# **International Journal of Applied Mathematics**

Volume 34 No. 3 2021, 485-506

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

doi: http://dx.doi.org/10.12732/ijam.v34i3.5

# ON THE CONSTRUCTION OF PERFECT CODES FOR n-DIMENSIONAL INTERLEAVING

Cibele Cristina Trinca<sup>1</sup>§, Reginaldo Palazzo Junior<sup>2</sup>

<sup>1</sup>Federal University of Tocantins (UFT)

Department of Biotechnology

and Bioprocess Engineering

Rua Badejós, Lote 7, Chácaras 69/72

Gurupi-TO - 77402-970, BRAZIL

<sup>2</sup>State University of Campinas (UNICAMP)

Department of Telematics

Av. Albert Einstein, 400

Cidade Universitária "Zeferino Vaz"

Campinas-SP - 13081-970, BRAZIL

**Abstract:** The authors propose an n-dimensional interleaving technique which spreads a cluster of errors having a quasicircular shape. Consequently simple one-dimensional random-error-correcting codes can be used to correct this kind of cluster instead of the more complex n-dimensional burst-error-correcting codes. Moreover let  $p \geq 2$  be a positive integer, whenever  $n \neq 1 \mod 3$ , the corresponding n-dimensional interleaving technique provides a perfect code. Also, whenever  $n = 3p - 2 = 1 \mod 3$ , the corresponding n-dimensional interleaving technique provides neither a perfect code nor a quasiperfect code.

AMS Subject Classification: 94B20

Key Words: perfect code; interleaved coding; burst correction codes

#### 1. Introduction

In [1] the authors propose an efficient two-dimensional interleaving technique

Received: October 2, 2020

© 2021 Academic Publications

<sup>§</sup>Correspondence author

that spreads a cluster of errors having a quasicircular shape.

Thenceforth this paper aims to extend the previous work by proposing an n-dimensional interleaving technique that spreads a cluster of errors having a quasicircular shape. Interleaving techniques are used in channels with memory to cope with burst errors. The combination of a (q, k, t) block code and the interleaving results in an equivalent  $(q^n, kq^{n-1}, t_l)$  interleaved block code, where  $n \geq 3$  is the dimension, q = 2n + 1 is the blocklength (order of the hypertorus which is  $q \times q \times \ldots \times q = q^n$ ), k is the number of information bits, t is the error-correcting code capability and  $t_l$  is the interleaved error-correcting code capability. Note that q = 2n + 1 may be interpreted as the cardinality of the set consisting of bi-orthogonal signal points (two antipodes signals per dimension including the signal at the origin), where each signal point is viewed as a square in 2D, a cube in 3D and a hypercube in nD. This signal set is the Lee sphere of radius one. So far effective interleaving techniques are well known for combating a one and two-dimensional burst of errors as is the case in data transmission systems [2], [6], [7]. However to the best of our knowledge little is known regarding interleaving techniques for combating n-dimensional cluster of errors, where n > 3.

The works [1] and [6] are based on the set partitioning concept introduced by Ungerboeck [11], however, this work is based on Golomb's et. al. congruence equation (equation (5)). As it will become clear both the set partitioning concept (Table 3) and Golomb's et al. congruence equation (equation (5)) are distinct techniques leading to the same solution. For lower dimensions either one of them may be used indistinctly. However, for higher dimensions, the latter one is less complex.

Thereby the purpose of this work is to propose an n-dimensional interleaving technique for spreading quasicircular clusters of errored bits. Consequently one-dimensional random-error-correcting codes can be used to correct the spread errors instead of the more complex n-dimensional burst-error-correcting codes.

The motivation to use this technique is related to the following applications, among others, such as magnetic or optical data storage, where clusters of errors can occur due to dust particles or defective regions, and in the construction of quantum computers to overcome decoherence, a phenomenon due to the interaction between the system and the surrounding environment, by use of quantum error-correcting codes such as the class of toric quantum codes,  $T^2$  quantum codes [8], [9] associated with the lattice  $\mathbb{Z}^2$  or the class of  $T^n$  quantum codes associated with the lattice  $\mathbb{Z}^n$ . We assume that the clusters of errors have a quasicircular shape (hypercubes) and the bits are cubic-shaped unit areas (hyperfaces).

Figure 1 shows the model of the n-dimensional storage system under consideration.

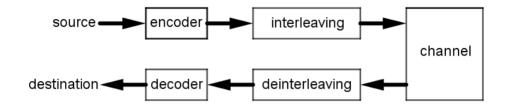


Figure 1: Model of an n-dimensional storage system, from [1]

Before presenting the n-dimensional interleaving technique, we consider the corresponding three, four, and five-dimensional interleaving technique.

#### 2. Lattices

A large class of the problems in coding theory is related to the properties of lattices [3, 4, 10, 17, 18, 19]. A lattice is simply an array of vectors (points) that algebraically forms a group under ordinary vector addition. This property leads to the study of subgroups (sublattices) and coset decomposition (partitions). An algebraic way to obtain sublattices from lattices is employing a scaled transformation matrix T. Given a lattice  $\Lambda$ , a sublattice  $\Lambda' = T\Lambda$  can be obtained by transforming each vector  $\lambda \in \Lambda$  to  $\lambda' \in \Lambda'$  according to  $\lambda' = T\lambda$ .

Let  $V(\Lambda)$  be the volume of the Voronoi region of the n-dimensional lattice  $\Lambda$ . For a sublattice  $\Lambda' = T\Lambda$ , we have that  $\frac{V(T\Lambda)}{V(\Lambda)} = |det(T)|$ .

