting Linear Codes. Springer, Berlin/New York (2006)
3. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symb. Comput. **24**, 235–265 (1997)
4. Cunningham, W.H.: Improved bounds for matroid partition and intersection algorithms. SIAM J. Comput. **15**, 948–957 (1986)
5. Edmonds, J.: Minimum partition of a matroid into independent subsets. J. Res. Nat. Bur. Standards Sect. B **69B**, 67–72 (1965)
6. Grassl, M.: Searching for linear codes with large minimum distance. In: Bosma, W., Cannon, J. (eds.) Discovering Mathematics with Magma – Reducing the Abstract to the Concrete, pp. 287–313. Springer, Berlin/New York (2006)
7. Kiermaier, M., Wassermann, A.: Minimum weights and weight enumerators of $\mathbb{Z}_4$-linear quadratic residue codes. IEEE Trans. Inf. Theory **58**, 4870–4883 (2012)
8. Lawler, E.L.: Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York (1976)
9. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam/New York (1977)
10. Oxley, J.: Matroid Theory, 2nd edn. Oxford University Press, Oxford/New York (2011)
11. Vardy, A.: The intractability of computing the minimum distance of a code. IEEE Trans. Inf. Theory **43**, 1757–1766 (1997)

On the Design of Storage Orbit Codes

Shiqiu Liu and Frédérique Oggier

Abstract We propose the use of orbit codes to design storage codes, that is, subspace codes obtained from orbits of the action of subgroups of $GL_n(\mathbb{F}_q)$ on m-dimensional subspaces of $\mathbb{F}_q^n$. We translate the storage code parameters into those of the algebraic objects involved, and construct a simple family of storage orbit codes.

Keywords Orbit codes • Storage codes • Group action

1 Introduction

Let $\mathbb{F}_q$ be the finite field with q elements, with q a prime power. The set of all subspaces of $\mathbb{F}_q^n$ of dimension m is called *Grassmannian* and is denoted by $G_q(m,n)$:

$$G_q(m,n) = \{\mathscr{U} \text{ subspace of } \mathbb{F}_q^n,\ \dim(\mathscr{U}) = m\}.$$

We denote by $GL_n(\mathbb{F}_q)$ the set of $n \times n$ invertible matrices with coefficients in $\mathbb{F}_q$. Multiplication by elements of $GL_n(\mathbb{F}_q)$ defines a group action from the right on $G_q(m,n)$ by

$$\begin{aligned} G_q(m,n) \times GL_n(\mathbb{F}_q) &\longrightarrow G_q(m,n) \\ (\mathscr{U}, A) &\longmapsto \mathscr{U}A = \{uA,\ u \in \mathscr{U}\} \end{aligned}$$

since any element of $GL_n(\mathbb{F}_q)$ maps an m-dimensional subspace to an m-dimensional subspace. In fact, as pointed out in [6], since any two m-dimensional subspaces can be mapped onto each other by an element of $GL_n(\mathbb{F}_q)$, $GL_n(\mathbb{F}_q)$ acts transitively on $G_q(m,n)$.

S. Liu (✉) • F. Oggier (✉)
Division of Mathematical Sciences, Nanyang Technological University, Singapore, Singapore
e-mail: sliu012@e.ntu.edu.sg; frederique@e.ntu.edu.sg

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_28

Codes whose codewords are elements of $G_q(m,n)$ are popular for error correction in network coding [1], where the distance of interest between codewords is the subspace distance. They are sometimes referred to as constant dimension codes. Constant dimension codes have been obtained from orbits of a cyclic group of $GL_n(\mathbb{F}_q)$ in [6], together with a decoding algorithm for a subclass of such codes. In this paper, we will consider the problem of designing codes for distributed storage systems.

Many methods have been proposed to construct storage codes, see e.g. [4] for a survey of different code designs and constructions. Here is an example to illustrate how codes are used in the context of distributed storage.

Example 1 Suppose a data object $(u_0, u_1) \in \mathbb{F}_2^2$ of size 2 needs to be stored across a set of 4 nodes. Use for example a (4,2)-binary cyclic with generator polynomial $x^2 + 1$ (note that $x^4 - 1 = (x^2 + 1)(x^2 - 1)$). Then a codeword is of the form

$$[u_0, u_1] \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = [u_0, u_1 u_0, u_1].$$

Then node 1 stores u_0, node 2 u_1, node 3 u_0 and node 4 u_1, that is every codeword coefficient is assigned to be stored by one node. This code protects the data object against one node failure: if any one node fails, it is still possible to recover the data object. Furthermore, any one node failure can be repaired by contacting one node.

We are interested in storage codes whose codewords are elements of $G_q(m,n)$, but whose design criterion varies from that of constant dimension codes. In [3], such storage codes were built from cliques within the Grassmannian graph. We will present in Sect. 2 another approach to the design of storage codes, that of orbit codes, following the terminology of [6]. In Sect. 3, we propose a family of cyclic orbit codes suitable for collaborative repair (the meaning of collaborative repair will be explain below).

2 Storage Codes

Fix $\mathscr{U} \in G_q(m,n)$, let G be a subgroup of $GL_n(\mathbb{F}_q)$, and let $\mathscr{U}G = \{\mathscr{U}g,\ g \in G\}$ be the orbit of $\mathscr{U}$ under the right action of G. We will refer to $\mathscr{U}G$ as an *orbit code*.

In order to represent an m-dimensional subspace $\mathscr{U}$ of the vector space $\mathbb{F}_q^n$, we fix a basis of $\mathscr{U}$, and use an $m \times n$ matrix U whose row space $\{vU,\ v \in \mathbb{F}_q^m\}$ is $\mathscr{U}$.

A storage code $\mathscr{C}$, in the context of networked distributed storage [4], aims at encoding a data object $\mathbf{o} \in \mathbb{F}_q^n$ into N network nodes, such that the object can be retrieved from a subset of live nodes in case of node failures. In the case of linear codes, every node stores the inner products of $\mathbf{o} \in \mathbb{F}_q^n$ with some (say m, $m \geq 1$) vectors in $\mathbb{F}_q^n$, thus the node may compute linear combinations of these inner products, and thus is seen as storing an m-dimensional vector space, the span of these m vectors which are assumed to be linearly independent, without loss of generality. The main difference between a storage code and an erasure code is that a storage code should be amenable to repair, namely, the code should be such that the data stored at one node can be computed from a (small) subset of other nodes, without (necessarily) having to decode the object first. Alternatively, in the case of collaborative repair, a subset of nodes can be computed from another subset of repair nodes, allowing these repair nodes to exchange data among each other [4].

Let us translate these storage parameters for an orbit code $\mathscr{U}G$:

1. The number of storage nodes is the cardinality $|\mathscr{U}G|$ of the orbit, and it is well known that
$$|\mathscr{U}G| = \frac{|G|}{|Stab_G(\mathscr{U})|},$$
where $Stab_G(\mathscr{U}) = \{g \in G,\ \mathscr{U}g = \mathscr{U}\}$ is a subgroup of G called the stabilizer of $\mathscr{U}$,
2. The storage capacity (or number of stored symbols in $\mathbb{F}_q$) for every node is m,
3. The size of the object to be stored is n.

Remark 2 Note that nodes are storing the inner product of the object $\mathbf{o}$ with basis vectors, thus we associate to a basis vector $\mathbf{v}$ the inner product between $\mathbf{o}$ and $\mathbf{v}$. Consequently, we say that a node stores $\mathbf{v}$ instead of saying that it stores $\mathbf{vo}^T$, which makes it easier to translate storage codes in terms of group action. However, in terms of storage, $\mathbf{vo}^T$ is one element of $\mathbb{F}_q$, not n.

A standard way to evaluate the performance of a storage code is to compare it with the optimal trade-off curve between the amount of storage per node, and the repair bandwidth. This trade-off curve is usually computed from a min-cut bound. We recall what is the min-cut bound for the most general setting, where the level of collaboration varies, going from no collaboration to full collaboration, and any regime in between. This min-cut bound is [2], keeping the original notation of the trade-off curve:

$$\begin{aligned} M \le \min_{\mathbf{u}\in P}(\sum_{i\in I} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta'\} \\ + \sum_{i\in \bar{I}} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta\}) \end{aligned} \tag{1}$$

where

$$I = \{i,\ t - s + 1 - u_i \ge 0\},\ \bar{I} = \{i,\ t - s + 1 - u_i < 0\}$$

and

$$P = \{\mathbf{u} = (u_0, \ldots, u_{g-1}),\ 1 \le u_i \le t \text{ and } \sum_{i=0}^{g-1} u_i = k\}.$$

Parameters in the above bound are:

M : *the size of an object*
k : *any choice of k nodes should allow the object retrieval*
α : *storag capacity per node*
γ : *repair bandwidth per node*
β : *the download repair bandwidth*
β' : *the collaboration repair bandwidth*
d : *number of live nodes contacted*
t : *the threshold at which a repair process is triggered*
$t - s$: *the number ofother repair nodes involved in collaboration*

At the two extreme regimes, the Minimum Storage Repair (MSR) and the Minimum Repair Bandwidth Repair (MBR), we have:

$$\boxed{\begin{array}{l} MSR : \alpha = \frac{M}{k},\ \gamma = \frac{M}{k}\frac{d+t-s}{d-k+t-s+1}, \\ \beta = \beta' = \frac{M}{k}\frac{1}{d-k+t-s+1}. \end{array}}$$

$$\boxed{\begin{array}{l} MBR : \alpha = \gamma = \frac{M}{k}\frac{2d+t-s}{2d-k+t-s+1}, \\ \beta = \frac{M}{k}\frac{2}{2d-k+t-s+1},\ \beta' = \frac{M}{k}\frac{1}{2d-k+t-s+1}. \end{array}}$$

When $s = 1$, we get full collaboration case, while $s = t$ corresponds to the situation with no collaboration.

Example 3 Let $G = \langle g \rangle$ be the subgroup of $GL_5(\mathbb{F}_2)$ generated by

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

Consider the orbit code $\mathscr{U}G$, where $\mathscr{U}$ has for basis

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Since g has order 4 and $Stab_G(\mathscr{U})$ is trivial, the orbit code $\mathscr{U}G$ is

$$U, \begin{bmatrix} 0&0&1&0&0 \\ 0&0&0&0&1 \\ 1&1&1&1&1 \end{bmatrix}, \begin{bmatrix} 1&1&1&1&1 \\ 0&1&0&0&0 \\ 0&0&0&1&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&1&0 \\ 1&0&0&0&0 \\ 0&0&0&0&1 \end{bmatrix}.$$

This corresponds to a storage code $\mathscr{C}$ where the size of the object $\mathbf{o}$ is $n = 5$, every storage node stores $m = 3$ symbols (e.g., the first node corresponding to the subspace $\mathscr{U}$ stores $(1, 0, 0, 0, 0)\mathbf{o}^T = o_1$, $(0, 1, 0, 0, 0)\mathbf{o}^T = o_2$, $(0, 0, 1, 0, 0)\mathbf{o}^T = o_3$ for $\mathbf{o} = (o_1, \ldots, o_5) \in \mathbb{F}_q^5$). The data object $\mathbf{o}$ may be retrieved out of any two nodes, since $\dim(\mathscr{U}g^i + \mathscr{U}g^j) = 5$ for all $i \neq j$. In case of one node failure, the subspace $\mathscr{U}g^i$ may be computed from the knowledge of a subspace of dimension 1 from each of the other $\mathscr{U}g^j$, $j \neq i$, or in other words $\dim(\mathscr{U}g^i \cap \mathscr{U}g^j) = 1$ for all $i \neq j$. This code instance has been reported in [5], which is an MBR code corresponding to the no collaboration scenario, with parameters (keeping the min-cut bound notation): the file size $M = 5$, the storage capacity $\alpha = m = 3$, $k = 2$, $d = 3$, $t = 2$, $\beta = 1$.

3 A Simple Instance of Cyclic Orbit Codes

We next propose a family of cyclic orbit codes and compute the parameters of the corresponding storage codes.

Lemma 4 *Let $G = \langle g \rangle$ be the subgroup of $GL_n(\mathbb{F}_q)$ generated by the $n \times n$ matrix*

$$g = \begin{bmatrix} 0\ 1\ 0 & & 0 \\ 0\ 0\ 1 & & 0 \\ & \ddots & \\ 0\ 0\ 0 & \ldots & 1 \\ 1\ 1\ 1 & \ldots & 1 \end{bmatrix},$$

and let U contain a canonical basis of $G_2(n-1, n)$. Then any two distinct elements $\mathscr{U}g^i$ and $\mathscr{U}g^j$ of the orbit $\mathscr{U}G$ of $\mathscr{U}$ under the right action of G intersect in a subspace of dimension $n-2$. Furthermore, the size of the orbit is $n+1$.

Proof The size of the orbit $\mathscr{U}G$ is the order of g which is $n+1$ since the stabilizer is trivial. Since

$$\dim(\mathscr{U}g^i + \mathscr{U}g^j) = 2(n-1) - \dim(\mathscr{U}g^i \cap \mathscr{U}g^j) \leq n,$$

it must be that

$$n-2 \le \dim(\mathscr{U}g^i \cap \mathscr{U}g^j)$$

which concludes the proof, since $\mathscr{U}g^i$ and $\mathscr{U}g^j$ are distinct for $i \neq j$. □

Proposition 5 *The orbit $\mathscr{U}G$ forms a code $\mathscr{C}$ such that*

1. *The object is retrieved by contacting any two nodes,*
2. *The repair of two failures is done by downloading $n-2$ elements of $\mathbb{F}_q$ from one node for each failure, and exchanging one element of $\mathbb{F}_q$ between the two repair nodes.*

Recall Remark 2 for counting the size of elements downloaded/repaired.

Proof Label the $n+1$ storage nodes from 0 to n. By assumption, the ith node stores the element $\mathscr{U}g^i$ of the orbit $\mathscr{U}G$ of $\mathscr{U}$ under the right action of G, $i = 0, 1, \dots, n$.

When contacting any two nodes, we get two elements of the orbit $\mathscr{U}G$, say $\mathscr{U}g^i$ and $\mathscr{U}g^j$, $i \neq j$, and we have

$$\begin{aligned}\dim(\mathscr{U}g^i \cup \mathscr{U}g^j) &= \dim(\mathscr{U}g^i) + \dim(\mathscr{U}g^j) - \dim(\mathscr{U}g^i \cap \mathscr{U}g^j)\\ &= (n-1) + (n-1) - (n-2) = n,\end{aligned}$$

hence we can retrieve the object.

Suppose two nodes have failed, say node s and node t, download from node i the subspace $A = \mathscr{U}g^i \cap \mathscr{U}g^s$ of dimension $n-2$, and from node j the subspace $B = \mathscr{U}g^j \cap \mathscr{U}g^t$ also of dimension $n-2$, by the above lemma. If $\dim(A \cup B) = n$, the repair process can be completed by collaboration, by exchanging the missing basis vector at each repair node. Indeed, since $\dim(A \cup B) = n$, write the missing basis vector a at node s in a basis $\{v_1, \dots, v_n\}$ of $A \cup B$ as $a = \sum_{i=1}^n a_i v_i$. If $v_1, \dots, v_l \in A$, ask the symbol $\sum_{i=l+1}^n a_i v_i$ from B. Iterate this process for the missing basis vector at node t.

We are left to show that $\dim(A \cup B) = n$, or in fact, since

$$\dim(A \cup B) = \dim(A) + \dim(B) - \dim(A \cap B) = 2(n-2) - \dim(A \cap B),$$

to show that $\dim(A \cap B) = n-4$.

Write $\mathscr{U}g^i = \langle g_1, \dots, g_{n-2}, g_{i_0} \rangle$, $\mathscr{U}g^j = \langle g_1, \dots, g_{n-2}, g_{j_0} \rangle$. Then $A = \mathscr{U}g^s \cap \mathscr{U}g^i = \langle g_1, \dots g_{s_0-1}, g_{s_0+1}, \dots, g_{n-2}, g_{i_0} \rangle$, otherwise the nodes i, j and s would intersect in the same subspace of dimension $n-2$, which is not possible by definition of G, and for the same reason $B = \mathscr{U}g^t \cap \mathscr{U}g^j = \langle g_1, \dots g_{t_0-1}, g_{t_0+1}, \dots, g_{n-2}, g_{j_0} \rangle$. Without loss of generality, suppose that $s_0 < t_0$. Then

$$A \cap B = \langle g_1, \dots, g_{s_0-1}, g_{s_0+1}, \dots, g_{t_0-1}, g_{t_0+1}, \dots g_{n-2} \rangle$$

which has dimension $n-4$. □

From the code construction above, we can see that

$$\boxed{\begin{array}{l} M = n,\ d = 1,\ k = 2,\ t = 2,\ t - s = 1, \\ \alpha = n - 1,\ \gamma = n - 1,\ \beta = n - 2,\ \beta' = 1. \end{array}}$$

This code satisfies (1), and is optimal when $n = 4$.

Example 6 Consider $G_2(3, 4)$, and the subspace $\mathscr{U}$, with basis

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Let $G = \langle g \rangle$ be the cyclic subgroup of $GL_4(\mathbb{F}_q)$ generated by

$$g = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

The order of g is 5. The orbit of $\mathscr{U} \in G_2(3, 4)$ is by definition

$$\mathscr{U}G := \{\mathscr{U}g^i,\ 0 \leq i \leq 4\}.$$

The elements of the orbit are explicitly given by

$$U,\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix},\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

We store them in node 0 to 4. Assume that node 1 and node 4 failed. To repair node 1, download $\{(0010), (0001)\}$ from node 2, to repair node 4, download $\{(1000), (0100)\}$ from node 0, then during collaboration, the node repairing node 1 can compute (0011) and send it, and the node repairing node 4 sends (0100) in exchange (Fig. 1). Note that the strategy is not unique. To repair node 1, download alternatively $\{(0001), (0110)\}$ from node 3, to repair node 4, get $\{(1100), (1111)\}$ from node 2, then the node repairing node 1 can get (0011) and the one repairing node 4 can get (0111) from the collaboration.

To compare with Example 1, consider $G_2(1, 2)$, and the subspace $\mathscr{U}$, with basis $U = [1\ 0]$. Let $G = \langle g \rangle$ be the cyclic subgroup of $GL_2(\mathbb{F}_q)$ generated by $g = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$. The order of g is 3. The orbit of $\mathscr{U} \in G_2(1, 2)$ is by definition $\mathscr{U}G := \{\mathscr{U}g^i,\ i = 0, 1, 2\}$. The elements of the orbit are explicitly given by

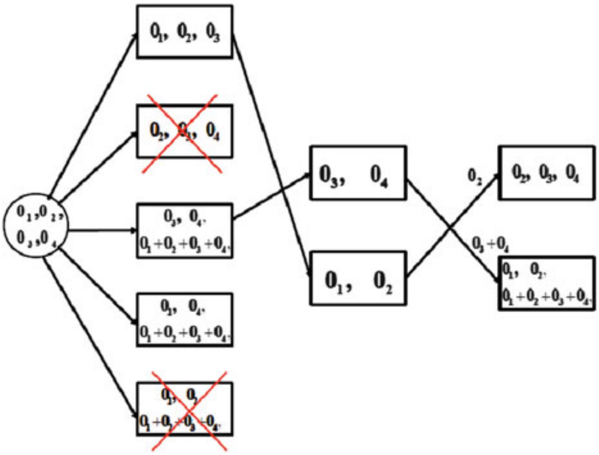


Fig. 1 A data object $\mathbf{o} = (\mathbf{o}_1, \ldots, \mathbf{o}_4)$ is stored in 5 nodes node 0 to node 4. Node 1 and node 4 failed. The two nodes that repair them connect to node 2 and node 0 to download 2 pieces of data, and exchange 1 piece of data among each other: $\alpha = \gamma = 3$. This strategy satisfies the full collaboration scenario with parameters at MBR point

[1 0], [0 1], [1 1]. Comparing with Example 1, our method also stores an object of size 2, however we only need 3 nodes to store the data object, thus improving the storage overhead, while still being able to tolerate one failure. When repairing only one node, our approach needs to connect 2 nodes, which is not as efficient as the code in Example 1.

Note finally that the main difference of our approach with respect to the use of classical erasure codes is that it gives a way to handle several symbols at each node (instead of 1, as in Example 1).

4 Future Work

In this paper, we translated the parameters of storage codes into those of orbit codes, illustrated how one known instance of storage code be seen as an orbit code, and constructed a family of storage codes suitable for collaborative repair.

Future research directions naturally include:

- The construction of other storage codes, using other cyclic groups, but also other finite groups,
- The analysis and comparison of the code parameters with both the codes available in the literature, but also the known bounds.

Acknowledgements The research of S. Liu is supported by the Singapore National Research Foundation under Research Grant NRF-RF2009-07. The research of F. Oggier for this work is supported by the MoE Tier-2 grant "eCODE: Erasure Codes for Datacenter Environments".

References

1. Koetter, R., Kschischang, F.R.: Coding for errors and erasures in random network coding. IEEE Trans. Inf. Theory **54**(8), 3579–3591 (2008)
2. Liu, S., Oggier, F.: On storage codes allowing partially collaborative repairs. In: Proceedings of the International Symposium on Information Theory, Honolulu, pp. 2440–2444. IEEE (2014)
3. Oggier, F.: Some constructions of storage codes from grassmann graphs. In: Proceedings of the International Zurich Seminar on Communications, Zürich (2014)
4. Oggier, F., Datta, A.: Coding techniques for repairability in networked distributed storage systems. Found. Trends® Commun. Inf. Theory **9**(4), 383–466 (2012). doi:10.1561/0100000068
5. Rashmi, K., Shah, N., Kumar, P., Ramchandran, K.: Explicit construction of optimal exact regenerating codes for distributed storage. In: Proceedings of the Allerton Conference on Communication, Control, and Computing, Monticello, vol. 1–2, pp. 1243–1249. IEEE, Urbana Champaign (2009)
6. Trautmann, A.L., Manganiello, F., Braun, M., Rosenthal, J.: Cyclic orbit codes. IEEE Trans. Inf. Theory **59**(11), 7386–7404 (2013)

Hadamard $\mathbb{Z}_2\mathbb{Z}_4 Q_8$-Codes: Rank and Kernel

Pere Montolio and Josep Rifà

Abstract Hadamard $\mathbb{Z}_2\mathbb{Z}_4 Q_8$-codes are Hadamard binary codes coming from a subgroup of the direct product of $\mathbb{Z}_2$, $\mathbb{Z}_4$ and Q_8 groups, where Q_8 is the quaternionic group. We characterize Hadamard $\mathbb{Z}_2\mathbb{Z}_4 Q_8$-codes as a quotient of a semidirect product of $\mathbb{Z}_2\mathbb{Z}_4$-linear codes and we show that all these codes can be represented in a standard form, from a set of generators. On the other hand, we show that there exist Hadamard $\mathbb{Z}_2\mathbb{Z}_4 Q_8$-codes with any given pair of allowable parameters for the rank and dimension of the kernel.

Keywords Dimension of the kernel • Error-correcting codes • Hadamard codes • Rank • $\mathbb{Z}_2\mathbb{Z}_4$-linear codes • $\mathbb{Z}_2\mathbb{Z}_4 Q_8$-codes

1 Introduction

Non-linear codes with a group structure (like $\mathbb{Z}_2\mathbb{Z}_4$-linear codes and $\mathbb{Z}_2\mathbb{Z}_4 Q_8$-codes) have received a great deal of attention since [2]. The codes in this paper can be characterized as the image of a subgroup, by a suitable Gray map, of the direct product of $\mathbb{Z}_2$, $\mathbb{Z}_4$ and Q_8, the quaternionic group of order 8 [6, 8].

Hadamard matrices with a subjacent algebraic structure have been deeply studied, as well as the links with other topics in algebraic combinatorics or applications [3]. We quote just a few papers about this subject [1, 4, 7], where we can find beautiful equivalences between Hadamard groups, 2-cocyclic matrices and relative difference sets. On the other hand, from a coding theory point of view, it is desirable that the algebraic structures we are dealing with preserves the Hamming distance. This is the case, for example, of the $\mathbb{Z}_2\mathbb{Z}_4$-linear codes which has been

P. Montolio (✉)
Computing, Multimedia and Telecommunication Studies, Universitat Oberta de Catalunya, Barcelona, Spain
e-mail: pma@uoc.edu

J. Rifà
Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Barcelona, Spain
e-mail: josep.rifa@uab.cat

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_29

intensively studied during the last years [2]. More generally, the propelinear codes and, specially those which are translation invariant, are particularly interesting because the subjacent group structure has the property that both, left and right product, preserve the Hamming distance. Translation invariant propelinear codes has been characterized as the image of a subgroup by a suitable Gray map of a direct product of $\mathbb{Z}_2$, $\mathbb{Z}_4$ and Q_8 [6].

In this paper we analyze codes that have both properties, being Hadamard and $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes. These codes were previously studied and classified [8] in five shapes. The aim of this paper is to go further. First of all by giving an standard form for a set of generators of the code, depending on the parameters, which helps to understand of the characteristics of each shape and then by putting the focus in an exact computation of the values of the rank and dimension of the kernel. One of the main results of this paper is to characterize the $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes as a quotient of a semidirect product of Hadamard $\mathbb{Z}2\mathbb{Z}_4$-linear codes. The second main result is to construct, using the above characterization, Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes whose values for the rank and dimension of the kernel are any allowable pair previously chosen.

The structure of the paper is as follows. Section 2 introduces the notation and preliminary concepts; Sect. 3 shows the standard form of generators that allows to represent any Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code in a unique way, this section finishes with two important theorems which characterizes a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code as a quotient of a semidirect product of $\mathbb{Z}_2\mathbb{Z}_4$-linear codes (Theorems 2 and 3). Finally, in Sect. 4 we give the constructions of $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes fulfilling the requirements for the prefixed values of the dimension of the kernel and rank. We finish the last section with a couple of examples about the constructions and achievement of codes with each allowable pair of values for the rank and dimension of the kernel.

2 Preliminaries

Let $\mathbb{Z}_2$ and $\mathbb{Z}_4$ denote the binary field and the ring of integers modulo 4, respectively. Any non-empty subset of $\mathbb{Z}_2^n$ is called a binary code and a linear subspace of $\mathbb{Z}_2^n$ is called a *binary linear code* or a *$\mathbb{Z}_2$-linear code*. Let $\mathrm{wt}(v)$ denote the *Hamming weight* of a vector $v \in \mathbb{Z}_2^n$ (i.e., the number of its nonzero components), and let $d(v,u) = \mathrm{wt}(v+u)$, the *Hamming distance* between two vectors $v, u \in \mathbb{Z}_2^n$.

Let Q_8 be the *quaternionic group* on eight elements. The following equalities provides a presentation and the list of elements of Q_8:

$$Q_8 = \langle \mathbf{a}, \mathbf{b} \,:\, \mathbf{a}^4 = \mathbf{a}^2\mathbf{b}^2 = \mathbf{1}, \mathbf{bab}^{-1} = \mathbf{a}^{-1}\rangle = \{\mathbf{1}, \mathbf{a}, \mathbf{a}^2, \mathbf{a}^3, \mathbf{b}, \mathbf{ab}, \mathbf{a}^2\mathbf{b}, \mathbf{a}^3\mathbf{b}\}.$$

Given three non-negative integers k_1, k_2 and k_3, denote as $\mathscr{G}$ the group $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times Q_8^{k_3}$. Any element of $\mathscr{G}$ can be represented as a vector where the first k_1 components belong to $\mathbb{Z}_2$, the next k_2 components belong to $\mathbb{Z}_4$ and the last k_3 components belong to Q_8.

We use the multiplicative notation for $\mathscr{G}$ and we denote by $\mathbf{e}$ the identity element of the group and by $\mathbf{u}$ the element with all components of order two. Hence, $\mathbf{e}=(0,\overset{k_1+k_2}{\ldots},0,\mathbf{1},\overset{k_3}{\ldots},\mathbf{1})$ and $\mathbf{u}=(1,\overset{k_1}{\ldots},1,2,\overset{k_2}{\ldots},2,\mathbf{a}^2,\overset{k_3}{\ldots},\mathbf{a}^2)$.

We call *Gray map* the function Φ:

$$\Phi\,:\;\mathbb{Z}_2^{k_1}\times\mathbb{Z}_4^{k_2}\times Q_8^{k_3}\;\longrightarrow\;\mathbb{Z}_2^{k_1+2k_2+4k_3},$$

acting componentwise in such a way that over the binary part is the identity, over the quaternary part acts as the usual Gray map, so $0\rightarrow(00)$, $1\rightarrow(01)$, $2\rightarrow(11)$, $3\rightarrow(10)$ and over the quaternionic part acts in the following way [8]: $\mathbf{1}\rightarrow(0,0,0,0)$, $\mathbf{b}\rightarrow(0,1,1,0)$, $\mathbf{a}\rightarrow(0,1,0,1)$, $\mathbf{ab}\rightarrow(1,1,0,0)$, $\mathbf{a}^2\rightarrow(1,1,1,1)$, $\mathbf{a}^2\mathbf{b}\rightarrow(1,0,0,1)$, $\mathbf{a}^3\rightarrow(1,0,1,0)$, $\mathbf{a}^3\mathbf{b}\rightarrow(0,0,1,1)$.

Note that $\Phi(\mathbf{e})$ is the all-zeros vector and $\Phi(\mathbf{u})$ is the all-ones vector.

Let $\mathscr{C}$ be a subgroup of $\mathbb{Z}_2^{k_1}\times\mathbb{Z}_4^{k_2}\times Q_8^{k_3}$. Binary codes $C=\Phi(\mathscr{C})$ are called $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes. In the specific case $k_3=0$, code C is called $\mathbb{Z}_2\mathbb{Z}_4$-linear. In this last case, note that $\mathscr{C}$ is isomorphic to $=\mathbb{Z}_2^{\gamma}\times\mathbb{Z}_4^{\delta}\subset\mathbb{Z}_2^{k_1}\times\mathbb{Z}_4^{k_2}$. We will say that $\mathscr{C}$ is of type $2^{\gamma}4^{\delta}$ [2].

We are interested in Hadamard binary codes $C=\Phi(\mathscr{C})$ where $\mathscr{C}$ is a subgroup of $\mathscr{G}=\mathbb{Z}_2^{k_1}\times\mathbb{Z}_4^{k_2}\times Q_8^{k_3}$ of length $n=2^m$. All through the paper we are assuming it.

The *kernel* of a binary code C of length n is $K(C)=\{z\in\mathbb{Z}_2^n : C+z=C\}$. The dimension of $K(C)$ is denoted by $k(C)$ or simply k. The *rank* of a binary code C is the dimension of the linear span of C. It is denoted by $r(C)$ or simply r.

A *Hadamard matrix* of order n is a matrix of size $n\times n$ with entries ±1, such that $HH^T=nI$. Any two rows (columns) of a Hadamard matrix agree in precisely $n/2$ components. If $n>2$ then any three rows (columns) agree in precisely $n/4$ components. Thus, if $n>2$ and there is a Hadamard matrix of order n then n is multiple of 4.

Two *Hadamard matrices* are *equivalent* if one can be obtained from the other by permuting rows and/or columns and multiplying rows and/or columns by -1. With the last operations we can change the first row and column of H into $+1$'s and we obtain an equivalent Hadamard matrix which is called normalized. If $+1$'s are replaced by 0's and -1's by 1's, the initial Hadamard matrix is changed into a (binary) Hadamard matrix and, from now on, we will refer to it when we deal with Hadamard matrices. The binary code consisting of the rows of a (binary) Hadamard matrix and their complements is called a (binary) *Hadamard code*, which is of length n, with $2n$ codewords, and minimum distance $n/2$.

Let $C=\Phi(\mathscr{C})$ be a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code of length 2^m. Set $|T(\mathscr{C})|=2^{\sigma}$, $|Z(\mathscr{C})/T(\mathscr{C})|=2^{\delta}$ and $|\mathscr{C}/Z(\mathscr{C})|=2^{\rho}$, where $T(\mathscr{C})$ is the subgroup of elements of order two, $Z(\mathscr{C})$ is the center of $\mathscr{C}$. Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes were studied in [8] and classified in five different shapes based on the parameters σ,δ,ρ.

There are two important tools that has been used in the technical proofs of the statements throughout this paper, the commutator and the swapper.

