Faculty of Mathematical Sciences

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands
Phone: +31-53-4893400
Fax: +31-53-4893114
Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1647

Error Patterns II

C. Hoede and Z. Li[1]

October, 2002

ISSN 0169-2690

[1]On leave from Shaanxi University, Xi'an, P.R. China

Error Patterns II

C. Hoede and Z. Li*

Department of Mathematical Sciences
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede, The Netherlands

Abstract

In coding theory the problem of decoding focuses on error vectors. In the simplest situation code words are $(0,1)$-vectors, as are the received messages and the error vectors. Comparison of a received word with the code words yields a set of error vectors. In deciding on the original code word, usually the one for which the error vector has minimum Hamming weight is chosen. This note is a continuation of a first investigation of error patterns. First we consider burstiness and distribution of error vectors without assuming cyclic conditions. Then we define some forms of perfectness of codes and pose the problem of finding semi-perfect codes. The results of a systematic search for small vector lengths are presented. Finally a link is laid between error vectors, graphs and balanced incomplete block designs.

Keywords: codes, graphs, error patterns.
2000 AMS Classification: 94B

1 Introduction

We refer to our first note on error patters for an introduction to the subject [3]. We consider $(0,1)$-vectors of length n. If $\mathbb{Z}_2$ is the Galois field on two elements, we consider essentially all elements of $\mathbb{Z}_2^n$. Any subset of these 2^n vectors forms a code, which we may call a *general* code. Usually more structure is considered to be present. A *linear* code is one in which the code words form a vector space. If with a specific code word $\underline{c}$ also all cyclic shifts are present the code is called *cyclic*. For the time being we consider only general codes.

We recall that for some $(0,1)$-vector we introduced the concept of *distribution* D of the ones of the vector. Assuming cyclic condition the k ones occur on distances $d_1, d_2, \ldots, d_k$.

*On leave from Shaanxi Normal University, Xi'an, P.R. China

Definition 1. The *distribution* D of a vector of length n with k ones is

$$\begin{aligned} D &= \frac{\prod_{i=1}^{k} d_i \cdot k^k}{n^k} && (k \geq 1) \\ D &= 1 && (k = 0) \quad . \end{aligned}$$

Definition 2. The *burstiness* B of a vector of length n with k ones is

$$B = 1 - D.$$

2 Distribution for vectors without cyclic conditions

Let us consider the situation in which cyclic conditions are assumed for some (error) vectors of length 9 of weight 3, so $k = 3$. The three ones are clearly distributed most evenly if $d_1 = d_2 = d_3 = 3$, as in the vector (0 1 0 0 1 0 0 1 0). We obtain $D = \frac{3^3 \cdot 3^3}{9^3} = 1$. For this vector $B = 1 - 1 = 0$.
Most likely such a vector is not due to burst errors but to three single errors.

Now let us assume that there are no cyclic conditions, then the most evenly distribution of the three ones is as in (1 0 0 0 1 0 0 0 1). Now there are two distances, both of value 4, that are to be considered.
As for the case of cyclic conditions we consider the product $P_{k-1} = d_1, d_2, \ldots, d_{k-1}$ of the distances between consecutively occurring ones. Completely analogous to the proof of Lemma 1 in [3], we obtain

Lemma 1. P_{k-1} has maximum value $(\frac{n-1}{k-1})^{k-1}$.

The maximum occurs for vectors in which first and last component are ones and the $k-2$ other ones are at equal distances consecutively, like in our example. Note that the maximum value is not always obtained, so strictly speaking we have here an upper bound on P_{k-1}. The value is used to normalize the product of distances and to achieve that $0 \leq D_a \leq 1$ for all vectors.