An essential property of a lattice is the (squared) minimum distance between its points. The lattice we consider in this paper is the  $\mathbb{Z}^n$ . Although not as important as the (squared) minimum distance for lattice comparisons, however, relevant for the possible interferences is the number of nearest neighbors of any given lattice point. Since a lattice consists of a collection of an infinite number of points and from the fact that in applications only a finite number of points are of interest, we define a basic cell of order q as an array of  $q \times q \times \cdots \times q = q^n$  points based on the  $\mathbb{Z}^n$  lattice.

#### 3. Binary quadratic forms

A polynomial equation with integer coefficients is called a *Diophantine equation* if the solutions are also integers. For instance, consider the polynomial equation

$$A(x,y) = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = ax^2 + (c+b)xy + dy^2, \tag{1}$$

with a, b, c and d integers associated with a given lattice  $\Lambda$ . Let  $V(\Lambda)$  be the volume of the Voronoi region of the lattice  $\Lambda$  associated with A(x,y), let  $V(T\Lambda)$  be the volume of the Voronoi region of the sublattice  $T\Lambda$  and let q be the number of cosets. The corresponding Diophantine equation can be written as

$$ax^{2} + (c+b)xy + dy^{2} = \frac{V(T\Lambda)}{V(\Lambda)} = |det(T)| = q.$$
(2)

From (2) we have the following case of interest:

• a = d = 1, b = c = 0. In this case we have the binary quadratic form  $x^2 + y^2 = q$  associated with the q-ary partitioning of the lattice  $\mathbb{Z}^2$ .

The mathematical model of the first level q-ary set partitioning problem for the rectangular grid is to consider

$$ax^{2} + dy^{2} = \frac{V(T\mathbb{Z}^{2})}{V(\mathbb{Z}^{2})} = |det(T)| = q,$$
 (3)

in words, knowing the number of cosets, find T.

To find T is equivalent to solving (3) for x and y integers. The existence of solutions to (3) is guaranteed by the following pair of classical theorems, see [5]:

**Theorem 1.** (Genus) The equation  $x^2 + y^2 = q$  can be solved for x and y integers with q prime if, and only if,  $q = 1 \mod 4$  or q = 2.

**Theorem 2.** (Composition) Let q be factorable by the product  $q_1q_2$  and let  $A(x',y')=q_1$  and  $A(x'',y'')=q_2$ . Then A(x,y)=q can be obtained by using

$$A(x',y')A(x'',y'') = A(x'x'' - y'y'', x'y'' + x''y').$$
(4)

When q can be written as  $x^2 + y^2$ , we say that q is representable.

## 4. Bi-dimensional interleaving

The set partitioning technique is used, in this case, to separate bits of a cluster of errors in a  $5 \times 5$  torus, [1]. From Theorem 1 the binary quadratic form is  $x^2 + y^2 = 5$  and the solutions are  $(\pm 2, \pm 1)$  and  $(\pm 1, \pm 2)$ . Table 3 illustrates the transformations leading to the  $5 \times 5$  array, as it is shown in Table 1, and Table 2 the code originated from them.

Table 1: Codewords of the  $5 \times 5$  array, see Table 3

	0	1	2	3	4
0	X				
1			X		
2					X
3		X			
3				X	

Table 2: Codewords of the  $5 \times 5$  array

	Codes	
Column		Row
00		00
13		12
21		24
34		31
42		43

## 5. Three-dimensional interleaving

In this section, the goal is to separate bits of a cluster of errors in a  $7 \times 7 \times 7$  hypertorus.

An n-space signal point has 2n neighbors signal points within a Lee distance one from it. Geometrically we may visualize a Lee sphere of radius 1 in n dimensions as a central hypercube which has 2n hyperfaces to which another

hypercube has been affixed to each of its hyperfaces. The three-dimensional Lee sphere of radius one consists of hepta cubes, as shown in Fig. 2.

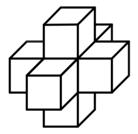


Figure 2: 3-dimensional Lee sphere of radius 1, from [12]

The following theorem establishes a single-error-correcting code in dimension three having 49 codewords.

**Theorem 3.** ([12], p. 306) 49 of the heptacubes shown in Fig. 2 can be used to close-pack the  $7 \times 7 \times 7$  hypertorus.

*Proof.* Specifically we look at a typical  $7 \times 7$  cross section of the solution, shown in Fig. 3:

The cross-sections of the heptacube will be either X-pentominoes or single squares. In the cross-section shown in Fig. 3 we see seven X-pentominoes and seven squares labeled A and seven squares labeled B. The A's are bottoms of heptacubes whose centers are in the plane *above* and the B's are tops of heptacubes protruding upward from the plane *below*. Since the seven A's are systematically translated (1 unit to the northwest) from the X-pentominoes centers, we are assured that in the next cross-section above the one we are examining, the X-pentomino sections fit together correctly. Similarly, the seven B's are systematically translated (1 unit to the southeast) from the seven X-pentominoes' centers and are, therefore, consistent as tops of heptacubes from the layer below. Finally, since seven is a prime, it is easy to see that these translations must lead to a periodicity of 7 in the third dimension.

An even more general result holds. Basically, it asserts that close-packed single error-correcting codes for the Lee metric exist in n dimensions, for all n, as it follows:

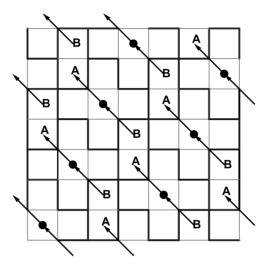


Figure 3: A cross section of the close-packed  $7 \times 7 \times 7$  hypertorus, from [12]

**Theorem 4.** ([12], p. 307) In n dimensions, the Lee spheres of radius 1 can be used to close-pack the hypertorus which is  $q \times q \times q \times \ldots \times q = q^n$ , where q = 2n + 1.

In the proof of Theorem 4 the set S of all points  $(a_1, a_2, \ldots, a_n)$  of the hypertorus, as centers of the spheres, satisfying

$$\sum_{i=1}^{n} ia_i \equiv 0 \pmod{2n+1} \tag{5}$$

are collected.

