On perfect arithmetic codes

Antoine C. Lobstein

Centre National de la Recherche Scientifique, Télécom Paris, Département Informatique, 46 rue Barrault, 75634 Paris Cedex 13, France

Received 20 November 1991
Revised 17 January 1992

Abstract


This short paper treats the perfect codes, in the case of arithmetic codes and Garcia-Rao modular distance.

1. Introduction

Arithmetic codes can detect or correct errors arising in arithmetic operations such as addition, multiplication, modular addition, . . . performed on integers which are represented in a fixed radix $r \geq 2$. The need for a measure of an error leads one to define the weight of an integer (and the distance between integers) in an appropriate way.

For any integer $I$, a form $I = \sum_i a_i \cdot r^i$ where $|a_i| < r$ for all $i$, is called a modified radix-$r$ form of $I$. Any such form is called minimal if it contains a minimal number of nonzero coefficients $a_i$; the arithmetic weight of $I$, $W(I)$, is the number of nonzero terms in a minimal form of $I$.

Let us now consider $\mathbb{Z}_m$, the ring of integers (mod $m$) ranging from 0 to $m-1$ ($m > 0$). The modular addition of two elements $I_1$ and $I_2$ in $\mathbb{Z}_m$ is $I_1 \oplus I_2 = I_1 + I_2$ if $I_1 + I_2 < m$, and $I_1 + I_2 - m$ otherwise. For the measure of errors occurring in this arithmetic operation, it is very natural to consider the following weight, called the Garcia-Rao modular weight and denoted by $w_m$: for any $I$ in $\mathbb{Z}_m$, $w_m(I) = \min\{W(I), W(m-I)\}$; and for any $I_1, I_2$ in $\mathbb{Z}_m$, $d_m(I_1, I_2) = w_m(|I_1 - I_2|)$ [7].

In general, $d_m$ does not satisfy the triangle inequality but from now on we shall only consider radices $r$ and moduli $m$ such that the triangle inequality holds (cf. [2]). This implies in particular that two spheres of radius $e$ with centres at distance $2e + 1$ have an empty intersection, and so perfect codes (a code is a subset of $\mathbb{Z}_m$)
can be studied. In [6] we surveyed this topic. We investigated the binary and ternary case: we gave two new perfect ternary codes as well as 'old' binary and ternary perfect codes (cf. [1] and [3]); we gave some results on the non-existence of perfect codes. Eventually we tried to find 'small' perfect codes and stated that for \( r = 2 \) and modulus \( m < 2^{14} - 2^{12} - 2^{10} \), the only possible \( e \)-error-correcting perfect codes having at most 20 codewords (apart from already known perfect codes), have the following parameters [6, Proposition 4]:

\[
\begin{align*}
e &= 1 \\
m &= 238 (14 \text{ words}) & m &= 255 (15) & m &= 1,528 (e = 2) (8 \text{ words}) \\
m &= 256 (16) & m &= 272 (16) & m &= 3,952 (e = 2) (16) \\
m &= 4,080 (e = 2) (17)
\end{align*}
\]

Our result is that none of these parameters provides a perfect code (whereas we would have preferred to find one (or more) perfect code(s) among these candidates). Our goal here is not to give the proof of this result (it can be found in [5]) but rather to show the general philosophy of it.

2. A guide line for the proof

In all seven cases we mentioned above, we prove that a perfect code cannot exist. Let us denote \( C = \{x_0, x_1, x_2, \ldots, x_k\} \) such a hypothetical code where, without loss of generality, \( x_0 = 0 < x_1 < x_2 < \cdots < x_k \) (with \( k = 7, 13, 14, 15 \) or 16).

If \( e = 1 \) (resp. \( e = 2 \)) we have to consider the set \( P_3 \) (resp. \( P_5 \)) of elements in \( \mathbb{Z}_m \) which have modular weight at least 3 (resp. 5) and pick \( k \) codewords \( x_1, \ldots, x_k \) such that any difference between codewords (including the codeword 0), \( \Delta_{i,j} := x_i - x_{i-j} \) (\( i = 1, 2, \ldots, k; \) \( j = 1, \ldots, i \)) belongs to \( P_3 \) (resp. \( P_5 \)). (In particular, with this notation, \( \Delta_{i,1} \) represents a difference between consecutive codewords.)

The main idea is the following: We choose a subset \( E_3 \) of \( P_3 \) (resp. \( E_5 \) of \( P_5 \)) which contains a certain number of the smallest elements in \( P_3 \) (resp. \( P_5 \)) and which is:
—sufficiently small to be handled and
—sufficiently large to necessarily provide a certain number of differences between consecutive codewords, thus giving constraints of which we prove that they cannot be fulfilled.