Definition 3. The *(acyclic) distribution* D_a of a vector of length n with k ones is

$$\begin{aligned} D_a &= \frac{\prod_{i=1}^{k-1} d_i \cdot (k-1)^{k-1}}{(n-1)^{k-1}} && (k \geq 1) \\ D_a &= 1 && (k = 0) \quad . \end{aligned}$$

Definition 4. The *(acyclic) burstiness* B_a of a vector length n with k ones is

$$B_a = 1 - D_a.$$

For our example we find, with $d_1 = 4,\ d_2 = 4,\ k = 3$, that $D_a = \frac{4 \cdot 4 \cdot 2^2}{8^2} = \frac{2^6}{2^6} = 1$.

Usually in literature the concept of burst length is considered.

Definition 5. The *burst length* L of an (error) vector with k ones is the distance between first and last occurring one. We assume $k \geq 2$.

Definition 6. *Equivalence* of the measures M_1 and M_2 means that, given two vectors $\underline{w}_1$ and $\underline{w}_2$, we have that if $M_1(\underline{w}_1) \geq M_1(\underline{w}_2)$ then $M_2(\underline{w}_1) \geq M_2(\underline{w}_2)$ and vice versa.

Lemma 2. The measures B and L are not equivalent.

Proof. We give a counterexample. Consider the vectors

$$\begin{aligned} \underline{w}_1 &= (1\,1\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0) \quad \text{and} \\ \underline{w}_2 &= (1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0), \end{aligned}$$

of length 16 with 4 ones.
We assume cyclic conditions and find $L(\underline{w}_1) = 7,\ L(\underline{w}_2) = 6$, so $L(\underline{w}_1) > L(\underline{w}_2)$ and, according to this measure, $\underline{w}_2$ is "burstier". However, the product P_4 of the distances gives $P_4(\underline{w}_1) = 1.5.1.9 = 45$, $P_4(\underline{w}_2) = 2.2.2.10 = 80$. Hence $D(\underline{w}_1) < D(\underline{w}_2)$, so $B(\underline{w}_1) > B(\underline{w}_2)$, which means that $\underline{w}_1$ is "burstier" according to that measure. □

3 Various forms of perfectness of codes

Given a general code C, the set of other words of $\mathbb{Z}_2^n$ may show special features. All words have a certain Hamming distance to one of the code words, namely the number of ones in the error vector with respect to the code word considered. When the code words have mutual distance greater than or equal to $d = 2e+1$, the code is e –error– correcting. This means that if we consider "spheres" around the code words of "radius" e, then all the words in a sphere can be uniquely decoded to the code word in the center of the sphere. Because we consider general codes we prefer to use this geometrical picture.
The first concept of perfectness involves the situation that all the spheres precisely "cover" all the words in $\mathbb{Z}_2^n$. A standard example is the code consisting of the words (0 0 0) and (1 1 1). All 6 other words have distance 1 to either (0 0 0) or (1 1 1). The spheres each contain 4 words, covering the 8 words precisely. Blahut [1] gives the following definition.

Definition 7. A *perfect code* is one for which there are equal radius spheres about the code words that are disjoint and that completely fill the space.

In general the set of non-overlapping spheres of some radius e will *not* completely fill the space $\mathbb{Z}_2^n$. In that case there are words that have distance $e+1$ or larger to the words of the code or, shortly, distance at least $e+1$ to the code. If this occurs again some special feature may be present. For example, all of these words outside the spheres may have distance $e+1$ to the code.

Definition 8. A *quasi-perfect* code is one in which spheres of radius e about each code word are disjoint and all words not in such a sphere are at a distance $e+1$ from *at least* one code word.

If some extra conditions are posed on the words we get another related concept.

Definition 9. A *nearly perfect* code is a quasi-perfect code with the extra property that every word that has distance at least e to the code; has distance e or $e+1$ to the same number of code words (namely $\lfloor n/(e+1) \rfloor$), see Cameron and van Lint [2].

We will introduce another concept of perfectness, which is called *semi-perfectness*. The reason is that, like described in our first paper, we want to investigate the situation that every word of $\mathbb{Z}_2^n$ can be decoded uniquely. This can be the case if all received words have a *unique* code word at minimum distance.