Besides the number of solutions to this congruence is clearly  $q^{n-1}$ , since any choice of  $a_2, a_3, \ldots, a_n$  may be made and then there is a unique value of  $a_1$  modulo q to satisfy the congruence.

Therefore by using equation (5) we can construct a single-error-correcting code in dimension three with 49 codewords. Table 5 shows the corresponding codewords.

Since the close-packed  $7 \times 7 \times 7$  hypertorus has a finite number of bits, we use operations modulo q = 2n + 1 = 7 (n = 3) to guarantee that the points of a given coset remain inside the hypertorus ([12], Theorem 3).

The  $7 \times 7 \times 7$  hypertorus consists of seven  $7 \times 7$  arrays, where each square

of these arrays means a cube. In Table 5 these seven  $7 \times 7$  arrays, which are the seven  $7 \times 7$  cross-sections, are labeled by the numbers 0, 1, 2, 3, 4, 5 and 6. Each column of this table shows the corresponding codewords of each  $7 \times 7$  cross-section and these codewords are featured by equation (5).

Next we explain how to separate 49 adjacent bits of the  $7 \times 7 \times 7$  hypertorus arranged in a  $7 \times 7$  array by the greatest distance.

By observing Table 4 the codewords of each  $7 \times 7$  array can be generated by the vector (0 1 4), that is, we can put them apart by one unit to the right in the horizontal direction and four units down in the vertical one. Now to rise in the third dimension to change from one array to the one above to continue the construction of the corresponding codewords we use the vector (1 0 2) as the corresponding generator. Thus, by using these vectors, we can construct the 49 codewords shown in Table 5.

Each codeword of the single-error-correcting code has the heptacube as being its Voronoi region. Consequently by using the Voronoi region and knowing how the codewords are constructed we can rearrange the  $343=49\times7$  total bits.

Thereby the first 49 adjacent bits are arranged as being the 49 codewords of the hypertorus. Now the other six blocks with 49 adjacent bits are arranged analogously. So suppose that we wish to separate another block with 49 adjacent bits in the hypertorus. As the Voronoi region has six cubes around the cube related to the codeword, we start arranging the first bit of the 49 bits in one of these six cubes related to the codeword  $(0\ 0\ 0)$ . Therefore if we use the vector  $(0\ 1\ 4)$  in each  $7\times 7$  array to arrange the plane and rise in the third dimension by using the vector  $(1\ 0\ 2)$  to change from one array to the one above we can distribute these 49 adjacent bits in the hypertorus, where each one of them belongs to a different Voronoi region and the same cube of each Voronoi region.

Therefrom, by using this distribution method, we can cyclically arrange each one of the seven blocks with 49 adjacent bits in the  $7 \times 7 \times 7$  hypertorus.

In Table 3 it is shown the solutions related to the Diophantine equation (3) and Theorem 1.

Then in this section we present the 3D-interleaving. In [12] it is conjectured that this is the only case for which a close-packing exists in dimension three.

We have that the codewords of each  $7 \times 7$  array are generated by the vector (0 1 4), that is, we can put them apart by one unit to the right in the horizontal direction and four units down in the vertical one, see column 0 in Table 5 and the transformation T for q = 7 in Table 3. Then we can observe that each  $7 \times 7$  array satisfies the rook domain property [13]. Table 4 provides us the codewords that

Table 3: The greatest minimum distance from the solution of equation (3)

q	$d_{min}^2$	Transformation $T$
3	2	$\left( egin{array}{cc} 2 & 1 \\ 1 & 2 \end{array}  ight)$
5	15	$\left(\begin{array}{cc}2&1\\1&3\end{array}\right) \left(\begin{array}{cc}2&1\\-1&2\end{array}\right)$
7	15	$\left(\begin{array}{cc}2&1\\1&4\end{array}\right) \left(\begin{array}{cc}3&2\\1&3\end{array}\right)$
9	9	$\left(\begin{array}{cc} 3 & 0 \\ 1 & 3 \end{array}\right)  \left(\begin{array}{cc} 1 & 6 \\ -1 & 3 \end{array}\right)$
11	10	$\left(\begin{array}{cc} 3 & 2 \\ -1 & 3 \end{array}\right)  \left(\begin{array}{cc} 4 & 1 \\ 1 & 3 \end{array}\right)$
13	13	$\begin{pmatrix} 5 & 2 \\ 1 & 3 \end{pmatrix}$
15	17	$\begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$

present the first digit as being 0, where X indicates a corresponding codeword, and we can check that no codeword from this  $7 \times 7$  array meet another in the horizontal direction.

Table 4: Codewords of the  $7 \times 7$  array labeled 0, see Table 3

	0	1	2	3	4	5	6
0	Χ						
1			X				
2					X		
3							X
4		X					
5				Χ			
6						X	

Besides to rise in the third dimension to change from one array to the one above to continue constructing the corresponding codewords we must use the

			ARRAYS			
"0"	"1"	"2"	"3"	"4"	"5"	"6"
000	102	204	306	401	503	605
014	116	211	313	415	510	612
021	123	225	320	422	524	626
035	130	232	334	436	531	633
042	144	246	341	443	545	640
056	151	253	355	450	552	654
063	165	260	362	464	566	661

Table 5: Arrays, see Table 4

vector (1 0 2) as the corresponding generator. By [1] we can observe that this vector also provides us the rook domain property [13] to the vertical direction because no codeword meets another in this direction.

Thereby we can conclude that the corresponding 3D-interleaving satisfies the rook-domain property [13].

Consequently one-dimensional single-error-correcting codes can be used to correct the spread errors instead of the more complex 3-dimensional burst-error-correcting codes. Therefore, in this case, we have a perfect code [12]. Other approaches on the construction of perfect codes can be found in [15, 16].