Two elements a and b of $\mathscr{C}$ commutes if and only if $ab=ba$. As an extension of this concept, the *commutator* of a and b is defined as the element $[a,b]$ such that

$ab = [a, b]ba$. Note that all commutators belong to $T(\mathscr{C})$ and any element of $T(\mathscr{C})$ commutes with all elements of $\mathscr{C}$.

We say that two elements a and b of $\mathscr{C}$ *swap* if and only if $\Phi(ab) = \Phi(a)+\Phi(b)$. As an extension of this concept, define the *swapper* of a and b as the element $(a{:}b)$ such that $\Phi((a{:}b)ab) = \Phi(a) + \Phi(b)$. Note that all swappers belong to $T(\mathscr{G})$ but they can be out of $\mathscr{C}$. In other words, for any element a of $\mathscr{C}$ we have $\Phi(a) \in K(C)$ if and only if $(a{:}b) \in \mathscr{C}$, for every $b \in \mathscr{C}$. Moreover, the linear span of C can be seen as $\Phi(\langle \mathscr{C} \cup S(\mathscr{C})\rangle)$, where $\langle \mathscr{C} \cup S(\mathscr{C})\rangle$ is the group generated by $\mathscr{C}$ and $S(\mathscr{C})$, the set of swappers of the elements in $\mathscr{C}$.

3 The Standard Form for the Generator Set of a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-Code

In this section, starting from a given a $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code we construct a standard generator set, which allow to characterize it.

Proposition 1 *Let $\mathscr{C}$ be a subgroup of $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times Q_8^{k_3}$ such that $C = \Phi(\mathscr{C})$ is a Hadamard code. We can always construct a standard set of generators $\{x_1, \dots, x_\sigma;$ $r_1, \dots, r_\tau; s_1, s_\upsilon\}$ of $\mathscr{C}$ such that:*

- *The elements x_i are of order two and generate $T(\mathscr{C})$.*
- *The elements r_i are of order four and commute with each other, $[r_i, r_j] = \mathbf{e}$ for every $1 \leq i, j \leq \tau$. When $\mathbf{u} \in \langle r_1 \dots r_\tau\rangle$ we will take $\mathbf{u} = r_1^2$ and we have $r_1^2 = \mathbf{u} \notin \langle r_2^2 \dots r_\tau^2\rangle$.*
- *The cardinal υ of the set $\{s_1, s_\upsilon\}$ is in $\{0, 1, 2\}$ and when $\upsilon = 2$ we have $s_1^2 = \mathbf{u} \neq s_2^2$, and $[s_1, s_2] = \mathbf{e}$. Moreover, when $r_1^2 = s_1^2 = \mathbf{u}$ then $[r_1, s_1] = \mathbf{u}$.*
- *Any element $c \in \mathscr{C}$ can be written in a unique way as*

$$c = \prod_{i=1}^{\sigma} x_i^{a_i} \prod_{j=1}^{\tau} r_j^{b_j} \prod_{k=1}^{\upsilon} s_k^{c_k}, \quad \text{where } a_i, b_j, c_k \in \{0, 1\}.$$

The next theorem shows that a subgroup $\mathscr{C}$, such that $\phi(\mathscr{C})$ is a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code, has an abelian maximal subgroup $\mathscr{A}$ which is normal in $\mathscr{C}$ and $\mathscr{C}/\mathscr{A}$ is an abelian group of order 2^a, for $a \in \{0, 1, 2\}$.

Theorem 2 *Let $\mathscr{C}$ be a subgroup of $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times Q_8^{k_3}$ such that $\phi(\mathscr{C}) = C$ is a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code. Then $\mathscr{C}$ has an abelian maximal subgroup $\mathscr{A}$ which is normal in $\mathscr{C}$ and $|\mathscr{C}/\mathscr{A}| \in \{1, 2, 4\}$. Further, $\mathscr{C}$ may be expressed as a quotient of a semidirect product of $\mathscr{A}$.*

The next result characterizes the maximal abelian subgroup $\mathscr{A}$ and, since Hadamard $\mathbb{Z}_2\mathbb{Z}_4$-linear codes are well known [5], it will make possible the construction of all Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes.

Table 1 Existence conditions and parameters k_1, k_2, k_3 depending on the shape of Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes of length $n = 2^m$, where $m = \sigma + \tau + \upsilon - 1$. For all starred shapes $r_1^2 = \mathbf{u}, \overline{\tau} = \tau - 1$ and for all non-starred shapes $r_1^2 \neq \mathbf{u}, \overline{\tau} = \tau$

Shape	$\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times Q_8^{k_3}$			$\mathscr{C}$	Existence
	k_1	k_2	k_3		
1*	0	$2^{\sigma+\tau-2}$	0	$\mathscr{A}$	$\forall \tau \leq \lfloor \frac{m+1}{2} \rfloor$; $\sigma = m - \tau + 1$
1	$2^{\sigma-1}$	$(2^\tau - 1)2^{\sigma-2}$	0	$\mathscr{A}$	$\forall \tau \leq \lfloor \frac{m}{2} \rfloor$; $\sigma = m - \tau + 1$
2	0	0	$2^{\sigma+\tau-2}$	$\mathscr{A} \rtimes \mathbb{Z}_4 / \langle (\mathbf{u}, s_1^2) \rangle$	$\forall \tau \leq \lfloor \frac{m}{2} \rfloor$; $\sigma = m - \tau$
3	0	$2^{\sigma-1}$	$(2^\tau - 1)2^{\sigma-2}$	$\mathscr{A} \rtimes \mathbb{Z}_4 / \langle (\mathbf{u}, s_1^2) \rangle$	$\forall \tau \leq \lfloor \frac{m-1}{2} \rfloor$; $\sigma = m - \tau$
4	$2^{\sigma-1}$	0	$2^{\sigma-3}$	$\mathscr{A} \rtimes \mathbb{Z}_4 / \langle (r_1^2, s_1^2) \rangle$	m even; $\tau = 1$; $\sigma = \frac{m}{2} + 1$
4*	0	2^σ	$2^{\sigma-1}$	$\mathscr{A} \rtimes \mathbb{Z}_4 / \langle (r_2^2, s_1^2) \rangle$	m even; $\tau = 2$; $\sigma = \frac{m}{2} - 1$
5	0	0	$2^{\sigma+1}$	$\mathscr{A} \rtimes (\mathbb{Z}_4 \times \mathbb{Z}_4) / \langle (r_1^2, s_1^2)(r_2^2, s_2^2) \rangle$	$\tau = 2$; $\sigma = m - 3$

Theorem 3 *Let $\mathscr{C}$ be a subgroup of $\mathbb{Z}_2^{k_1} \times \mathbb{Z}_4^{k_2} \times Q_8^{k_3}$ such that $\phi(\mathscr{C}) = C$ is a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code and $\mathscr{A}$ the abelian maximal subgroup in $\mathscr{C}$. Then $\phi(\mathscr{A})$ can be described as a duplication of a Hadamard $\mathbb{Z}_2\mathbb{Z}_4$-linear code when $\upsilon = 1$ or as a quadruplication of a Hadamard $\mathbb{Z}_4$-linear code, if $\upsilon = 2$.*

Depending on the values of the parameters σ, τ, υ the Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes are classified in several shapes, as we can see in Table 1. In fact there are two big classes of Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes. Despite all codes contains the all one vector $\mathbf{u}$, there are codes where there exist an element r_1 such that $r_1^2 = \mathbf{u}$ (codes of shape 1*, 2, 4* and 5) and there are codes where $\mathbf{u}$ is not the square of any other element (codes of shape 1, 3 and 4). We will define the new parameter $\overline{\tau} = \tau - 1$ in the first case ($r_1^2 = \mathbf{u}$) and $\overline{\tau} = \tau$ in the second case ($r_1^2 \neq \mathbf{u}$). The existence conditions for Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes easily come from Theorem 3 and [5], where it was stated the existence conditions for Hadamard $\mathbb{Z}_2\mathbb{Z}_4$-linear codes.

Table 1 summarizes what we have done in this section.

4 Construction of Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-Codes

In this section we deal with the construction of Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes with any allowable pair of values for the rank and the dimension of the kernel. We do not include all the constructions but, as a summary, we include Theorem 4, where it is described what are the allowable parameters for the dimension of the kernel and, for each one of these values, it is said what is the range of values for the rank. For each one of the possible pair of allowable values for the dimension of the kernel and rank, we construct a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code fulfilling it. As an illustration of the constructions we include two examples at the end of the section.

Let C a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code of length 2^m; let $T(\mathscr{C})$ be the subgroup of elements in $\mathscr{C}$ of order two; let $\mathscr{A}(\mathscr{C}) = \langle x_1, \dots, x_\sigma, r_1, \dots, r_\tau \rangle$ and let $\mathscr{R}(\mathscr{C})$ be defined by

$$\begin{cases} \mathscr{R}(\mathscr{C}) = \langle x_1 \dots x_\sigma, r_2 \dots r_\tau \rangle; & \text{if } r_1^2 = \mathbf{u} \\ \mathscr{R}(\mathscr{C}) = \mathscr{A}(\mathscr{C}); & \text{if } r_1^2 \neq \mathbf{u} \end{cases}$$

Theorem 4 *Let C a Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-code of length 2^m and $|T(\mathscr{C})| = 2^\sigma$; $|\mathscr{C}/\mathscr{A}(\mathscr{C})| = 2^\upsilon$; $|\mathscr{A}(\mathscr{C})/T(\mathscr{C})| = 2^\tau$; $|\mathscr{R}(\mathscr{C})/T(\mathscr{C})| = 2^{\overline{\tau}}$; $|\mathscr{C}/T(\mathscr{C})| = 2^{\tau+\upsilon}$ and $m+1 = \sigma+\tau+\upsilon$. Then the rank r and the dimension of the kernel k of C satisfy the following conditions.*

1. *The values of the dimension of the kernel are $1 \neq m+1-k \in \{0, 4, \tau-1, \tau, \tau+1\}$. The specific case $m+1-k = 0$ is obtained in codes where $\overline{\tau} \leq 1$ or in codes of shape 5. The specific case $m+1-k = 4$ is obtained in codes of shape 5.*

2.(a) *If $m+1-k = 0$ then $r-(m+1) = 0$,*
   (b) *If $m+1-k = 4$ and $\upsilon = 2$ then $r-(m+1) = 2$,*
   (c) *If $m+1-k = \tau-1 \geq 2$ then $r-(m+1) = \binom{\tau-1}{2}$,*
   (d) *If $m+1-k = \tau \geq 2$ then $r-(m+1) = \binom{\tau}{2}$,*
   (e) *If $m+1-k = \tau+1$ and $\overline{\tau} <= 1$ then $r-(m+1) = \tau$.*
   (f) *If $m+1-k = \tau+1$ and $\overline{\tau} = \tau-1 \geq 2$ then $r-(m+1) \in \{\binom{\tau-1}{2}, \dots \binom{\tau}{2}+1\}$.*
   (g) *If $m+1-k = \tau+1$ and $\overline{\tau} = \tau \geq 2$ then $r-(m+1) \in \{\binom{\tau}{2}+1, \dots \binom{\tau+1}{2}\}$.*

Example 5 The following example shows constructions of codes of length $n = 2^m = 2^6 = 64$, with $\tau = 3 \geq \overline{\tau} = 2 \geq 2, \upsilon = 1$ and $\sigma = 3$. The resulting codes are of shape 2 and, before the Gray map, subgroups of Q_8^{16}.

$$\begin{aligned} r_1 &= (\mathbf{a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a}) \\ r_2 &= (\mathbf{a\ a\ a^3\ a^3\ a\ a\ a^3\ a^3\ 1\ 1\ a^2\ a^2\ 1\ 1\ a^2\ a^2}) \\ r_3 &= (\mathbf{a\ a^3\ a\ a^3\ 1\ a^2\ 1\ a^2\ a\ a^3\ a\ a^3\ 1\ a^2\ 1\ a^2}) \end{aligned}$$

The codes with all possible pairs of values rank,dimension of the kernel are generated by r_1, r_2, r_3 and s_1. We show the vector s_1 and the values of the pair rank, dimension of the kernel.

$$\begin{array}{l|l} s_1 = (\mathbf{b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b}) & (k,r) = (5,8) \\ s_1 = (\mathbf{b\ b\ b\ ab\ b\ b\ b\ ab\ b\ b\ b\ ab\ b\ b\ b\ ab}) & (k,r) = (3,11) \\ s_1 = (\mathbf{b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ ab\ b\ b\ b\ ab}) & (k,r) = (3,10) \\ s_1 = (\mathbf{b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ ab} & (k,r) = (3,9) \\ s_1 = (\mathbf{b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ b\ ab\ ab\ ab\ ab}) & (k,r) = (3,8) \end{array}$$

Example 6 The following example shows constructions of codes of length 64, with $\tau = \overline{\tau} = 2$, $\upsilon = 1$ and $\sigma = 4$. The resulting codes are of shape 3 and, before the Gray map, subgroups of $\mathbb{Z}_4^8 Q_8^{12}$. All possible pairs of rank and dimension of the kernel are presented.

Take the following vectors in $\mathbb{Z}_4^8 Q_8^{16}$:

$$r_1 = (0\ 2\ 0\ 2\ 0\ 2\ 0\ 2\ \mathbf{1}\ \mathbf{a}^2\ \mathbf{a}\ \ \mathbf{a}\ \ \mathbf{a}\ \ \mathbf{a}\ \ \mathbf{1}\ \mathbf{a}^2\ \mathbf{a}\ \ \mathbf{a}\ \ \mathbf{a}\ \ \mathbf{a})$$
$$r_2 = (0\ 0\ 2\ 2\ 0\ 0\ 2\ 2\ \mathbf{a}\ \ \mathbf{a}\ \ \mathbf{1}\ \mathbf{a}^2\ \mathbf{a}\ \mathbf{a}^3\ \mathbf{a}\ \ \mathbf{a}\ \ \mathbf{1}\ \mathbf{a}^2\ \mathbf{a}\ \mathbf{a}^3)$$

The codes with all possible pairs of values rank, dimension of the kernel are generated by r_1, r_2 and s_1. We show the vector s_1 and the values of the pair rank, dimension of the kernel.

$$\begin{array}{l|l}
s_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}) & (k, r) = (5, 8) \\
s_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{b}\ \mathbf{ab}\ \mathbf{b}\ \mathbf{ab}\ \mathbf{b}\ \mathbf{ab}\ \mathbf{b}\ \mathbf{ab}\ \mathbf{b}\ \mathbf{ab}\ \mathbf{b}\ \mathbf{ab}) & (k, r) = (4, 10) \\
s_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{b}\ \mathbf{ab}\ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \mathbf{ab}\ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}\ \ \mathbf{b}) & (k, r) = (4, 9)
\end{array}$$

Acknowledgements This work has been partially supported by the Spanish MICINN grant TIN2013-40524-P and the Catalan AGAUR grant 2014SGR-691.

References

1. Flannery, D.: Cocyclic Hadamard matrices and Hadamard groups are equivalent. J. Algebra **192**(2), 749–779 (1997). doi:http://dx.doi.org/10.1006/jabr.1996.6949
2. Hammons, A.R., Kumar, P.V., Calderbank, A.R., Sloane, N.J.A., Solé, P.: The F_4-linearity of kerdock, preparata, goethals, and related codes. IEEE Trans. Inf. Theory **40**(2), 301–319 (1994). doi:http://dx.doi.org/10.1109/18.312154
3. Horadam, K.J.: Hadamard matrices and their applications: progress 2007–2010. Cryptogr. Commun. **2**(2), 129–154 (2010). doi:http://dx.doi.org/10.1007/s12095-010-0032-0
4. Ito, N.: On Hadamard groups. J. Algebra **168**(3), 981–987 (1994). doi:http://dx.doi.org/10.1006/jabr.1994.1266
5. Phelps, K.T., Rifà, J., Villanueva, M.: On the additive (F_4-linear and non-F_4-linear) Hadamard codes: rank and kernel. IEEE Trans. Inf. Theory **52**(1), 316–319 (2006). doi:http://dx.doi.org/10.1109/TIT.2005.860464
6. Rifà, J., Pujol, J.: Translation-invariant propelinear codes. IEEE Trans. Inf. Theory **43**(2), 590–598 (1997). doi:http://dx.doi.org/10.1109/18.556115
7. de Launey, W., Flannery, D.L., Horadam, K.J.: Cocyclic Hadamard matrices and difference sets. Discret. Appl. Math. **102**(1–2), 47–61 (2000). doi:http://dx.doi.org/10.1016/S0166-218X(99)00230-9
8. del Río, Á., Rifà, J.: Families of Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes. IEEE Trans. Inf. Theory **59**(8), 5140–5151 (2013). doi:http://dx.doi.org/10.1109/TIT.2013.2258373

2-Designs and Codes from Simple Groups $L_3(q)$ and Higman-Sims Sporadic Simple Group *HS*

Jamshid Moori and Georges F. Randriafanomezantsoa Radohery

Abstract We discuss the methods used in constructing designs and codes from the fixed points of the Sylow p-subgroups of the 2 points stabilizers in the 2-transitive permutation representation of finite groups. To illustrate the methods we apply them to the simple groups $L_3(q)$ ($q \geq 3$) and Higman-Sims sporadic simple group HS. This talked is based on the results included in an article entitled “2-designs and codes from 2-transitive simple groups” which is appearing in the Utilitas Mathematica.

Keywords Designs • Codes • 2-transitive simple groups • Sylow subgroups • Maximal subgroups

1 Introduction

In 1937, Ernst Witt developed a method to construct a Steiner system from a t-transitive group (See [16] and [15]). In 1965, D.R. Hughes generalized Witt's method to construct not only a Steiner system but a t-design in general [9].

In our work we have explored designs and codes that can be constructed by applying Witt's and Hughes's methods on various finite simple groups, such as $L_3(q)$ ($q \geq 3$), $L_2(11)$, $U_3(4)$, $U_3(9)$, A_n ($n \geq 6$), $S_6(2)$, $S_8(2)$ and HS. Some of these structures are well known and have been constructed elsewhere using other methods. In this talk to illustrate the methods we apply them to the simple groups $L_3(q)$ ($q \geq 3$) and Higman-Sims sporadic simple group HS. This talked is based on the results included in an article entitled “2-designs and codes from 2-transitive simple groups” which is appearing in the Utilitas Mathematica. Section 2 provides the notation that we are following. Section 3 presents some background results and

J. Moori (✉)
School of Mathematical Sciences, North-West University (Mafikeng), Mmabatho 2375,
South Africa
e-mail: jamshid.moori@nwu.ac.za

G.F.R. Radohery
African Institute for Mathematical Sciences, Muizenberg 7945, South Africa
e-mail: georges@aims.ac.za

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_30

the details on the Witt's and Hughes's methods. In Sect. 4 we will apply Witt's method to the group $L_3(q)$ ($q \geq 3$) and hence we will construct a 2-design which is a projective plane of order q. The code from a projective plane have been fully studied and it is a generalized Reed-Muller code.

We will see that Witt's method does not apply to alternating groups A_n ($n \geq 6$) acting naturally on n points, the Higman-Sims group HS and the symplectic groups. In Sect. 5 using Hughes's method on HS, we will construct a new 2-(176,6,36) design which admits HS itself as the full automorphism group. From that design we will construct a $[176, 115, 6]_2$ code and its dual $[175, 21, 56]_2$, both admit HS as full automorphism group.

2 Terminologies and Notations

For the structure of groups and their maximal subgroups we follow the $\mathbb{ATLAS}$ notation [7]. The groups $G{:}H$ and $G{}^{\cdot}H$ denote a split extension and a non-split extension, respectively. For a prime p, p^n denotes the elementary abelian group of order p^n. Suppose that G is a finite group acting on a finite set Ω, the action of G on Ω gives a permutation representation π with corresponding permutation character χ_π denoted by $\chi(G|\Omega)$. Let U be a subgroup of G, if Ω is the set of all conjugates of U in G then we denote $\chi(G|\Omega)$ by χ_U. The fixed points of U is the set $\mathscr{F}(U)$ defined by $\mathscr{F}(U) = \{x \in X{:}x^g = x \text{ for all } g \in U\}$. The normalizer of U in G is denoted by $N_G(U)$. Let's consider a subgroup H of G such that $U < H < G$. The group U is called a **S-subgroup** of H if for any $g \in G$ such that $U^g < H$ then we can find $h \in H$ such that $U^g = U^h$.

An incidence structure is a triple $I = (\mathscr{P}, \mathscr{B}, \mathscr{I})$, $\mathscr{P}$ is called the point set, $\mathscr{B}$ is called the block set and $\mathscr{I}$ is an incidence relation between $\mathscr{P}$ and $\mathscr{B}$. A **t-design** or more precisely a **t-(v, k, –) design** is an incidence structure $\mathscr{D} = (\mathscr{P}, \mathscr{B}, \mathscr{I})$ such that $|\mathscr{P}| = v$, every block $B \in \mathscr{B}$ is incident with precisely k points and every t distinct points are together incident with precisely λ blocks. We will say that the design is **symmetric** if it has the same number of points and blocks. Let $\mathscr{D} = (\mathscr{P}, \mathscr{B}, \mathscr{I})$ be a t-(v, k, λ) design with b blocks and let the points be labeled $\{p_1, p_2, \ldots, p_v\}$ and the blocks $\{B_1, B_2, \ldots, B_b\}$. The **incidence matrix** of $\mathscr{D}$ is a $v \times b$ matrix $D = (d_{ij})$ $(1 \leq i \leq v,\ 1 \leq j \leq b)$ such that $d_{ij} = 1$ if $(x_i, B_j) \in \mathscr{I}$ and $d_{ij} = 0$ otherwise. Two designs $\mathscr{D}_1$ and $\mathscr{D}_2$ are isomorphic if there is an incidence-preserving bijection sending the point set of $\mathscr{D}_1$ to the point set $\mathscr{D}_2$ and sending the block set of $\mathscr{D}_1$ to the block set of $\mathscr{D}_2$. An automorphism of a design $\mathscr{D}$ is an isomorphism from $\mathscr{D}$ to $\mathscr{D}$. A **S(t, k, v) Steiner System** is a t-$(v, k, 1)$ design. A Steiner triple system is a 2-$(v, 3, 1)$ design, v is called the order of the Steiner triple system. A necessary and sufficient condition for the existence of such a design is $v \equiv 1 \text{ or } 3 \mod 6$ (see [11]).

If q is a prime power and $F = \mathbb{F}_q$ the finite field of order q, then a **q-ary linear block code** C of length n and dimension k is a subspace of dimension k of the n-dimensional vector space F^n. The elements of C are called **codewords**.

In this work all codes are linear block codes. The **dual** code $C^{\perp}$ of a code C is its orthogonal space with respect to the standard inner product of F^n. The **hull** of a code C is the intersection $C \cap C^{\perp}$. We endow the vector space F^n with the Hamming distance defined by $d(x, y) = |\{i | x_i \neq y_i\}|$ for $x = (x_1, x_2, \ldots, x_n)$, $y = (y_1, y_2, \ldots, y_n) \in F^n$. The **weight** of a code word x is defined by $w(x) = d(x, \mathbf{0})$. The all one vector will be denoted by $\mathbf{j}$, and is the constant vector of weight the length of the code. The minimum weight d of a code C is defined by $d = \min_{x \in C, x \neq \mathbf{0}} w(x)$. A $\mathbf{[n, k, d]_q}$ **code** C is a q-ary linear block code of length n, dimension k and minimum weight d. The **code** C_F of the t-design $\mathscr{D}$ over F is the space spanned by the rows of the incidence matrix of $\mathscr{D}$ over the field F. The length of C_F is the size of the point set of $\mathscr{D}$ and the dimension of C_F is the rank of the incidence matrix of $\mathscr{D}$ over F. Two codes with the same length and dimension are said to be isomorphic if one can be obtained from the other by permuting the coordinate positions. The automorphism group of C_F is the group of isomorphisms from C_F to C_F.

3 Preliminary Results

The method of constructions of the designs are based on the following results

Theorem 1 (Witt's method) *Let X be a faithful t-transitive G-set, where $t \geq 2$. Let H be the stabilizer of t points $x_1, x_2, \ldots, x_t$ in X, and let U be a Sylow p-subgroup of H for some prime p.*

(i) $N_G(U)$ acts t-transitively on $\mathscr{F}(U)$.
(ii) If $k = |\mathscr{F}(U)| > t$ and U is a nontrivial normal subgroup of H, then $(X, \mathscr{B})$ is a Steiner system of type $S(t, k, v)$, where $|X| = v$ and

$$\mathscr{B} = \{\mathscr{F}(U^g) : g \in G\}.$$

Proof [14, Therorem 9.66] □

Theorem 2 (Hughes's method) *Let G be a t-transitive permutation group on a set of v points X, H the stabilizer of t points and U an S-subgroup of H such that $k = |\mathscr{F}(U)| > t$ then*

(i) $N_G(U)$ acts t-transitively on $\mathscr{F}(U)$.
(ii) $(X, \mathscr{B})$ is a t-(v, k, λ) design, where $\lambda = [H : H_{\mathscr{F}(U)}]$, $\mathscr{B} = \{\mathscr{F}(U^g) : g \in G\}$. Furthermore G is t-flag-transitive on this design.

Proof [9, Theorem 3.3] □

Remark 3 • In Theorem 1, G is t-flag-transitive on the Steiner system.

- In Theorem 2, the Sylow p-subgroups of H are S-subgroups of H. If U is the unique Sylow p-subgroup of H then $N_G(U) = G_{\mathscr{F}(U)}$. Indeed if $g \in G_{\mathscr{F}(U)}$

then $\mathscr{F}(U^g) = \mathscr{F}(U)$ that is U^g fixes the t points fixed by H and $U^g < H$, therefore $U^g = U$ that is $g \in N_G(U)$ and $G_{\mathscr{F}(U)} \leq N_G(U)$. In any case $N_G(U) \leq G_{\mathscr{F}(U)}$. Now $H_{\mathscr{F}(U)} = H \cap G_{\mathscr{F}(U)} = H \cap N_G(U) = N_H(U) = H$, thus if U is a normal Sylow p-subgroup of H then $\lambda = 1$ and we get back to Theorem 1.

Theorems 1 and 2 will be applied to some 2-transitive simple groups to construct 2-designs. For the list of all 2-transitive simple groups the readers are referred to [6]. All symmetric 2-designs with 2-transitive automorphism groups have been classified by William M. Kantor :

Theorem 4 ([10]) *Let $\mathscr{D}$ be a symmetric design with $v > 2k$ such that $Aut(\mathscr{D})$ is 2-transitive on points. Then $\mathscr{D}$ is one of the following:*

(i) *A projective space;*
(ii) *The unique Hadamard design with $v = 11$, and $k = 5$;*
(iii) *A unique design with $v = 176$, $k = 50$ and $\lambda = 14$; or*
(iv) *A design with $v = 2^{2m}$, $k = 2^{m-1}(2^m - 1)$ and $\lambda = 2^{m-1}(2^{m-1} - 1)$, of which there is exactly one for each $m \geq 2$.*

We will see that the designs in (i) can be explicitly constructed from Theorem 1, the design in (iii) can be constructed indirectly from Theorem 2 and (ii) can be constructed from a modification of the method described in Theorem 2.

4 Some Designs and Codes from $L_3(q)$

In this section we examine the general parameters of the designs and codes that are built by applying Witt's method to $G = L_3(q)$.

Theorem 5 *Let q be some power of a prime number p. Consider $L_3(q)$ as a faithful 2-transitive group on a set Ω with $|\Omega| = q^2+q+1$. Let $H \lesssim L_3(q)$ be the stabilizer of two points.*

(i) *If $q = p^n$ then H is a subgroup of order $q^2(q-1)^2/(3, q-1)$ having a normal Abelian Sylow p-subgroup U of order p^{2n}.*
(ii) $|\mathscr{F}(U)| = q + 1$.

Proof According to [3] any 2-transitive permutation representation of $L_3(q)$ on a set of size $q^2 + q + 1$ is isomorphic to the action of G on the points of the projective space $PG(2, q)$.

(i) We have $|L_3(q)| = q^3(q^3 - 1)(q^2 - 1)/(3, q - 1)$ and $|H| = |L_3(q)|/q(q + 1)(q^2 + q + 1) = q^2(q - 1)^2/(3, q - 1)$. According to [13], H is a subgroup of a line stabilizer K which is a maximal subgroup of order $(q - 1)^2q^3(q + 1)/(3, q - 1)$ of $L_3(q)$. Let's consider the line $\ell = [1{:}0{:}0]$. The stabilizer of ℓ are the 3×3 matrices of the form $\left(\begin{smallmatrix} \alpha & \mathbf{v}_2 \\ \mathbf{0} & M_2 \end{smallmatrix}\right)$ with $\alpha \in F_q$, $\mathbf{v}_2$ a vector of length 2 with entries in F_q and M_2 a 2×2 matrix over F_q such that $\alpha \cdot \det(M_2) = 1$. The set U of matrices of the form $\left(\begin{smallmatrix} \alpha & \mathbf{v}_2 \\ \mathbf{0} & I_2 \end{smallmatrix}\right)$ is an Abelian normal subgroup of order q^2 of K and it is the Sylow p-subgroup of H.

(ii) Consequence of the fact that $U < K$ and each line of $PG(2,q)$ has $q+1$ points. □

Hence we can apply Witt's method to $G = L_3(q)$ and we have the following result:

Corollary 6 *From the simple group $G = L_3(q)$ we obtain a 2-$(q^2+q+1, q+1, 1)$ design $\mathscr{D}$ on which G acts 2-transitively on points and transitively on blocks.*

Proof Direct application of Theorem 1 to $L_3(q)$. □

Remark 7 If B is a block in $\mathscr{B}$ then $|\mathscr{B}| = [G : G_B]$. Since a block is just a line of $PG(2,q)$, $|G_B| = (q-1)^2 q^3 (q+1)/(3, q-1)$, $|\mathscr{B}| = q^2+q+1$ and $\mathscr{D}$ is symmetric. We have $G \lesssim \mathrm{Aut}(\mathscr{D})$. The question is whether $\mathrm{Aut}(G) \lesssim \mathrm{Aut}(\mathscr{D})$ or not. In [8] we can find the list of the maximal subgroups of $G = L_3(q)$. The group G has a maximal subgroup of the form $q^2{:}GL(2,q)$ which is of index q^2+q+1, the action of G on the cosets of that maximal subgroup gives us a permutation representation of degree q^2+q+1 of G. The subgroup $q^2{:}GL(2,q)$ is the point stabilizer in that representation and $H < q^2{:}GL(2,q)$. Considering that the set of subgroups of $L_3(q)$ isomorphic to $q^2{:}GL(2,q)$ split into two conjugacy classes, if the outer automorphism of G contains an involution which fuses these two conjugacy classes then $\mathrm{Aut}(G) \not\lesssim \mathrm{Aut}(\mathscr{D})$. In general it suffices to remove that involution from $\mathrm{Aut}(G)$ to get the automorphism group of $\mathscr{D}$.

Theorem 8 *Let p be a prime number and C the $\mathbb{F}_p$-code of a symmetric design 2-$(m^2+m+1, m+1, 1)$.*

(i) If $p|m$, then $2 \le \dim C \le \frac{1}{2}(m^2+m+2)$.
(ii) If $p \nmid m$ and $p|m+1$, then $\dim C = m(m+1)$.
(iii) If $p \nmid m$ and $p \nmid m+1$, then $\dim C = m^2+m+1$.

Proof Direct application of [12, Proposition 2.6]. □

Remark 9 We see that from $L_3(q)$ using Theorem 1, we can obtain a $\mathbb{F}_p$-code C with $\dim(C) < m(m+1)$ only for the prime p such that $q = p^n$.

Theorem 10 *Let p be any prime, $q = p^n$, and $\mathscr{D}$ a $2-(q^2+q+1, q+1, 1)$ symmetric design. Then the linear code C_p over $\mathbb{F}_p$ derived from $\mathscr{D}$ is a generalized Reed-Muller code and has dimension $\binom{p+1}{2}^n + 1$. The minimum-weight words of C_p are the scalar multiples of the incidence vectors of the blocks. Further, $Hull_p(\mathscr{D})$ has minimum weight $2q$ with the minimum-weight vectors the scalar multiples of the differences of the incidence vectors of two distinct blocks of $\mathscr{D}$. The minimum weight d of $C_p^{\perp}$ satisfies $q+p \le d \le 2q$, with equality at the lower bound if $p = 2$.*

Proof [2] and [1, Theorem 6.3.1 and Theorem 6.4.2] □

Remark 11 Table 1 gives the result that we obtained by computation using MAGMA [4] for different values of q.