Definition 10. A *d-semi-perfect* code is a code with the property that every word has a *unique* code word at minimum distance.

So for such a code the Hamming distance d enables one to determine the most likely code word that was sent.

Example 1. We consider $\mathbb{Z}_2^4$, the set of words of length 4 and the code $C_1 = \{(0\ 0\ 0\ 0), (1\ 1\ 1\ 0)\}$. The two code words are on distance 3, and the code is therefore 1-error-correcting. The two spheres are:

$$\begin{aligned} S_1 &= \{(0\ 0\ 0\ 0), (1\ 0\ 0\ 0), (0\ 1\ 0\ 0), (0\ 0\ 1\ 0), (0\ 0\ 0\ 1)\} \\ S_2 &= \{(1\ 1\ 1\ 0), (0\ 1\ 1\ 0), (1\ 0\ 1\ 0), (1\ 1\ 0\ 0), (1\ 1\ 1\ 1)\}. \end{aligned}$$

There are 6 words left:
$(0\ 0\ 1\ 1), (1\ 0\ 0\ 1), (0\ 1\ 0\ 1)$, of weight 2, and $(1\ 1\ 0\ 1), (1\ 0\ 1\ 1), (0\ 1\ 1\ 1)$, of weight 3. The first three words have distance 2 to $(0\ 0\ 0\ 0)$, but distance 3 to $(1\ 1\ 1\ 0)$. Therefore they can be decoded, by maximum likelihood, as $(0\ 0\ 0\ 0)$. The second three words have distance 3 to $(0\ 0\ 0\ 0)$, but distance 2 to $(1\ 1\ 1\ 0)$, and can therefore be decoded as $(1\ 1\ 1\ 0)$.
In this example we see that C is not perfect, but C is quasi-perfect and d-semi-perfect. □

Example 2. Let us now consider the code $C_2 = \{(0\ 0\ 0\ 0), (1\ 1\ 1\ 1)\}$. The two spheres of radius 1 contain all words of weight 1 respectively weight 3. But all six words of weight 2 have the same distance to both code words, namely 2. This means that the code is quasi-perfect however, C_2 is not d-semi-perfect as there is no unique code word on distance 2. □

If we consider Example 2, there is a way in which the decoding might be done in a unique way. As we pointed out in our first paper we may introduce a second measure next to the Hamming distance. We proposed to use the burstiness measure B or, equivalently, the distribution measure D. In case a word has equal Hamming distance to two, or more, code words, the measure B may be different for the respective error vectors. If that is the case, the code word for which B has the highest value, if it is unique, may then be chosen for the decoding. So not only the maximum likelihood is considered for the distance, but it is also assumed that error patterns with a higher burstiness, for a specific code word, are indicating a more likely message of that code word. In principle occurrence of bursts is considered more likely than the occurrence of the same number of evenly distributed errors.

Definition 11. A *semi-perfect* code is a code for which a word either has a unique code word on minimum distance or, in case there are more code words on minimum distance than one, there is a unique code word for which the error vector has highest burstiness B.

In our Example 2 there still is no way to distinguish between the two code words.
It will be clear that instead of considering a measure like B or B_a, we might also consider other measures in combination with the Hamming distance in order to be able to uniquely decode a received word. One could e.g. consider the burst length L or any other measure applied to the error vector. We want to focus on the *pattern* of the errors of the error vectors.

4 A systematic search for semi-perfect codes

We now report on a systematic search for 1-error-correcting codes with code-vectors of length $n = 1, 2, \ldots$, etc. We are particularly interested in examples that are semi-perfect or even d-semi-perfect, so where the Hamming distance alone allows unique decoding. The main result is the discovery of an infinite class of d-semi-perfect codes. We will follow the search in our presentation.

$n = 1$. There are only two words, (0) and (1). Only one word, say (0), can be chosen as code word. The received word (1) contains one error and is decoded as (0). This *trivial* code is in fact 1-error-correcting. The sphere of radius 1 about (0) contains all words so, strictly speaking, we have a perfect code! We obtain a trivial perfect code for any value of n, by just considering only the code word with zeroes only.