### 6. Four-dimensional interleaving

In this section the congruence equation (5) is used to separate bits of a cluster of errors in a  $9 \times 9 \times 9 \times 9$  hypertorus.

A point in the 4-space has 8 other points within a Lee distance 1. Geometrically we may visualize a Lee sphere of radius 1 in 4 dimensions as a central hypercube which has 8 hyperfaces to which another hypercube has been affixed to each of its hyperfaces.

Since the close-packed  $9 \times 9 \times 9 \times 9$  hypertorus has a finite number of bits, we use operations modulo q=2n+1=9 (n=4) to guarantee that the points of a given coset remain inside the hypertorus ([12], Theorem 3). Thus by using equation (5) we develop an algorithm that by straightforward computation lists the 729 codewords.

The  $9 \times 9 \times 9 \times 9$  hypertorus consists of nine  $9 \times 9 \times 9$  hypertorus, where

each square of these hypertorus means a cube. These nine  $9 \times 9 \times 9$  hypertorus which are  $9 \times 9 \times 9$  cross-sections are labeled by the numbers 0, 1, 2, 3, 4, 5, 6, 7 and 8. The cross-section labeled j, where j = 0, 1, 2, 3, 4, 5, 6, 7 and 8, provides the 81 codewords that present the first digit as being j. As we already know these codewords are featured by equation (5).

Next we explain how to separate 729 adjacent bits of the  $9 \times 9 \times 9 \times 9$  hypertorus arranged in a  $9 \times 9 \times 9$  hypertorus by the most significant distance.

By observing the codewords that were found by the algorithm mentioned previously, we can see that the vectors (0 0 1 6), (0 1 1 1) and (1 0 0 2) can generate all the codewords as it follows: we start with the codeword (0 0 0 0) and, by using these three vectors, we can generate all the other codewords; we sum the vector (0 0 1 6) eight times to generate nine codewords that present the first digit as being 0, if we sum this vector nine times, we obtain the null vector. Now we sum the ninth codeword with the vector (0 1 1 1) to find the tenth one and, after that, we sum again the vector (0 0 1 6) eight times to generate more nine codewords that present the first digit as being 0. Thus we continue this procedure until we obtain 81 codewords. Whenever we sum a codeword with the vector (0 1 1 1), the second digit of the new codeword is changed and its value is the second digit of the codeword plus 1. Then we apply the generator (0 0 1 6) to generate nine codewords for each new second digit. Hence, for each first digit j, the second digit changes nine times and, when we find the last codeword related to the first digit j, if we sum it with the vector (0 1 1 1), we obtain the null vector. Now we need to obtain the first codeword that presents the first digit as being 1. Therefore, for that, we sum the vector (1 0 0 2) with the codeword (0 0 0 0) which is the first codeword that presents the first digit as being 0. After that we obtain the codeword (1 0 0 2) and we perform the same procedure related to the all-zero codeword to find the 81 codewords which present the first digit as being one. Consequently we use this systematic way to find all the other codewords. We can observe that if we sum the first codeword that presents the first digit as being eight with the corresponding vector (1 0 0 2), we also obtain the null vector. We must observe the corresponding cyclicalities because all the codewords are featured by equation (5) which uses operations modulo q = 9.

Hence to generate all the codewords of a cross-section we must use the vectors  $(0\ 0\ 1\ 6)$  and  $(0\ 1\ 1\ 1)$ . Now to rise in the fourth dimension to change from one cross-section to the other above to continue the construction of the corresponding codewords, we must use the vector  $(1\ 0\ 0\ 2)$  as the corresponding generator. So by using these vectors we can construct the 729 codewords.

Each codeword of the single-error-correcting code has a hypercube as being

its Voronoi region. Then by using the Voronoi region and knowing how the codewords can be constructed we can rearrange the  $6561 = 729 \times 9$  total bits.

Thereby the first 729 adjacent bits are arranged as being the 729 codewords of the  $9 \times 9 \times 9 \times 9$  hypertorus. Now the other eight blocks with 729 adjacent bits are arranged analogously. So suppose that we wish to separate another block with 729 adjacent bits in the hypertorus. As the Voronoi region has eight cubes around the hypercube related to the codeword, we start the arrangement putting the first bit of the 729 bits in one of these eight cubes around the codeword (0 0 0 0). Thus if we use the vectors (0 0 1 6) and (0 1 1 1) in each  $9 \times 9 \times 9$  cross-section to make the arrangement in it and rise in the fourth dimension by using the vector (1 0 0 2) to change from one cross-section to the other above, we can distribute these 729 adjacent bits in the hypertorus where each one of them belongs to a different Voronoi region and the same cube of each Voronoi region.

Hence by using this method of distribution we can cyclically arrange each one of the nine  $9 \times 9 \times 9$  cross-sections with 729 adjacent bits in the  $9 \times 9 \times 9 \times 9$  hypertorus.

So in this section we present the four-dimensional interleaving. In [12] it is conjectured that this is the only case for which a close-packing exists in dimension four.

We have that the codewords of each  $9 \times 9 \times 9$  cross-section are generated by the vector (0 0 1 6), that is, we can put them apart by 1 unit to the right in the horizontal direction and six units down in the vertical one. Thus Table 6 provides us the first nine codewords that present the first digit as being 0, where X indicates a corresponding codeword, and we can check that three codewords from this  $9 \times 9$  array meet each other in the horizontal direction. Then we can observe that each  $9 \times 9 \times 9$  cross-section does not satisfy the rook domain [13] property.

Besides to rise in the fourth dimension to change from one cross-section to the other above to continue constructing the corresponding codewords, we must use the vector  $(1\ 0\ 0\ 2)$  as the corresponding generator. By [1] we can observe that this vector provides us the rook domain property [13] to the "vertical direction" (fourth dimension) because no codeword meets another in this direction.