Table 1 Some designs and codes from $L_3(q)$

G Aut(G)	Degree	H = $G_{\alpha\beta}$	p	U	$\|\mathscr{F}(U)\|$	Design ($\mathscr{D}$)	Code (C)	Aut($\mathscr{D}$)	Aut(C)
$L_3(3)$ $L_3(3)$:2	13	$S_3 \times S_3$	3	3^2	4	2-(13,4,1)	$[13, 7, 4]_3$	$L_3(3)$	$2 \times L_3(3)$
$L_3(4)$ $L_3(4)$:($2 \times S_3$)	21	2^4:3	2	2^4	5	2-(21,5,1)	$[21, 10, 5]_2$	$L_3(4)$:S_3	$L_3(4)$:S_3
$L_3(5)$ $L_3(5)$:2	31	$(5{:}\mathbb{Z}_4) \times (5{:}\mathbb{Z}_4)$	5	5^2	6	2-(31,6,1)	$[31, 16, 6]_5$	$L_3(5)$	$2^2 \times L_3(5)$
$L_3(7)$ $L_3(7)$:S_3	57	$((7^2{:}3){:}2) : 2$	7	7^2	8	2-(57,8,1)	$[57, 29, 8]_7$	$L_3(7) : 3$	$6 \times (L_3(9){:}3)$
$L_3(8)$ $L_3(8)$:$\mathbb{Z}_6$	73	$(2^3{:}7) \times (2^3{:}7)$	2	2^6	9	2-(73,9,1)	$[73, 28, 9]_2$	$L_3(8) : 3$	$L_3(8) : 3$
$L_3(9)$ $L_3(9)$:2^2	91	$(3^2{:}8) \times (3^2{:}8)$	3	3^4	10	2-(91,10,1)	$[91, 37, 10]_3$	$L_3(9) : 2$	
$L_3(11)$ $L_3(11)$:2	133	$(11{:}\mathbb{Z}_{10}) \times (11{:}\mathbb{Z}_{10})$	11	11^2	12	2-(133,12,1)	$[133, 67, 12]_{11}$	$L_3(11)$	

5 Some Designs and Codes from the Higman-Sims Group HS

According to $\mathbb{ATLAS}$ [7], HS has a 2-transitive permutation representation of degree 176. In that representation Theorem 1 does not apply to HS. Indeed, the point stabilizer is the group $U_3(5)$:2 and the two points stabilizer H is $A_6{}^{\cdot}2^2$ and none of the Sylow p-subgroups of H is normal. More precisely, H has only Sylow 2-subgroups, Sylow 3-subgroups and Sylow 5-subgroups. A Sylow 3-subgroup or a Sylow 5-subgroup of H cannot obviously be normal since they are subgroups of A_6. We can easily show that (using MAGMA) H has 45 Sylow 2-subgroups and hence a Sylow 2-subgroup of H cannot be normal in H.

Now we apply Theorem 2 to HS. A Sylow 2-subgroup of H is isomorphic to $(2 \times D_8)$:2 and it fixes exactly 2 points. Hence by applying Theorem 2 to HS with a Sylow 2-subgroup of H we obtain a trivial 2-(176,2,1) 2-design. A Sylow 3-subgroup of H is isomorphic to 3^2 and it fixes exactly 2 points and by applying Theorem 2 in this case we obtain a trivial 2-(176,2,1) 2-design isomorphic to the previous design. A Sylow 5-subgroup of H is a cyclic group of order 5 which fixes 6 points and we have the following result.

Proposition 12 *From HS we can construct a 2-(176,6,36) design $\mathscr{D}$ with Aut($\mathscr{D}$) = HS.*

Proof We apply Theorem 2 to HS with a Sylow 5-subgroup of H to construct the design $\mathscr{D}$. Computation with MAGMA shows that $|\mathrm{Aut}(\mathscr{D})| = |HS|$ so $\mathrm{Aut}(\mathscr{D}) \cong HS$. □

Remark 13 The code C associated to the design $\mathscr{D}$ is a $[176, 155, 6]_2$ code with Aut(C) =HS. The dual $C^\perp$ is a $[176, 21, 56]_2$ code. In [5], $C^\perp$ is presented as a subspace of a linear code $[176, 22, 50]_2$ which is constructed from a monomial representation of HS.

Now a word of weight i is denoted by w_i and the set of words of weight i is denoted by W_i. By acting HS on W_i, the orbits form the blocks of a 2-$(176, i, k_i)$ design $\mathscr{D}_{w_i}$ where $k_i = |(w_i)^{\mathrm{HS}}| \times i/176$. The number of blocks of $\mathscr{D}_{w_i}$ is $[\mathrm{HS} : (\mathrm{HS})_{w_i}]$. Table 2 gives the parameters of the design $\mathscr{D}_{w_i}$, and the structure of HS_{w_i} for the words of minimum weight of C. Table 3 gives the same results for the weight distribution of $C^\perp$.

Remark 14 Under the action of HS the words of weight 6 of C split into 2 orbits W_{6_1} and W_{6_2} of length 36,960 and 92,400, respectively. The code words in W_{6_1} form a 2-(176,6,36) design $\mathscr{D}_{w_{6_1}}$ and the generators of the code C belong to W_{6_1}. The code

Table 2 2-design $\mathscr{D}_{w_i}$ from C and the stabilizer in HS of a word w_i of C

i	$\mathscr{D}_{w_i}$	No. of blocks	$(\mathrm{HS})_{w_i}$	Maximality
6_1	2-(176,6,36)	36,960	$(5{:}4) \times A_5$	Yes
6_2	2-(176,6,90)	92,400	$(SL_2(5){:}2){:}2$	No

Table 3 2-design $\mathscr{D}_{w_i}$ from $C^\perp$ and the stabilizer in HS of a word w_i of $C^\perp$

i	$\mathscr{D}_{w_i}$	No. of blocks	$(\mathrm{HS})_{w_i}$	Maximality
56	2-(176,56,110)	1,100	$L_3(4){:}2_1$	Yes
64	2-(176,64,540)	4,125	$2^3.(2^3.L_2(7))$	No
72	2-(176,72,2556)	15,400	$2 \times A_6 \cdot 2^2$	Yes
$(80)_1$	2-(176,80,790)	3,850	$2^4.S_6$	Yes
$(80)_2$	2-(176,80,47400)	231,000	$(((2{\times}D_8){:}2){:}3){:}2$	No
$(80)_3$	2-(176,80,75840)	369,600	S_5	No
$(88)_1$	2-(176,88,38280)	154,000	$(A_4{\times}A_4){:}2$	No
$(88)_2$	2-(176,88,172260)	693,000	$((\mathbb{Z}_4 \times \mathbb{Z}_4){:}2){:}2$	No
$(96)_1$	2-(176,96,1140)	3,850	$2^4.S_6$	Yes
$(96)_2$	2-(176,96,68400)	231,000	$(((2 \times D_8){:}2){:}3){:}2$	No
$(96)_3$	2-(176,96,109440)	369,600	S_5	No
104	2-(176,104,5356)	15,400	$2 \times A_6 \cdot 2^2$	Yes
112	2-(176,112,1665)	4,125	$2^3.(2^3.L_2(7))$	No
120	2-(176,120,510)	1,100	$L_3(4){:}2_1$	Yes

words in W_{6_2} form a 2-(176,6,90) design $\mathscr{D}_{w_{6_2}}$ with corresponding $[176,154,6]_2$ code C'. The dual of C' is a $[176,22,50]_2$ code and with its words of weight 50 we can construct the unique symmetric 2-(176,50,14) design admitting HS as full automorphism group mentioned in Theorem 4.

Acknowledgements Supports from NRF, AIMS and North-West University (Mafikeng) are acknowledged.

References

1. Assmus, Jr., E.F., Key, J.D.: Designs and Their Codes. Cambridge Tracts in Mathematics, vol. 103. Cambridge University Press, Cambridge (1992)
2. Assmus, Jr., E.F., Key, J.D.: Designs and codes: an update. Des. Codes Cryptogr. **9**(1), 7–27 (1996); Second Upper Michigan Combinatorics Workshop on Designs, Codes and Geometries, Houghton (1994). doi:10.1023/A:1027359905521
3. Bannai, E.: Doubly transitive permutation representations of the finite projective special linear groups PSL(n, q). Osaka J. Math. **8**, 437–445 (1971)
4. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symb. Comput. **24**(3–4), 235–265 (1997); Computational Algebra and Number Theory, London (1993). doi:10.1006/jsco.1996.0125
5. Calderbank, A.R., Wales, D.B.: A global code invariant under the Higman-Sims group. J. Algebra **75**(1), 233–260 (1982). doi:10.1016/0021-8693(82)90073-4
6. Cameron, P.J.: Permutation Groups. London Mathematical Society Student Texts, vol. 45. Cambridge University Press, Cambridge (1999). 10.1017/CBO9780511623677. doi:10.1017/CBO9780511623677
7. Conway, J.H., Curtis, R.T., Norton, S.P., Parker, R.A., Wilson, R.A.: Atlas of finite groups. Oxford University Press, Eynsham (1985) (Maximal subgroups and ordinary characters for simple groups, With computational assistance from J. G. Thackray)

8. Gill, N.: PSL$(3, q)$ and line-transitive linear spaces. Beitr. Algebra Geom. **48**(2), 591–620 (2007)
9. Hughes, D.R.: On t-designs and groups. Am. J. Math. **87**, 761–778 (1965)
10. Kantor, W.M.: Classification of 2-transitive symmetric designs. Graphs Comb. **1**(2), 165–166 (1985). doi:10.1007/BF02582940
11. Kirkman, T.P.: On a problem in combinations. Camb. Dublin Math. J. **2**, 191–204 (1847)
12. Lander, E.S.: Symmetric designs: an algebraic approach. London Mathematical Society Lecture Note Series, vol. 74. Cambridge University Press, Cambridge (1983). doi:10.1017/CBO9780511662164
13. Mitchell, H.H.: Determination of the ordinary and modular ternary linear groups. Trans. Am. Math. Soc. **12**(2), 207–242 (1911). doi:10.2307/1988576
14. Rotman, J.J.: An Introduction to the Theory of Groups. Springer, New York (1995)
15. Witt, E.: Die 5-fach transitiven gruppen von Mathieu. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg **12**, 256–264 (1937)
16. Witt, E.: Über Steinersche systeme. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg **12**, 265–275 (1937)

New Variant of the McEliece Cryptosystem

Hamza Moufek and Kenza Guenda

Abstract The purpose of this paper is to present a new version of the McEliece cryptosystem based on punctured convolutional codes and the pseudo-random generators. We use the modified self-shrinking generator to fill the punctured pattern. More precisely we propose to fill out the pattern punctured by the bits generated using a pseudo random generator LFSR.

Keywords Punctured convolutional code • McEliece cryptosystem • Self shrinking generator

1 Introduction

In 1978 Robert J. McEliece invented the first cryptosystem based on algebraic coding theory [10]. Since then different variants have been proposed [1, 2, 5].

Different attacks were made against these schemes. Among them, we mention the attack on the original McEliece system by Canteaut and Sendrier [6] and the attack on the cryptosystem based on convolutional codes by Landais and Tillich [7].

The purpose of this paper is to present a new version of the McEliece cryptosystem based on punctured convolutional codes and the pseudo-random generators. Instead of using time-varying convolutional codes as it was given in [8] and broken by [7], we use the modified self-shrinking generator to fill the punctured pattern. More precisely we propose to fill out the pattern punctured by the bits generated using a pseudo random generator LFSR.

H. Moufek • K. Guenda (✉)
Faculty of Mathematics, USTHB, University of Science and Technology of Algiers,
Bab Ezzouar, Algeria
e-mail: moufekhamza@hotmail.com; ken.guenda@gmail.com

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_31

2 Our New Variant

In this section we will give the description of our new variant McEliece Cryptosystem. To hide the structure of the convolutional code, we follow the method of puncturing described in [9]. Starting from a convolutional code of parameters (n, k, K), we construct an equivalent code of parameters (Mn, Mk, K_p), called grouped code. With the puncturing pattern T, whose the number of its coefficient corresponds to the number of columns of the grouped matrix, we obtain the generator matrix of punctured code with parameters $(N^{'}, Mk, K_p)$.

Since we use the modified self-shrinking generator to fill the punctured pattern, in the next paragraph we describe the modified self shrinking generator given by Kanso [6].

2.1 *The Modified Self Shrinking Generator*

In [6] Kanso modified the method of Meier and Staffelbach [11] by using one LFSR of length s which operates as follows:

Let A be an LFSR that generates the sequence $a_t = a_0, a_1, a_2, \ldots$

At time i, we consider the triplet $(a_{3i}, a_{3i+1}, a_{3i+2})$. If the bit $a_{3i} \bigoplus a_{3i+1} = 1$, the output of the LFSR is a_{3i+2}. Else no output is produced.

The puncturing pattern is of size $n \times M$. So we proceed so that the modified self-shrinking generator product an output sequences of period greater or equal than $n \times M$.

2.2 *Description of Our Cryptosystem*

Algorithm 1: Key Generation

1. Choose a generator matrix G for a convolutional code of parameters: its length n and its dimension k.
2. Write the polyphase decomposition of elements of G and then forming the polycyclic pseudocirculant matrix.
3. Replace the polynomials $g_{i,j}(D)$ of G by their Mth PCPC matrix and interlacing the lines and columns at depth M.
4. Generate a random sequence of bits in a modified self-shrinking generator, and fill the matrix T by the $n \times M$ output elements.
5. Apply the function ϕ to the matrix $G^{[M]}(D)$ to get the secret matrix G_p.
6. Choose randomly tow matrix: P the permutation matrix and S an invertible matrix.
7. Compute the public matrix $G^{'} = SG_pP$.

Algorithm 2: Encryption
For sending a message x to someone, we calculate: $c = xG' + e$

Algorithm 3: Decryption
To decrypt the message, you must:

1. Compute $z = cP^{-1}$
2. Determining z' by correcting errors of u.
3. Compute $x = z'S^{-1}$

To correct errors of z, the Viterbi decoding algorithm is used, because an algorithm which decodes the parent code, it also decodes the punctured code.

3 Security of Our Scheme

There are several ways to attack the McEliece encryption system. Among them we find algebraic methods and probabilistic methods.

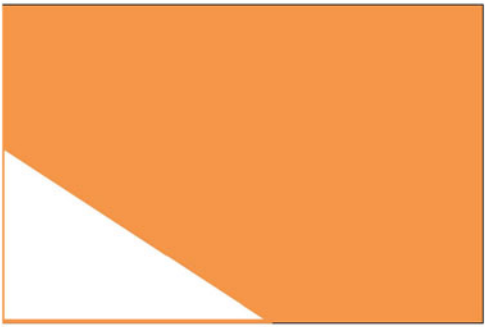
In this section we give a proof that our scheme is secure against the structural and decoding attacks.

3.1 Structural Attacks

The objective of a structural attack is to find an equivalent code to the public code whose a polynomial decoding algorithm is known. For this, we used a puncturing pattern to hide the structure of the code, whose the attacker cannot imagine. Moreover, the equivalence of punctured codes is an NP-complete problem [12]. This makes impossible the cryptanalysis of the system.

The authors have described an algorithm that is used to find an equivalent generator matrix to the secret matrix ([7], section 3). By applying this method to our public matrix we will obtain a matrix of the form (Fig. 1):
which is not equivalent to the matrix G_p

Fig. 1 The generator matrix obtained by the attack of G. Landais and J. Tillich



3.1.1 Exhaustive Search Attack

In our scheme, the private key (G_p, S, P, T) is obtained randomly. In this section we will show that our scheme is secure against the exhaustive search attack. This is equivalent to show that it is a difficult task to find the private key. For that we start by showing that the choice of the matrices S and P is very large. Namely, since the number of invertible matrices in $\mathbb{F}_q$ is $\prod_{i=1}^{k}(q^k - q^{i-1})$ and the number of permutation matrices of size n is equal to $n!$. Then in order to find the two selected matrices we have to try $n! \prod_{i=1}^{k}(q^k - q^{i-1})$ matrices.

Now, we will show that the complexity of finding the matrix G_p is very large. For that, let A be an r-sequence generated by a primitive LFSR of length s and let L be the set of positions of the columns removed during the puncturing step from the matrix G. To find a subset $L^{'} \subseteq L$, it is necessary to know a part of the sequence generated by the LFSR.

3.2 Information Set Decoding

For security level around 2^{80} measured by Canteaut-Chabaud's algorithm [4], we propose the following set of parameters.

Let C be an $(400, 343)$-convolutional code. After a puncturing of depth 7 in 56 positions, we obtain a punctured code of length 2744 and dimension 2401.

For a code of rate $R = 3/4$, we propose a puncturing of depth 3 for an $(570, 421)$ – convolutional code in 26 positions. Thereafter we will have an $(1684, 1263)$-punctured code.

In Table 1 we give examples of different (n, k) – convolutional code and their $(N^{'}, k^{'})$-punctured code associated with different security level.

Table 1 Suggested parameters of our cryptosystem for different security levels

Security level	n	k	M	Number of deleted columns	$k^{'} = kM$	$N^{'}$	Rate
80	305	150	4	20	600	1200	1/2
	284	71	5	30	355	1420	1/4
	570	421	3	26	1263	1684	3/4
	125	1050	2	100	250	2000	1/8
100	316	154	5	40	770	1540	1/2
	625	155	3	15	465	1860	1/4
	730	540	3	30	1620	2160	3/4
	68	550	5	30	340	2720	1/8

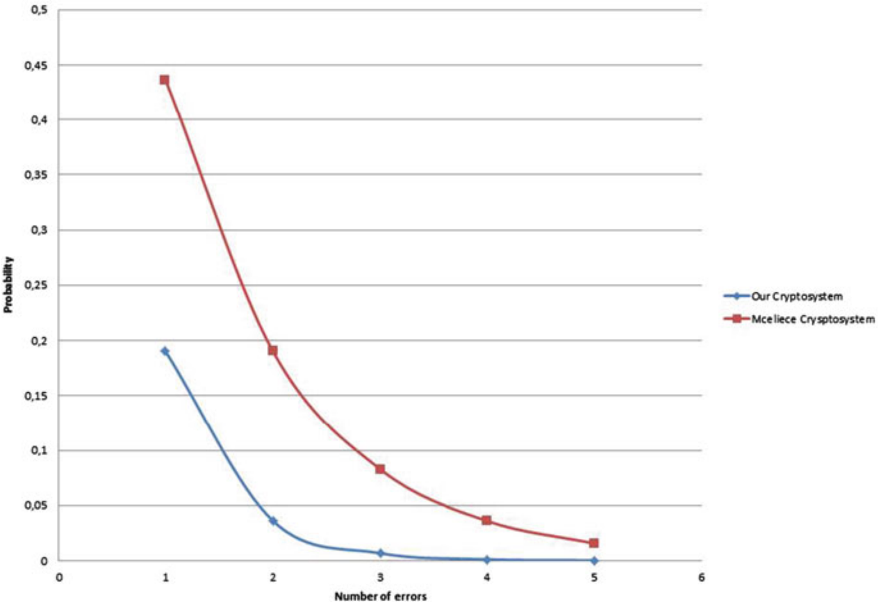


Fig. 2 Probability of guessing k ungarbled columns from those indexed by L_0 depending on the number of errors

3.3 Message-Resent Attack

The message-resent attack was given in [3] and is described as follow:

Suppose now that, through some accident, or as a result of action in the part of the cryptanalyst, both $c_1 = mSGP + e_1$ and $c_2 = mSGP + e_2$ with $e_1 \neq e_2$ are sent. We call this a message-resent condition.

Using an $(1684, 1263)$-punctured code with a free distance $d_{free} = 131$. We compare the effect of this attack on our system and the McEliece cryptosystem, we get the following graph (Fig. 2):

These results are better than the result obtained by attacking Mceliece crptosystem.

We remark that whenever we increase the number of errors, the probability of avoiding this attack increases.

References

1. Barbier, M., Baretto, P.S.L.M.: Key reduction of McEliece's cryptosystem using list decoding. In: International Symposium of Information Theory ISIT 2011, Saint-Pettersburg (2011)
2. Berger, T.P., Cayrel, P.L., Gaborit, P., Otmani, A.: Reducing key length of the McEliece cryptosystem. In: Preneel, B. (Ed.) AFRICACRYPT 2009, Gammarth. Volume 5580 of Lecture Notes in Computer Science, pp. 77–97. Springer Berlin/Heidelberg (2009)

3. Berson, T.A.: Failure of the McEliece public-key cryptosystem under message-resend and related-message attack. In: Kaliski, B.S., Jr. (ed.) Advances in Cryptology-CRYPTO '97, Santa Barbara, California, USA, 17–21 Aug 1997. Volume 1294 of Lecture Notes in Computer Science, pp. 213–220. Springer (1997)
4. Canteaut, A., Chabaud, F.: A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. IEEE Trans. Inf. Theory **44**(1), 367–378 (1998)
5. Johannesson, R., SZigangirov, K.: Fundamentals of Convolutional Coding. IEEE, New York (1999)
6. Kanso, A.: Modified self-shrinking generator. Comput. Electr. Eng. **36**(5), 993–1001 (2010)
7. Landais, G., Tillich, J.P.: An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In: Gaborit, P. (ed.) PQCrypto 2013. Lecture Notes in Computer Science, vol. 7932, pp. 102–117. Springer, Berlin/Heidelberg (2013)
8. Londahl, C., Johansson, T.: A new version of McEliece PKC based on convolutional codes. In: International Conference on Information and Communications Security ICICS 2012, Hong-Kong, Oct 2012
9. Marazin, M., Gautier, R., Burel, G.: Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream. IET Signal Process, **6**(2), 122–131 (2012)
10. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DSN Progress Report 42–44, pp. 114–116 (1978)
11. Meier, W., Statfelbach, O.: The self-shrinking generator. In: De Santis, A. (ed.) Advances in Cryptology, Eurocrypt 94. Lecture Note in Computer Science, vol. 950, pp. 205–214. Springer, Berlin (1995)
12. Wieschebrink, C.: Two NP-complete problems in coding theory with an application in code based cryptography. In: International Symposium on Information Theory (ISIT06), Seattle, July 2006

Power Decoding of Reed–Solomon Codes Revisited

Johan S.R. Nielsen

Abstract Power decoding, or "decoding by virtual interleaving", of Reed–Solomon codes is a method for unique decoding beyond half the minimum distance. We give a new variant of the Power decoding scheme, building upon the key equation of Gao. We show various interesting properties such as behavioural equivalence to the classical scheme using syndromes, as well as a new bound on the failure probability when the powering degree is 3.

Keywords Reed-Solomon code • Algebraic decoding • Power decoding

1 Introduction

Power decoding was originally developed by Schmidt, Sidorenko and Bossert for low-rate Reed–Solomon codes (RS) [5], and is usually capable of decoding almost as many errors as the Sudan decoder [8] though it is a unique decoder. If an answer is found, this is always the closest codeword, but in some cases the method will fail; in particular, this happens if two codewords are equally close to the received. With random errors this seems to happen exceedingly rarely, though a bound for the probability has only been shown for the simplest case of powering degree 2 [5, 10].

The algorithm rests on the surprising fact that a received word coming from a low-rate RS code can be "powered" to give received words of higher-rate RS codes having the same error positions. For each of these received words, one constructs a classical key equation by calculating the corresponding syndromes and solves them simultaneously for the same error locator polynomial.

Gao gave a variant of unique decoding up to half the minimum distance [1]: in essence, his algorithm uses a different key equation and with this finds the information polynomial directly. We here show how to easily derive a variant of Power decoding for Generalised RS (GRS) codes, Power Gao, where we obtain multiple of Gao's type of key equation, and we solve these simultaneously.

J.S.R. Nielsen (✉)
Institute of Communications Engineering, Ulm University, Ulm, Germany
e-mail: jsrn@jsrn.dk

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_32

We then show that Power Gao is *equivalent* to Power syndromes in the sense that they will either both fail or both succeed for a given received word. Power Gao has some "practical" advantages, though: it extends Power decoding to the case of using 0 as an evaluation point (which Power syndromes does not support); and the information is obtained directly when solving the key equations, so finding roots of the error locator and Forney's formula is not necessary.

The main theoretical advantage is that Power Gao seems easier to analyse: in particular, we show two new properties of Power decoding: (1) that whether Power decoding fails or not depends only on the error and not on the sent codeword; and (2) a new bound on the failure probability when the powering degree is 3.

We briefly sketched Power Gao already in [2], but its behaviour was not well analysed and its relation to Power syndromes not examined. In Sect. 2 we derive the powered Gao key equations, and in Sect. 3 we describe the complete algorithm and discuss computational complexity issues. In Sect. 4 we show the behavioural equivalence to Power syndromes as well as the new properties on Power decoding. Section 5 describes an explicit family of errors for which Power decoding will fail.

2 The Key Equations

Consider some finite field $\mathbb{F}$. The $[n, k, d]$ Generalised Reed-Solomon (GRS) code is the set

$$\mathscr{C} = \big\{\big(\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n)\big) \mid f \in \mathbb{F}[x] \wedge \deg f < k\big\}$$

where $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ are distinct, and the $\beta_1, \ldots, \beta_n \in \mathbb{F}$ are non-zero (not necessarily distinct). The α_i are called *evaluation points* and the β_i *column multipliers*. $\mathscr{C}$ has minimum distance $d = n - k + 1$ and the code is therefore MDS.

Consider now that some $\boldsymbol{c} = (c_1, \ldots, c_n)$ was sent, resulting from evaluating some $f \in \mathbb{F}[x]$, and that $\boldsymbol{r} = (\beta_1 r_1, \ldots, \beta_n r_n) = \boldsymbol{c} + (\beta_1 e_1, \ldots, \beta_n e_n)$ was the received word with (normalised) error $\boldsymbol{e} = (e_1, \ldots, e_n)$. Let $\mathscr{E} = \{i \mid e_i \neq 0\}$ and $\epsilon = |\mathscr{E}|$. In failure probability considerations, we consider the $|\mathbb{F}|$-ary symmetric channel.

Introduce $G \triangleq \prod_{i=1}^{n}(x - \alpha_i)$, and for any integer $t \geq 1$, let $R^{(t)}$ be the Lagrangian polynomial through the "powered" $\boldsymbol{r}$, i.e. the minimal degree polynomial satisfying $R^{(t)}(\alpha_i) = r_i^t$ for $i = 1, \ldots, n$. Naturally, we have $\deg R^{(t)} \leq n - 1$ and $R^{(t)}$ can be directly calculated by the receiver. As usual for key equation decoders, the algorithm will revolve around the notion of error locator: $\Lambda = \prod_{j \in \mathscr{E}}(x - \alpha_j)$. Choose now some $\ell \in \mathbb{N}$ subject to $\ell(k - 1) < n$. Then we easily derive the powered Gao key equations:

Proposition 1 $\Lambda R^{(t)} \equiv \Lambda f^t \mod G$

Proof Polynomials are equivalent modulo G if and only if they have the same evaluation at $\alpha_1, \ldots, \alpha_n$. For α_i where $e_i \neq 0$, both sides of the above evaluate to zero, while for the remaining α_i they give $\Lambda(\alpha_i){r_i}^t = \Lambda(\alpha_i) f(\alpha_i)^t$. □

3 The Decoding Algorithm

The key equations of Proposition 1 are non-linear in Λ and f, so the approach for solving them is to relax the equations into a linear system, similarly to classical key equation decoding. We will ignore the structure of the right hand-sides and therefore seek polynomials λ and $\psi^{(1)}, \ldots, \psi^{(\ell)}$ such that $\lambda R^{(t)} \equiv \psi^{(t)} \mod G$ as well as $\deg \lambda + t(k-1) \geq \deg \psi^{(t)}$ for $t = 1, \ldots, \ell$. We will call such $(\lambda, \psi^{(1)}, \ldots, \psi^{(\ell)})$ *a solution* to the key equations.

Clearly $(\Lambda, \Lambda f, \ldots, \Lambda f^\ell)$ is a solution. There are, however, infinitely many more, so the strategy is to find a solution such that $\deg \lambda$ is minimal; we will call this the *minimal solution*. Thus decoding can only succeed when Λ has minimal degree of all solutions. The probability of this occurring will be discussed in Sect. 4.

Conceptually, Power Gao decoding is then straightforward: pre-calculate G and from the received word, calculate $R^{(1)}, \ldots, R^{(\ell)}$. Find then a minimal solution $(\lambda, \psi_1, \ldots, \psi_\ell)$ with λ monic. If this has the valid structure of $(\Lambda, \Lambda f, \ldots, \Lambda f^\ell)$, then return f. Otherwise, declare decoding failure.

For Power syndromes, the key equations are similar to ours except that the modulo polynomials are just powers of x. In this case, finding a minimal solution is known as multi-sequence shift-register synthesis, and the fastest known algorithm is an extension of the Berlekamp–Massey algorithm [5] or the Divide-&-Conquer variant of this [6]. These can not handle the modulus G that we need, however.

A generalised form of multi-sequence shift-register synthesis was considered in [2], and several algorithms for finding a minimal solution were presented. The key equations for our case fit into this framework. We refer the reader to [2] for the details on these algorithms, but the asymptotic complexities when applied to Power Gao decoding are given in Table 1. The same complexities would apply to Power syndromes and also match the algorithms [5, 6] mentioned before. The other

Table 1 Complexities of solving the key equations for the three approaches discussed in [2]

Algorithm	O-complexity
Mulders–Storjohann	$\ell^2 n^2$
Alekhnovich	$\ell^3 n \log^2 n \log\log n$
Demand–Driven[a]	$\ell n^2 [\log n \log\log n]$

[a] If $\mathscr{C}$ is cyclic, then $G = x^n - 1$ since the α_i form a multiplicative group, and in this case the log-factors in square brackets can be removed.

steps of the decoding are easily seen to be cheaper than this; e.g. the calculation of $R^{(1)}, \dots, R^{(\ell)}$ by Lagrangian interpolation can be done trivially in $O(\ell n^2)$ or using fast Fourier techniques in $O(\ell n \log^2 n)$ [9, p. 231]. Thus Power Gao decoding is asymptotically as fast as Power syndromes.

4 Properties of the Algorithm

Power Gao will fail if $(\Lambda, \Lambda f, \dots, \Lambda f^\ell)$ is not the found minimal solution, so the question is when one can expect this to occur. Since the algorithm returns at most one codeword, it *must* fail for some received words whenever $\epsilon \geq d/2$. Whenever an answer is found, however, this must correspond to a closest codeword: any closer codeword would have its own corresponding error locator and information polynomial, and these would yield a smaller solution to the key equations.

We first show that Power syndromes is behaviourally equivalent to Power Gao. We will need to assume that the evaluation points $\alpha_i \neq 0$ for all i, which is a condition for Power syndromes decoding. This implies $x \nmid G$. We will use a "coefficient reversal" operator defined for any $p \in \mathbb{F}[x]$ as $\overline{p} = x^{\deg p} p(x^{-1})$.