$n = 2$. There are four words, (0 0), (0 1), (1 0) and (1 1). We should have code words on distance at least 3, so again only one code word, say (0 0), can be chosen. The sphere about (0 0), of radius 1, contains (0 0), (0 1) and (1 0), but not the word (1 1). However the sphere of radius 2 does. The code is perfect. It is, of course, also d-semi-perfect, as there is only one candidate to decode to for all received words, namely the code word (0 0).

$n = 3$. There are eight words of which e.g. (0 0 0) and (1 1 1) can be chosen as code words on distance 3. The two spheres of radius 1 contain 4 words each and therefore contain all words. The code is therefore perfect (as well as quasi-perfect and d-semi-perfect).

$n = 4$. We refer to the Examples 1 and 2. The words (0 0 0 0) and (1 1 1 0) formed a code that was not perfect, but was quasi-perfect respectively d-semi-perfect. The reasons are quite different. The code is quasi-perfect because the words not in the two spheres are on distance 2 to the code, i.e. the minimum distance to a code word is 2 for all of such vectors. The code is d-semi-perfect because the words are on minimum distance to a *unique* code word, which happens to be 2. This difference between the two concepts became clear in Example 2 as well.
For our main result we should already point out that the two words (0 0 0 | 0) and (1 1 1 | 0) can be seen as composition of the perfect code words found for $n = 3$ and the digit 0, which can be seen as the perfect code word found for $n = 1$!

$n = 5$. In constructing the code one is inclined to choose as many code words as possible on mutual distance at least 3. So let us choose $\underline{c}_1 = (0\ 0\ 0\ 0\ 0)$, $\underline{c}_2 = (1\ 1\ 1\ 0\ 0)$, without loss of generality, and try to add a third code word. After some puzzling we might find $\underline{c}_3 = (0\ 1\ 0\ 1\ 1)$. A general procedure would be to construct a graph on 2^5 vertices, corresponding to the 32 possible $(0, 1)$-vectors, and add edges between two vertices whenever the distance between the corresponding words is at least 3. In the resulting graph we may then determine a maximum clique,
But let us investigate the code $C = \{(0\ 0\ 0\ 0\ 0), (1\ 1\ 1\ 0\ 0), (0\ 1\ 0\ 1\ 1)\}$. The three spheres of radius 1 about them contain 18 of the 32 words, showing zero or one error. We will now list all 14 other words and give their error vectors with respect to the three code words.

nr.	word	$\underline{c}_1 = 0\,0\,0\,0\,0$	$\underline{c}_2 = 1\,1\,1\,0\,0$	$\underline{c}_3 = 0\,1\,0\,1\,1$
1.	1 0 0 0 1	1 0 0 0 1	0 1 1 0 1	1 1 0 1 0
2.	1 0 0 1 0	1 0 0 1 0	0 1 1 1 0	1 1 0 0 1
3.	0 0 1 0 1	0 0 1 0 1	1 1 0 0 1	0 1 1 1 0
4.	0 0 1 1 0	0 0 1 1 0	1 1 0 1 0	0 1 1 0 1
5.	1 1 0 1 0	1 1 0 1 0	0 0 1 1 0	1 0 0 0 1
6.	1 1 0 0 1	1 1 0 0 1	0 0 1 0 1	1 0 0 1 0
7.	1 0 1 1 0	1 0 1 1 0	0 1 0 1 0	1 1 1 0 1
8.	1 0 1 0 1	1 0 1 0 1	0 1 0 0 1	1 1 1 1 0
9.	1 0 0 1 1	1 0 0 1 1	0 1 1 1 1	1 1 0 0 0
10.	0 1 1 0 1	0 1 1 0 1	1 0 0 0 1	0 0 1 1 0
11.	0 0 1 1 1	0 0 1 1 1	1 1 0 1 1	0 1 1 0 0
12.	0 1 1 1 0	0 1 1 1 0	1 0 0 1 0	0 0 1 0 1
13.	1 0 1 1 1	1 0 1 1 1	0 1 0 1 1	1 1 1 0 0
14.	1 1 1 1 1	1 1 1 1 1	0 0 0 1 1	1 0 1 0 0

Table 1: Error patterns for 14 words.