So as the vector (0 0 1 6) does not yield us the rook domain [13] property, then we can conclude that the corresponding 4D-interleaving does not satisfy the rook-domain property [13].

Consequently one-dimensional random-error-correcting codes that correct three errors can be used to correct the spread errors instead of

	0	1	2	3	4	5	6	7	8
0	X			X			X		
1									
2									
3			Χ			X			X
4									
5									
6		X			X			X	
7									
8									

Table 6: Codewords of the  $9 \times 9$  array labeled 0, see Table 3

the more complex 4-dimensional burst-error-correcting codes. Therefore, in this case, we have neither a perfect code [12] nor a quasi-perfect code [14].

# 7. Five-dimensional interleaving

In this section bits of a cluster of errors are separated in an  $11 \times 11 \times 11 \times 11 \times 11$  hypertorus.

A point in the 5-space has 10 other points within a Lee distance 1. Geometrically we may visualize a Lee sphere of radius 1 in 5 dimensions as a central hypercube which has 10 hyperfaces to which another hypercube has been affixed to each of its hyperfaces.

Using equation (5) of the Theorem 4, we can construct a single-error-correcting code in dimension five with 14641 codewords.

Since the close-packed  $11 \times 11 \times 11 \times 11$  hypertorus has a finite number of bits, we use operations modulo q = 2n + 1 = 11 (n = 5) to guarantee that the points of a given coset remain inside the hypertorus ([12], Theorem 3). Thus by using equation (5) we develop an algorithm that by straightforward computation shows us the 14641 codewords.

The  $11 \times 11 \times 11 \times 11 \times 11$  hypertorus consists of eleven  $11 \times 11 \times 11 \times 11$  hypertorus where each square of these hypertorus means a cube. These eleven  $11 \times 11 \times 11 \times 11$  hypertorus which are  $11 \times 11 \times 11 \times 11$  cross sections are labeled by the numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. The cross section labeled j, where j = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10, provides the 1331 codewords that

present the first digit as being j. As we already know these codewords are featured by equation (5).

Next we explain how to separate 14641 adjacent bits of the  $11 \times 11 \times 11 \times 11 \times 11 \times 11 \times 11$  hypertorus arranged in an  $11 \times 11 \times 11 \times 11$  hypertorus by the most significant distance.

By observing the codewords that were found by the algorithm mentioned previously, we can see that all the codewords can be generated by the vectors  $(0\ 0\ 0\ 1\ 8),\ (0\ 0\ 1\ 1\ 3),\ (0\ 1\ 1\ 1\ 7)$  and  $(1\ 0\ 0\ 0\ 2)$  as it follows: we start with the codeword  $(0\ 0\ 0\ 0\ 0)$  and by using these four vectors we are able to generate all the other codewords; we sum the vector  $(0\ 0\ 0\ 1\ 8)$  ten times to generate 11 codewords that present the first digit as being 0, if we sum this vector eleven times we obtain again the null vector. Now we sum the eleventh codeword with the vector  $(0\ 0\ 1\ 1\ 3)$  to generate the twelfth one and, after that, we sum again the vector  $(0\ 0\ 1\ 8)$  ten times to generate more 11 codewords that present the first digit as being 0. The twelfth codeword has the third digit as being 1. Thus we continue this procedure until we obtain 121 codewords. Whenever we sum a codeword with the vector  $(0\ 0\ 1\ 1\ 3)$ , the third digit of the new codeword is changed and its value is the third digit of the codeword plus 1. Then for each new third digit we apply the generator  $(0\ 0\ 1\ 8)$  to generate 11 codewords.

Hence for each first digit j the third digit changes eleven times and when we find the last codeword related to the third digit, if we sum it with the vector  $(0\ 0\ 1\ 1\ 3)$ , we obtain the null vector.

After finding these 121 codewords we must sum the last codeword with the vector  $(0\ 1\ 1\ 1\ 7)$  to change the second digit from the value 0 to 1. For each new second digit we must apply the previous procedure to obtain more 121 codewords again. For each first digit j the second digit also changes eleven times and when we find the last codeword related to the second digit, if we sum it with the vector  $(0\ 1\ 1\ 1\ 7)$ , we obtain the null vector. So we can observe that we have the 1331 codewords of each cross-section which is labeled j.

Now we need to obtain the first codeword that presents the first digit as being 1. Therefore, for that, we sum the vector  $(1\ 0\ 0\ 0\ 2)$  with the codeword  $(0\ 0\ 0\ 0)$  which is the first codeword that presents the first digit as being 0. After that we obtain the codeword  $(1\ 0\ 0\ 0\ 2)$  and we perform the same procedure related to the all-zero codeword to find the 1331 codewords which present the first digit as being 1. Thus we use this systematic way to find all the other codewords. We can observe that if we sum the first codeword that presents the first digit as being 10 with the corresponding vector  $(1\ 0\ 0\ 0\ 2)$ , we also obtain the null vector. We must observe the corresponding cyclicalities because all the codewords are featured by equation (5) which uses operations

modulo q = 11.

Hence to generate all the codewords of a cross section we must use the vectors  $(0\ 0\ 0\ 1\ 8)$ ,  $(0\ 0\ 1\ 1\ 3)$  and  $(0\ 1\ 1\ 1\ 7)$ . Now to rise in the fifth dimension to change from one cross section to the other above to continue the construction of the corresponding codewords, we must use the vector  $(1\ 0\ 0\ 0\ 2)$  as the corresponding generator. So by using these vectors we can construct the 14641 codewords.

Each codeword of the single-error-correcting code has a hypercube as being its Voronoi region. Then by using the Voronoi region and knowing how the codewords can be constructed we can rearrange the  $161051=14641\times11$  total bits.