In Power syndromes decoding, one considers $\boldsymbol{r}^{(t)} = (\beta_1 r_1^t, \dots, \beta_n r_n^t)$ for $t = 1, \dots, \ell$ as received words of GRS codes with parameters $[n, t(k-1)+1, n-t(k-1)]$, resulting from evaluating f^t; these "virtual" codes have the same evaluation points and column multipliers as $\mathscr{C}$. The $\boldsymbol{r}^{(t)}$ will therefore have the same error positions as $\boldsymbol{r}$, so the same error locator applies. For each t, we can calculate the syndrome $S^{(t)}$ corresponding to $\boldsymbol{r}^{(t)}$, which can be written as

$$S^{(t)} = \Big(\sum_{i=1}^{n} \frac{r_i^t \zeta_i}{1 - x\alpha_i} \bmod x^{n-t(k-1)+1} \Big)$$

where $\zeta_i = \prod_{j \neq i} (\alpha_i - \alpha_j)^{-1}$; see e.g. [4, p. 185]. By insertion one sees that

$$\overline{\Lambda} S^{(t)} \equiv \Omega^{(t)} \mod x^{n-t(k-1)+1}, \quad t = 1, \dots, \ell$$

where $\Omega^{(t)}$ is a certain polynomial satisfying $\deg \Omega^{(t)} < \deg \Lambda$. Note that we are using Λ reversed; indeed, one often defines error-locator as $\prod_{i \in \mathscr{E}} (1 - x\alpha_i) = \overline{\Lambda}$ when considering the syndrome key equation. The decoding algorithm follows simply from finding a minimal degree polynomial $\overline{\lambda}$ such that $\omega^{(t)} = (\overline{\lambda} S^{(t)} \bmod x^{n-t(k-1)+1})$ satisfies $\deg \lambda > \deg \omega^{(t)}$ for all t. The decoding method fails if $\overline{\lambda} \neq \gamma \overline{\Lambda}, \forall \gamma \in \mathbb{F}$. We now have:

Proposition 2 *Decoding using Power Gao fails if and only if decoding using Power syndromes fails.*

Proof Note first that $R^{(t)} = \sum_{i=1}^{n} r_i^t \zeta_i \prod_{j \neq i} (x - \alpha_j)$. By insertion we get $S^{(t)} \equiv \overline{R}^{(t)} \overline{G}^{-1} \mod x^{n-t(k-1)+1}$ (since $x \nmid G$). Power Gao fails if there is some $\lambda \in \mathbb{F}[x]$ which is not a constant times Λ and such that $\deg \lambda \leq \deg \Lambda$ and $\psi^{(t)} = (\lambda R^{(t)} \bmod G)$ has $\deg \psi^{(t)} < \deg \lambda + t(k-1) + 1$ for each $t = 1, \ldots, \ell$. This means there must be some $\omega^{(t)}$ with $\deg \omega^{(t)} \leq \deg \lambda - 1$ such that

$$\begin{aligned} \lambda R^{(t)} - \omega^{(t)} G &= \psi && \Longleftrightarrow \\ \overline{\lambda}\,\overline{R}^{(t)} - \overline{\omega}^{(t)}\overline{G} &= \overline{\psi}^{(t)} x^{\deg G + \deg \lambda - 1 - (\deg \lambda + t(k-1))} && \Longrightarrow \\ \overline{\lambda}\,\overline{R}^{(t)} &\equiv \overline{\omega}^{(t)}\overline{G} \mod x^{n - t(k-1) - 1} \end{aligned}$$

Dividing by $\overline{G}$, we see that $\overline{\lambda}$ and the $\overline{\omega}^{(t)}$ satisfy the congruences necessary to form a solution to the Power syndromes key equation, and they also satisfy the degree bounds. Showing the proposition in the other direction runs analogously. □

Corollary 3 (Combining [5] and Proposition 2) *Power Gao decoding succeeds if $\epsilon < d/2$. Let*

$$\tau(\ell) = \tfrac{\ell}{\ell+1} n - \tfrac{1}{2}\ell(k-1) - \tfrac{\ell}{\ell+1}$$

Then decoding will fail with high probability if $\epsilon > \tau(\hat{\ell})$, where $1 \leq \hat{\ell} \leq \ell$ is chosen to maximise $\tau(\ell)$.[1]

Between the above two bounds, Power decoding will sometimes succeed and sometimes fail. Simulations indicate that failure occurs with quite small probability. The only proven bound so far is for $\ell = 2$ where for exactly ϵ errors occurring, we have $P_f(\epsilon) < ({}^{q}/_{q-1})^{\epsilon} q^{3(\epsilon - \tau(2))}/(q-1)$, [5, 10].

We will give a new bound for $P_f(\epsilon)$ when $\ell = 3$, but we will first show a property which allows a major simplification in all subsequent analyses.

Proposition 4 *Power Gao decoding fails for some received word $\boldsymbol{r}$ if and only if it fails for $\boldsymbol{r} + \hat{\boldsymbol{c}}$ where $\hat{\boldsymbol{c}}$ is any codeword.*

Proof We will show that Power Gao decoding fails for $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ if and only if it fails for $\boldsymbol{e}$ as received word; since $\boldsymbol{c}$ was arbitrary, that implies the proposition.

Let $R_e^{(t)}$ be the power Lagrangians for $\boldsymbol{e}$ as received word, i.e. $R_e^{(t)}(\alpha_i) = e_i^t$ for each i and t, and let $R_e = R_e^{(1)}$. Consider a solution to the corresponding Power Gao key equations $(\lambda, \psi_1, \ldots, \psi_\ell)$; i.e. $\lambda R_e^{(t)} \equiv \psi_t \mod G$ and $\deg \lambda + t(k-1) + 1 > \deg \psi_t$. Let as usual $R^{(t)}$ be the power Lagrangians for $\boldsymbol{r}$ as received word and $R = R^{(1)}$. Note now that $R^{(t)} \equiv R^t \mod G$ since both sides of the congruence evaluate to the same at all α_i; similarly $R_e^{(t)} \equiv R_e^t \mod G$. Since $r_i = f(\alpha_i) + e_i$ linearity implies that $R = f + R_e$. Define $\psi_0 = \lambda$ and note that then also for $t = 0$ we have $\deg \lambda + t(k-1) + 1 > \deg \psi_t$. We then have the chain of congruences modulo G:

$$\lambda R^{(t)} \equiv \lambda R^t \equiv \lambda (f + R_e)^t \equiv \lambda \textstyle\sum_{s=0}^{t} \binom{t}{s} f^s R_e^{t-s} \equiv \sum_{s=0}^{t} \binom{t}{s} f^s \psi_{t-s} \mod G$$

[1] Decoding may succeed in certain degenerate cases, see [3, Proposition 2.39]. Failure is certain when using the method of [5] since what it considers “solutions” are subtly different than here.

Each term in the last sum has degree $s \deg f + \deg \psi_{t-s} < s(k-1) + \deg \lambda + (t-s)(k-1) + 1 = \deg \lambda + t(k-1) + 1$, which means that

$$\Big(\lambda,\ \textstyle\sum_{s=0}^{1} \binom{1}{s} f^s \psi_{1-s},\ \dots,\ \sum_{s=0}^{\ell} \binom{\ell}{s} f^s \psi_{\ell-s}\Big)$$

is a solution to the Power Gao key equations with $\boldsymbol{r}$ as a received word. The same argument holds in the other direction, so any solution to one of the key equations induces a solution to the other with the same first component; obviously then, their minimal solutions must be in bijection, which directly implies that they either both fail or neither of them fail. □

For the new bound on the failure probability, we first need a technical lemma:

Lemma 5 *Let $U \in \mathbb{F}[x]$ of degree N, and let $K_1 < K_2 < K_3 < N$ be integers. Let $S = \{(f_1, f_2, f_3) \mid f_1 f_3 \equiv f_2^2 \mod U,\ f_2 \text{ monic},\ \forall t. \deg f_t < K_t\}$. Then*

$$\begin{aligned} |S| &\le 3^{K_2-1} q^{K_2} && \text{if } K_1 + K_3 - 2 < N \\ |S| &\le 2^{K_1+K_3-2} q^{K_1+K_2+K_3-N-2} && \text{if } K_1 + K_3 - 2 \ge N \end{aligned}$$

Proof If $K_1 + K_3 - 2 < N$, then $f_1 f_3 \equiv f_2^2 \mod U$ implies $f_1 f_3 = f_2^2$. We can choose a monic f_2 in $(q^{K_2} - 1)/(q-1)$ ways. For each choice, then f_2 has at most $K_2 - 1$ prime factors, so the factors of f_2^2 can be distributed among f_1 and f_3 in at most 3^{K_2-1} ways. Lastly, the leading coefficient of f_1 can be chosen in $q-1$ ways.

If $K_1 + K_3 - 2 \ge N$, then for each choice of f_2, the product $f_1 f_3$ can be among $\{f_2^2 + gU \mid \deg g \le K_1 + K_3 - 2 - N\}$. This yields at most $q^{K_1+K_2+K_3-N-2}/(q-1)$ candidates for $f_1 f_3$; each of these has at most $K_1 + K_3 - 2$ unique prime factors, which can then be distributed among f_1 and f_3 in at most $2^{K_1+K_3-2}$ ways. Again, the leading coefficient of f_1 leads to a factor $q-1$ more. □

Proposition 6 *For $\ell = 3$, the probability that Power decoding (Gao or Syndrome) fails when $\epsilon > d/2$ is at most*

$$\begin{aligned} (q/(q-1))^{\epsilon} (3/q)^{2\epsilon-(n-2k+1)} q^{3(\epsilon-\tau(2))+k-1} && \text{if } \epsilon < \tau(2) - \tfrac{1}{3}k + 1 \\ (q/(q-1))^{\epsilon} 2^{2(2\epsilon-d)+2(k-1)} q^{4(\epsilon-\tau(3))-2} && \text{if } \epsilon \ge \tau(2) - \tfrac{1}{3}k + 1 \end{aligned}$$

Proof By Proposition 4, we can assume that $\boldsymbol{c} = 0$, i.e. that $\boldsymbol{r} = \boldsymbol{e}$. That means $R^{(t)}(\alpha_i) = 0$ for $i \notin \mathcal{E}$, so we can write $R^{(t)} = E^{(t)} \Upsilon$ for some $E^{(t)}$ with $\deg E^{(t)} < \epsilon$, where $\Upsilon = G/\Lambda$ is the "truth-locator". Power Gao decoding fails if and only if there exists $(\lambda, \psi_1, \psi_2, \psi_3)$ such that $\lambda \ne \Lambda$, $\deg \lambda \le \deg \Lambda$, $\deg \lambda + t(k-1) + 1 > \deg \psi_t$ for $t = 1, 2, 3$ as well as

$$\lambda R^{(t)} \equiv \psi_t \mod G \qquad \Longleftrightarrow \qquad \lambda E^{(t)} \equiv \hat{\psi}_t \mod \Lambda$$

where $\hat{\psi}_t = \psi_t / \Upsilon$. Note that ψ_t must be divisible by Υ since both the modulus and the left-hand side of the first congruence is.

Denote by E the unique polynomial with degree less than ϵ having $E(\alpha_i) = e_i$ for $i \in \mathcal{E}$. For any $i \in \mathcal{E}$ then $(\lambda E^{(t)})(\alpha_i) = \lambda(\alpha_i)\Upsilon(\alpha_i)^{-1}e_i^t$, which means $\lambda E^{(t)} \equiv \hat{\lambda}E^t \mod \Lambda$ for some polynomial $\hat{\lambda}$.

After having chosen error positions, drawing error values uniformly at random is the same as drawing uniformly at random from possible E. So given the error positions, the probability that Power decoding will fail is $T_\Lambda/(q-1)^\epsilon$, where T_Λ is the number of choices of E such that there exist $\hat{\lambda}, \hat{\psi}_1, \hat{\psi}_2, \hat{\psi}_3$ having

$$\hat{\lambda}E^t \equiv \hat{\psi}_t \mod \Lambda, \quad t = 1, 2, 3$$

as well as $\deg \hat{\psi}_t < \deg \Lambda + t(k-1) + 1 - (n - \deg \Lambda) = 2\epsilon - (n - t(k-1) - 1)$.

Note that these congruences imply $\hat{\psi}_1\hat{\psi}_3 \equiv \hat{\psi}_2^2 \mod \Lambda$. Denote by $\hat{T}_\Lambda$ the number of triples $(\hat{\psi}_1, \hat{\psi}_2, \hat{\psi}_3) \in \mathbb{F}[x]^3$ satisfying just this congruence as well as the above degree bounds. Then $\hat{T}_\Lambda \geq T_\Lambda$: for if $\gcd(\hat{\lambda}, \Lambda) = 1$ then two different values of E could not yield the same triple since $E \equiv \hat{\psi}_2/\hat{\psi}_1 \mod \Lambda$ uniquely determines E. Alternatively, if $\gcd(\hat{\lambda}, \Lambda) = g \neq 1$ then the congruences imply $g \mid \hat{\psi}_t$ for all t, so that $E \equiv (\hat{\psi}_2/g)/(\hat{\psi}_1/g) \mod \Lambda/g$. This leaves a potential $q^{\deg g}$ possible other choices of E yielding the same triple; but all these possibilities are counted in the triples since $(t\psi_1/g, t\psi_2/g, t\psi_3/g)$ will be counted for any $t \in \mathbb{F}[x]$ with $\deg t < \deg g$.

In fact, we have $\hat{T}_\Lambda \geq (q-1)T_\Lambda$, since whenever $(\hat{\psi}_1, \hat{\psi}_2, \hat{\psi}_3)$ is counted, so is $(\beta\hat{\psi}_1, \beta\hat{\psi}_2, \hat{\psi}_3)$, and this doesn't change the fraction $\hat{\psi}_1/\hat{\psi}_2$. Thus, we over-estimate instead $\hat{T}_\Lambda/(q-1)$ by counting the number of triples where $\hat{\psi}_2$ is monic. Lemma 5 gives an upper bound for exactly this number, setting $N = \epsilon$ and $K_t = 2\epsilon - (n - t(k-1) - 1)$. Divided by $(q-1)^\epsilon$, this is then an upper bound on the failure probability given the error positions. But since this probability is independent of the choice of Λ, it is also the failure probability over all errors vectors of weight ϵ. □

By experimentation, one can demonstrate that the bound is not tight: for instance, for a $[250, 30, 221]$ GRS code, the bound is greater than 1 for $\epsilon > 143$, while simulation indicate almost flawless decoding up to 147 errors. However, in a relative and asymptotic sense the above bound is strong enough to show that up to $\tau(3)$ errors can be corrected with arbitrary low failure probability:

Corollary 7 *Having $\ell = 3$, then for any $\delta > 0$, with $n \to \infty$ while keeping q/n, k/n and ϵ/n constant, the probability that Power decoding fails goes to 0 when $\epsilon/n < \tau(3)/n - \delta$.*

Proof (Proof sketch) We consider only the high-error failure probability of Proposition 6. For $n \to \infty$, the failure probability bound will approach

$$2^{2(2\epsilon-d)+2(k-1)}q^{4(\epsilon-\tau(3))} \leq (q^n)^{4(\epsilon/n-\tau(3)/n)+(2(2\epsilon/n-d/n)+2k/n)/\log q}$$

The contribution $(2(2\epsilon/n - d/n) + 2k/n)/\log q$ goes to 0 as $n \to \infty$, leaving $(q^n)^a$ for $a = 4(\epsilon/n - \tau(3)/n) < -4\delta$. □

5 A Family of Bad Errors

Power decoding will usually fail when the powered key equations are linearly dependent; in particular, it will fail if one of the key equations is trivially satisfied.

An anonymous reviewer of this paper suggested the following construction of errors where, for a given sent codeword, the second key equation will be trivial: let $\mathbb{F}$ be a non-binary field, and let $\hat{\boldsymbol{c}} \in \mathscr{C}$ be some non-zero codeword, obtained as the evaluation of $\hat{f}$ with $\deg \hat{f} < k$. Choose $d/2 \leq \epsilon \leq \tau(2)$ positions for which $\hat{\boldsymbol{c}}$ is non-zero. Let then $\boldsymbol{e} = (e_1, \ldots, e_n)$ be given by $e_i = -2\hat{c}_i$ when i is one of the chosen position, and $e_i = 0$ otherwise. If $\hat{\boldsymbol{r}} = \hat{\boldsymbol{c}} + \boldsymbol{e}$ is received, then the second Lagrangian $\hat{R}^{(2)}$ equals $\hat{f}^2$, i.e. $\deg \hat{R}^{(2)} < 2k - 1$ (in other words, we have $\hat{\boldsymbol{r}}^{(2)} = \hat{\boldsymbol{c}}^{(2)}$ so the squared received word is a codeword in the "squared" GRS code). That means that for *any* $\lambda \in \mathbb{F}[x]$, then $\deg(\lambda \hat{R}^{(2)} \bmod G) \leq \deg \lambda + 2k - 1$, and so the second key equation is useless.

Clearly then, if $\hat{\boldsymbol{r}}$ is received, then almost surely[2] Power decoding will fail when $\ell = 2$, and it is easy to show that it will also fail when $\ell > 2$.

From Proposition 4 it follows that decoding will also fail when receiving $\boldsymbol{c} + \boldsymbol{e}$ for *any* sent codeword $\boldsymbol{c} \in \mathscr{C}$; in particular when sending $\boldsymbol{0}$ and receiving $\boldsymbol{e}$. This might at first seem counter-intuitive since the second Lagrangian $E^{(2)}$ when $\boldsymbol{e}$ is the received word does not have low degree (i.e. $\boldsymbol{e}^{(2)}$ is *not* in the squared GRS code). However, in this case the key equation involving $E^{(2)}$ will be linearly dependent on that involving $E = E^{(1)}$, and so will not add further requirements. This can be seen directly as follows: since $\boldsymbol{e} = \hat{\boldsymbol{r}} - \hat{\boldsymbol{c}}$ then $\boldsymbol{e}^{(2)} = \hat{\boldsymbol{r}}^{(2)} + \hat{\boldsymbol{c}}^{(2)} - 2\hat{\boldsymbol{c}} \star \hat{\boldsymbol{r}} = 2\hat{\boldsymbol{c}}^{(2)} - 2\hat{\boldsymbol{c}} \star \hat{\boldsymbol{r}}$, where $\star$ is the component-wise product. Thus $E^{(2)} \equiv 2\hat{f}^2 - 2\hat{f}R \mod G$. So if $\lambda \in \mathbb{F}[x]$ satisfies the first key equation, i.e. $\deg(\lambda E \bmod G) \leq \deg \lambda + k - 1$, then we get

$$\deg(\lambda E^{(2)} \bmod G) = \deg(\lambda \hat{f}^2 + \lambda E \hat{f} \bmod G) \leq \deg \lambda + 2(k-1)$$

So λ satisfies the second key equation.

The "bad error" construction can easily be generalised for higher ℓ whenever $\mathbb{F}$ has ℓ'th roots of unity different from 1: then e_i can be chosen as $(\xi_i - 1)\hat{c}_i$ where $\xi_i \neq 1$ is any of those roots of unity. Then for $\hat{\boldsymbol{r}} = \hat{\boldsymbol{c}} + \boldsymbol{e}$ we get $\hat{\boldsymbol{r}}^{(\ell)} = \hat{\boldsymbol{c}}^{(\ell)}$ and so $\deg R^{(\ell)} \leq \ell(k-1)$.

A full Power Gao decoder has been implemented in Sage v5.13 [7] and is available for download at http://jsrn.dk/code-for-articles. Also implemented is randomly constructing "bad errors" $\boldsymbol{e}$ as above (for any ℓ), and a demonstration that Power decoding fails for $\hat{\boldsymbol{r}}$, $\boldsymbol{e}$ and $\boldsymbol{c} + \boldsymbol{e}$ for any random codeword $\boldsymbol{c}$.

[2] As in Theorem 3, failure is not certain but extremely unlikely for just a few errors beyond $d/2$.

References

1. Gao, S.: A new algorithm for decoding Reed-Solomon codes. In: Communications, Information and Network Security, no. 712 in S. Eng. and Comp. Sc., pp. 55–68. Springer (2003)
2. Nielsen, J.S.R.: Generalised multi-sequence shift-register synthesis using module minimisation. In: Proceedings of IEEE ISIT, Istanbul, 882–886 (2013)
3. Nielsen, J.S.R.: List decoding of algebraic codes. Ph.D. thesis, Technical University of Denmark (2013). Available at jsrn.dk
4. Roth, R.: Introduction to Coding Theory. Cambridge University Press, New York (2006)
5. Schmidt, G., Sidorenko, V., Bossert, M.: Syndrome decoding of Reed-Solomon codes beyond half the minimum distance based on shift-register synthesis. IEEE Trans. Inf. Theory **56**(10), 5245–5252 (2010)
6. Sidorenko, V., Bossert, M.: Fast skew-feedback shift-register synthesis. Designs, Codes Cryptogr. **70**, 55–67 (2014)
7. Stein, W.A.: Sage Mathematics Software. http://www.sagemath.org (2014)
8. Sudan, M.: Decoding of Reed–Solomon codes beyond the error-correction bound. J. Complex. **13**(1), 180–193 (1997)
9. von zur Gathen, J., Gerhard, J.: Modern Computer Algebra, 3rd edn. Cambridge University Press, Cambridge (2012)
10. Zeh, A., Wachter, A., Bossert, M.: Unambiguous decoding of generalized Reed–Solomon codes beyond half the minimum distance. In: Proceedings of IZS, Zurich, 63–66 (2012)

On Fibre Products of Kummer Curves with Many Rational Points over Finite Fields

Ferruh Özbudak, Burcu Gülmez Temür, and Oğuz Yayla

Abstract We determined the number of rational points of fibre products of two Kummer covers over a rational point of the projective line in a recent work of F. Özbudak and B.G. Temür (Des Codes Cryptogr 70(3):385–404, 2014), where we also constructed explicit examples, including a record and two new entries for the current Table of Curves with Many Points (manYPoints: Table of curves with many points. http://www.manypoints.org (2014). Accessed 30 Sep 2014). Using the methods given in Özbudak and Gülmez Temür (Des Codes Cryptogr 70(3):385–404, 2014), we made an exhaustive computer search over $\mathbb{F}_5$ and $\mathbb{F}_7$ by the contributions of O. Yayla and at the end of this search we obtained 12 records and 6 new entries for the current table; in particular, we observed that the fibre product with genus 7 and 36 rational points coincides with the Ihara bound, thus we concluded that the maximum number $N_7(7)$ of $\mathbb{F}_7$-rational points among all curves of genus 7 is 36 (Özbudak et al., Turkish J Math 37(6):908–913, 2013). Recently, we made another exhaustive computer search over $\mathbb{F}_{11}$. In this paper we are representing the results as three records and three new entries for the current table.

Keywords Curves with many points • Algebraic function fields

F. Özbudak (✉)
Department of Mathematics and Institute of Applied Mathematics, Middle East Technical University, Dumlupınar Bul., No:1, 06800 Ankara, Turkey
e-mail: ozbudak@metu.edu.tr

B. Gülmez Temür
Department of Mathematics, Atılım University, İncek, Gölbaşı, 06836 Ankara, Turkey
e-mail: burcu.temur@atilim.edu.tr

O. Yayla
Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Altenberger Strasse 69, A-4040 Linz, Austria

Department of Mathematics, Hacettepe University, Beytepe, 06800 Ankara, Turkey
e-mail: oguz.yayla@hacettepe.edu.tr

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_33

1 Introduction

Let $\mathbb{F}_q$ be a finite field with $q = p^n$ elements, where p is a prime number. For an absolutely irreducible, nonsingular and projective curve χ defined over $\mathbb{F}_q$, let N be the number of $\mathbb{F}_q$-rational points of χ and $g(\chi)$ be its genus. The number N is bounded by the Hasse-Weil bound

$$N \leq q + 1 + 2g(\chi)\sqrt{q}. \tag{1}$$

If the bound in (1) is attained then χ is called a maximal curve. There are some improvements on (1) especially when $g(\chi)$ is large [4, 5, 8, 12, 14]. Let $N_q(g)$ denote the maximum number of $\mathbb{F}_q$-rational points among the absolutely irreducible, nonsingular and projective curves of genus g defined over $\mathbb{F}_q$. It is an important problem to determine $N_q(g)$ and to construct explicit curves with many rational points (see [3], and [7] for the current tables). There are many applications to areas including coding theory, cryptography and quasi-random points [5, 8, 9, 13, 14]. Some types of fibre products of Kummer covers of the projective line were studied and such explicit curves with many points were found [2, 6] and [12].

The theory of algebraic curves is essentially equivalent to the theory of algebraic function fields. Throughout this paper we will use the language of function fields [12]. We will call a degree one place of an algebraic function field a *rational place* of the function field. Consider the fibre product

$$\begin{aligned} y_1^{n_1} &= h_1(x), \\ y_2^{n_2} &= h_2(x), \end{aligned} \tag{2}$$

where $n_1, n_2 \geq 2$ are integers and $h_1(x), h_2(x) \in \mathbb{F}_q(x)$. Let E be the algebraic function field $E = \mathbb{F}_q(x, y_1, y_2)$ with the system of equations in (2). If the number of rational places of E is greater than $N_{max,q,g}/\sqrt{2}$, where $N_{max,q,g}$ is the best known upper bound for $N_q(g)$ (Hasse-Weil, Serre, Ihara, Oesterlé etc.) – this is the case if there is no entry for the lower bound in the tables [7] – then we call it a *new entry*. If the number of rational places of E is greater than the existing lower bound in the tables [7], then we call it a *record*. We have given explicit examples of fibre products of Kummer covers with many rational points in [10], in particular Examples 4 and 5 (in Sect. 3) were new entries for the table [7].

We made an exhaustive computer search on n_1, n_2, h_1 and h_2 to find such function fields $E = \mathbb{F}_q(x, y_1, y_2)$ with many rational places over the finite fields $\mathbb{F}_5$, $\mathbb{F}_7$, at the end of that search, we obtained 12 records and 6 new entries for the current tables [7] presented in Tables 1–3 (see [11]). Recently, we made another exhaustive computer search over $\mathbb{F}_{11}$ and in this paper we are representing these new results in Tables 4 and 5. In all the exhaustive computer searches mentioned above

Table 1 Algebraic function fields with many rational places over $\mathbb{F}_5$ (records)

n_1	n_2	$h_1(x) = \frac{h_{1,1}(x)}{h_{1,2}(x)}$	$h_2(x) = \frac{h_{2,1}(x)}{h_{2,2}(x)}$	g	N	$N_{min,q,g}$
2	2	$\frac{3x^3+2x^2+2x+1}{x^2+2x+4}$	$\frac{2x^3+4x^2+1}{x^2+2x+4}$	6	22	21
4	4	$\frac{(x)(x^2+x+2)}{x+4}$	$\frac{(x+4)(x^2+2x+4)}{x}$	25	56	52
4	4	$\frac{(x+4)(x^2+4x+2)}{x+3}$	$\frac{4(x+4)(x^2+3x+4)}{(x+3)^2}$	27	56	52
4	4	$\frac{x^6+3x^4+4x^3+x^2+2x+2}{x+2}$	$\frac{3x^4+4x^3+2x^2+x+1}{1}$	29	64	52

Table 2 Algebraic function fields with many rational places over $\mathbb{F}_7$ (records)

n_1	n_2	$h_1(x) = \frac{h_{1,1}(x)}{h_{1,2}(x)}$	$h_2(x) = \frac{h_{2,1}(x)}{h_{2,2}(x)}$	g	N	$N_{min,q,g}$
3	2	$\frac{4x^2+4x+5}{1}$	$\frac{2(x^2+x+3)(x^2+3x+1)}{1}$	5	26	24
2	3	$\frac{6(x+6)(x^2+1)}{1}$	$\frac{4(x+5)(x^2+1)^2}{1}$	6	27	25
3	3	$\frac{5(x+2)(x+5)}{x}$	$\frac{3x^2(x+5)}{x+3}$	7	36	30
3	3	$\frac{x^2+1}{x}$	$\frac{x^2+4}{1}$	10	39	36
3	6	$\frac{6(x^2+1)}{1}$	$\frac{(x+1)(x+6)^2}{x+5}$	16	54	45
2	6	$\frac{6(x+3)(x^2+x+3)}{1}$	$\frac{4(x+3)^2(x^2+3x+6)}{x+2}$	18	52	51
3	6	$\frac{x(x+1)}{x+4}$	$\frac{(x+4)^3}{x(x+5)}$	19	63	54
6	6	$\frac{3x^2(x+1)}{x+3}$	$\frac{2x(x+1)(x+3)}{x+1}$	22	72	63

Table 3 Algebraic function fields with many rational places over $\mathbb{F}_7$ (new entries)

n_1	n_2	$h_1(x) = \frac{h_{1,1}(x)}{h_{1,2}(x)}$	$h_2(x) = \frac{h_{2,1}(x)}{h_{2,2}(x)}$	g	N	$\lceil \frac{N_{max,q,g}}{\sqrt{2}} \rceil$
2	6	$\frac{6(x+3)(x^2+x+3)}{1}$	$\frac{4x^2(x^2+x+3)}{x+5}$	14	44	41
2	6	$\frac{2(x+3)(x+4)(x+6)}{1}$	$\frac{3(x+3)^2(x^2+2x+3)}{x+4}$	15	52	43
2	6	$\frac{4(x+2)(x^2+4)}{1}$	$\frac{2(x+2)^2(x+5)(x^2+x+3)}{1}$	20	54	53
3	6	$\frac{6(x+6)(x^2+6x+4)}{x+4}$	$\frac{3(x+6)^2(x^2+5x+5)}{1}$	28	72	68
6	6	$\frac{3x(x+2)(x+3)}{1}$	$\frac{6x^2(x+4)}{(x+3)^2}$	40	108	90
6	6	$\frac{4(x+1)(x+5)(x+6)}{1}$	$\frac{3(x+6)^2(x^2+4x+5)}{x+1}$	49	114	107

Table 4 Algebraic function fields with many rational places over $\mathbb{F}_{11}$ (records)

n_1	n_2	$h_1(x) = \frac{h_{1,1}(x)}{h_{1,2}(x)}$	$h_2(x) = \frac{h_{2,1}(x)}{h_{2,2}(x)}$	g	N	$N_{min,q,g}$
2	5	$\frac{2x^2+7x+2}{x+7}$	$\frac{6x^2+9x+2}{x+2}$	6	41	40
10	2	$\frac{6x^2+x+1}{7x^3+1}$	$x^2 + 9 * x + 4$	8	44	42
2	5	$\frac{2x^2+1}{x+2}$	$10x^3 + 5x^2 + x + 7$	13	62	60

Table 5 Algebraic function fields with many rational places over $\mathbb{F}_{11}$ (new entries)

n_1	n_2	$h_1(x) = \frac{h_{1,1}(x)}{h_{1,2}(x)}$	$h_2(x) = \frac{h_{2,1}(x)}{h_{2,2}(x)}$	g	N	$\lceil \frac{N_{max,q,g}}{\sqrt{2}} \rceil$
5	5	$2x^2 + x$	$\frac{5x^2+3x+3}{x^2}$	16	75	64
2	10	$\frac{8x^2+6x+9}{x+2}$	$2x^4 + 8x^3 + 10x^2 + 4x$	22	84	81
10	5	$9x^3 + 7x^2 + 3x + 5$	$6x^4 + 6x^3 + 4x + 1$	36	150	119

we used the method given in [10] in order to determine the number of rational places of E over $\mathbb{F}_q$.

2 Fibre Products of Kummer Covers

Before Theorem 2 we need to develop some tools that we use in its proof.

Proposition 1 ([10]) *Let C_1, C_2 be subgroups of $\mathbb{F}_q^*$ with $|C_1| = \bar{n}_1$, $|C_2| = \bar{n}_2$. Let m be a positive integer with $m \mid (q-1)$ and N be an arbitrary integer. Let $\mathscr{S} = \{(x_1, x_2) \in C_1 \times C_2 :$ there exists $s \in \mathbb{F}_q^*$ such that $x_1^N x_2 = s^m\}$. Then the cardinality $|\mathscr{S}|$ of $\mathscr{S}$ is*

$$|\mathscr{S}| = \gcd(\bar{n}_1, N) \gcd\left(\frac{\bar{n}_1}{\gcd(\bar{n}_1, N)}, \bar{n}_2\right) \gcd\left(\operatorname{lcm}\left(\frac{\bar{n}_1}{\gcd(\bar{n}_1, N)}, \bar{n}_2\right), \frac{q-1}{m}\right).$$

Moreover let C be the subset of $\mathbb{F}_q^*$ defined as

$$C = \left\{y \in \mathbb{F}_q^* \text{:there exists } (x_1, x_2) \in C_1 \times C_2 \text{ and } s \in \mathbb{F}_q^* \text{ such that } y = x_1^N x_2 s^m\right\}.$$

Then C is a subgroup of $\mathbb{F}_q^*$ with the cardinality

$$|C| = \operatorname{lcm}\left(\operatorname{lcm}\left(\frac{\bar{n}_1}{\gcd(\bar{n}_1, N)}, \bar{n}_2\right), \frac{q-1}{m}\right).$$

Proof (see Proposition 1 in [10]) □

The following theorem was one of our main results (see Theorem 4 and Corollary 1 in [10]).