The following words have unique code words on minimum distance: nrs. 1,2,3, 4,7,8,9,11 (all with distance 2). The code is not perfect, but also not quasi-perfect as word nr. 13 has distance 3 to the code. The interesting cases are 5,6,10,12,13 and 14, as in these cases the distances to $\underline{c}_2$ and $\underline{c}_3$ are equal, namely 2 or 3. This means that the code is also not d-semi-perfect. Let us have a look at the error vectors in these six cases.

	$\underline{c}_2$	$\underline{c}_3$
5.	0 0 1 1 0	1 0 0 0 1
6.	0 0 1 0 1	1 0 0 1 0
10.	1 0 0 0 1	0 0 1 1 0
12.	1 0 0 1 0	0 0 1 0 1
13.	0 1 0 1 1	1 1 1 0 0
14.	0 0 0 1 1	1 0 1 0 0

Let us investigate whether the error patterns allow distinction between $\underline{c}_2$ and $\underline{c}_3$ for the decoding. We choose B as measure and consider the word nr. 13. The error patterns are different. D has factors 2, 1 and 2 for $\underline{c}_2$ and factors 1, 2 and 3 for $\underline{c}_3$. Hence the error vector for $\underline{c}_3$ is burstier and may be chosen for the decoding. Also word. nr. 14 can be decoded in this way, as D has factors 1 and 4 for $\underline{c}_2$ and factors 2 and 3 for $\underline{c}_3$, so that word nr. 14 can be decoded as $\underline{c}_2$.
For the words 5, 6, 10 and 12 we meet the situation that the values for D are the same. Hence for this choice of a measure, next to the Hamming distance, we obtain the conclusion that the code is not semi-perfect.
Now let us choose B_a, the acyclic burstiness. All four remaining cases are now resolved and we have unique decoding. So for Hamming distance and B_a we have indeed a semi-perfect code. Note that with the burst length L we would obtain the same result.

We have chosen a code consisting of three words. However we may consider the code consisting of two words: (0 0 0 0 0) and (1 1 1 1 1). This code is a 2-error-correcting perfect code, as the two spheres of radius 2 contains precisely all 32 words. A code like this may be indicate for all odd values of n. We will call them *simple* perfect codes.

For our discussion it is also important to consider the code consisting of the two words (0 0 0 | 0 0) and (1 1 1 | 0 0), where we, again, indicated that we have partitioned the five digits into a set of three and a set of two digits. The set of three digits shows the perfect simple code for $n = 3$, whereas the set of two digits shows the perfect trivial code for $n = 2$.
Now consider an arbitrary received word, say (0 1 1 | 1 1). Partitioning this word we obtain the partial words (0 1 1) and (1 1). As the code on the first set of three digits is perfect (0 1 1) has (1 1 1) as the unique code word on distance 1. As the code on the second set of two digits is perfect (1 1) has (0 0) as the unique code word on distance 2. It is clear that (0 1 1 | 1 1) is to be decoded as (1 1 1 | 0 0) with distance $1 + 2 = 3$ as the distance to (0 0 0 | 0 0)

is $2+2=4$. The code is d-semi-perfect but not quasi-perfect. This feature will now be investigated further.