Thereby the first 14641 adjacent bits are arranged as being the 14641 codewords of the  $11 \times 11 \times 11 \times 11 \times 11$  hypertorus. Now the other ten blocks with 14641 adjacent bits are arranged analogously. So suppose that we wish to separate another block with 14641 adjacent bits in the hypertorus. As the Voronoi region has ten cubes around the hypercube related to the codeword, we start the arrangement putting the first bit of the 14641 bits in one of these ten cubes around the codeword (0 0 0 0). Thus if we use the vectors (0 0 0 1 8), (0 0 1 1 3) and (0 1 1 1 7) in each  $11 \times 11 \times 11 \times 11$  cross-section to make the arrangement in it and rise in the fifth dimension by using the vector (1 0 0 0 2) to change from one cross-section to the other above, we can distribute these 14641 adjacent bits in the hypertorus where each one of them belongs to a different Voronoi region and to the same cube of each Voronoi region.

Then by using this method of distribution we can cyclically arrange each one of the eleven  $11 \times 11 \times 11 \times 11$  cross-sections with 14641 adjacent bits in the  $11 \times 11 \times 11 \times 11 \times 11$  hypertorus.

So in this section we present the five-dimensional interleaving. In [12] it is conjectured that this is the only case for which a close-packing exists in dimension five.

We have that the codewords of each  $11 \times 11 \times 11 \times 11$  cross-section are generated by the vector (0 0 0 1 8), that is, we can put them apart by 1 unit to the right in the horizontal direction and eight units down in the vertical one. Thus Table 7 provides us the first 11 codewords that present the first digit as being 0, where X indicates a corresponding codeword, and we can check that no codeword from this  $11 \times 11$  array meet another in the horizontal direction. Then we can observe that each  $11 \times 11 \times 11 \times 11$  cross-section satisfies the rook domain [13] property.

Besides to rise in the fifth dimension to change from one cross-section to the other above to continue constructing the corresponding codewords, we must use

	0	1	2	3	4	5	6	7	8	9	10
0	Χ										
1								X			
2				X							
3											X
4							X				
5			X								
6										X	
7						X					
8		X									
9									Χ		
10					Χ						

Table 7: Codewords of the  $11 \times 11$  array labeled 0, see Table 3

the vector (1 0 0 0 2) as the corresponding generator. By [1] we can observe that this vector provides us the rook domain property [13] to the "vertical direction" (fifth dimension) because no codeword meets another in this direction.

So as the vector  $(0\ 0\ 0\ 1\ 8)$  yields us the rook domain property [13], then we can conclude that the corresponding 5D-interleaving satisfies the rook-domain property [13].

Consequently one-dimensional single-error-correcting codes can be used to correct the spread errors instead of the more complex 5-dimensional burst-error-correcting codes. So, in this case, we have a perfect code [12].

Table 8 summarizes the cases considered previously.

#### 8. The *n*-dimensional interleaving

The congruence equation (5), in this case, is used to separate bits of a cluster of errors in a  $q^n = q \times q \times \ldots \times q$  (n times) hypertorus, where q = 2n + 1.

A point in the *n*-space has q-1=2n other points within a Lee distance 1 of it. Geometrically we may visualize a Lee sphere of radius 1 in *n* dimensions as a central hypercube which has q-1=2n hyperfaces to which another hypercube has been affixed to each of its hyperfaces.

Since the close-packed  $q^n = q \times q \times ... \times q$  (n times) hypertorus has a finite number of bits, we use operations modulo q = 2n + 1 to guarantee that the

Table 8: Summary

q	Generator	Adjustments
5	12	None
7	014	Add 102 when changing column
9	0016	Add 0111 when changing 2nd digit
		Add 1002 when changing column
11	00018	Add 01117 when changing 2nd digit
		Add 01113 when changing 3rd digit
		Add 10002 when changing column
13	00001 <b>10</b>	Add 011112 when changing 2nd digit
		Add 0011111 when changing 3rd digit
		Add 000115 when changing 4th digit
		Add 100002 when changing column
15	000001 <b>12</b>	Add 01111110 when changing 2nd digit
		Add 0011116 when changing 3rd digit
		Add 0001110 when changing 4th digit
		Add 0000117 when changing 5th digit
		Add 1000002 when changing column
17	000000114	Add 01111113 when changing 2nd digit
		Add 001111116 when changing 3rd digit
		Add 000111110 when changing 4th digit
		Add 00001112 when changing 5th digit
		Add 00000119 when changing 6th digit
		Add 10000002 when changing column

points of a given coset remain inside the hypertorus ([12], Theorem 3).

The  $q^n = q \times q \times \ldots \times q$  (n times) hypertorus consists of q  $q^{n-1} = q \times q \times \ldots \times q$  (n-1 times) hypertorus, where each square of these hypertorus means

a cube. These q  $q^{n-1} = q \times q \times \ldots \times q$  (n-1 times) hypertorus which are  $q^{n-1} = q \times q \times \ldots \times q$  (n-1 times) cross sections are labeled by the numbers  $0, 1, 2, \ldots, q-1$ . The cross section labeled j, where  $j = 0, 1, 2, \ldots, q-1$ , provides the  $q^{n-2}$  codewords that present the first digit as being j. As we already know these codewords are featured by equation (5).

From now on we explain how to separate  $q^{n-1}$  adjacent bits of the  $q^n = q \times q \times \ldots \times q$  (n times) hypertorus arranged in a  $q^{n-1} = q \times q \times \ldots \times q$  (n - 1 times) hypertorus by the greatest distance.

The codewords of each  $q^{n-1} = q \times q \times q \times \ldots \times q$  (n-1 times) cross section are generated by the vector v that has the following features:

- $\diamond v$  has n coordinates;
- $\diamond$  The number n + (n-2) belongs to the last coordinate of the vector v;
- ♦ The coordinate previous to the last one is fixed by the number 1;
- $\diamond$  The other coordinates which are the first (n-2) ones are filled by the number zero.