Theorem 2 *Under the notations and assumptions of Proposition 1 we further define $\hat{m} = \gcd\left(\frac{q-1}{A}, m\right)$, where $A = \operatorname{lcm}\left(\frac{\bar{n}_1}{\gcd(\bar{n}_1, N)}, \bar{n}_2\right)$. Then we have that*

$|\mathscr{S}| = \dfrac{\bar{n}_1\bar{n}_2}{m}\hat{m}$. *Moreover, let* $m_2 = \gcd(n_2', n_1')$ *and* $E = \mathbb{F}_q(x, y_1, y_2)$ *be the algebraic function field with*

$$\begin{aligned} y_1^{n_1} &= h_1(x), \\ y_2^{n_2} &= h_2(x). \end{aligned} \tag{3}$$

Assume that the full constant field of E *is* $\mathbb{F}_q$ *and* $[E : \mathbb{F}_q(x)] = n_1 n_2$ *and assume that* $\bar{n}_1 \mid (q-1), \bar{n}_2 \mid (q-1)$ *and* $m_2 \mid (q-1)$. *Let* $\bar{n}_i = \gcd(n_i, a_i)$, $n_i' = \frac{n_i}{\bar{n}_i}$, *and* $a_i' = \frac{a_i}{\bar{n}_i}$ *for* $i = 1, 2$. *As* $\gcd(n_1', a_1') = 1$, *we choose integers* A_1 *and* B_1 *such that* $A_1 n_1' + B_1 a_1' = 1$. *Let*

$$A = \operatorname{lcm}\left(\frac{\bar{n}_1}{\gcd(-a_2' B_1, \bar{n}_1)}, \bar{n}_2\right).$$

Let $\hat{m}_2 = \gcd\left(\frac{q-1}{A}, m_2\right)$. *Then there exist either no or exactly* $(\bar{n}_1\bar{n}_2\hat{m}_2)$ *rational places of* E *over* P_0. *For* $1 \le i \le 2$, *the evaluation of* $f_i(x) \in \mathbb{F}_q(x)$ *at* P_0 *is denoted by* $f_i(u)$. *Furthermore, there exists a rational place of* E *over* P_0 *if and only if all of the following conditions hold:*

C1: $f_1(u)$ *is an* $\bar{n}_1$*-power in* $\mathbb{F}_q$.

C2: $f_2(u)$ *is an* $\bar{n}_2$*-power in* $\mathbb{F}_q$.

C3: *Assume that the conditions in items C1, C2 above hold and let* $\alpha_1, \alpha_2 \in \mathbb{F}_q^*$ *such that* $\alpha_1^{\bar{n}_1} = f_1(u)$ *and* $\alpha_2^{\bar{n}_2} = f_2(u)$. *Let*

$$B = \operatorname{lcm}\left(A, \frac{q-1}{m_2}\right).$$

Then

$$\left(\alpha_1^{-a_2' B_1}\alpha_2\right)^B = 1.$$

Proof (see [10]) □

Remark 3 Figures 1–3 below representing the ramification and inertia indices of degree one places lying over P_0 on some intermediate fields in between $\mathbb{F}_q(x)$ and E may give an idea about the proof of Theorem 2.

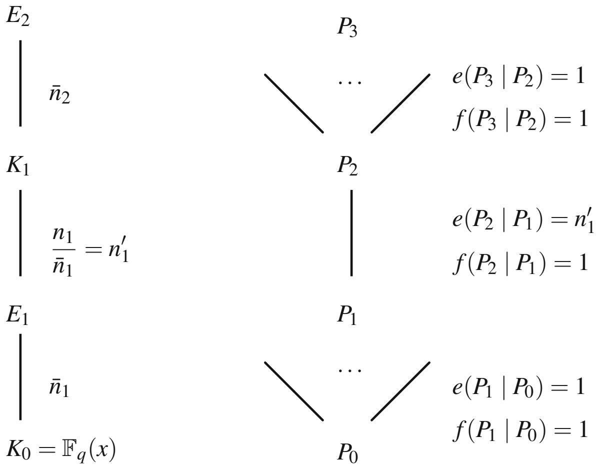


Fig. 1 E_2 extension over $\mathbb{F}_q(x)$

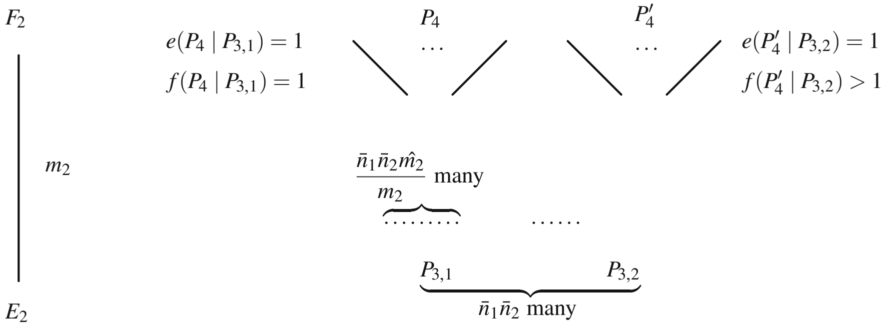

F_2
m_2
E_2
$e(P_4 \mid P_{3,1}) = 1$
$f(P_4 \mid P_{3,1}) = 1$
P_4
$\dots$
P'_4
$\dots$
$e(P'_4 \mid P_{3,2}) = 1$
$f(P'_4 \mid P_{3,2}) > 1$
$\frac{\bar{n}_1\bar{n}_2\hat{m}_2}{m_2}$ many
$\dots\dots$
$P_{3,1}$
$P_{3,2}$
$\bar{n}_1\bar{n}_2$ many

Fig. 2 F_2 extension over E_2

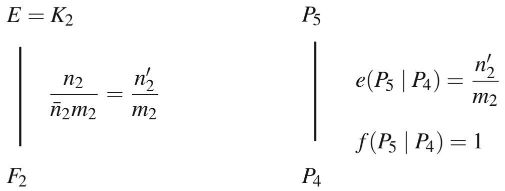

$E = K_2$
$\frac{n_2}{\bar{n}_2 m_2} = \frac{n'_2}{m_2}$
F_2
P_5
$e(P_5 \mid P_4) = \frac{n'_2}{m_2}$
$f(P_5 \mid P_4) = 1$
P_4

Fig. 3 K_2 extension over F_2

3 Examples

In this section, we present explicit examples of fibre products of Kummer extensions with many rational places obtained by using Theorem 2 (see Section 5 in [10], in particular Examples 4 and 5 below were new entries for the tables in [7]).

Example 4 Let $E = \mathbb{F}_{5^3}(x, y_1, y_2)$ be the function field over $\mathbb{F}_{5^3}$ given by the following equations:

$$\begin{aligned} y_1^2 &= x^3 + x \\ y_2^2 &= x^3 + x + 2 \end{aligned}$$

The genus of E is $g(E) = 4$ and $N(E) = 170$.

Example 5 Let $E = \mathbb{F}_{11^2}(x, y_1, y_2)$ be the function field over $\mathbb{F}_{11^2}$ given by the following equations:

$$\begin{aligned} y_1^2 &= x^3 + x \\ y_2^{12} &= x^2(1 - x^2) \end{aligned}$$

The genus of E is $g(E) = 31$ and $N(E) = 612$.

Example 6 Let $E = \mathbb{F}_{13^2}(x, y_1, y_2)$ be the function field over $\mathbb{F}_{13^2}$ given by the following equations:

$$\begin{aligned} y_1^2 &= x^7 + 1 \\ y_2^7 &= -x^7 - 1 \end{aligned}$$

The genus of E is $g(E) = 36$ and $N(E) = 1106$. This function field is maximal.

4 Search for Curves with Many Points

First we represent the genus computation for fibre products of two Kummer covers over finite fields $\mathbb{F}_q$.

Proposition 7 ([11]) *Let $F_1 = \mathbb{F}_q(x, y_1)$ and $F_2 = \mathbb{F}_q(x, y_2)$ be the algebraic function fields with $y_1^{n_1} = h_1(x) = \frac{h_{1,1}(x)}{h_{1,2}(x)}$ and $y_2^{n_2} = h_2(x) = \frac{h_{2,1}(x)}{h_{2,2}(x)}$ respectively, where $h_{1,1}(x), h_{1,2}(x)$, $h_{2,1}(x), h_{2,2}(x) \in \mathbb{F}_q[x]$ then the compositum*

$F_1F_2 = E = \mathbb{F}_q(x, y_1, y_2)$ and the genus $g(E)$ of E is equal to:

$$g(E) = 1 - n_1n_2 + \tfrac{1}{2}n_1n_2\left(1 - \frac{1}{\operatorname{lcm}\left(\frac{n_1}{\gcd(n_1,|d_1|)}, \frac{n_2}{\gcd(n_2,|d_2|)}\right)}\right) + \tfrac{1}{2}n_1n_2\sum_{p(x)\in R}\left(1 - \frac{1}{\operatorname{lcm}\left(\frac{n_1}{\gcd(n_1,a_{p,1})}, \frac{n_2}{\gcd(n_2,a_{p,2})}\right)}\right)\deg(p(x)).$$

where $d_1 = \deg h_{1,2}(x) - \deg h_{1,1}(x)$, $d_2 = \deg h_{2,2}(x) - \deg h_{2,1}(x)$, R is the set of all irreducible polynomials in the polynomial ring $\mathbb{F}_q[x]$ and $a_{p,i}$ is the multiplicity of $p(x) \in R$ as a zero or pole of $h_i(x)$ for $i = 1, 2$. If $p(x) \in R$ is neither a zero nor a pole of $h_i(x)$ then obviously $a_{p,i} = 0$ and the summation is finite as each rational function has finitely many zeros and poles.

Remark 8 The proposition above can be proved using Proposition 3.7.3 in [12] on Kummer extensions and Abhyankar's lemma (see Proposition 3.8.9 in [12]).

We made an exhaustive computer search for finding new curves with many points over $\mathbb{F}_5$, $\mathbb{F}_7$ (see [11]) and $\mathbb{F}_{11}$ (with the help of Magma [1]), totally we have 15 records and 9 new entries for the current tables [7]. We represent our latest contributions in Tables 4 and 5 above to the current Table of Curves with Many Points [7].

Acknowledgements We would like to thank the anonymous reviewers for their useful suggestions. The authors were partially supported by TÜBİTAK under Grant No. TBAG-109T672.

References

1. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symb. Comput. **24**(3–4), 235–265 (1997). Computational algebra and number theory (London, 1993)
2. Garcia, A., Garzon, A.: On Kummer covers with many rational points over finite fields. J. Pure Appl. Algebra **185**(1–3), 177–192 (2003)
3. van der Geer, G., van der Vlugt, M.: Tables of curves with many points. Math. Comput. **69**(230), 797–810 (2000)
4. Hirschfeld, J.W.P.: Projective Geometries over Finite Fields. Oxford Mathematical Monographs, 2nd edn. The Clarendon Press/Oxford University Press, New York (1998)
5. Hirschfeld, J.W.P., Korchmáros, G., Torres, F.: Algebraic Curves over a Finite Field. Princeton Series in Applied Mathematics. Princeton University Press, Princeton (2008)
6. Kawakita, M.Q.: Kummer curves and their fibre products with many rational points. Appl. Algebra Eng. Commun. Comput. **14**(1), 55–64 (2003)
7. manYPoints: Table of curves with many points. http://www.manypoints.org (2014). Accessed 30 Sep 2014
8. Niederreiter, H., Xing, C.: Rational Points on Curves over Finite Fields: Theory and Applications. London Mathematical Society Lecture Note Series, vol. 285. Cambridge University Press, Cambridge (2001)

9. Niederreiter, H., Xing, C.: Algebraic Geometry in Coding Theory and Cryptography. Princeton University Press, Princeton (2009)
10. Özbudak, F., Gülmez Temür, B.: Finite number of fibre products of Kummer covers and curves with many points over finite fields. Des. Codes Cryptogr. **70**(3), 385–404 (2014)
11. Özbudak, F., Gülmez Temür, B., Yayla, O.: An exhaustive computer search for finding new curves with many points among fibre products of two Kummer covers over $\mathbb{F}_5$ and $\mathbb{F}_7$. Turkish J. Math. **37**(6), 908–913 (2013)
12. Özbudak, F., Stichtenoth, H.: Curves with many points and configurations of hyperplanes over finite fields. Finite Fields Appl. **5**(4), 436–449 (1999)
13. Stichtenoth, H.: Algebraic Function Fields and Codes. Graduate Texts in Mathematics, vol. 254, 2nd edn. Springer, Berlin (2009)
14. Tsfasman, M., Vlăduţ, S., Nogin, D.: Algebraic Geometric Codes: Basic Notions. Mathematical Surveys and Monographs, vol. 139. American Mathematical Society, Providence (2007)

Hyperbolic Lattices with Complete Labeling Derived from $\{4g, 4g\}$ Tessellations

Cátia Quilles Queiroz, Cintya Benedito, José Carmelo Interlando, and Reginaldo Palazzo Jr.

Abstract Hyperbolic lattices $\mathcal{O}$ are the basic entities used in the design of signal constellations in the hyperbolic plane. Once the identification of the arithmetic Fuchsian group in a quaternion order is made, the next step is to present the codewords of a code over a graph, or a signal constellation (quotient of an order by a proper ideal). However, in order for the algebraic labeling to be complete, it is necessary that the corresponding order be maximal. An order $\mathcal{M}$ in a quaternion algebra $\mathcal{A}$ is called *maximal* if $\mathcal{M}$ is not contained in any other order in $\mathcal{A}$ (Reiner, Maximal Orders. Academic, London, 1975). The main objective of this work is to describe the maximal orders derived from $\{4g, 4g\}$ tessellations, for which we have hyperbolic lattices with complete labeling.

Keywords Hyperbolic geometry • Fuchsian group • Quaternion algebra • Quaternion order • Tessellation of the hyperbolic plane

1 Introduction

The theory of lattices in the context of designing signal constellations for power and bandwidth efficient digital communication systems was strongly stimulated by the connection with number theory, group theory and coding theory. As a consequence,

C. Quilles Queiroz (✉)
Department of Mathematics, ICEx, Universidade Federal de Alfenas, UNIFAL, Alfenas, Brazil
e-mail: catia.quilles@gmail.com; catia_quilles@hotmail.com

C. Benedito
Department of Telematics, FEEC, Universidade Estadual de Campinas, UNICAMP, Campinas, Brazil
e-mail: cintyawink@gmail.com

J. Carmelo Interlando
Department of Mathematics and Statistics, San Diego State University, San Diego, CA, USA
e-mail: carmelo.interlando@sdsu.edu

R. Palazzo Jr. (✉)
Department of Communications, FEEC, Universidade Estadual de Campinas, UNICAMP, Campinas, Brazil
e-mail: palazzo@decom.fee.unicamp.br; palazzo@dt.fee.unicamp.br

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_34

the theory of lattices proved to be a tool of great importance for the problem of sphere packing and for the problem of constructing codes with larger codeword lengths within the context of the work of Nyquist and Shannon, [5]. However, these lattices belong to surfaces with genus $g = 1$ (or equivalently, surfaces with constant curvature zero). Going one step farther, that is, to surfaces with genus $g > 1$, it is known that due to the large number of isometries in spaces with negative constant curvature, it is always possible to construct a covering (tiling, tessellation) of such spaces, [10] and [11]. These tessellations, denoted by $\{p, q\}$, are characterized by p-sided regular polygons, where each vertex is covered by q such regular polygons. As an example, consider the tessellation $\{6, 3\}$. This tessellation covers the Euclidean plane with regular hexagons, where each vertex has three hexagons as neighbors. Interesting approaches regarding the $\{p, q\}$ tessellations in spaces with constant curvature can be seen in [6] and [4]. In a pioneering work in the context of communication theory [2] considers the self-dual tessellations $\{p, p\}$ in the design of communication systems in the hyperbolic plane. Such tessellations are an important subset of the $\{p, q\}$ tessellations due to the fact that they are characterized as geometrically uniform tessellations, [7].

The arithmetic Fuchsian groups, Γ_{4g}, obtained in [3], for $g = 2, 3$, were extended to $g = 2^n$, 3.2^n, 5.2^n in [16], where g denotes the genus of an oriented compact surface and n a positive integer. As a consequence, the generators of the corresponding groups Γ_{4g} consisting of the side-pairing transformations of the fundamental hyperbolic polygon P_{4g} were determined. A fundamental polygon P_{4g} covering the hyperbolic plane $\mathbb{D}^2$ is associated with a corresponding quaternion order. Thus, knowing the quaternion order, in [13] geometrically uniform codes derived from graphs over the quotient of these orders by proper ideals were constructed in the hyperbolic plane. However, to have a complete labeling of each one of these codes, the quaternion order necessarily has to be maximal. This is the reason of our interest in maximal orders.

The aim of this paper is to provide a procedure to determine the maximal quaternion orders associated with the self-dual $\{4g, 4g\}$ tessellations, for which a complete labeling may be obtained.

This paper is organized as follows. In Sect. 2, some basic concepts which are necessary to define quaternion orders, tessellations and Fuchsian groups are presented. In addition, the generators of the arithmetic Fuchsian groups are considered. In Sect. 3, the identification of the Fuchsian groups Γ_{4g} derived from a quaternion algebra $\mathscr{A}$ over a number field $\mathbb{K}$ are established. In Sect. 4 the main results are presented. Finally, in Sect. 5 some conclusions are drawn.

2 Preliminary Results

In this section some basic and important concepts to the development of this paper such as quaternion algebras, quaternion orders, hyperbolic lattices and arithmetic Fuchsian groups are presented. For a detailed description of these concepts we refer the reader to [8, 9] and [14].

Let $\mathbb{K}$ be a field. A quaternion algebra $\mathscr{A}$ over $\mathbb{K}$ is a $\mathbb{K}$-vector space of dimension 4 with a $\mathbb{K}$-base $\mathfrak{B} = \{1, i, j, k\}$, where $i^2 = a$, $j^2 = b$, $ij = -ji = k$, $a, b \in \mathbb{K} - \{0\}$, and denoted by $\mathscr{A} = (a, b)_{\mathbb{K}}$.

Let $\alpha \in \mathscr{A}$ be given by $\alpha = a_0 + a_1 i + a_2 j + a_3 ij$, where $a_0, a_1, a_2, a_3 \in \mathbb{K}$. The conjugate of α, denoted by $\bar{\alpha}$, is defined by $\bar{\alpha} = a_0 - a_1 i - a_2 j - a_3 ij$. Thus, the reduced norm of $\alpha \in \mathscr{A}$, denoted by $Nrd_{\mathscr{A}}(\alpha)$, or simply $Nrd(\alpha)$ when there is no confusion, is defined by

$$Nrd(\alpha) = \alpha.\bar{\alpha} = a_0^2 - aa_1^2 - ba_2^2 + aba_3^2, \tag{1}$$

and the reduced trace of α by

$$Trd(\alpha) = \alpha + \bar{\alpha} = 2a_0. \tag{2}$$

Let $\mathscr{A} = (a, b)_{\mathbb{K}}$ be a quaternion algebra over a field $\mathbb{K}$ and $\varphi : \mathbb{K} \longrightarrow \mathbb{F}$ a field homomorphism. Define

$$\mathscr{A}^{\varphi} = (\varphi(a), \varphi(b))_{\varphi(\mathbb{K})} \quad \text{and} \quad \mathscr{A}^{\varphi} \otimes \mathbb{F} = (\varphi(a), \varphi(b))_{\mathbb{F}}, \tag{3}$$

where $\mathscr{A}^{\varphi} \otimes \mathbb{F}$ denotes the tensor product of the algebra $\mathscr{A}^{\varphi}$ by the field $\mathbb{F}$, [12]. Each homomorphism φ in the algebra $\mathscr{A}^{\varphi} = (\varphi(a), \varphi(b))_{\varphi(\mathbb{K})}$ is called place of the quaternion algebra $\mathscr{A}$.

Let $\mathbb{K}$ be a totally real algebraic number field of degree n. This means that the n monomorphisms φ_i, $i = 1, \ldots, n$ are all real, that is, $\varphi_i(\mathbb{K}) \subset \mathbb{R}$. Therefore, the n distinct places are defined by $\mathbb{R}$-isomorphisms

$$\rho_1 : \mathscr{A}^{\varphi_1} \otimes \mathbb{R} \longrightarrow M_2(\mathbb{R}) \quad \text{and} \quad \rho_i : \mathscr{A}^{\varphi_i} \otimes \mathbb{R} \longrightarrow \mathscr{H}, \tag{4}$$

where φ_1 is the identity, φ_i an embedding of $\mathbb{K}$ on $\mathbb{R}$, for $i = 1, \ldots, n$, and $\mathscr{H}$ a division subalgebra of $M_2(\mathbb{K}(\sqrt{a}))$. Hence, $\mathscr{A}$ is not ramified at the place φ_1 and ramified at the places φ_i, for $2 \leq i \leq n$.

Let $Nrd_{\mathscr{H}}$ and $Trd_{\mathscr{H}}$ be the reduced norm and the reduced trace in $\mathscr{H}$, respectively. Given $\alpha \in \mathscr{A}$, it is easy to verify that

$$Nrd_{\mathscr{H}}(\alpha) = det(\rho_1(\alpha)) \quad \text{and} \quad Trd_{\mathscr{H}}(\alpha) = tr(\rho_1(\alpha)). \tag{5}$$

Given $\mathscr{A}$, a quaternion algebra over $\mathbb{K}$, and R a ring of $\mathbb{K}$, an R-order $\mathscr{O}$ in $\mathscr{A}$ is a subring with unity of $\mathscr{A}$ which is a finitely generated R-module such that $\mathscr{A} = \mathbb{K}\mathscr{O}$. Hence, if $\mathscr{A} = (a, b)_{\mathbb{K}}$ and $I_{\mathbb{K}}$, the integer ring of $\mathbb{K}$, where $a, b \in I_{\mathbb{K}} - \{0\}$, then $\mathscr{O} = \{a_0 + a_1 i + a_2 j + a_3 ij : a_0, a_1, a_2, a_3 \in I_{\mathbb{K}}\}$, is an order in $\mathscr{A}$ denoted by $\mathscr{O} = (a, b)_{I_{\mathbb{K}}}$.

Let $\mathscr{A} = (a, b)_{\mathbb{K}}$ be a quaternion algebra over $\mathbb{K}$, R a ring of $\mathbb{K}$, and $\mathscr{O}$ an R-order in $\mathscr{A}$. We also call $\mathscr{O}$ a hyperbolic lattice due to its identification with an arithmetic Fuchsian group.

The lattices $\mathscr{O}$ are used as the basic entity in generating the signals of a signal constellation in the hyperbolic plane. Since $\mathscr{O}$ is an order in $\mathscr{A}$, then there exists a basis $\{e_1, e_2, e_3, e_4\}$ of $\mathscr{A}$ and an R-ideal $\mathfrak{a}$ such that, $\mathscr{O} = \mathfrak{a}e_1 \oplus Re_2 \oplus Re_3 \oplus Re_4$, where $\oplus$ denotes direct sum. Note that by definition, given $x, y \in \mathscr{O}$, we have $x \cdot y \in \mathscr{O}$. Furthermore, since every $x \in \mathscr{O}$ is integral over R, [14], it follows that $Nrd(x), Trd(x) \in R$, [8], as we can see in the next result.

Proposition 1 ([8]) *Given a R-quaternion order $\mathscr{O}$ in a quaternion algebra $\mathscr{A}$. If $x \in \mathscr{O}$, then $Trd(x), Nrd(x) \in R$.*

One of the main objectives of this paper is to identify the maximal orders. An order $\mathscr{M}$ in a quaternion algebra $\mathscr{A}$ is called *maximal* if $\mathscr{M}$ is not contained in any other order in $\mathscr{A}$, [14].

The arithmetic Fuchsian groups were considered in the construction of geometrically uniform signal constellations in the hyperbolic plane, where the arithmetic and geometric concepts, inherent to these groups, are merged in this process.

A Fuchsian group Γ is a discrete subgroup of $PSL(2, \mathbb{R})$, that is, Γ consists of the orientation preserving isometries $T : \mathbb{H}^2 \to \mathbb{H}^2$, acting on $\mathbb{H}^2$ by homeomorphisms. Analogously, the discrete group Γ_p consisting of the orientation preserving isometries $T : \mathbb{D}^2 \to \mathbb{D}^2$ is also a Fuchsian group, given by the transformations $T_p \in \Gamma_p < PSL(2, \mathbb{C})$ such that

$$T_p(z) = \frac{az + c}{\bar{c}z + \bar{a}},\ a, b \in \mathbb{C},\ |a|^2 - |c|^2 = 1.$$

For each order $\mathscr{O}$ in $\mathscr{A}$, consider $\mathscr{O}^1$ as the set $\mathscr{O}^1 = \{\alpha \in \mathscr{O} : Nrd_{\mathscr{H}}(\alpha) = 1\}$. Note that $\mathscr{O}^1$ is a multiplicative group. Now, the Fuchsian groups may be obtained by the isomorphism ρ_1 in (4). In fact, from (5) we have $Nrd_{\mathscr{H}}(\alpha) = det(\rho_1(\alpha))$. Furthermore, we know that $\mathscr{O}^1$ is a multiplicative group, and so $\rho_1(\mathscr{O}^1)$ is a subgroup of $SL(2, \mathbb{R})$, that is, $\rho_1(\mathscr{O}^1) < SL(2, \mathbb{R})$. Therefore, the group derived from a quaternion algebra $\mathscr{A} = (a, b)_{\mathbb{K}}$ and whose order is $\mathscr{O}$, denoted by $\Gamma(\mathscr{A}, \mathscr{O})$, is given by

$$\Gamma(\mathscr{A}, \mathscr{O}) = \frac{\rho_1(\mathscr{O}^1)}{\{\pm Id_2\}} < \frac{SL(2, \mathbb{R})}{\{\pm Id_2\}} \cong PSL(2, \mathbb{R}). \tag{6}$$

As a consequence,

Theorem 2 ([15]) *$\Gamma(\mathscr{A}, \mathscr{O})$ is a Fuchsian group.*

These previous concepts and results lead to the concept of arithmetic Fuchsian groups. Since every Fuchsian group may be obtained in this way, we say that a Fuchsian group is derived from a quaternion algebra if there exists a quaternion algebra $\mathscr{A}$ and an order $\mathscr{O} \subset \mathscr{A}$ such that Γ has finite index in $\Gamma(\mathscr{A}, \mathscr{O})$. The group Γ is called an arithmetic Fuchsian group.

Theorem 3 establishes the necessary and sufficient conditions for arithmecity of Fuchsian groups and its characterization makes use of the set consisting of the traces of its elements, that is, $Tr(\Gamma) = \{\pm Tr(T) : T \in \Gamma\}$.

Theorem 3 ([9, 15]) *Let Γ be a Fuchsian group where the fundamental region has finite area, that is, $\mu(\mathbb{H}^2/\Gamma) < \infty$. Then Γ is derived from a quaternion algebra $\mathscr{A}$ over a totally real number field $\mathbb{K}$, if and only if, the following conditions are satisfied by Γ:*

1. *If $\mathbb{K}_1 = \mathbb{Q}(Tr(T) : T \in \Gamma)$, then $\mathbb{K}_1$ is an algebraic number field of finite degree, and $Tr(\Gamma)$ is contained in $I_{\mathbb{K}_1}$, the ring of integers of $\mathbb{K}_1$;*
2. *If φ is an embedding of $\mathbb{K}_1$ in $\mathbb{C}$ such that $\varphi \neq Id$, then $\varphi(Tr(\Gamma))$ is bounded in $\mathbb{C}$.*

3 Identification of the Fuchsian Groups Γ_{4g} from the Quaternion Orders

In this section we identify the arithmetic Fuchsian groups Γ_{4g} derived from a quaternion algebra $\mathscr{A}$ over a number field $\mathbb{K}$, where $g = 2.2^n, 3.2^n, 5.2^n$ and $3.5.2^n$, for $n \geq 0$, denotes the genus of the surface $\mathbb{D}^2/\Gamma_{4g}$ in a quaternion order.

Theorem 4 ([1, 3, 16]) *For each value of g and $n \geq 0$, the arithmetic Fuchsian group Γ_{4g} is derived from a quaternion algebra $\mathscr{A}$ over a totally real number field $\mathbb{K} = \mathbb{Q}(\theta)$ and the elements of Γ_{4g} are identified, by an isomorphism, with the elements of the order $\mathscr{O} = (\theta, -1)_{I_{\mathbb{K}}}$, where $I_{\mathbb{K}}$ denotes the integer ring of $\mathbb{K}$ and θ is given by:*

1. $\theta = \sqrt{2+\sqrt{2+\ldots+\sqrt{2+\sqrt{2}}}}$, *containing $n+1$ roots, for $g = 2.2^n$;*
2. $\theta = \sqrt{2+\sqrt{2+\ldots+\sqrt{2+\sqrt{3}}}}$, *containing $n+1$ roots, for $g = 3.2^{n+1}$;*
3. $\theta = \sqrt{2+\sqrt{2+\ldots+\sqrt{2+\frac{\sqrt{10+2\sqrt{5}}}{2}}}}$, *containing $n+2$ roots, for $g = 5.2^{n+1}$;*
4. $\theta = \sqrt{2+\sqrt{2+\ldots+\sqrt{7+\sqrt{5}+\frac{\sqrt{30+6\sqrt{5}}}{2}}}}$, *containing $n+4$ roots, for $g = 3.5.2^n$.*

We are going to verify that the Fuchsian groups associated with the orders established in Theorem 4 are in fact arithmetic. For that, it suffices to show that the quaternion algebra is not ramified at φ_1 and it is ramified at the remaining places.

Given the Fuchsian group Γ_{4g} and the quaternion algebra $\mathscr{A} = (\theta, -1)_{\mathbb{K}}$, the elements of $T \in \Gamma$ are given by:

$$T = \frac{1}{2^s}\begin{pmatrix} x_l + y_l\sqrt{\theta} & z_l + w_l\sqrt{\theta} \\ -z_l + w_l\sqrt{\theta} & x_l - y_l\sqrt{\theta} \end{pmatrix},$$

where $s \in \mathbb{N}$, $x_l, y_l, z_l, w_l \in \mathbb{Z}[\theta]$ and θ is given by Theorem 4. Since φ_1 is the identity, it follows that $\mathscr{A} \simeq M_2(\mathbb{K})$ is not ramified at φ_1. Now, observe that θ is square-free for $\mathbb{K} = \mathbb{Q}(\theta)$, that is, there is no $t \in \mathbb{K} - \{0\}$ such that $t^2 = \theta$. Therefore, $\mathscr{A}$ is ramified at all places φ_i, except at φ_1.

Now, the order $\mathscr{O} = (\theta, -1)_{I_{\mathbb{K}}}$ is not a maximal order in the quaternion algebra $\mathscr{A} = (\theta, -1)_{\mathbb{K}}$. Since we are interested in realizing a complete algebraic labeling, we have to find an order that contains the order $\mathscr{O}$ in $\mathscr{A}$ and that it is maximal.

4 Maximal Quaternion Orders Derived from $\{4g, 4g\}$ Tessellations

In this section the characterization of maximal quaternion orders is considered. The interest of this structure derives from the fact that with these orders we have hyperbolic lattices with complete labeling.

Let the Fuchsian group Γ_{4g}, where $g = 2.2^n$, 3.2^{n+1}, 5.2^{n+1} or $3.5.2^n$, for $n \geq 0$, by Theorem 4 we have that Γ_{4g} is derived from a quaternion algebra $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$. Given p_m the minimal polynomial of degree m of θ, where $p_m(x) = x^m + a_{m-1}x^{m-1} + \ldots + a_1 x + a_0$, it follows that the discriminant of the maximal order $\mathscr{M}$ in $\mathscr{A}$ depends on the a_0 coefficient of the minimal polynomial. Thus, the basis of the maximal order in $\mathscr{A}$ that contains $\mathscr{O} = (\theta, -1)_{\mathbb{Z}[\theta]}$ is obtained, as shown in the next results. We observed that in the cases $g = 3$ and 5, the basis has not the same form of its generalizations $g = 3.2^n$ or 5.2^n, respectively, and we do not deal with these cases in this work. Due to space limitation the proofs will be omitted.