$n=6$. We have $2^6=64$ words and we consider the composition of two times the perfect code for $n=3$. So we partition the words as follows: $(a\ b\ c \mid d\ e\ f)$. For the first set of three digits we choose (0 0 0) or (1 1 1) and for the second set we do the same. This yields the following code C of four code words.

$$C=\{(0\ 0\ 0 \mid 0\ 0\ 0), (0\ 0\ 0 \mid 1\ 1\ 1), (1\ 1\ 1 \mid 0\ 0\ 0), (1\ 1\ 1 \mid 1\ 1\ 1)\}.$$

Although there can be codes with more words than four on mutual distance at least 3, this code has the nice property that it is d-semi-perfect. Any received word can be partitioned into two sets of three digits. If the word is $(a\ b\ c \mid d\ e\ f)$, then both sets $(a\ b\ c)$ and $(d\ e\ f)$ have a unique code word in the perfect code on the corresponding digits. The code word for which $(a\ b\ c)$ is nearest to the word on the corresponding digits and $(d\ e\ f)$ is nearest evidently is the unique code word on minimum distance. So if we receive (1 0 1 | 0 1 0) the word (1 0 1) is nearest to (1 1 1), whereas (0 1 0) is nearest to (0 0 0). Hence the received word is nearest to (1 1 1 | 0 0 0) and this is the unique code word on minimum distance.

$n=7$. For $n=7$ we might e.g. consider the composition of three perfect codes for a partitioning $7=3+3+1$ or a composition of two perfect codes for a partitioning $7=5+2$.
In both cases we obtain a d-semi-perfect code, of four and two code words respectively, namely

$$C_1 = \{(0\ 0\ 0 \mid 0\ 0\ 0 \mid 0), (0\ 0\ 0 \mid 1\ 1\ 1 \mid 0), (1\ 1\ 1 \mid 0\ 0\ 0 \mid 0), (1\ 1\ 1 \mid 1\ 1\ 1 \mid 0)\}$$

$$C_2 = \{(0\ 0\ 0\ 0\ 0 \mid 0\ 0), (1\ 1\ 1\ 1\ 1 \mid 0\ 0)\}.$$

The reader should by now easily follow us.

Definition 12. A *partition code* C with code words of length n is a code in which n is partitioned into integers $n_1, n_2, n_3, \ldots, n_k$.
Let $C_1, C_2, c_3, \ldots, C_k$ be codes with code words of lengths $n_1, n_2, n_3, \ldots, n_k$, then C consists of all possible combinations of these code words.

Clearly we have $|C| = \prod_{i=1}^{k} |C_i|$. We will be mainly interested in partition codes for which the codes C_i, $i=1,\ldots,k$, are perfect. We will call this a *partition code of perfect codes*.

Theorem 1. A partition code of perfect codes is d-semi-perfect.

Proof. Let n be partitioned into the number n_i, $i=1,\ldots,k$ and let there be perfect codes with code words of length n_i. Any received word $\underline{r}$ can be partitioned according to the partitioning of the code words. Let us call the k sets of digits the parts of the word $\underline{r}$. Then each part has a unique nearest code word in the perfect codes on the corresponding digits. Therefore there is a unique code word in C with minimum distance to $\underline{r}$. □

Remark 1. The theorem presupposed that there are indeed perfect codes with code-words of lengths n_i, $i = 1, \ldots, k$. As we have seen from our systematic search we have trivial or simple perfect codes for $n = 1, 2, 3, 5, 7, 9, \ldots$ etc. This means that we can construct many d-semi-perfect codes already.

Remark 2. The essential difference between the way of looking at codes that are perfect or quasi-perfect on one hand and that are d-semi-perfect or semi-perfect on the other hand is that in the first case the focus is on spheres about the code words that do not overlap. When the minimum distance is e.g. 3 then only spheres of radius 1 can be considered. In the second case, however, the focus is not on spheres, but on sets of words that have a certain code word as the code word on unique minimum distance.

Definition 13. A *semi-sphere* is a set of words consisting of a codeword $\underline{c}$ and all words that have $\underline{c}$ as the unique code word on minimum distance.

With this definition we may formulate d-semi-perfectness as the property that the union of the semi-spheres of the code words contains precisely all words, in distinction from perfectness where the union of all spheres contains precisely all words.

