

# IBM Research Report

## Generalized and Extended Product Codes

**Mario Blaum, Steven Hetzler**

IBM Research Division  
Almaden Research Center  
650 Harry Road  
San Jose, CA 95120-6099  
USA



Research Division  
Almaden – Austin – Beijing – Cambridge – Dublin - Haifa – India – Melbourne - T.J. Watson – Tokyo - Zurich

# Generalized and Extended Product Codes

Mario Blaum and Steven Hetzler  
IBM Almaden Research Center  
San Jose, CA 95120

September 28, 2016

## Abstract

Generalized Product (GPC) Codes, a new construction unifying Product Codes and Integrated Interleaved (II) Codes, are presented. Product Codes and II Codes are special cases of GPC codes. Applications for approaches requiring local and global parities are described, like in the case of Locally Recoverable (LRC) Codes. The more general problem of extending product codes by adding global parities is studied and optimal solutions for one, two and three global parities are presented. Tradeoffs between the small field size required for GPC codes and optimality of more general EPC codes are discussed.

**Keywords:** Erasure-correcting codes, product codes, Reed-Solomon (RS) codes, generalized concatenated codes, integrated interleaving, MDS codes, PMDS codes, maximally recoverable codes, local and global parities, locally recoverable (LRC) codes.

## 1 Introduction

There has been considerable research lately on codes with local and global properties for erasure correction (see for instance [1][2][3][7][10][12][18][19][20][21][22][23][24][26] and references within). In general, data symbols are divided into sets and parity symbols (i.e., local parities) are added to each set (often, using an MDS code). This way, when a number of erasures not exceeding the number of parity symbols occurs in a set, such erasures are rapidly recovered. In addition to the local parities, a number of global parities are also added. Those global parities involve all of the data symbols and may include the local parity symbols as well. The global parities can correct situations in which the erasure-correcting power of the local parities has been exceeded.

The interest in erasure correcting codes with local and global properties arises mainly from two applications. One of them is the cloud. A cloud configuration may consist of many storage devices, of which some of them may even be in different geographical locations and the data is distributed across them. In the case that one or more of those devices fails, it

is desirable to recover its contents “locally,” that is, using a few parity devices within a set of limited size in order to affect performance as little as possible. However, the local parity may not be enough. In case the erasure-correcting capability of a local set is exceeded, extra protection is needed. In order to handle this situation, some devices containing global parities are incorporated, and when the local correction power is exceeded, the global parities are invoked and correction is attempted. If such a situation occurs, there will be an impact on performance, but data loss may be averted. It is expected that the cases in which the local parity is exceeded are relatively rare events, so the aforementioned impact on performance does not occur frequently. As an example of this type of application, we refer the reader to the description of the Azure system [11] or to the Xorbas code presented in [22].

A second application occurs in the context of Redundant Arrays of Independent Disk (RAID) architectures [5]. In this case, a RAID architecture protects against one or more storage device failures. For example, RAID 5 adds one extra parity device, allowing for the recovery of the contents of one failed device, while RAID 6 protects against up to two device failures. In particular, if those devices are Solid State Drives (SSDs), like flash memories, their reliability decays with time and with the number of writes and reads [15]. The information in SSDs is generally divided into pages, each page containing its own internal Error-Correction Code (ECC). It may happen that a particular page degrades and its ECC is exceeded. However, this situation may not be known to the user until the page is accessed (what is known as a silent failure). Assuming an SSD has failed in a RAID 5 scheme, if during reconstruction a silent page failure is encountered in one of the surviving SSDs, then data loss will occur. A method around this situation is using RAID 6. However, this method is costly, since it requires two whole SSDs as parity. It is more desirable to divide the information in a RAID type of architecture into  $m \times n$  stripes:  $m$  represents the size of a stripe, and  $n$  is the number of SSDs. The RAID architecture can be viewed as consisting of a large number of stripes, each stripe encoded and decoded independently. Certainly, codes like the ones used in cloud applications can be used as well for RAID applications. In practice, the choice of code depends on the statistics of errors and on the frequency of silent page failures. RAID systems, however, may behave differently than a cloud array of devices, in the sense that each column represents a whole storage device. When a device fails, then the whole column is lost, a correlation that may not occur in cloud applications. For that reason, RAID architectures may benefit from a special class of codes with local and global properties, the so called Sector-Disk (SD) codes, which take into account such correlations [16][17].

From now on, we call symbols the entries of a code with local and global properties. Such symbols can be whole devices (for example, in the case of cloud applications) or pages (in the case of RAID applications for SSDs). We assume that each symbol is protected by one local group, but a natural extension is to consider multiple localities [21][24][28]. A special case of multiple localities is given by product codes [14]: any symbol is protected by either horizontal or vertical parities.