Theorem 5 *Given $n = 0$ and $g = 2$, the Fuchsian group Γ_8 is derived from the quaternion algebra $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$, where $\theta = \sqrt{2}$. The minimal polynomial of θ has degree 2 and a basis $\mathscr{B}$ of the maximal order $\mathscr{M}$ in $\mathscr{A}$ is given by:*

$$\left\{1, i, \frac{1}{2}\left((\sqrt{2}+1) + \sqrt{2}i + j\right), \frac{1}{2}\left((\sqrt{2}+1)i + k\right)\right\}.$$

Theorem 6 *Given $n > 0$ and $g = 2.2^n$, the Fuchsian group Γ_{4g} is derived from the quaternion algebra $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$, where $\theta = \sqrt{2+\sqrt{2+\ldots+\sqrt{2+\sqrt{2}}}}$ contains $n+1$ roots, and the minimal polynomial of θ has degree $m = 2^{n+1}$. If $\mathscr{M} \supseteq (\theta, -1)_{\mathbb{Z}[\theta]}$ with a basis $\mathscr{B}$ given by*

$$\left\{1, i, \frac{1}{2}\left((\theta^{m-1} + \theta^{m-2} + \ldots + \theta^{\frac{m}{2}} + 1 - \theta^{m-3}) + \theta^{m-1}i + j\right),\right.$$
$$\left.\frac{-1}{2\theta}\left(2 + (\theta^{m-1} + \theta^{m-2} + \ldots + \theta^{\frac{m}{2}} - 1 - \theta^{m-3})i + k\right)\right\}$$

is such that $Nrd(\alpha) \in \mathbb{Z}[\theta]$ *for all* $\alpha \in \mathscr{B}$*, then* $\mathscr{M}$ *is a maximal quaternion order in* $\mathscr{A}$*.*

Theorem 7 *Given* $n \geq 0$ *and* $g = 3.2^{n+1}$*, the Fuchsian group* Γ_{4g} *is derived from the quaternion algebra* $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$*, where* $\theta = \sqrt{2+\sqrt{2+\ldots+\sqrt{2+\sqrt{3}}}}$ *contains* $n+1$ *roots and the minimal polynomial has order* $m = 2^{n+1}$*. If* $\mathscr{M} \supseteq \mathscr{O} = (\theta, -1)_{\mathbb{Z}[\theta]}$ *with a basis* $\mathscr{B}$ *given by*

$$\left\{1, \frac{-1}{\theta}i, \frac{1}{2}\left((\theta^{m-1} + \theta^{m-2} + \ldots + \theta + 1 - \theta^{\frac{m}{2}}) + (\theta^{m-1} + \theta^{m-2} + \ldots + \theta + 1)i + j\right),\right.$$
$$\left.\frac{1}{2}\left((\theta^{m-1} + \theta^{m-2} + \ldots + \theta + 1) + (\theta^{m-1} + \theta^{m-2} + \ldots + \theta + 1 - \theta^{\frac{m}{2}})i + k\right)\right\}$$

is such that $Nrd(\alpha) \in \mathbb{Z}[\theta]$ *for all* $\alpha \in \mathscr{B}$*, then* $\mathscr{M}$ *is a maximal quaternion order in* $\mathscr{A}$*.*

Theorem 8 *Given* $n \geq 0$ *and* $g = 5.2^{n+1}$*, the Fuchsian group* Γ_{4g} *is derived from the quaternion algebra* $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$*, where* $\theta = \sqrt{2+\sqrt{2+\ldots+\sqrt{2+\frac{\sqrt{10+2\sqrt{5}}}{2}}}}$ *contains* $n+2$ *roots and the minimal polynomial has order* $m = 2^{n+2}$*. If* $\mathscr{M} \supseteq (\theta, -1)_{\mathbb{Z}[\theta]}$ *with a basis* $\mathscr{B}$ *given by*

$$\left\{1, \frac{-1}{\theta}i, \frac{1}{2}(\theta^{3.2^n} + j), \frac{1}{2}(\theta^{3.2^n} i + k)\right\}$$

is such that $Nrd(\alpha) \in \mathbb{Z}[\theta]$ *for all* $\alpha \in \mathscr{B}$*, then* $\mathscr{M}$ *is a maximal quaternion order in* $\mathscr{A}$*.*

Example 9 Let $g = 10 = 5 \cdot 2$. By Theorem 4, the quaternion order associated is $\mathscr{O} = (\theta, -1)_{\mathbb{Z}[\theta]}$, where $\theta = \sqrt{2 + \frac{\sqrt{10+2\sqrt{5}}}{2}}$. The minimal polynomial is

$$p_8(x) = x^8 - 8x^6 + 19x^4 - 12x^2 + 1,$$

then $a_0 = 1$. Thus, by Theorem 8, the basis of the maximal order $\mathscr{M}$ in the quaternion algebra $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$ is given by

$$\left\{1, -\frac{i}{\theta}, \frac{(\theta^6 + j)}{2}, \frac{(\theta^6 i + k)}{2}\right\}.$$

Theorem 10 *Given* $n \geq 0$ *and* $g = 3.5.2^n$*, the Fuchsian group* Γ_{4g} *is derived from the quaternion algebra* $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$*, where* $\theta = \sqrt{2+\sqrt{2+\ldots+\frac{\sqrt{7+\sqrt{5}+\sqrt{30+6\sqrt{5}}}}{2}}}$

contains $n + 4$ roots, $n \geq 0$ and the minimal polynomial has degree $m = 2^{n+3}$. If $\mathscr{M} \supseteq (\theta, -1)_{\mathbb{Z}[\theta]}$ with a basis $\mathscr{B}$ given by

$$\left\{1, \frac{-1}{\theta}i, \frac{1}{2}\left((\theta^{5.2^n} + \theta^{3.2^n} + \theta^{2^n}) + j\right), \frac{1}{2}\left((\theta^{5.2^n} + \theta^{3.2^n} + \theta^{2^n})i + k\right)\right\}$$

is such that $Nrd(\alpha) \in \mathbb{Z}[\theta]$ for all $\alpha \in \mathscr{B}$, then $\mathscr{M}$ is a maximal quaternion order in $\mathscr{A}$.

Example 11 Let $g = 15 = 3.5.2^0$, by Theorem 4 the quaternion order associated is $\mathscr{O} = (\theta, -1)_{\mathbb{Z}[\theta]}$, where $\theta = \frac{\sqrt{7+\sqrt{5}+\sqrt{30+6\sqrt{5}}}}{2}$. The minimal polynomial is $p_8(x) = x^8 - 7x^6 + 14x^4 - 8x^2 + 1$, then the free coefficient is $a_0 = 1$. Thus, by Theorem 10, the basis of the maximal order $\mathscr{M}$ in the quaternion algebra $\mathscr{A} = (\theta, -1)_{\mathbb{Q}(\theta)}$ is given by

$$\left\{1, \frac{-1}{\theta}i, \frac{1}{2}\left((\theta^5 + \theta^3 + \theta) + j\right), \frac{1}{2}\left((\theta^5 + \theta^3 + \theta)i + k\right)\right\}.$$

5 Conclusions

In this paper we have identified maximal quaternion orders derived from $\{4g, 4g\}$ tessellations for which hyperbolic lattices with complete labeling were obtained.

Acknowledgements C. Quilles Queiroz, C. Benedito and R. Palazzo Jr. are supported by FAPESP under grant 2008/04992-0, CNPq under grant 306617/2007-2, 303059/2010-9 and FAPEMIG under grant PEE-01223-14.

References

1. Benedito, C.W.O., Palazzo, R., Jr.: A necessary condition for obtaining arithmetic Fuchsian groups derived from quaternion orders. In: Program and Abstracts from XXII Brazilian Algebra Meeting, Salvador, p. 71. UFBA. Printed in Bulgaria: Bulgarian Academy of Sciences (2012)
2. Brandani, E.: Signal constellations and performance analyses in the hyperbolic plane. Doctoral Dissertation, FEEC-UNICAMP (in Portuguese) (2000)
3. Carvalho, E.D.: Construction and labeling of geometrically uniform signal constellations in Euclidean and hyperbolic spaces. Doctoral Dissertation, FEEC-UNICAMP (in Portuguese) (2001)
4. Cavalcante, R.G.: Performance analyses of signal constellations in Riemannian manifolds. Master Dissertation, FEEC-UNICAMP, Campinas, Brasil (in Portuguese) (2002)
5. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups, 2nd edn. Springer, New York (1991)

6. Faria, M.B.: Sphere packing in hiperbolic spaces. Master Dissertation, IMECC-UNICAMP, Campinas, Brasil (in Portuguese) (2001)
7. Forney, G.D.: Geometrically uniform codes. IEEE Trans. Inf. Theory **IT-37**, 1241–1260 (1991)
8. Johansson, S.: A description of quaternion algebra. www.math.chalmers.se/~sj/forskning.html (1997)
9. Katok, S.: Fuchsian Groups. The University of Chicago Press, Chicago (1992)
10. Magnus, W.: Non Euclidean Tesselations and Their Groups. Academic Press, New York (1970)
11. Massey, W.S.: Algebraic Topology: An Introduction. Springer-Verlag, New York (1967)
12. O'Meara, O.T.: Introduction to Quadratic Forms. Springer, Berlin/New York (1973)
13. Quilles Queiroz, C.R.O., Palazzo, R., Jr.: Codes over graphs derived from quotient rings of the quaternion orders. Int. Sch. Res. Netw. – ISRN Algebra **2012** 1–14 (2012). Article ID 956017. doi:10.5402/2012/956017
14. Reiner, I.: Maximal Orders. Academic, London (1975)
15. Takeuchi, K.: A characterization of arithmetic Fuchsian groups. J. Math. Soc. Jpn. **27**(4), 600–612 (1975)
16. Vieira, V.L.: Arithmetic Fuchsian groups identified over the quaternion orders for the construction of signal constellations. Doctoral Dissertation, FEEC-UNICAMP (in Portuguese) (2007)

On Quasi-symmetric 2-$(64, 24, 46)$ Designs Derived from Codes

Bernardo G. Rodrigues and Vladimir D. Tonchev

Abstract The paper studies quasi-symmetric 2-$(64, 24, 46)$ designs supported by minimum weight codewords in the dual code of the binary code spanned by the lines of $AG(3, 2^2)$. We classify up to isomorphism all designs invariant under automorphisms of odd prime order in the full automorphism group G of the code, being of order $|G| = 2^{13} \cdot 3^4 \cdot 5 \cdot 7$. We show that there is exactly one isomorphism class of designs invariant under an automorphisms of order 7, 15 isomorphism classes of designs with an automorphism of order 5, and no designs with an automorphism of order 3. Any design in the code that does not admit an automorphism of odd prime order has full group of order 2^m for some $m \leq 13$, and there is exactly one isomorphism class of designs with full automorphism group of order 2^{13}.

Keywords Quasi-symmetric designs • Automorphism groups • Linear codes

1 Introduction

We assume familiarity with the basic concepts and notation from combinatorial design theory and coding theory [1, 2, 6, 11]. For properties of quasi-symmetric designs we refer the reader to [10].

This paper summarizes computational results concerning quasi-symmetric designs with parameters 2-$(64, 24, 46)$ and block intersection numbers 8 and 12, whose blocks are supports of codewords of weight 24 in the dual code of the binary code spanned by the incidence vectors of the lines in the 3-dimensional affine geometry $AG(3, 2^2)$. Our main computational tools were Magma [4] and

B.G. Rodrigues (✉)
School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, 4041 Durban, South Africa
e-mail: rodrigues@ukzn.ac.za

V.D. Tonchev
Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931, USA
e-mail: tonchev@mtu.edu

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_35

Cliquer [8]. We note that a somewhat similar approach for finding quasi-symmetric designs in codes was used in [7]. The main difference is that here we search for designs in a well known affine geometry code, while the code used in [7] was spanned by the incidence matrix of a previously known quasi-symmetric design. Our study is motivated by a previously known design with parameters 2-$(64, 24, 46)$, being a member of an infinite family of quasi-symmetric designs discovered by Blokhuis and Haemers [3].

2 Designs in the Dual Code of the Code of Lines of $AG(3, 2^t)$

In [3], Blokhuis and Haemers, inspired by a method for constructing balanced incomplete block designs based on resolvable designs due to Shrikhande and Raghavarao [9] gave an elegant construction of a quasi-symmetric design $D = D(q)$ with parameters 2-$(q^3, q^2(q-1)/2, q(q^3-q^2-2)/4)$ and block intersection numbers $q^2(q-2)/4$ and $q^2(q-1)/4$, where q is a power of 2.

Any block U of $D(q)$ can be viewed as the union of $q(q-1)/2$ parallel lines in the 3-dimensional affine geometry $AG(3, q)$ belonging to a parallel class P, and the q^2 blocks of $D(q)$ associated with P correspond to a symmetric 2-$(q^2, q(q-1)/2, q(q-2)/4)$ design D' with points labeled by the lines of P, and blocks being maximal arcs in $AG(2, q)$.

A crucial property of the blocks of $D(q)$ is that for any $q > 2$ every block U meets every line L of $AG(3, q)$ in an even number of points [3]. This implies the following.

Lemma 1 *If $q > 2$, then every block U of $D(q)$ is the support of a codeword of weight $q^2(q-1)/2$ in the dual code $C^\perp$ of the binary code C of length q^3 spanned by the incidence vectors of the lines in $AG(3, q)$.*

The dimension of the code C (hence of $C^\perp$) can be computed easily using Hamada's formula [5] for the 2-rank of the incidence matrix of the lines of $AG(3, 2^t)$.

In this paper, we study quasi-symmetric designs arising in this manner from the code related to $AG(3, 2^2)$, namely, quasi-symmetric 2-$(64, 24, 46)$ designs with intersection numbers 8 and 12, supported by codewords of weight 24 in the dual code $C^\perp$ of the binary code C spanned by the block by point incidence matrix of the 2-$(64, 4, 1)$ design D, having as blocks the lines of $AG(3, 2^2)$. By Hamada's rank formula [5], the dimension of C is 51, hence $C^\perp$ is a binary $[64, 13]$ code. The weight distribution $\{A_i\}_{i=0}^{64}$ of $C^\perp$ is given in Table 1.

We define a graph Γ having as vertices the 1008 codewords of $C^\perp$ of weight 24, where two codewords are adjacent in Γ whenever they share exactly 8 or 12 nonzero positions. Any quasi-symmetric 2-$(64, 24, 46)$ design with intersection numbers 8 or 12, whose 336 blocks are supports of codewords of $C^\perp$, corresponds to a clique of Γ of size 336. A search for 336-cliques in Γ performed on a personal computer using the Cliquer program [8] indicates that Γ contains a huge number of cliques

Table 1 Weight distribution of $C^{\perp}$

Weight i	A_i
0	1
24	1008
32	6174
40	1008
64	1

of this size. After running the Cliquer non-stop for several days, millions of 336-cliques were found with no indication that the search is coming to an end. Because of that, we restricted our search to finding designs in $C^{\perp}$ which are invariant under various subgroups of the automorphism group of $C^{\perp}$. The results of this search are described in the following sections.

3 Designs with an Automorphism of Order 3

The automorphism group G of the code C, which is also the automorphism group of the dual code $C^{\perp}$, coincides with the group of the affine space $AG(3, 2^2)$, being of order

$$|G| = 23224320 = 2 \times 4^3(4^3 - 1)(4^3 - 4)(4^4 - 4^2) = 2^{13} \cdot 3^4 \cdot 5 \cdot 7. \qquad (1)$$

The group G contains five conjugacy classes of subgroups of order 3: two classes, 3_1^a and 3_1^b, fixing one point and having 21 cycles of length 3 on the set of 64 code coordinates; two classes, 3_4^a and 3_4^b fixing 4 points; and one class 3_{16} fixing 16 points. We used Magma [4] to find representatives of these conjugacy classes.

A subgroup H_1^a from the class 3_1^a partitions the 1008 vertices of Γ into 336 orbits of length 3 (that is, H_1^a does not fix any codeword of $C^{\perp}$ of weight 24). A total of 210 out of these 336 orbits are 3-cliques. We call these 210 orbits "good" orbits, because any 336-clique of Γ which is stabilized by H_1^a must be a union of 112 such (good) orbits. We define a new graph Γ_{210} having as vertices the 210 good orbits, where two vertices are adjacent in Γ_{210} if the corresponding orbits are *compatible*, that is, the six vertices of Γ belonging to these two orbits form a 6-clique in Γ. A quick search with Cliquer establishes that Γ_{210} contains no 112-cliques (the maximum clique size of Γ_{210} turned out to be 84).

A subgroup H_1^b from the class 3_1^b partitions the 1008 vertices of Γ into 346 orbits: 331 of length 3, and 15 fixed vertices. Out of the 331 orbits of length 3, 325 are good. The maximum clique size in the graph Γ_{325} of the 325 good orbits of length 3 is 109, and any such 109-clique gives a 327-clique in Γ. The 15 vertices of Γ which are fixed by H_1^b form a 15-clique in Γ. Thus, if Γ contains a 336-clique Q invariant under H_1^b, Q must consist of 107 3-orbits plus 15 fixed vertices, or 108 3-orbits plus 12 fixed vertices, or 109 3-orbits plus 9 fixed vertices.

We define now a graph Γ_{340} on 340 vertices, being the 325 good orbits of length 3 plus the 15 vertices of Γ fixed by H_1^b.

A 336-clique in Γ invariant under H_1^b consisting of 109 3-orbits plus 9 fixed vertices corresponds to a 118-clique in Γ_{340}. Similarly, a 336-clique in Γ invariant under H_1^b consisting of 108 3-orbits plus 12 fixed vertices corresponds to a 120-clique in Γ_{340}, and a 336-clique in Γ invariant under H_1^b consisting of 107 3-orbits plus 15 fixed vertices corresponds to a 122-clique in Γ_{340}. A Cliquer search shows that Γ_{340} does not contain any cliques of size 118.

Consequently, the code $C^\perp$ does not contain any quasi-symmetric designs with an automorphism of order 3 fixing exactly one point.

A subgroup H_4^a from the class 3_4^a partitions the 1008 vertices of Γ into 336 orbits of length 3, 330 out of these 336 orbits being "good", that is, 3-cliques of Γ. The graph having as vertices these 330 good orbits does not contain any 112-clique, hence $C^\perp$ does not support any quasi-symmetric design invariant under a subgroup of order 3 from class 3_4^a.

A subgroup H_4^b from the class 3_4^b partitions the 1008 vertices of Γ into 338 orbits: 335 orbits of length 3, and 3 fixed vertices. All 335 orbits of length 3 turned out to be good. Clearly, a 336-clique of Γ which is preserved by H_4^b has to consist of either 111 orbits of length 3 plus three fixed vertices, or 112 orbits of length 3. However, the graph Γ_{335} of the 335 orbits of length 3 does not contain any cliques of size 111. Consequently, $C^\perp$ does not contain any quasi-symmetric design invariant under H_4^b.

Finally, a subgroup H_{16} of order 3 from the class 3_{16}, fixing 16 of the 64 code positions, partitions the vertices of Γ into 368 orbits: 320 orbits of length 3, and 48 fixed vertices. All 320 orbits of length 3 are good. The maximum clique size in the graph defined on the 320 orbits of length 3 is 100, while the maximum size of a clique in the subgraph of Γ on the 48 vertices fixed by H_{16} is 16. Therefore, $C^\perp$ does not contain any quasi-symmetric design invariant under H_{16}.

The following statement summarizes our findings concerning automorphisms of order 3.

Theorem 2 *The code $C^\perp$ does not contain any quasi-symmetric* 2-$(64, 24, 46)$ *design that admits an automorphism of order 3.*

We note that the group G is transitive on the set of all 1008 codewords of $C^\perp$ of weight 24, hence the code $C^\perp$ does not contain any quasi-symmetric 2-$(64, 24, 46)$ design invariant under G. The group G contains seven conjugacy classes of maximal subgroups. The orders of these maximal subgroups are: $2^7 \cdot 3^3 \cdot 7$, $2^7 \cdot 3^4 \cdot 5 \cdot 7$, $2^{10} \cdot 3^4$, $2^{13} \cdot 3^3 \cdot 5$, $2^{13} \cdot 3^3 \cdot 5$, $2^{13} \cdot 3^3 \cdot 5 \cdot 7$, and $2^{12} \cdot 3^4 \cdot 5 \cdot 7$. Since all these orders are divisible by 3, we have the following.

Corollary 3 *The code $C^\perp$ does not contain any quasi-symmetric* 2-$(64, 24, 46)$ *design invariant under a maximal subgroup of G.*

4 Designs with an Automorphism of Order 5

It is clear from (1) that the group G has only one conjugacy class of subgroups of order 5. A subgroup H_5 of order 5 partitions the vertices of Γ into 204 orbits: there are 201 orbits of length 5, and 3 fixed vertices. All 201 orbits of length 5 are good. We define a graph Γ_{204} having as vertices the 204 orbits. Any quasi-symmetric design invariant under H_5 corresponds to a 68-clique of Γ_{204} (67 orbits of length 5 plus one fixed vertex of Γ). Cliquer returns 243 68-cliques in Γ_{204}, each such clique yielding a quasi-symmetric 2-$(64, 24, 46)$ design with an automorphism of order 5.

Using Magma, we computed the normalizer $N(H_5)$ of H_5 in G. The order of $N(H_5)$ is $720 = 2^4 \cdot 3^2 \cdot 5$. The set of 243 336-cliques is partitioned into 15 orbits under the action of $N(H_5)$: there are three orbits of length 9, and 12 orbits of length 18. This implies that there are at most 15 nonisomorphic quasi-symmetric designs with an automorphism of order 5 supported by codewords of weight 24 in $C^\perp$.

Further computations with Magma show that there are exactly 15 nonisomorphic classes of designs invariant under an automorphism of order 5. Some data concerning these designs is given in Table 2.

We note that the block graphs of designs no. 5 and 9 are isomorphic, and so are those of designs 4 and 15 respectively.

The following statement summarizes our findings concerning automorphisms of order 5.

Theorem 4 *The code $C^\perp$ contains exactly fifteen isomorphism classes of quasi-symmetric* 2-$(64, 24, 46)$ *designs admitting an automorphism of order 5. The full automorphism groups of these designs have orders* 20480 *(two designs),* 1280 *(one design), and* 640 *(12 designs).*

Table 2 Non-isomorphic designs invariant under an automorphism of order 5

D_i	$\lvert\mathrm{Aut}(D_i)\rvert$	2-rank (D_i)	# of cliques of size 5 in the block graph.
1	20480	12	78592
2	20480	13	116480
3	1280	13	97280
4	640	13	91136
5	640	13	92032
6	640	13	90432
7	640	13	83456
8	640	13	86400
9	640	13	92032
10	640	13	76672
11	640	13	89600
12	640	13	98496
13	640	13	84416
14	640	13	93376
15	640	13	91136

5 Designs with an Automorphism of Order 7

The group G contains one conjugacy class of subgroups of order 7. A subgroup H_7 of order 7 partitions the 1008 vertices of Γ into 144 orbits of length 7. Cliquer finds exactly 27 48-cliques in the graph defined on the 144 orbits, each such clique giving a quasi-symmetric design.

The normalizer $N(H_7)$ of H_7 in G is a group of order $378 = 2 \cdot 3^3 \cdot 7$. It follows from Theorem 2 that the subgroup of order 27 of $N(H_7)$ is transitive on the set of 27 48-cliques. Consequently, the 27 quasi-symmetric designs corresponding to the 27 cliques are isomorphic, and we have the following result.

Theorem 5 *The code $C^{\perp}$ contains exactly one isomorphism class of quasi-symmetric* 2-$(64, 24, 46)$ *designs admitting an automorphism of order 7. The full automorphism group of a design from this isomorphism class is of order* $896 = 2^7 \cdot 7$.

6 Designs Invariant Under 2-Subgroups of G

According to the results of the preceding sections, any quasi-symmetric 2-$(64, 24, 46)$ design supported by the dual code of $AG(3, 2^2)$ which does not have an automorphism of odd prime order must have a full automorphism group of order 2^m for some $m \leq 13$.

A computation with Magma shows that a Sylow 2-subgroup of G (of order 2^{13}, cf. (1)) partitions the set of 1008 codewords of weight 24 into six orbits of lengths 16, 32, 64, 128, 256, and 512 respectively. There is only one union of orbits containing exactly 336 codewords: $336 = 16 + 64 + 256$. A quick computation shows that this union is indeed the incidence matrix of a quasi-symmetric 2-$(64, 24, 46)$ design. Thus we have the following.

Theorem 6 *The code $C^{\perp}$ contains exactly one isomorphism class of quasi-symmetric* 2-$(64, 24, 46)$ *designs admitting a Sylow* 2*-subgroup of G as its full automorphism group.*

The 2-rank of a design admitting a Sylow 2-subgroup of G as a full automorphism group is 12, and the weight distribution of the binary $[64, 12]$ code C' spanned by its blocks is

$$a_0 = 1,\ a_{24} = 496,\ a_{32} = 3102,\ a_{40} = 496,\ a_{64} = 1. \tag{2}$$

The full automorphism group of the code C' is of order $2^{13} \cdot 3^2 \cdot 5$.

Acknowledgements The first author gratefully acknowledges support by the National Research Foundation of South Africa through Grants # 84470 and #91495.

The second author would like to thank the University of KwaZulu-Natal for the warm hospitality during his visit. The research of this author was partially supported by an NSA grant, and a Fulbright grant #5869. The authors wish to thank the unknown referees for their useful remarks.

References

1. Assmus, E.F., Jr, Key, J.D.: Designs and Their Codes. Cambridge University Press, Cambridge (1992)
2. Beth, T., Jungnickel, D., Lenz, H.: Design Theory, 2nd edn. Cambridge University Press, Cambridge (1999)
3. Blokhuis, A., Haemers, W.H.: An infinite family of quasi-symmetric designs. J. Stat. Plan. Inference **95**(1), 117–119 (2001)
4. Bosma, W., Cannon, J., Playoust., C.: The Magma algebra system I: the user language. J. Symb. Comput. **24**(3,4), 235–265 (1997)
5. Hamada, N.: On the p-rank of the incidence matrix of a balanced or partially balanced incomplete block design and its application to error correcting codes. Hiroshima Math. J. **3**, 153–226 (1973)
6. MacWilliams, F., Sloane, N.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
7. Munemasa, A., Tonchev, V.: A new quasi-symmetric 2-(56, 16, 6) design obtained from codes. Discret. Math. **284**, 231–234 (2004)
8. Niskanen, S., Ostergard, P.R.J.: Cliquer user's guide, version 1.0. Technical Report T48. Communications Laboratory, Helsinki University of Technology, Espoo (2003)
9. Shrikhande, S.S., Raghavarao, D.: A method of construction of incomplete block designs. Sankhya **25**, 399–402 (1963)
10. Shrikhande, M., Sane, S.S.: Quasi-symmetric designs. Cambridge University Press, Cambridge (1996)
11. Tonchev, V.: Combinatorial Configurations. Longman-Wiley, New York (1988)

Fractional Repetition and Erasure Batch Codes

Natalia Silberstein

Abstract Batch codes are a family of codes that represent a distributed storage system (DSS) of n nodes so that any batch of t data symbols can be retrieved by reading at most one symbol from each node. Fractional repetition codes are a family of codes for DSS that enable efficient uncoded repairs of failed nodes. In this work these two families of codes are combined to obtain fractional repetition batch (FRB) codes which provide both uncoded repairs and parallel reads of subsets of stored symbols. In addition, new batch codes which can tolerate node failures are considered. This new family of batch codes is called erasure combinatorial batch codes (ECBCs). Some properties of FRB codes and ECBCs and examples of their constructions based on transversal designs and affine planes are presented.

Keywords Fractional repetition codes • Batch codes • Transversal designs • Affine planes

1 Introduction

In distributed storage systems (DSS) information is stored across a network of nodes in such a way that a user (data collector) can retrieve the stored data even if some system nodes fail. To provide reliability against node failures, data redundancy based on different types of erasure codes is introduced in such systems. Moreover, to provide an efficient repair of a single failed node (the most common case in DSS), a new family of erasure codes for DSS, called *regenerating* codes, was presented in [4]. Two types of regenerating codes, *minimum storage regenerating* (MSR) and *minimum bandwidth regenerating* (MBR) [4] codes, were introduced to optimize the storage overhead and repair bandwidth, respectively (for constructions

This research was supported in part by the Fine Fellowship and by the Israeli Science Foundation (ISF), Jerusalem, Israel, under Grant 10/12.

N. Silberstein (✉)
Department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel
e-mail: natalys@cs.technion.ac.il

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_36

see [4, 5, 12, 13] and references therein). In particular, a regenerating code C is used to store a file on n nodes, where each node stores α symbols from a finite field $\mathbb{F}_q$, such that a data collector can recover the stored file from any set of $k < n$ nodes. A single failed node can be repaired by downloading $\beta \leq \alpha$ symbols from any node in a set of size d, $k \leq d \leq n-1$, of surviving nodes. Note that any random set of d nodes can be used to repair a failed node.

Fractional repetition (FR) codes [6] are a family of codes for DSS which allow for uncoded repairs (no decoding is needed), while relaxing the requirement of random d-set for repairs by making it table based instead. This relaxation allows for increasing the amount of data that can be stored by using FR codes when compared to MBR codes, while having the same repair bandwidth. When an (n, k, α, ρ) FR code C is used to store a file $\mathbf{f} \in \mathbb{F}_q^M$ of size M, $\mathbf{f}$ is first encoded to a codeword $c_\mathbf{f}$ of a (θ, M) maximum distance separable (MDS) code [8], with $\theta = n\alpha/\rho$. Next, θ symbols of the MDS codeword $c_\mathbf{f}$ are placed on n nodes, each of size α, as follows. Let $N_1, \ldots, N_n$ be a collection of subsets of size α of the set $[\theta] := \{1, 2, \ldots, \theta\}$, such that every element in $[\theta]$ appears in exactly ρ subsets. Then node i stores the symbols of $c_\mathbf{f}$ indexed by the subset N_i. An FR code should satisfy the requirement that from any set of k nodes it is possible to reconstruct the stored file, that is, $M = \min_{|I|=k} | \cup_{i \in I} N_i|$. Note that for FR codes it holds that $\alpha = d$ and $\beta = 1$, since when some node i fails, it can be repaired by using α other nodes which store common symbols with node i. Constructions of FR codes based on different types of regular graphs and combinatorial designs can be found in [6, 9, 11, 14].

Batch codes [7] are a family of codes for DSS which store θ (encoded) data symbols on n system nodes in such a way that any batch of t data symbols can be decoded by reading at most one symbol from each node, while keeping the total storage over all n nodes equal to N. A *ρ-uniform combinatorial batch code* (CBC), denoted by $\rho-(\theta, N = \rho\theta, t, n)$, is a batch code where each node stores a subset of data symbols, that is decoding is performed only by reading items from the nodes, and each symbol is stored in exactly ρ nodes [7, 10]. These codes were studied in [2, 3, 7, 10, 15].

In this work, we consider two new families of codes for DSS. The first family, called *fractional repetition batch* (FRB) codes, is based on the combination of FR and combinatorial batch codes and hence has the properties of both FR and batch codes simultaneously: FRB codes allow for uncoded efficient repairs and load balancing in partial data reconstruction which can be performed by several users independently and in parallel. The second family of codes, called *erasure combinatorial batch codes* (ECBCs), allow for recovery of any batch of t data symbols even in presence of nodes failures, by reading at most one symbol from the remaining available nodes. ECBCs generalize the original batch codes [7, 10] which require *all* the nodes in a system to be always available for accessing their stored data. We analyze the properties of incidence matrices of FRB codes and ECBCs and present the necessary and sufficient conditions on the structure of these codes. We provide constructions for FRB codes and ECBCs based on transversal designs and affine planes.

The rest of this paper is organized as follows. In Sect. 2 we define FRB codes, consider properties of their incidence matrices and provide some examples of their constructions. In Sect. 3 we define ECBCs, discuss their properties and describe codes based on affine planes and transversal designs. Conclusions and problems for future research are given in Sect. 4.

2 Fractional Repetition Batch Codes

In this section we consider a new family of codes for DSS, called FRB codes, which combine the properties of both FR and combinatorial batch codes.

Let $\mathbf{f} \in \mathbb{F}_q^M$ be a file of size M and let $c_\mathbf{f} \in \mathbb{F}_q^\theta$ be a codeword of an (θ, M) MDS code which encodes the data $\mathbf{f}$. Let $\{N_1, \ldots, N_n\}$ be a collection of α-subsets of a set $[\theta]$. A $\rho - (n, M, k, \alpha, t)$ *fractional repetition batch* (FRB) code C represents a system of n nodes with the following properties:

1. Every node i, $1 \leq i \leq n$, stores α symbols of $c_\mathbf{f}$ indexed by N_i;
2. Every symbol of $c_\mathbf{f}$ is stored on ρ nodes;
3. From any set of k nodes it is possible to reconstruct the stored file $\mathbf{f}$, in other words, $M = \min_{|I|=k} |\cup_{i \in I} N_i|$;
4. Any batch of t symbols from $c_\mathbf{f}$ can be retrieved by downloading at most one symbol from each node.