Example 3. Consider the example we gave for $n = 6$. The code C consists of the four words $(0\ 0\ 0 \mid 0\ 0\ 0), (0\ 0\ 0 \mid 1\ 1\ 1), (1\ 1\ 1 \mid 0\ 0\ 0)$ and $(1\ 1\ 1 \mid 1\ 1\ 1)$. We calculate the semi-sphere about $(0\ 0\ 0 \mid 0\ 0\ 0)$. The 16 words that have this code word as unique code word on minimum distance are, in order of weight, $(0\ 0\ 0 \mid 0\ 0\ 0)$, the code word itself.

$$(1\ 0\ 0 \mid 0\ 0\ 0), (0\ 1\ 0 \mid 0\ 0\ 0), (0\ 0\ 1 \mid 0\ 0\ 0),$$
$$(0\ 0\ 0 \mid 1\ 0\ 0), (0\ 0\ 0 \mid 0\ 1\ 0), (0\ 0\ 0 \mid 0\ 0\ 1),$$

the words on distance 1.

$$(1\ 0\ 0 \mid 1\ 0\ 0), (1\ 0\ 0 \mid 0\ 1\ 0), (1\ 0\ 0 \mid 0\ 0\ 1),$$
$$(0\ 1\ 0 \mid 1\ 0\ 0), (0\ 1\ 0 \mid 0\ 1\ 0), (0\ 1\ 0 \mid 0\ 0\ 1),$$
$$(0\ 0\ 1 \mid 1\ 0\ 0), (0\ 0\ 1 \mid 0\ 1\ 0), (0\ 0\ 1 \mid 0\ 0\ 1),$$

the words on distance 2.

In all there are $1 + 6 + 9 = 16$ words in this semi-sphere. Note that in the sphere about $(0\ 0\ 0 \mid 0\ 0\ 0)$ of radius 2 we also have a word like $(1\ 1\ 0 \mid 0\ 0\ 0)$. However, this code is closer to $(1\ 1\ 1 \mid 0\ 0\ 0)$, and belongs to the semi-sphere about that code word. □

5 A relation between word patterns, graphs and block designs

Given a specific value for n we can pose the problem to determine all different error patterns, or even just all different patterns of words. This can be done assuming cyclic conditions or acyclic conditions. We will assume cyclic conditions.

Let us consider a simple example of a cyclic (7,4) Hamming code, namely the code consisting of the code words

$$(0\ 0\ 0\ 0\ 0\ 0\ 0)$$
$$(0\ 0\ 0\ 1\ 0\ 1\ 1)$$
$$(1\ 1\ 1\ 0\ 1\ 0\ 0)$$
$$(1\ 1\ 1\ 1\ 1\ 1\ 1)$$

and their cyclic shifts.

As shifting (0 0 0 0 0 0 0) and (1 1 1 1 1 1 1) yields the same vectors there are 16 code words in all as both (0 0 0 1 0 1 1) and (1 1 1 0 1 0 0) yields 6 other vectors on shifting cyclically. Moreover all 16 code words have mutual distance at least 3, as we shall see in a moment, so we may consider the 16 spheres of radius 1 about them that each contains 8 words. So there are 16 spheres that together contain $16 \times 8 = 2^4 \times 2^3 = 2^7$ words, so the code is perfect.
We can relate this code to subgraphs of the complete graph K_7. We let the components of the vectors corresponding with the 7 vertices of the K_7, numbering these with the numbers $0, 1, 2, 3, 4, 5, 6$. Note that there are 7 edges for which the difference between these numbers for the vertices are 1 modulo 7, 7 for which this difference is 2 modulo 7 and 7 for which it is 3 modulo 7, see Figure 1.

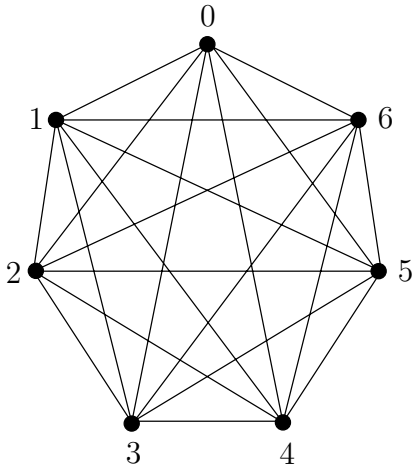

Figure 1: Labeled K_7.