Two examples will show what we mean and how to choose \( E_5 \) or \( E_3 \).

3. Two examples

**Example 1.** A very short example is given by the case \( e = 2, \ m = 1,528 \) (8 codewords, so \( k = 7 \)), which is almost immediately 'killed' by our method: \( P_5 \)
contains 180 elements, and its smallest elements are

171, 173, 179, 181, 203, 205, 211, 213, 299, 301, 307,
309, 331, 333, 339, 341, 347, 349, 355, 357, 363, 365,
395, 397, 403, 405, 411, 413, 419, 421, 427. Stop!

This is all you need to know to be convinced that no perfect code exists.

Note that 171 is the smallest element in \( P_5 \) (and 1,528 - 171 = 1,357 is the
largest). Choose \( E_5 = \{171, 173, 179, 181, 203, 205, 211, 213\} \) (8 elements only); note that 299 is the smallest element in \( P_5 - E_5 \).

Consider the seven differences \( \Delta_{i,1} \) \((i = 1, \ldots, 7)\) of consecutive codewords. Because \( 4 \cdot 171 + 3 \cdot 299 > 1,357 \) and \( \sum_{1 \leq i \leq 7} \Delta_{i,1} = x_7 \equiv 1,357 \) at least five among
these \( \Delta_{i,1} \) belong to \( E_5 \) (this is an example of what we meant by 'sufficiently large
to necessarily provide a certain number of differences between consecutive
codewords'). So necessarily there exists one \( i \) between 1 and 6 such that \( \Delta_{i,1} \) and
\( \Delta_{i+1,1} \) belong to \( E_5 \). But \( \Delta_{i,1} + \Delta_{i+1,1} = x_i - x_{i-1} + x_{i+1} - x_i = x_{i+1} - x_{i-1} = \Delta_{i+1,2} \in P_5 \),
which is impossible: \( x, y \in E_5 \Rightarrow x + y \notin P_5 \), as it is easy to check since
the smallest integers in \( P_5 \), up to 427, are odd (see above; this is an example of
what we meant by 'constraints of which we prove that they cannot be fulfilled').
So we have a contradiction.

**Example 2.** \( e = 1, m = 255 \) (15 codewords, so \( k = 14 \)). \( P_3 \) contains 158 elements,
and its smallest 13 elements are

11, 13, 19, 21, 22, 23, 25, 26, 27, 29, 35, 37, 38.

Note that 11 is the smallest element in \( P_3 \) (and 255 - 11 = 244 is the largest).

Choose \( E_3 = \{11, 13\} \) (2 elements only!); note that 19 is the smallest element in
\( P_3 - E_3 \). We have the following result:

\[ \exists i \in \{1, \ldots, 13\} \text{ such that } \Delta_{i,1} \text{ and } \Delta_{i+1,1} \text{ belong to } E_3. \]

**Proof.** Let us assume the contrary (remember that the codewords are \( x_0 = 0 < x_1 < x_2 < \cdots < x_{14} \)).

- \( \Delta_{i,1} = 11 \Rightarrow \Delta_{j+1,2} \geq 37 \) (if \( j \neq 14 \) because \( \Delta_{j+1,2} = \Delta_{j,1} + \Delta_{j+1,1} \), \( \Delta_{j+1,1} = 11 \) or
13 is impossible by hypothesis, and \( \Delta_{j+1,1} = 19, 21, 22, 23 \) or 25 \( \Rightarrow \Delta_{j+1,2} = \)
30, 32, 33, 34 or 36 which are not in \( P_3 \). For the same reasons, \( \Delta_{i,1} = 11 \Rightarrow \Delta_{j,2} \geq \)
37 (if \( j \neq 1 \)).
- In the same way, \( \Delta_{j,1} = 13 \Rightarrow \Delta_{j+1,2} \geq 35 \) (if \( j \neq 14 \) and \( \Delta_{j,2} \geq 35 \) (if \( j \neq 1 \)).
- And \( \Delta_{j,1} = 19 \Rightarrow \Delta_{j+1,2} \geq 35 \) (if \( j \neq 14 \) and \( \Delta_{j,2} \geq 35 \) (if \( j \neq 1 \)).

Now \( x_{14} = \sum_{i \leq j \leq 7} \Delta_{2i,2} \geq 35 \cdot 7 = 245 \), a contradiction. \( \square \)

Since \( 11 + 13 = 24 \notin P_3 \), \( \Delta_{i,1}, \Delta_{i+1,1} \in E_3 \Rightarrow \Delta_{i,1} = \Delta_{i+1,1} \) and there must be a sequence of three consecutive codewords
This information, with little further investigation, is sufficient to prove that code $C$ cannot exist.

References