Note that the total storage over all n nodes needed to store a file $\mathbf{f}$ equals to $n\alpha = \theta\rho$. The general coding scheme for an FRB code is shown in Fig. 1.

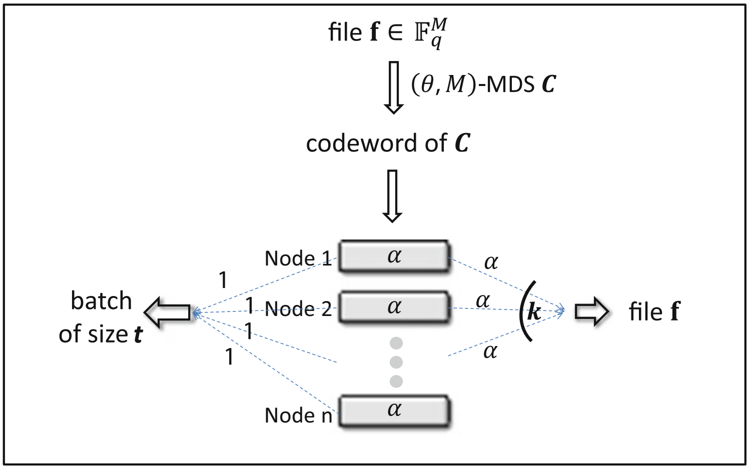

file **f** ∈ $\mathbb{F}_q^M$
(θ, M)-MDS $\boldsymbol{C}$
codeword of $\boldsymbol{C}$
Node 1
Node 2
Node n
α
α
α
1
1
1
1
α
α
α
$\boldsymbol{k}$
batch of size $\boldsymbol{t}$
file **f**

Fig. 1 The coding scheme for an FRB code

Remark 1 Note that while in a classical batch code any t *data* symbols can be retrieved, in a FRB code any batch of t *coded* symbols can be retrieved. In particular, when a systematic MDS code is chosen for an FRB code, the data symbols can be easily retrieved.

Now we consider the matrix representation of FRB codes which follows from the matrix representation of FR and combinatorial batch codes. The incidence matrix of a $\rho - (n, M, k, \alpha, t)$ FRB code C, denoted by $\mathbf{I}(C)$, is a binary $n \times \theta$ matrix with rows and columns indexed by the nodes and symbols of an MDS codeword, respectively, such that $(\mathbf{I}(C))_{i,j} = 1$ if and only if node i contains symbol j of $c_\mathbf{f}$. In other words, the ith row of $\mathbf{I}(C)$ is the incidence vector of the set N_i. Note that the number of ones in each row is α and the number of ones in each column is ρ in this matrix.

In the following, we obtain the necessary and sufficient conditions on a binary matrix to be the incidence matrix of an FRB code. Let A be a binary matrix, and let S and T be some subsets of rows and columns of A, respectively. Let $A_{S,T}$ be a submatrix of A with rows and columns indexed by S and T. We say that a set T of columns *covers* a set S of rows if there is no all-zero row in $A_{S,T}$. Similarly, a set S of rows *covers* a set T of columns if there is no all-zero column in $A_{S,T}$.

The next theorem follows from the properties of incidence matrices for combinatorial batch and FR codes (see [2, 10, 14] for details).

Theorem 2 *An $n \times \theta$ binary matrix A with α* ones *in each row and ρ* ones *in each column is the incidence matrix of a $\rho - (n, M, k, \alpha, t)$ FRB code if and only if the following two conditions hold:*

1. *Any i columns of A, $1 \leq i \leq t$, cover at least i rows;*
2. *Any k rows of A cover at least M columns.*

If we consider the incidence matrix of an FRB code as the biadjacency matrix of a bipartite graph, where the left vertex set L corresponds to the rows (nodes) and the right vertex set R corresponds to the columns (codeword symbols) of the matrix, then the conditions of Theorem 2 can be formulated as follows.

Corollary 3 *A biadjacency matrix of a bipartite graph $G = (L \cup R, E)$, $|L| = n$, $|R| = \theta$, with the left degree α and right degree ρ, is the incidence matrix of a $\rho - (n, M, k, \alpha, t)$ FRB code if and only if the following two conditions hold:*

1. *Any subset $T \subseteq R$ of at most t vertices has at least $|T|$ neighbours in L;*
2. *Any subset $S \subseteq L$ of k vertices has at least M neighbours in R.*

Remark 4 The construction of batch codes based on unbalanced expander graphs was proposed in [7]. To construct an FRB code, we need a bipartite expander with two different expansion factors, 1 and M/k, for two sides R and L of a graph, respectively.

2.1 Constructions of FRB Codes

In this subsection, we consider constructions of FRB codes based on optimal FR codes and optimal uniform CBCs. We say that an FR code is an optimal code if it can store a file of maximum size, i.e. it maximizes $M = M(n, k, \alpha, \rho)$ (see [6, 14] for details). We say that a uniform combinatorial batch code is an optimal code if it stores the maximum number of symbols, i.e., it maximizes $\theta = \theta(n, \rho, t)$ (see [2, 10, 15]).

It was proved recently [15] that combinatorial batch codes based on some transversal designs are (near) optimal CBCs. Moreover, it was shown that FR codes based on transversal designs are optimal FR codes [14]. Therefore, it is natural to consider FRB codes based on transversal designs.

A *transversal design* (TD) of group size h and block size ℓ, denoted by TD(ℓ, h), is a triple $(\mathscr{P}, \mathscr{G}, \mathscr{B})$, where

1. $\mathscr{P}$ is a set of ℓh *points*;
2. $\mathscr{G}$ is a partition of $\mathscr{P}$ into ℓ sets (*groups*), each one of size h;
3. $\mathscr{B}$ is a collection of ℓ-subsets of $\mathscr{P}$ (*blocks*);
4. Each block meets each group in exactly one point;
5. Any pair of points from different groups is contained in exactly one block.

It follows from the definition of TD that the number of blocks in TD(ℓ, h) is h^2 and the number of blocks that contain a given point is h [1]. The incidence matrix $\mathbf{I}_{TD}$ of TD(ℓ, h) is the $\ell h \times h^2$ binary matrix where columns are incidence vectors of the blocks. A TD(ℓ, h) is called *resolvable* if the set $\mathscr{B}$ can be partitioned into sets $\mathscr{B}_1, \dots, \mathscr{B}_h$, each one contains h blocks, such that each element of $\mathscr{P}$ is contained in exactly one block of each $\mathscr{B}_i$. Resolvable TD(ℓ, h) is known to exist for any $\ell \le h$ and prime power h [1].

Next we consider an FRB code C_{TD} such that its incidence matrix is the incidence matrix of TD. Based on the properties of uniform CBCs and FR codes constructed from different TDs [14, 15], we obtain the following result.

Theorem 5 *1. Let TD$(2, \alpha)$ be a TD with $\alpha > 2$. Then C_{TD} is a $2-(2\alpha, M, k, \alpha, 5)$ FRB code with $M = k\alpha - \left\lfloor \frac{k^2}{4} \right\rfloor$.*

2. Let TD$(\alpha - 1, \alpha)$ be a resolvable TD, for a prime power α. Then C_{TD} is a $(\alpha - 1) - (\alpha^2 - \alpha, M, k, \alpha, \alpha^2 - \alpha - 1)$ FRB code with $M \ge k\alpha - \binom{k}{2} + (\alpha - 1)\binom{x}{2} + xy$, where $x, y \ge 0$ are integers which satisfy $k = x(\alpha - 1) + y$, $y \le \alpha - 2$.

Example 6 We consider the FRB code based on TD$(3, 4)$. By Theorem 5, for $k = 4$ we have a $3 - (12, 11, 4, 4, 11)$ FRB code, which stores a file of size 11 and which allows for parallel reads of any (coded) 11 symbols.

In general, when a given FR code is considered as a batch code, determining its parameter t (the number of symbols that can be read in parallel) is a nontrivial task. Similarly, for a given batch code it is difficult to find the parameter M (the file size) for any k. In the following, we consider an FRB code based on TD$(3, \alpha)$, where every symbol is replicated 3 times. For this code, the parameter M is given in [14]. We obtain the upper and lower bounds on t in the following theorem.

Theorem 7 *The FRB code based on TD$(3, \alpha)$ is a $3-(3\alpha, M, k, \alpha, t)$ code, where $6 \leq t \leq 2\alpha + 1$ for $\alpha \geq 7$ and $t = 12$ for $\alpha = 5$. The file size is given by $M = k\alpha - \binom{k}{2} + 3\binom{x}{2} + xy$, for $x, y \geq 0$ such that $k = 3x + y$ and $y \leq 2$.*

Proof The parameters n, ρ and M follow from the properties of TD$(3, \alpha)$ and FR codes based on TDs [14]. The lower bound on t follows from Theorem 5.1. To prove the upper bound on t one can consider a specific structure of an incidence matrix for TD and show that there are $2\alpha + 2$ columns that cover only $2\alpha + 1$ rows. □

In the rest of this section we consider FRB codes obtained from affine planes. The optimality of uniform combinatorial batch codes based on affine planes was proved in [15].

An *affine plane* of order s, denoted by $A(s)$, is a set system $(X, \mathscr{B})$, where X is a set of $|X| = s^2$ *points*, $\mathscr{B}$ is a collection of s-subsets (*blocks*) of X of size $|\mathscr{B}| = s(s + 1)$, such that each pair of points in X occur together in exactly one block of $\mathscr{B}$. An affine plane is called *resolvable*, if the *set* $\mathscr{B}$ can be partitioned into $s + 1$ sets of size s, called parallel classes, such that every element of X is contained in exactly one block of each class. It is well known [1] that if q is a prime power, then there exists a resolvable affine plane $A(q)$.

Theorem 8 *Let $A(q)$ be an affine plane and let $\boldsymbol{I}(A)$ be its $q^2 \times (q^2 + q)$ incidence matrix. Then the FRB code C_A with the incidence matrix equal to $\boldsymbol{I}(A)$ is a $q - (q^2, k(q + 1) - \binom{k}{2}, k, q + 1, q^2)$ FRB code.*

Proof The parameters ρ, n, α and t follow from the properties of the batch code based on $A(q)$ [15]. Since any two points of $A(q)$ belong to exactly one block and hence any two rows of $\mathbf{I}(A)$ intersect, it follows that the file size is $k(q + 1) - \binom{k}{2}$. □

3 Erasure Combinatorial Batch Codes

In this section we consider uniform combinatorial batch codes which can tolerate node failures (erasures). We call such batch codes *erasure batch* codes. Specifically, we define a $\rho - (\theta, N = \rho\theta, t, n, \Delta)$ *uniform erasure combinatorial batch* code (ECBC) to be a code which stores θ data symbols on n nodes, such that each symbol is stored on ρ nodes and for any given set of Δ failed nodes, any batch of t symbols can be retrieved by reading at most one symbol from each one of $n - \Delta$ available nodes, while keeping the total storage equal to N. Note that it should hold that $\Delta \leq \rho - 1$.

Remark 9 Note that if any set of Δ nodes contains at most t different symbols, then it is possible to correct any Δ erasures, i.e., to repair Δ failed nodes by reading at most one symbol from every available node.

Similarly to Theorem 2, we provide the necessary and sufficient conditions on a binary matrix to be the incidence matrix of a uniform ECBC.

Theorem 10 *An $n \times \theta$ binary matrix A with ρ* ones *in each column is the incidence matrix of a $\rho - (\theta, N, t, n, \Delta)$ uniform ECBC if and only if any i columns of A, $1 \leq i \leq t$, cover at least $i + \Delta$ rows.*

Based on Theorem 10 and resolvability of $A(q)$ [1] we have the following result.

Theorem 11 *Let $A(q)$ be an affine plane and let $\boldsymbol{I}(A)$ be its $q^2 \times (q^2+q)$ incidence matrix. Then the code C_A^E with the incidence matrix equal to $\boldsymbol{I}(A)$ is a $q - (q^2 + q, q^3 + q^2, t, q^2, q-1)$ uniform ECBC, where $\frac{q^2-q+2}{2} \leq t \leq q^2 - q$.*

Proof The parameters ρ, θ, N, n follow from the properties of $A(q)$, and Δ is the largest possible. To prove the upper bound on t we consider a set of erased nodes which correspond to $q-1$ points of a block b of $A(q)$. Let $p \in b$ be the point which was not erased. If we take one block in the parallel class which contains b and $q-1$ blocks which do not contain p in each one of q other parallel classes, then the corresponding q^2-q+1 columns of $\mathbf{I}(A)$ cover at most q^2-1 rows, thus by Theorem 10, $t \leq q^2-q$. To prove the lower bound on t we note that any q columns of $\mathbf{I}(A)$ cover at least $q^2 - \binom{q}{2}$ rows (since there are q blocks of $A(q)$ which pairwise intersect). Then since $\frac{q^2-q+2}{2} \geq q$ for $q \geq 2$, any i columns, where $q \leq i \leq \frac{q^2-q+2}{2}$, cover at least $q^2 - \binom{q}{2} = \frac{q^2-q+2}{2} + (q-1) \geq i + (q-1)$ rows. For $i \leq q-1$ it holds that any i columns cover at least $iq - \binom{i}{2} \geq i + (q-1)$ rows, which completes the proof. □

Now we consider a uniform ECBC C_{TD}^E based on a transversal design, i.e., the code with the incidence matrix equal to the incidence matrix of TD. Similarly to Theorems 5 and 7 one can prove the following result.

Theorem 12

- *Let TD$(2, \alpha)$ be a TD with $\alpha > 2$. Then the code C_{TD}^E is a $2 - (\alpha^2, 2\alpha^2, 3, 2\alpha, 1)$ uniform ECBC.*
- *Let TD$(3, \alpha)$ be a TD with $\alpha > 3$. Then the code C_{TD}^E is a $3 - (\alpha^2, 3\alpha^2, t, 3\alpha, 2)$ uniform ECBC, where $4 \leq t \leq 2\alpha - 2$ for $\alpha \geq 6$, $t = 9$ for $\alpha = 5$, and $t = 8$ for $\alpha = 4$.*

4 Conclusion and Future Work

This paper introduces two new families of erasure codes for distributed storage systems, namely fractional repetition batch codes and uniform erasure combinatorial batch codes. FRB codes have the properties of both FR and batch codes allowing for uncoded repairs of failed system nodes and parallel reads of subsets of data symbols. Uniform ECBCs have the properties of combinatorial batch codes even in presence of system nodes failures. We provide the matrix description of these codes and present constructions based on transversal designs and affine planes.

We conclude with a list of open problems for future research.

1. Find an upper bound on t and M given other parameters $\{n, \rho, \alpha, k\}$ for an FRB code;
2. Given the set of parameters $\{n, \rho, \alpha, k\}$, construct a $\rho-(n, M, k, \alpha, t)$ FRB code with the maximum M and t;
3. Find the exact values of t for FRB codes and ECBCs based on transversal designs and affine planes.

Acknowledgements The author thanks Tuvi Etzion and Mark Silberstein for the valuable discussions. The author also wishes to thank COST Action IC1104 "Random Network Coding and Designs over GF(q)" on travel support to present this work.

References

1. Anderson, I.: Combinatorial Designs and Tournaments. Clarendon Press, Oxford (1997)
2. Bhattacharya, S., Ruj, S., Roy, B.K.: Combinatorial batch codes: a lower bound and optimal constructions. Adv. Math. Commun. **3**(1), 165–174 (2012). doi:10.3934/amc.2012.6.165
3. Bujtás, C., Tuza, Z.: Optimal batch codes: Many items or low retrieval requirement. Adv. Math. Commun. **5**(3), 529–541 (2011). doi:10.3934/amc.2011.5.529
4. Dimakis, A., Godfrey, P., Wu, Y., Wainwright, M., Ramchandran, K.: Network coding for distributed storage systems. IEEE Trans. Inf. Theory **56**(9), 4539–4551 (2010). doi:10.1109/TIT.2010.2054295
5. Dimakis, A.G., Ramchandran, K., Wu, Y., Suh, C.: A survey on network codes for distributed storage. Proc. IEEE **99**, 476–489 (2011)
6. El Rouayheb, S., Ramchandran, K.: Fractional repetition codes for repair in distributed storage systems. In: Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Urbana-Champaign, pp. 1510–1517 (2010)
7. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Batch codes and their applications. In: Proceedings of the 36th Annual ACM Symposium on the Theory of computing (STOC'04), Chicago, pp. 262–271 (2004). doi:10.1145/1007352.1007396
8. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1978)
9. Olmez, O., Ramamoorthy, A.: Repairable replication-based storage systems using resolvable designs. In: Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, pp. 1174–1181 (2012). doi:10.1109/Allerton.2012.6483351
10. Paterson, M.B., Stinson, D.R., Wei, R.: Combinatorial batch codes. Adv. Math. Commun. **3**(1), 13–27 (2009). doi:10.3934/amc.2009.3.13
11. Pawar, S., Noorshams, N., El Rouayheb, S., Ramchandran, K.: Dress codes for the storage cloud: simple randomized constructions. In: Proceedings of the 2011 IEEE International Symposium on Information Theory (ISIT 2011), St. Petersburg, pp. 2338–2342 (2011). doi:10.1109/ISIT.2011.6033980

12. Rashmi, K.V., Shah, N., Kumar, P.: Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. IEEE Trans. Inf Theory **57**(8), 5227–5239 (2011). doi:10.1109/TIT.2011.2159049
13. Shah, N., Rashmi, K.V., Kumar, P., Ramchandran, K.: Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff. IEEE Trans. Inf. Theory **58**(3), 1837–1852 (2012). doi:10.1109/TIT.2011.2173792
14. Silberstein, N., Etzion, T.: Optimal fractional repetition codes (2014). arXiv:1401.4734
15. Silberstein, N., Gál, A.: Optimal combinatorial batch codes based on block designs (2013). Accepted for publication in Designs, Codes and Cryptography (Springer) http://dx.doi:10.1007/s10623-014-0007-9

Idempotents Generators for Minimal Cyclic Codes of Length $p^n q$

Gustavo Terra Bastos and Marinês Guerreiro

Abstract Let p and q be distinct positive prime numbers and ℓ a positive integer such that $\gcd(\ell, pq) = 1$. For a natural number $n \geq 1$, let $\mathscr{C}_{p^n q}$ be a cyclic group of order $p^n q$ and $\mathbb{F}_\ell$ a finite field with ℓ elements. In this paper we explicitly present the primitive idempotents of the group algebra $\mathbb{F}_\ell \mathscr{C}_{p^n q}$ under some further restrictions on ℓ, p and q. These idempotents generate the minimal ideals of $\mathbb{F}_\ell \mathscr{C}_{p^n q}$, hence the minimal cyclic codes of length $p^n q$. Our computation is based on techniques developed by Bakshi and Raka (Finite Fields Appl 9(4):432–448, 2003) and Ferraz and Polcino Milies (Finite Fields Appl 13:382–393, 2007). A particular example for codes of length 245 is computed and we believe that this points out some mistakes in current literature on this subject.

Keywords Minimal cyclic codes • Primitive idempotents

1 Introduction

Cyclic codes are usually characterized as ideals in quotient rings of polynomials which are easily proved to be isomorphic to ideals in group algebras of cyclic groups. There are advantages in both approaches but the last one has been quite useful lately, mainly on the problem of explicitly (and correctly) compute the primitive idempotents that generate minimal codes (see [2] and [4]).

Work partially supported by CNPq/(Brasil).

Work partially supported by FAPEMIG/MG(Brasil).

G.T. Bastos (✉)
Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 13083-970 Campinas, SP, Brasil
e-mail: gtbastos@yahoo.com.br

M. Guerreiro
Departamento de Matemática, Universidade Federal de Viçosa, 36570-900 Viçosa, MG, Brasil
e-mail: marines@ufv.br

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_37

For a natural number $n \geq 1$, p and q distinct positive prime numbers and ℓ a positive integer such that $\gcd(\ell, pq) = 1$, let $\mathscr{C}_{p^n q}$ be a cyclic group of order $p^n q$ and $\mathbb{F}_\ell$ a finite field with ℓ elements. In this paper we use a mix of both polynomial and group algebra techniques to compute a complete set of primitive idempotents of the group algebra $\mathbb{F}_\ell \mathscr{C}_{p^n q}$, under some further restrictions which are stated below in (1).

In Sect. 2 we summarize an argument for the number of simple components of a semisimple abelian group algebra and apply it to our study case. In Sect. 3, we present a method to construct the primitive idempotents of the group algebra $\mathbb{F}_\ell \mathscr{C}_{p^n}$, for $\mathscr{C}_{p^n}$ a cyclic group of order p^n such that $\gcd(\ell, p) = 1$. This is the main tool which is used in Sect. 4 to finally compute the primitive idempotents that generate the minimal codes of $\mathbb{F}_\ell \mathscr{C}_{p^n q}$.

2 The Number of Simple Components of $\mathbb{F}_\ell \mathscr{C}_{p^n q}$

Let $\mathbb{F}_\ell$ be a finite field with ℓ elements and G a finite abelian group such that $\gcd(\ell, |G|) = 1$. In [3] Ferraz established the conditions under which the number of simple components of $\mathbb{F}_\ell G$ is equal to the number of cyclic subgroups of G and presented a method to compute the number of simple components of a semisimple group algebra. Here we summarize the important results from [3] and [4] that we apply in the case of our interest.

Definition 1 Let G be a finite abelian group and $g \in G$. The **ℓ-cyclotomic class of g in G** is the set $C_g = \left\{ g^{\ell^j} / 0 \leq j \leq t_g - 1 \right\}$, where t_g is the smallest positive integer number such that $\ell^{t_g} \equiv 1 (\text{mod } o(g))$ and $o(g)$ denotes the order of g in G, that is, the least positive integer t such that $g^t = 1$.

Theorem 2 ([4], Theorem 2.1) *If* $\gcd(\ell, |G|) = 1$, *then the number of simple components of* $\mathbb{F}_\ell G$ *is equal to the number of* ℓ*-cyclotomic classes of* G.

Given positive integers r e m, denote by $\overline{r} \in \mathbb{Z}_m$ the image of r in $\mathbb{Z}_m$, the ring of integers modulo m and $U(\mathbb{Z}_m)$ the group of the units of $\mathbb{Z}_m$. Then $\mathscr{G}_g = \left\{ g^r | \overline{r} \in U\left(\mathbb{Z}_{o(g)}\right) \right\}$.

Theorem 3 *Let* $\mathbb{F}_\ell$ *be a finite field with* ℓ *elements and* G *a finite abelian group with exponent* e *such that* $\gcd(\ell, |G|) = 1$. *Then* $C_g = \mathscr{G}_g$, *for all* $g \in G$, *if only if* $U(\mathbb{Z}_e)$ *is a cyclic group generated by* $\overline{\ell} \in \mathbb{Z}_e$.

The following theorem gives us the conditions on the exponent e of the group G and the size ℓ of the finite field that satisfies the necessary part of Theorem 3.

Corollary 4 ([6], Teorema 7.10) *Let* $\mathbb{F}_\ell$ *be a finite field with* ℓ *elements and* G *a finite abelian group with exponent* e *such that* $\gcd(\ell, |G|) = 1$. *Then* $C_g = \mathscr{G}_g$, *for all $g \in G$, if only if one of following conditions holds, where ϕ denotes Euler's totient function:*

(a) $e = 2$ *and* ℓ *is odd;*
(b) $e = 4$ *and* $\ell \equiv 3 (\mathrm{mod}\ 4)$*;*
(c) $e = p^n$ *and* $o(\ell) = \phi(p^n)$ *in* $U\left(\mathbb{Z}_{p^n}\right)$*;*
(d) $e = 2p^n$ *and* $o(\ell) = \phi(p^n)$ *in* $U\left(\mathbb{Z}_{2p^n}\right)$*.*

Theorem 5 *Let $n \geq 1$ be a natural number, p and q distinct positive prime numbers, ℓ a positive integer such that* $\gcd(\ell, pq) = 1$ *and*

$$\begin{aligned} &\gcd(p-1, q-1) = 2 \\ &\ell \text{ generates both unit groups } U(\mathbb{Z}_{p^n}) \text{ and } U(\mathbb{Z}_q) \\ &\gcd(p-1, q) = \gcd(p, q-1) = 1. \end{aligned} \tag{1}$$

Let $\mathscr{C}_{p^n q}$ be a cyclic group of order $p^n q$ and $\mathbb{F}_\ell$ a finite field with ℓ elements. Then the number of the simple components of $\mathbb{F}_\ell \mathscr{C}_{p^n q}$ is $3n + 2$ and the ℓ-cyclotomic classes of $\mathscr{C}_{p^n q}$ are:

$$C_1,\ C_{g^{p^n}},\ C_{g^{p^t}},\ C_{g^{p^t q}},\ \text{and } C_{g^{dp^t}}, \tag{2}$$

where $0 \leq t \leq n-1$ *and d is a fixed integer satisfying* $\gcd(d,\ pq\ell) = 1$*,* $1 < d < pq$*,* $d \not\equiv \ell^k (\mathrm{mod}\, pq)$ *for any k,* $0 \leq k \leq \frac{\phi(pq)}{2} - 1$.

Proof It follows from combinatorial and counting arguments. A more general proof for finite cyclic groups can be read in [5]. □

3 Primitive Idempotents in $\mathbb{F}_\ell \mathscr{C}_{p^n}$

In this section, we present a method to construct the primitive idempotents of the group algebra $\mathbb{F}_\ell G$, for G a cyclic group of order p^n. The main references for this section are [4] and [8].

Let H be a finite subgroup of G. If $\gcd(\ell, |H|) = 1$, define $\hat{H} = \frac{1}{|H|} \sum_{g \in H} g$.

The element $\hat{H}$ is an idempotent in $\mathbb{F}_\ell G$, according to [7, Lemma 3.6.6]. For $H = \langle h \rangle$ (the cyclic subgroup generated by h), sometimes we shall use $\hat{h}$ to denote $\hat{H}$. In particular, $\hat{G}$ is called **principal idempotent** of $\mathbb{F}_\ell G$.

Theorem 6 ([4], Lemma 3) *Let $G = \langle g \rangle$ be a cyclic group with order p^n and $\mathbb{F}_\ell$ a finite field with ℓ elements such that $\overline{\ell}$ generates $U\left(\mathbb{Z}_{p^n}\right)$. Consider*

$$G = G_0 \supset G_1 \supset \ldots \supset G_n = \{1\}$$

a descending chain on all subgroups of G. Then a complete set of primitive idempotents in $\mathbb{F}_\ell G$ is:

$$e_0 = \frac{1}{p^n}\sum_{g\in G} g \quad \textit{and} \quad e_i = \widehat{G_i} - \widehat{G_{i-1}},\ \textit{for}\ 1 \le i \le n.$$

A direct computation shows us that the expressions for the primitive idempotents of $\mathbb{F}_\ell G$ given in Theorem 6 are the same as the ones given polinomially by Arora and Pruthi in [8].

4 Primitive Idempotents in $\mathbb{F}_\ell \mathscr{C}_{p^n q}$

In this section we combine two different techniques to compute all primitive idempotents of the group algebra $\mathbb{F}_\ell \mathscr{C}_{p^n q}$, under the restrictions stated in (1). Considering the isomorphism of group algebras $\mathbb{F}_\ell \mathscr{C}_{p^n q} \cong \mathbb{F}_\ell(\mathscr{C}_{p^n} \times \mathscr{C}_q) \cong \mathbb{F}_\ell \mathscr{C}_{p^n} \otimes \mathbb{F}_\ell \mathscr{C}_q$, the main idea is to compute the products of all primitive idempotents in $\mathbb{F}_\ell \mathscr{C}_{p^n}$ with all primitive idempotents in $\mathbb{F}_\ell \mathscr{C}_q$, using the expressions obtained in Sect. 3.

We shall prove that a product of a primitive idempotent in $\mathbb{F}_\ell \mathscr{C}_{p^n}$ with the principal idempotent in $\mathbb{F}_\ell \mathscr{C}_q$ (and vice-versa) gives us a primitive idempotent in $\mathbb{F}_\ell \mathscr{C}_{p^n q}$. If the idempotents $e \in \mathbb{F}_\ell \mathscr{C}_{p^n}$ and $f \in \mathbb{F}_\ell \mathscr{C}_q$ that we multiply are both no principal, then their product in $\mathbb{F}_\ell \mathscr{C}_{p^n q}$ is not a primitive idempotent and we use the expressions obtained by Bakshi and Raka in [1], with our notation of group algebra, to write this product as a sum of two primitive idempotents in $\mathbb{F}_\ell \mathscr{C}_{p^n q}$. Here is where the condition $\gcd(p-1, q-1) = 2$ of (1) comes into play. If we did not suppose this, the product of these two no principal idempotents could split into a sum of more than two primitive idempotents in $\mathbb{F}_\ell \mathscr{C}_{p^n q}$, which is much harder to compute. For an element x in a finite abelian group G, denote by $\mathscr{S}_x = \sum\limits_{h\in C_x} h$ the sum of all elements belonging to the ℓ-cyclotomic class of x in G. We need the following technical results.

Lemma 7 *Let $G = \langle g\rangle$ be a finite cyclic group of order $p^n q$. Then*

(a) $$\sum_{i=1}^{p^{j+1}-1}\left(g^{qp^{n-j-1}}\right)^i = \sum_{i=1}^{p^j-1}\left(g^{qp^{n-j}}\right)^i + \mathscr{S}_{g^{qp^{n-j-1}}}.$$

(b) $$\sum_{i=1}^{p^j-1}\left(g^{qp^{n-j}}\right)^i = \sum_{i=n-j}^{n-1}\mathscr{S}_{g^{p^i q}}.$$

(c) $$\sum_{k=1}^{q-1}\left(g^{p^n}\right)^k \sum_{i=1}^{p^j-1}\left(g^{qp^{n-j}}\right)^i = \sum_{i=n-j}^{n-1}\mathscr{S}_{g^{p^i}} + \sum_{i=n-j}^{n-1}\mathscr{S}_{g^{dp^i}},$$

where d is like in Theorem 5.

Proof We omit the proofs of *(a)* and *(b)*. For *(c)*, is easy to prove that $\sum_{k=1}^{q-1} \left(g^{p^n}\right)^k = \mathscr{S}_{g^{p^n}}$. Thus, by (ii), we have

$$\sum_{k=1}^{q-1} \left(g^{p^n}\right)^k \sum_{i=1}^{p^j-1} \left(g^{qp^{n-j}}\right)^i = \mathscr{S}_{g^{p^n}} \sum_{i=n-j}^{n-1} \mathscr{S}_{g^{p^i q}}. \tag{3}$$

For an arbitrary element $\mathscr{S}_{g^{p^{n-(j-w)}}}$, for $0 \leq w \leq j-1$, we prove

$$\mathscr{S}_{g^{p^n}} \mathscr{S}_{g^{qp^{n-(j-w)}}} = \mathscr{S}_{g^{p^{n-(j-w)}}} + \mathscr{S}_{g^{dp^{n-(j-w)}}}, \tag{4}$$

Indeed, the product

$$\left(g^{p^n} + g^{lp^n} + \ldots + g^{\ell^{\phi(q)-1} p^n}\right) \left(g^{p^{n-(j-w)}q} + g^{lp^{n-(j-w)}q} + \ldots \right. \\ \left. + g^{\ell^{\phi\left(p^{j-w}\right)-1} p^{n-(j-w)}q}\right) \tag{5}$$

may be rewritten as

$$g^{p^{n-(j-w)}\left(p^{j-w}+q\right)} + g^{p^{n-(j-w)}\left(p^{j-w}+lq\right)} + \ldots + g^{p^{n-(j-w)}\left(\ell^{\phi(q)-1} p^{j-w}+\ell^{\phi\left(p^{j-w}\right)-1} q\right)}, \tag{6}$$

where we are summing exactly $\phi\left(q\right) \phi\left(p^{j-w}\right) = \phi\left(qp^{j-w}\right)$ distinct elements and this is the same number of distinct elements that appear in the sum $\mathscr{S}_{g^{p^{n-(j-w)}}} + \mathscr{S}_{g^{dp^{n-(j-w)}}}$. The remaining of the proof is given by an *exclusion argument*.