Each word, of the 2^7 words, corresponds to a subgraph of this K_7. We choose the vertices indicating the positions for which the component of the word is 1. This yields the empty graph, without vertices for the word (0 0 0 0 0 0 0), the K_7 itself for the word (1 1 1 1 1 1 1) and subgraphs K_3 and K_4 for the other 14 words of the considered code.

The code word (0 0 0 1 0 1 1) corresponds with the subgraph K_3 on the vertices 3,5 and 6. The code word (1 1 1 0 1 0 0) corresponds with the subgraph K_4 on the vertices 0,1,2, and 4. The set $\{3, 5, 6\}$ forms a *difference set*. The differences are $\{1, 2, 3\}$ modulo 7, as $6 - 5 = 1$, $5 - 3 = 3$ and $6 - 3 = 3$. So we have precisely one type each of the three types that we distinguished for the edges. Note that $3 - 6 = -3 \equiv 4$ modulo 7, so that the vertices 3 and 6 determine a difference 3, but along the cycle C_7 with edges of type 1 they can be seen as being on distance 4. This is how we calculated the distances for our measure D.
Anyhow, shifting the code word cyclically corresponds to shifting the K_3 around in the K_7 cyclically. The 7 K_3's each have precisely one edge of each type and together they contain each edge precisely once. But the properties of these 7 K_3's can be worded as follows. There are $b = 7$ blocks of $k = 3$ elements out of $v = 7$ elements, such that each element occurs $r = 3$ times and each pair of elements occurs precisely $\lambda = 1$ times. Such a configuration is called a *Balanced Incomplete Block Design* (b, v, r, k, λ) or BIBD, and here we meet a (7,7,3,3,1)-BIBD.

The word (1 1 1 0 1 0 0) has corresponding vertices $\{0, 1, 2, 4\}$. Now there are six differences, namely $\{1, 1, 2, 2, 3, 3\}$. Shifting this K_4 cyclically gives $b = 7$ blocks of $k = 4$ elements, out of $v = 7$ elements, each occurring $v = 4$ times each pair occurring precisely $d = 2$ times. Hence these seven words determine a (7,7,4,4,2)-BIBD.

By considering the 14 code words as K_3's respectively K_4's one can easily see that two of them have at least 3 vertices in common. The K_3 on $\{3, 5, 6\}$ and the K_4 on $\{0, 1, 2, 4\}$ have even no vertex in common and the corresponding code words have therefore Hamming distance 7. The two K_3's on $\{3, 5, 6\}$ and $\{4, 6, 0\}$ have only one vertex, 6, in common and have distance 4, as words.

For our problem, of getting grip on error patterns, or on word patterns in general, the interpretation of a word as a subgraph of a complete graph may turn out to be fruitful. If we consider e.g. $n = 5$ and words of weight 3, then one type of word is (1 1 1 0 0) and another type is (1 1 0 1 0). All $\binom{5}{3} = 10$ words are of one of these types. One type corresponds to K_3's with edges of types 1, 1 and 2 whereas the other types corresponds to K_3's with edges of type 1, 2 and 2 of the embedding K_5, that has only two types of edges.

References

[1] Blahut, R.E., *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Massachusetts (1983), ISBN 0–201–10102–5.

[2] Cameron, P.J. and J.H. van Lint, *Designs, Graphs, Codes and their links*, London Mathematical Society Student Texts 22, Cambridge University Press, (1991), ISBN 0–521–42385–6.

[3] Hoede, C. and Z. Li, *Error Patterns*, Memorandum nr. 1588 Faculty of Mathematical Sciences, University of Twente, Enschede, The Netherlands (2001), ISSN 0169–2690.