We restrict the algorithm developed to generate only the codewords whose first digit is equal to zero. Thus by using equation (6) we have the vector  $v_{n-i}$  which changes the (n-i)-th digit and, after applying the generator v, we have  $q^i$  codewords that present the first digit as being zero.

For i = 2, ..., (n-3), (n-2), we have

$$v_{n-i} = v_{(q^{(i-1)}+1)} - v_{q^{(i-1)}}, (6)$$

where  $v_{(q^{(i-1)}+1)}$  and  $v_{q^{(i-1)}}$  denote the  $(q^{(i-1)}+1)$  and  $q^{(i-1)}$ -th codewords whose first digit is equal to zero, respectively.

Therefore, for each  $j, j = 0, 1, 2, \ldots, q-1$ , whenever we sum the corresponding codeword with the vector  $v_{n-i}$ , the (n-i)-th digit of the new codeword is changed and its value is the (n-i)-th digit of the correspondent codeword plus 1. Then for each new (n-i)-th digit we apply the generator v to obtain a total of  $q^i$  codewords.

So the sequence of vectors used to change the digits in each cross section j is the following:

$$v_{(n-2)}, v_{(n-3)}, \dots, v_{(n-(n-2))} = v_2.$$
 (7)

Hence, for each j,  $j=0,1,2,\ldots,q-1$ , we apply the generator v and to change the digits  $(n-i)=n-2,n-3,\ldots,2$  we apply the vectors  $v_{n-i}$   $(i=2,3,\ldots,n-2)$ , respectively, and after each changing we apply the generator v to obtain a total of  $q^i$  codewords. Thus we obtain  $q^{n-2}$  codewords in each cross section labeled j.

Then we have that the vectors v and  $v_{n-i}$ , where  $i=2,3,4,\ldots,n-2$ , can generate all the codewords of a cross-section j and we start the procedure with the all-zero codeword.

Now to rise in the *n*-th dimension to change from one cross section to the other above to continue the construction of the corresponding codewords, we must use the vector  $(1\ 0\ 0\ \cdots\ 0\ 2)$  as the corresponding generator. So by using the vectors  $v, v_{n-i}$   $(i=2,3,4,\ldots,n-2)$  and  $(1\ 0\ 0\ \cdots\ 0\ 2)$  we can construct all the  $q^{n-1}$  codewords.

Each codeword of the single-error-correcting code has a hypercube as being its Voronoi region. Then by using the Voronoi region and knowing how the codewords can be constructed we can rearrange the  $q^n$  total bits.

Thereby the first  $q^{n-1}$  adjacent bits are arranged as being the  $q^{n-1}$  codewords of the  $q^n = q \times q \times \ldots \times q$  (n times) hypertorus. Now the other q-1 blocks with  $q^{n-1}$  adjacent bits are arranged analogously. So suppose that we wish to separate another block with  $q^{n-1}$  adjacent bits in the hypertorus. As the Voronoi region has q-1=2n cubes around the hypercube related to the codeword, then we start the arrangement putting the first bit of the  $q^{n-1}$  bits in one of these q-1=2n cubes related to the codeword ( $0 \times 0$ ). Thus if we use the vectors v and  $v_{n-i}$  ( $i=2,3,\ldots,n-2$ ) in each  $q^{n-1}=q\times q\times \ldots\times q$  (n-1 times) cross-section to make the arrangement in it and rise in the n-th dimension by using the vector ( $1 \times 0$  2) to change from one cross-section to the other above, we can distribute these  $q^{n-1}$  adjacent bits in the hypertorus where each one of them belongs to a different Voronoi region and to the same cube of each Voronoi region.

Then by using this distribution method we can cyclically arrange each one of the cross-sections with  $q^{n-1}$  adjacent bits in the  $q^n$  hypertorus.

So in this section we present the n-dimensional interleaving. In [12] it is conjectured that this is the only case for which a close-packing exists in dimension n.

We have that the codewords of each  $q^{n-1} = q \times q \times ... \times q$  (n-1 times) cross-section are generated by the vector v, that is, we can put them apart by 1 unit to the right in the horizontal direction and (n + (n-2)) units down in the vertical one.

Let  $p \geq 2$  be a positive integer. Consequently if  $n \neq 1 \mod 3$ , then the rook domain property [13] is satisfied because (n+(n-2)) is not a multiple of 3, that is,  $2n-2 \neq 0 \mod 3$ , and so no codeword from this  $q \times q$  array meet another codeword in the horizontal direction. However if  $n=3p-2=1 \mod 3$ , then (n+(n-2)) is a multiple of 3 and at least one codeword from this  $q \times q$  array meet another codeword in the horizontal direction. Then in

this case each  $q^{n-1} = q \times q \times \ldots \times q$  (n-1 times) cross section does not satisfy the rook domain property [13]. We have these features by observing that (2n+1)-(n+(n-2))=3, where q=2n+1, and the codewords of each cross section are generated by the vector v, that is, we can put them apart by 1 unit to the right in the horizontal direction and (n+(n-2)) units down in the vertical one.

Besides to rise in the n-th dimension to change from one cross-section to the one above to continue the construction of the corresponding codewords, we must use the vector  $(1\ 0\ \cdots\ 0\ 2)$  as the corresponding generator. By [1] we can observe that this vector provides the rook domain property [13] to the "vertical direction" (n-th dimension) because no codeword meets another codeword in this direction.

So, for (n+(n-2)) not a multiple of 3, the vector v yields the rook domain property [13]. Then in this case we can conclude that the corresponding n-dimensional interleaving satisfies the rook-domain property [13].

As a consequence one-dimensional single-error-correcting codes can be used to correct the spread errors instead of the more complex n-dimensional burst-error-correcting codes and we have a perfect code [12].