We recall from the proof of Theorem 5 that the cyclic group $\mathscr{C}_{p^n q}$ can be partitioned in the following ℓ-cyclotomic classes: C_1, $C_{g^{p^n}}$, $C_{g^{p^t}}$, $C_{g^{p^t q}}$, and $C_{g^{dp^t}}$, where $0 \leq t \leq n-1$.

We first note that no element in (6) is equal to g, hence no element of the ℓ-cyclotomic class C_1 appears in (6). Now we claim that no element in (6) belongs to the following ℓ-cyclotomic classes: $C_{g^{p^n}}$, $C_{g^{p^t q}}$, $C_{g^{p^t}}$ and $C_{g^{dp^t}}$, with $t \neq j-w$. Indeed, otherwise, the numbers written as $\left(\ell^r p^{j-w} + \ell^s q\right)$, for $0 \leq r \leq \phi(q)-1$ and $0 \leq s \leq \phi\left(p^{j-w}\right)-1$, appearing as powers in (6), would have to be multiples of p or q. This would contradict the hypothesis $\gcd(\ell, pq) = 1$. Therefore, all elements appearing in (6) must belong exactly to the ℓ-cyclotomic classes $C_{g^{p^{n-(j-w)}}}$ and $C_{g^{dp^{n-(j-w)}}}$. This proves (4). □

Now we are ready to prove the main result. The proof given here is an adaptation of the proof of [2, Theorem III.1] ($\ell \neq 2$), jointly with [1, Theorem 3].

Theorem 8 *Under the same hypothesis of Theorem 5, let $\mathscr{C}_{p^n q} = \langle g \rangle$ be a cyclic group of order $p^n q$, $\mathscr{C}_{p^n} = \langle a \rangle = \langle g^q \rangle$ and $\mathscr{C}_q = \langle b \rangle = \left\langle g^{p^n} \right\rangle$. Then*

$$\left\{ \hat{a} \cdot \hat{b},\ \hat{a} \cdot (1 - \hat{b}),\ \left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}} \right) \cdot \hat{b},\ e_{*}^{n-j},\ e_{**}^{n-j},\ \textit{for}\ 0 \leq j \leq n-1 \right\} \tag{7}$$

is a complete set of primitive orthogonal idempotents of $\mathbb{F}_\ell \mathscr{C}_{p^n q} \cong \mathbb{F}_\ell \left(\mathscr{C}_{p^n} \times \mathscr{C}_q \right)$, with

$$e^{n-j} = \left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}} \right) \left(1 - \hat{b} \right) = e_{*}^{n-j} + e_{**}^{n-j},\ \textit{for all}\ 0 \leq j \leq n-1,$$

where

$$e_{*}^{n-j} = \frac{(p-1)(q-1)}{2p^{j+1}q} \left[1 + \sum_{i=n-j}^{n-1} \mathscr{S}_{g^{p^i q}} \right] + \frac{A_{n-1}}{p^{n+j}q} \mathscr{S}_{g^{p^{n-j-1}}} + \frac{B_{n-1}}{p^{n+j}q} \mathscr{S}_{g^{dp^{n-j-1}}}$$
$$- \frac{(p-1)}{2p^{j+1}q} \left[\sum_{i=n-j}^{n} \mathscr{S}_{g^{p^i}} + \sum_{i=n-j}^{n-1} \mathscr{S}_{g^{dp^i}} \right] - \frac{(q-1)}{2p^{j+1}q} \mathscr{S}_{g^{p^{n-j-1}q}} \tag{8}$$

$$e_{**}^{n-j} = \frac{(p-1)(q-1)}{2p^{j+1}q} \left[1 + \sum_{i=n-j}^{n-1} \mathscr{S}_{g^{p^i q}} \right] + \frac{B_{n-1}}{p^{n+j}q} \mathscr{S}_{g^{p^{n-j-1}}} + \frac{A_{n-1}}{p^{n+j}q} \mathscr{S}_{g^{dp^{n-j-1}}}$$
$$- \frac{(p-1)}{2p^{j+1}q} \left[\sum_{i=n-j}^{n} \mathscr{S}_{g^{p^i}} + \sum_{i=n-j}^{n-1} \mathscr{S}_{g^{dp^i}} \right] - \frac{(q-1)}{2p^{j+1}q} \mathscr{S}_{g^{p^{n-j-1}q}}, \tag{9}$$

and the constants A_{n-1} e B_{n-1} are given as in [1, Theorem 3]:

Proof Let $G = \mathscr{C}_{p^n q}$. Consider the following decomposition

$$\begin{aligned} \mathbb{F}_\ell G &\cong \mathbb{F}_\ell \mathscr{C}_{p^n} \otimes \mathbb{F}_\ell \mathscr{C}_q \\ &\cong \left[\mathbb{F}_\ell \mathscr{C}_{p^n} \hat{a} \oplus \ldots \oplus \mathbb{F}_\ell \mathscr{C}_{p^n} \left(1 - \widehat{a^{p^{n-1}}} \right) \right] \otimes \left[\mathbb{F}_\ell \mathscr{C}_q \hat{b} \oplus \mathbb{F}_\ell \mathscr{C}_q \left(1 - \hat{b} \right) \right] \\ &\cong \mathbb{F}_\ell G\ \hat{a}\hat{b} \oplus \ldots \oplus \mathbb{F}_\ell G \left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}} \right) (1 - \hat{b}) \oplus \ldots \oplus \mathbb{F}_\ell G \\ &\quad \left(1 - \widehat{a^{p^{n-1}}} \right) (1 - \hat{b}) \end{aligned}$$

The principal idempotent $\hat{g} = \hat{a}\hat{b}$ is always primitive.

Moreover, $\left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right)\hat{b}$ and $\hat{a}\left(1-\hat{b}\right)$ are also primitive idempotents. Indeed,

$$\left(\left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right)\hat{b}\right)^2 = \left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right)^2 \left(\hat{b}\right)^2 = \left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right)\hat{b}. \tag{10}$$

Furthermore, $\mathbb{F}_\ell G\hat{b}\left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right) \cong \left(\mathbb{F}_\ell\mathscr{C}_{p^n}\right)\left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right)$ is a simple ideal, according to Theorem 6. Hence $\left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right)\hat{b}$ is a primitive idempotent of $\mathbb{F}_\ell G$. Using a similar argument, we verify that $\hat{a}\left(1-\hat{b}\right)$ is also a primitive idempotent of $\mathbb{F}_\ell G$.

Finally, we shall prove that $e^{n-j} = \left(\widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-(j+1)}}}\right)\left(1-\hat{b}\right)$ decomposes as a sum of two primitive idempotents in $\mathbb{F}_\ell G$. We use [1] to present two primitive idempotents e_*^{n-j} and e_{**}^{n-j} such that $e_*^{n-j} + e_{**}^{n-j} = e^{n-j}$. Indeed,

$$\begin{aligned}
e^{n-j} &= \widehat{a^{p^{n-j}}} - \widehat{a^{p^{n-j}}}\hat{b} - \widehat{a^{p^{n-(j+1)}}} + \widehat{a^{p^{n-(j+1)}}}\hat{b} \\
&= \frac{1}{p^j}\sum_{i=0}^{p^j-1}\left(g^{qp^{n-j}}\right)^i - \frac{1}{p^j q}\sum_{i=0}^{p^j-1}\left(g^{qp^{n-j}}\right)^i \sum_{k=0}^{q-1}\left(g^{p^n}\right)^k - \frac{1}{p^{j+1}}\sum_{t=0}^{p^{j+1}-1}\left(g^{qp^{n-j-1}}\right)^t \\
&+ \frac{1}{p^{j+1}q}\sum_{t=0}^{p^{j+1}-1}\left(g^{qp^{n-j-1}}\right)^t \sum_{k=0}^{q-1}\left(g^{p^n}\right)^k \\
&= \frac{1}{p^{j+1}q}\left\{(p-1)(q-1)\left[1+\sum_{i=n-j}^{n}\mathscr{S}_{g^{p^i q}}\right] - (q-1)\mathscr{S}_{g^{qp^{n-j-1}}}\right. \\
&+ \left.\sum_{k=1}^{q-1}\left(g^{p^n}\right)^k\left[-(p-1) + \sum_{t=1}^{p^{j+1}-1}\left(g^{qp^{n-j-1}}\right)^t - p\sum_{i=1}^{p^j-1}\left(g^{qp^{n-j}}\right)^i\right]\right\} \\
&= \frac{1}{p^{j+1}q}\left\{(p-1)(q-1)\left[1+\sum_{i=n-j}^{n}\mathscr{S}_{g^{p^i q}}\right] - (q-1)\mathscr{S}_{g^{qp^{n-j-1}}}\right. \\
&- \left.(p-1)\left[\sum_{i=n-j}^{n}\mathscr{S}_{g^{p^i}} + \sum_{i=n-j}^{n-1}\mathscr{S}_{g^{dp^i}}\right] + \sum_{i=n-j}^{n}\mathscr{S}_{g^{p^{n-j-1}}} + \sum_{i=n-j}^{n}\mathscr{S}_{g^{dp^{n-j-1}}}\right\}.
\end{aligned}$$

Now, with an easy computation, we verify the equality $e_*^{n-j} + e_{**}^{n-j} = e^{n-j}$. Therefore, we explicitly presented all primitive idempotents of the group algebra $\mathbb{F}_\ell G$. □

As an example, we explicitly compute all primitive idempotents of $\mathbb{F}_3(\mathscr{C}_{49} \times \mathscr{C}_5)$. This particular example allowed us to identify some possible errors that appeared in the computation of idempotents in [5] and made us to develop the general setting of the previous section.

For $p = 7$, $q = 5$ and $\ell = 3$, we have $\langle \overline{3} \rangle = U(\mathbb{Z}_{49})$ and $\langle \overline{3} \rangle = U(\mathbb{Z}_5)$. Moreover, the other hypothesis of (1) are also satisfied. Hence, by Theorem 8, the primitive idempotents of $\mathbb{F}_3(\mathscr{C}_{49} \times \mathscr{C}_5)$ are

$$e_0 = 2S_{\mathscr{C}_1} + 2S_{\mathscr{C}_g} + 2S_{\mathscr{C}_{g^2}} + 2S_{\mathscr{C}_{g^7}} + 2S_{\mathscr{C}_{g^{14}}} + 2S_{\mathscr{C}_{g^{49}}} + 2S_{\mathscr{C}_{g^5}} + 2S_{\mathscr{C}_{g^{35}}}$$
$$e_1 = 2S_{\mathscr{C}_1} + S_{\mathscr{C}_g} + S_{\mathscr{C}_{g^2}} + S_{\mathscr{C}_{g^7}} + S_{\mathscr{C}_{g^{14}}} + S_{\mathscr{C}_{g^{49}}} + 2S_{\mathscr{C}_{g^5}} + 2S_{\mathscr{C}_{g^{35}}}$$

$$\begin{array}{ll}
e_2 = S_{\mathscr{C}_g} + S_{\mathscr{C}_{g^2}} + S_{\mathscr{C}_{g^5}} & e_3 = S_{\mathscr{C}_{g^7}} + S_{\mathscr{C}_{g^{14}}} + S_{\mathscr{C}_{g^{35}}} \\
e_4^* = 2S_{\mathscr{C}_g} + 2S_{\mathscr{C}_{g^5}} & e_4^{**} = 2S_{\mathscr{C}_{g^2}} + 2S_{\mathscr{C}_{g^5}} \\
e_5^* = 2S_{\mathscr{C}_{g^7}} + 2S_{\mathscr{C}_{g^{35}}} & e_5^{**} = 2S_{\mathscr{C}_{g^{14}}} + 2S_{\mathscr{C}_{g^{35}}}.
\end{array}$$

References

1. Bakshi, G.K., Raka, M.: Minimal cyclic codes of length $p^n q$. Finite Fields Appl. **9**(4), 432–448 (2003)
2. Chalom, G., Ferraz, R.A., Guerreiro, M., Polcino Milies, C.: Minimal binary abelian codes of length $p^m q^n$. Preprint in arXiv:1205.5699
3. Ferraz, R.: Simple components and central units in group algebras. J. Algebra **279**, 191–203 (2004)
4. Ferraz, R., Polcino Milies, C.: Idempotents in group algebras and minimal abelian codes. Finite Fields Appl. **13**, 382–393 (2007)
5. Kumar, P., Arora, S.K.: $\lambda-$ Mapping and primitive idempotents in semisimple ring $\mathfrak{R}_m$. Commun. Algebra **41**, 3679–3694 (2013)
6. Martin, P.A.: Grupos, Corpos e Teoria de Galois. Editora Livraria da Física, São Paulo (2010)
7. Polcino Milies, C., Sehgal, S.K.: An Introduction to Group Rings. Kluwer Academic, Dordrecht (2002)
8. Pruthi, M., Arora, S.K.: Minimal cyclic codes of prime power length. Finite Fields Appl. **3**(2), 99–113 (1997)

Reconstruction of Eigenfunctions of q-ary n-Dimensional Hypercube

Anastasia Vasil'eva

Abstract We investigate eigenfunctions on the graph of n-dimensional q-ary Hamming space. First, we mention the formula of the interdependence for local weight enumerators of an eigenfunction in two orthogonal faces. Then we develop methods to reconstruct an eigenfunction in a ball by its values in the corresponding sphere. We use as an example the simplest case and obtain numerical conditions for reconstructing in this case.

Keywords q-ary Hamming scheme • Eigenfunctions • Reconstruction • Krawtchouk polynomials

1 Introduction

We study eigenfunctions of n-dimensional q-ary hypercube. We apply an explicit formula for local distributions in two orthogonal faces [10]. The local distributions were considered in [5, 7–9] for 1-error correcting perfect codes, perfect colorings and eigenfunctions of the hypercube in binary case ($q = 2$). In case $q > 2$ they are investigated in [2] for 1-error-correcting codes. In [6] more general case of direct product of graphs is studied; however, the formula is not extended for classes of graphs. The reconstruction problems in binary case were studied, for example, in [4, 9]. Earlier it was obtained in [1] that any perfect code of length n is uniquely determined by its codewords of weight $(n - 1)/2$.

The paper is organized as follows: in Sect. 2 we give some necessary notations and facts; in Sect. 3 we mention the formula for local weight enumerators of eigenfunctions in a pair of orthogonal faces and prove some necessary lemmas; using results of Sect. 3 we obtain in Sect. 4 the main Theorem 5 on reconstruction of eigenfunctions and compare it (Theorem 7) with the previously obtained Theorem 6 for binary case.

A. Vasil'eva (✉)
Sobolev Institute of Mathematics, 630090 Novosibirsk, prosp. Ak. Koptyuga 4, Russia

Novosibirsk State University, 630090 Novosibirsk, Pirogova str. 2, Russia
e-mail: vasilan@math.nsc.ru

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_38

2 Preliminaries

Let $q > 2$ be a positive integer, not necessary prime power.

Consider the set $\mathbf{F}_q = \{0, 1, \ldots, q-1\}$ as the group by modulo q and the hypercube $\mathbf{F}_q^n$ as the abelian group $\mathbf{F}_q \times \ldots \times \mathbf{F}_q$. We investigate functions on vertices of the graph $\mathbf{F}_q^n$ of n-dimensional q-ary hypercube, in this graph two vertices are adjacent if the Hamming distance between them equals 1.

Here and elsewhere I denotes a subset of $\{1, \ldots, n\}$ and $\overline{I} = \{1, \ldots, n\} \backslash I$. Take a vertex $\alpha \in \mathbf{F}_q^n$. Denote by $s(\alpha)$ the support of a vertex α, i.e. the set of nonzero positions of α; the cardinality of the support is equal to the Hamming weight of α. Write $W_i(\alpha)$ for the set of all vertices β that differ from α in i positions; i.e., the Hamming distance $\rho(\alpha, \beta)$ between vertices α and β is equal to i. By definition, put

$$\Gamma_I(\alpha) = \{\beta \in \mathbf{F}_q^n \ : \ \beta_i = \alpha_i \ \forall \ i \notin I\},$$

then $\Gamma_I(\alpha)$ is said to be a $|I|$-dimensional face, it has a structure of $\mathbf{F}_q^{|I|}$. Write simply W_i and Γ_I instead of $W_i(\alpha)$ and $\Gamma_I(\alpha)$ in case α is all-zero vertex. We say that two faces $\Gamma_I(\alpha)$ and $\Gamma_J(\beta)$ are orthogonal if $J = \overline{I}$. It is easy to see that orthogonal faces have exactly one common vertex.

The Hamming association scheme (the introduction can be find in [3]) consists of the set $\mathbf{F}_q^n$ with $n+1$ associations $R_0, \ldots, R_n$ and for any $\alpha, \beta \in \mathbf{F}_q^n$ $(\alpha, \beta) \in R_i$, $i = 0, \ldots, n$, iff the Hamming distance $\rho(\alpha, \beta)$ between α and β equals i.

Let $D_i = D_i^{q,n}$ be the matrix of i-th association R_i, i.e. $(q^n \times q^n)$-matrix with the entries

$$(D_i^{q,n})_{\alpha,\beta} = \begin{cases} 1, \ \rho(\alpha, \beta) = i \\ 0, \ \rho(\alpha, \beta) \neq i \end{cases}$$

Obviously, $D = D_1^{q,n}$ is the adjacency matrix of the hypercube $\mathbf{F}_q^n$. We will say that λ is an eigenvalue of the graph if it is an eigenvalue of its adjacency matrix. In particular, λ is called an eigenvalue of the hypercube $\mathbf{F}_q^n$ if it is an eigenvalue of $D^{q,n}$. It is known [3] that the eigenvalues of $D^{q,n}$ are equal to $(q-1)n - qi$, $i = 0, 1, \ldots, n$. The corresponding eigenfunctions (we call them λ-functions) satisfy an equation

$$\sum_{\beta \in W_1(\alpha)} f(\beta) = ((q-1)n - qi) f(\alpha), \quad \alpha \in \mathbf{F}_q^n. \tag{1}$$

Rewrite this equations in a matrix form:

$$Df = \lambda f,$$

here f is a vector of values of the function. Furthermore, all matrices $D_i^{q,n}$, $i = 0, \ldots, n$, have the same eigensubspaces V_h, $h = 0, \ldots, n$, and the eigenvalues of

$D_i^{q,n}$ on V_h is equal to $P_i^{(q)}(h; n)$, where

$$P_m^{(q)}(t; N) = \sum_{j=0}^{m} (-1)^j (q-1)^{m-j} \binom{t}{j} \binom{N-t}{m-j}$$

is the Krawtchouk polynomial in t. The values of Krawtchouk polynomials can be described as coefficients of a polynomial in x and y:

$$(x-y)^t (x+(q-1)y)^{N-t} = \sum_{m=0}^{N} P_m^{(q)}(t; N) y^m x^{N-m}.$$

3 Local Distributions

Consider the space of all complex functions on the q-ary n-dimensional hypercube:

$$V = \left\{ f : \mathbf{F}_q^n \longrightarrow \mathbb{C} \right\}.$$

The functions can be considered as q^n-dimensional vectors:

$$f \leftrightarrow (f(0,\dots,0), f(0,\dots,0,1),\dots, f(q-1,\dots,q-1))^T$$

Introduce the concept of a local distribution. By definition, put

$$v_j^{I,f}(\alpha) = \sum_{\beta \in \Gamma_I(\alpha) \bigcap W_j(\alpha)} f(\beta),$$

the vector $v^{I,f}(\alpha) = (v_0^{I,f}(\alpha), \dots, v_{|I|}^{I,f}(\alpha))$ is called the local distribution of the function f in the face $\Gamma_I(\alpha)$ with respect to the vertex α, or shortly (I, α)-local distribution of f. We say that the polynomial $g_f^{I,\alpha}$ in variables x, y is a (I, α)-local weight enumerator of the function f if

$$g_f^{I,\alpha}(x, y) = \sum_{j=0}^{|I|} v_j^{I,f}(\alpha) y^j x^{|I|-j} = \sum_{\beta \in \Gamma_I(\alpha)} f(\beta) y^{|s(\beta-\alpha)|} x^{|I|-|s(\beta-\alpha)|}.$$

To simplify notations we omit α if it is the all-zero vertex and f if obvious. We describe the interdependence between the local weight enumerators of an eigenfunction of the hypercube in two orthogonal faces.

Theorem 1 ([10]) *Let λ be an eigenvalue of $\mathbf{F}_q^n$, f be a λ-function, $h = \frac{(q-1)n-\lambda}{q}$ and $\alpha \in \mathbf{F}_q^n$. Then*

$$(x + (q-1)y)^{h-|\overline{I}|} g_f^{\overline{I},\alpha}(x, y) = (x' + (q-1)y')^{h-|I|} g_f^{I,\alpha}(x', y'),$$

where $x' = x + (q-2)y$, $y' = -y$.

Present $(\overline{I}, \alpha)$-local weight enumerator in terms of (I, α)-local weight enumerator:

$$g^{\overline{I},\alpha}(x, y) = (x - y)^{h-|I|}(x + (q-1)y)^{|\overline{I}|-h} g^{I,\alpha}(x + (q-2)y, -y) \qquad (2)$$

If the set I is not "too large" then it is possible to represent the components of $(\overline{I}, \alpha)$-local distribution of f in terms of the components of (I, α)-local distribution of f. Specify the size of the set I that is not "too large": the right hand side expression of (2) should be a polynomial. In this case the components of the $(\overline{I}, \alpha)$-local distribution can be represented in terms of components of (I, α)-local distribution. More precisely, let

$$h = h(\lambda) = \frac{(q-1)n - \lambda}{q}, \qquad l(\lambda) = min\{h, n-h\}.$$

We get the following lemma from the formula (2).

Lemma 2 *If $|I| = k \le l(\lambda)$ then for any λ-function f*

$$v_j^{\overline{I}}(\alpha) = \sum_{i=0}^{j} r_{ij} v_i^I(\alpha), \qquad (3)$$

where

$$r_{ij} = (-1)^i \sum_{l=0}^{j-i} P_{j-i-l}^{(q)}(h - k; n - 2k)(q-2)^l \binom{k-i}{l} \qquad (4)$$

Proof The formula is obtained from (2) by direct calculations. □

In conclusion of this section we give a simple technical lemma. Let $\alpha \in F_q^n$ be a vertex of weight k. Put $I = s(\alpha)$ and consider the face $\Gamma_I(\alpha)$. This face has the dimension k and contains the all-zero vertex. The following lemma says that any component $v_i^I(\alpha)$ can be decomposed into the sum σ^+ over vertices of weight k and the sum σ^- over vertices of weight less than k:

Lemma 3 *For any $i \le k$*

$$v_i^I(\alpha) = \sigma_i^-(\alpha) + \sigma_i^+(\alpha),$$

where

$$\sigma_i^+(\alpha) = \sum_{\beta \in W_k \bigcap \Gamma_I(\alpha) \bigcap W_i(\alpha)} f(\beta)$$

and the value of $\sigma_i^-(\alpha)$ *does not depend on the values* $f(\beta),\ \beta \in W_k$*, and depends only on values* $f(\beta),\ \beta \in W_0 \bigcup \ldots \bigcup W_{k-1}$.

Proof By definition,

$$v_i^I(\alpha) = \sum_{\beta \in \Gamma_I(\alpha) \bigcap W_i(\alpha)} f(\beta) =$$

$$= \sum_{s=k-i}^{k-1} \sum_{\beta \in W_s \bigcap \Gamma_I(\alpha) \bigcap W_i(\alpha)} f(\beta) + \sum_{\beta \in W_k \bigcap \Gamma_I(\alpha) \bigcap W_i(\alpha)} f(\beta) = \sigma_i^-(\alpha) + \sigma_i^+(\alpha).$$

□

4 Reconstruction

Let λ be an eigenvalue of the hypercube $\mathbf{F}_q^n$ and $d \le l(\lambda)$ and f be a λ-function.

We deal with the following question: we know the values $f(\alpha)$ for all α with Hamming weight d, whether it is possible to determine uniquely the values $f(\alpha)$ for all α with Hamming weight less than d or not. We consider the first case, where $d \le l(\lambda)$. In cases $l(\lambda) \le d \le n - l(\lambda)$ and $d \ge n - l(\lambda)$ the formulae and the evaluations are more sophisticated.

At first, we note that

$$\sum_{\alpha \in W_d} f(\alpha) = P_d^{(q)}(h; n) f(\mathbf{0}),$$

this formula follows from one of the basic properties of Hamming association scheme (see, for example, [3]):

$$D_d^{q,n} f = P_d^{(q)}(h; n) f.$$

It means that we know the value at the vertex of weight 0:

$$f(\mathbf{0}) = \frac{\sum_{\alpha \in W_d} f(\alpha)}{P_d^{(q)}(h; n)} \tag{5}$$

if it holds

$$P_d^{(q)}(h;n) \neq 0. \tag{6}$$

Then, try to reconstruct the values of our function at the vertices with weight 1, 2, 3 etc. Use induction upon the weight of the vertices. We already have the base of induction. Then suppose that for an arbitrary $k < d$ the values $f(\alpha), \alpha \in W_0 \bigcup \ldots \bigcup W_{k-1}$ are uniquely determined under some conditions. Define all values $f(\alpha), \alpha \in W_k$ and some additional condition for determining these values.

Lemma 4 *Let $I \subseteq \{1, \ldots, n\}$, $|I| = k$, U^I be the set of all vertices with the support I and Φ^I be the vector of all $f(\alpha)$, $\alpha \in U^I$. Then*

$$\sum_{i=0}^{k} r_{i,d-k} D_i^{q-1,k} \Phi^I = \Psi^I, \tag{7}$$

where $D_i^{q-1,k}$ are incidence matrices of $(q-1)$-ary k-dimensional Hamming scheme, and the vector Ψ^I does not depend on Φ^I.

Proof Take an arbitrary vertex $\alpha \in U^I$. Using Lemmas 2 and 3 get:

$$v_{d-k}^{\overline{I}}(\alpha) = \sum_{i=0}^{d-k} r_{i,d-k} \left(\sigma_i^-(\alpha) + \sigma_i^+(\alpha)\right).$$

It means that

$$\psi(\alpha) = \sum_{i=0}^{d-k} r_{i,d-k} \sum_{\beta \in W_k \bigcap \Gamma_I(\alpha) \bigcap W_i(\alpha)} f(\beta),$$

where

$$\psi(\alpha) = v_{d-k}^{\overline{I}}(\alpha) - \sum_{i=0}^{d-k} r_{i,d-k} \sigma_i^-(\alpha).$$

The set U^I with distance associations has the structure of the $(q-1)$-ary k-dimensional Hamming scheme. So in vector form we obtain (7). □

We are ready to state the main theorem and give a sketch of proof. This theorem allows us reconstructing a λ-function into the ball by its values into the corresponding sphere under some conditions.

Theorem 5 *Let λ be an eigenvalue of $\mathbf{F}_q^n$, $d \le l(\lambda)$ and $\varphi : W_d \longrightarrow \mathbb{C}$ be a function. Suppose that f is a λ-function such that for any $\alpha \in W_d$*

$$f(\alpha) = \varphi(\alpha).$$

Then for any α with Hamming weight less than d the value $f(\alpha)$ is uniquely determined if for all $k = 0, \dots, d$ and $l = 0, \dots, k$

$$\sum_{i=0}^{k} r_{i,d-k} P_i^{(q-1)}(l;k) \neq 0, \tag{8}$$

where r_{ij} are defined as in (4).

Proof The proof is done by induction upon Hamming weight of vertices. The base of induction is given by (5) and (6).

Suppose that (8) holds and the values of f at all vertices of weight no more than $k-1$ are uniquely determined. For any k-subset I of $\{1, \dots, n\}$ write the system (7) of equations. We are interested in Φ^I and the vector Ψ^I depends only on values of f at the vertices of weight no more than $k-1$ and values of function φ. The system has the unique solution iff its matrix

$$\sum_{i=0}^{k} r_{i,d-k} D_i^{q-1,k} \tag{9}$$

has full rank. (Note that a solution exists by virtue of the hypothesis of the Theorem.) Represent incidence matrices in terms of primitive idempotents $J_l^{q-1,k}$ of Hamming association scheme:

$$D_i^{q-1,k} = \sum_{l=0}^{k} P_i^{(q-1)}(l;k) J_l^{q-1,k}$$

and substitute in (7).

The matrix (9) has full rank iff all coefficients at primitive idempotents are nonzero. These coefficients are presented in (8). □

The approach of the proof allows us representing the main Theorem by analogy with the binary case [9]. In the binary case put

$$m_d(\lambda) = -1 + \min\left\{k \le l(\lambda) \;:\; P_{d-k}^{(2)}\left(\frac{n-\lambda}{2} - k; n-2k\right) = 0\right\}.$$

Theorem 6 ([9]) *Let λ be an eigenvalue of $\mathbf{F}_2^n$, $d \le l(\lambda)$ and $\varphi : W_d \longrightarrow \mathbb{C}$ be an arbitrary function. Suppose that f is a λ-function such that for any $\alpha \in W_d$*

$$f(\alpha) = \varphi(\alpha).$$

Then for any α with Hamming weight less than $m_d(\lambda)$ (and more than $n - m_d(\lambda)$) the value $f(\alpha)$ is uniquely determined.

In general case, where $q > 2$, the main Theorem 5 gives us the following. Put

$$m_d^q(\lambda) = -1 + \min \left\{ k \le l(\lambda) \ : \ \exists l \in \{0, \dots, k\} \sum_{i=0}^{k} r_{i,d-k} P_i^{(q-1)}(l;k) = 0 \right\}.$$

Theorem 7 *Let λ be an eigenvalue of $\mathbf{F}_2^n$, $d \le l(\lambda)$ and $\varphi : W_d \longrightarrow \mathbb{C}$ be an arbitrary function. Suppose that f is a λ-function such that for any $\alpha \in W_d$*

$$f(\alpha) = \varphi(\alpha).$$

Then for any α with Hamming weight less than $m_d^q(\lambda)$ the value $f(\alpha)$ is uniquely determined.

In conclusion we can say that in this paper we develop methods to study cases $l(\lambda) \le d \le n - l(\lambda)$ and $d \ge n - l(\lambda)$ using the case $d \le l(\lambda)$ as an example. These remaining cases seem to be more sophisticated but more interesting. They include the hypotheses of the complete reconstruction of an q-ary 1-perfect code by its vertices of Hamming weight $d = h = \frac{(q-1)n+1}{q}$.

Acknowledgements The author is supported by the Grant of the Russian Scientific Fund no. 14-11-00555.

References

1. Avgustinovich, S.: On a property of perfect binary codes. Diskretn. Anal. Issled. Oper. **2**(1), 4–6 (1995)
2. Choi, S., Hyun, J., Kim, H.: Local duality theorem for q-ary 1-perfect codes. Des. Codes Cryptogr. **70**(3), 305–311 (2014). doi:10.1007/s10623-012-9683-5
3. Delsarte, P.: An algebraic approach to the association schemes of coding theory. Philips Res. Rep. Suppl. **10** (1973)
4. Heden, O.: On the reconstruction of perfect codes. Discret. Math. **256**(1–2), 479–485 (2002)
5. Hyun, J.: Local duality for equitable partitions of a hamming space. J. Combin. Theory Ser. A **119**(2), 476–482 (2012). doi:10.1016/j.jcta.2011.10.006
6. Krotov, D.: On weight distributions of perfect colorings and completely regular codes. Des. Codes Cryptogr. **61**(3), 315–329 (2011). doi:10.1007/s10623-010-9479-4
7. Vasil'eva, A.: Local spectra of perfect binary codes. Discret. Appl. Math. **135**(1–3), 301–307 (2004). doi:10.1016/S0166-218X(02)00313-X

8. Vasil’eva, A.: Local and interweight spectra of completely regular codes and of perfect colourings. Probl. Inf. Trans. **45**(2), 151–157 (2009). doi:10.1134/S0032946009020069
9. Vasil’eva, A.: Local distribution and reconstruction of hypercube eigenfunctions. Probl. Inf. Trans. **49**(1), 32–39 (2013). doi:10.1134/S0032946013010031
10. Vasil’eva, A.: Local distributions of q-ary eigenfunctions and of q-ary perfect colorings. In: Proceedings of Seventh International Workshop on Optimal Codes and Related Topics OC2013, Albena, pp. 181–186. Institute of Mathematics and Informatics, Sofia (2013)

Author Index

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series
in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5