Now, for (n + (n - 2)) a multiple of 3, the vector v does not yield the rook domain property [13]. Then in this case we can conclude that the corresponding n-dimensional interleaving does not satisfy the rook-domain property [13].

Consequently one-dimensional random-error-correcting codes that correct three errors can be used to correct the spread errors instead of the more complex n-dimensional burst-error-correcting codes and we have neither a perfect code [12] nor a quasi-perfect code [14].

#### 9. Conclusion

A significant n-dimensional interleaving scheme for combating quasicircular clusters of errors using simple error-correcting codes was proposed in this work. Such a scheme makes use of the Golomb et al. congruence equation (equation (5)). Also both the set partitioning concept (Table 3) and Golomb's et al. congruence equation (equation (5)) are distinct techniques leading to the same solution. For lower dimensions either one of them may be used indistinctly. However, for higher dimensions, the latter one is less complex. Among many applications for which this technique can be employed we mention magnetic or optical data storage and channel coding.

## Acknowledgment

The authors would like to thank the financial Brazilian agency FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), under grant no. 2013/03976-9, for the funding support and Dr. Mario Enrique Duarte González for the support of the computational computations.

#### References

- [1] C. de Almeida and R. Palazzo Jr., Efficient two-dimensional interleaving technique by use of the set partitioning concept, *IET Eletronics Letters*, **32**, No 6 (1996), 538-539.
- [2] S. Lin and D.J. Costello Jr., Error Control Coding, Prentice Hall (2004).
- [3] C.C. Trinca Watanabe, J.-C. Belfiore, E.D. de Carvalho, J. Vieira Filho, R. Palazzo Jr. and R.A. Watanabe, Construction of complex nested ideal lattices for complex-valued channel quantization, *International Journal of Applied Mathematics*, **31**, No 4 (2018), 549-585; doi: 10.12732/ijam.v31i4.4.
- [4] C.C. Trinca Watanabe, J.-C. Belfiore, E.D. de Carvalho, J. Vieira Filho and R.A. Watanabe, Construction of nested real ideal lattices for interference channel coding, *International Journal of Applied Mathematics*, 32, No 2 (2019), 295-323; doi: 10.12732/ijam.v32i2.11.
- [5] H. Cohn, Advanced Number Theory, Dover Publications, New York (1980).
- [6] M. Blaum, P.G. Farrell and H.C.A. Van Tilborg, Handbook of Coding Theory, Elsevier, Amsterdam (1998).
- [7] W. Zhang and J.K. Wolf, A Class of burst error-correcting quasi-cyclic codes, *IEEE Transactions on Information Theory*, **34**, No 3 (1988), 463-479; doi: 10.1109/18.6026.
- [8] A.Y. Kitaev, Fault-tolerant quantum computation by anyons, *Annals of Physics*, **303**, No 1 (2003), 2-30.
- [9] C.D. Albuquerque, R. Palazzo Jr. and E.B. da Silva, Construction of new toric quantum codes, *Contemporary Mathematics*, **518**, No 1 (2010), 1-10.

- [10] C.C. Trinca Watanabe, J.-C. Belfiore, E.D. de Carvalho and J. Vieira Filho,  $E_8$ -Lattice via the cyclotomic field  $\mathbb{Q}_{\xi_{24}}$ , International Journal of Applied Mathematics, **31**, No 1 (2018), 63-71; doi: 10.12732/ijam.v31i1.6.
- [11] G. Ungerboeck, Channel coding with multilevel/phase signals, *IEEE Transactions on Information Theory*, **28**, No 1 (1982), 55-67; doi: 10.1109/TIT.1982.1056454.
- [12] S.W. Golomb and L.R. Welch, Perfect codes in the lee metric and the packing of polyominoes, *SIAM Journal on Applied Mathematics*, **18**, No 2 (1970), 302-317; doi: 10.1137/0118025.
- [13] S.W. Golomb and E.C. Posner, Rook domains, latin squares, affine planes and error-distributing codes, *IEEE Transactions on Information Theory*, 10, No 3 (1964), 196-208; doi: 10.1109/TIT.1964.1053680.
- [14] C.Q. Queiroz, C. Camarero, C. Martinez and R. Palazzo Jr., Quasi-perfect codes from cayley graphs over integer rings, *IEEE Transactions on Information Theory*, **59**, No 9 (2013), 5905-5916; doi: 10.1109/TIT.2013.2266398.
- [15] F. Oggier, G. Rekaya, J.-C. Belfiore and E. Viterbo, Perfect space-time block codes, *IEEE Transactions on Information Theory*, **52**, No 9 (2006), 3885-3902; doi: 10.1109/TIT.2006.880010.
- [16] C.C. Trinca, E.D. de Carvalho, J. Vieira Filho and A.A. Andrade, On the construction of perfect codes from HEX signal constellations, Journal of the Franklin Institute, 349, No 10 (2012), 3060-3077; doi: 10.1016/j.jfranklin.2012.09.007.
- [17] C. Alves, W.L.S. Pinto and A.A. Andrade, Well-rounded lattices via polynomials with real roots, *International Journal of Applied Mathematics*, **33**, No 4 (2020), 663-672; doi: 10.12732/ijam.v33i4.10.
- [18] J. Carmelo Interlando, J.O.D. Lopes and T.P. da Nóbrega Neto, A new number field construction of the  $D_4$ -Lattice, International Journal of Applied Mathematics), **31**, No 2 (2018), 299-305; doi: 10.12732/ijam.v31i2.11.
- [19] J. Carmelo Interlando, A.A. Andrade, B.G. Malaxechebarría, A.J. Ferrari and R.R. de Araújo, Fully-diverse lattices from ramified cyclic extensions of prime degree, *International Journal of Applied Mathematics*, 33, No 6 (2020), 1009-1015; doi: 10.12732/ijam.v33i6.4.