Product codes by themselves may also be used in RAID type of architectures: the horizontal parities protect a number of devices from failure. The vertical parities allow for rapid

recovery of a page or sector within a device (a first responder type of approach). However, if the number of silent failures exceeds the correcting capability of the vertical code, and the horizontal code is unusable due to device failure, data loss will occur. For that reason, it may be convenient to incorporate a number of global parities to the product code.

In effect, assume that we have a product code consisting of  $m \times n$  arrays such that each column has  $v$  parity symbols and each row has  $h$  parity symbols. If in addition to the horizontal and vertical parities we have  $g$  extra parities, we say that the code is an Extended Product (EPC) code and we denote it by  $EP(m, v; n, h; g)$ . Notice that if  $g = 0$ , we have a regular product code. Similarly, if  $v = 0$ , we have a Locally Recoverable (LRC) code.

Constructions of LRC codes involve different issues and tradeoffs, like the size of the field and optimality criteria. The same is true for EPC codes, of which, as we have seen above, LRC codes are a special case. In particular, one goal is to keep the size of the finite field required small, since operations over a small field have less complexity than over a larger field due to the smaller look-up tables required. For example, Integrated Interleaved (II) codes [9][25] over  $GF(q)$ , where  $q \geq \max\{m, n\}$ , were proposed in [2] as LRC codes (II codes are closely related to Generalized Concatenated Codes [4][29]). Let us mention the construction in [13], which also reduces field size when failures are correlated. Similarly, we will propose a new family of codes that we call Generalized Product (GPC) codes, of which both product codes and II codes are special cases.

As LRC codes, EPC codes also have optimality issues. For example, LRC codes optimizing the minimum distance were presented in [24], and except for special cases, in general II codes are not optimal as LRC codes, but the codes in [24] require a field of size at least  $mn$ , so there is a tradeoff. The same happens with GPC codes: except for the special case of one global parity, they do not optimize the minimum distance. We examine some cases of EPC codes that do optimize the minimum distance for two and three global parities, but a larger field is required.

There are stronger criteria for optimization than the minimum distance in LRC codes. For example, PMDS codes [1][3][6][11] satisfy the Maximally Recoverable (MR) property [6][8]. The definition of the MR property is extended for EPC codes in [8], but it turns out that EPC codes with the MR property are difficult to obtain. For example, in [8] it was proven that an EPC code  $EP(m, 1; n, 1; 1)$  (i.e., one vertical and one horizontal parity per column and row and one global parity) with the MR property requires a field that is superlinear on the size of the array (and no explicit construction is given). We will not address EPC codes with the MR property here.

Although the constructions can be extended to finite fields of any characteristic, for simplicity, in this paper we assume that the finite fields have characteristic 2.

The paper is structured as follows: in Section 2 we present the definition of GPC codes and prove its properties, like the erasure-correcting capability, the minimum distance and encoding and decoding algorithms. In Section 3, we present constructions of EPC codes with two and three parities and discuss optimality issues. We end the paper by drawing some conclusions.

## 2 Generalized Product (GPC) Codes

We start by defining Generalized Product Codes, which unify product codes with II codes. These codes also consist of  $m \times n$  arrays whose elements are in a finite field  $GF(q)$  and it has similar characteristics to a  $t$ -level II code, except that the last  $m - k$  rows are devoted to parity in such a way that each column in the code belongs in an  $[m, m - k]$  MDS code. Explicitly,

**Definition 2.1** Take  $t$  integers  $1 \leq u_0 < u_1 < \dots < u_{t-1} \leq n - 1$  and let  $\underline{u}$  be the following vector of length  $m = s_0 + s_1 + \dots + s_{t-1}$ , where  $s_i \geq 1$  for  $0 \leq i \leq t - 1$ :

$$\underline{u} = \left( \overbrace{u_0, u_0, \dots, u_0}^{s_0}, \overbrace{u_1, u_1, \dots, u_1}^{s_1}, \dots, \overbrace{u_{t-1}, u_{t-1}, \dots, u_{t-1}}^{s_{t-1}} \right). \quad (1)$$

Consider a set  $\{\mathcal{C}_i\}$  of  $t$  nested  $[n, n - u_i, u_i + 1]$ ,  $0 \leq i \leq t - 1$ , Reed-Solomon [14] (RS) codes with elements in a finite field  $GF(q)$ ,  $q > \max\{m, n\}$ , such that a parity-check matrix for  $\mathcal{C}_i$  is given by

$$H_i = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{u_i-1} & \alpha^{2(u_i-1)} & \dots & \alpha^{(u_i-1)(n-1)} \end{pmatrix}, \quad (2)$$

where  $\alpha$  is an element of order  $\mathcal{O}(\alpha) \geq n$  in  $GF(q)$ .

For  $0 \leq m - k < s_{t-1}$ , let  $\mathcal{C}(n; k, \underline{u})$  be the code consisting of  $m \times n$  arrays over  $GF(q)$  such that, for each array in the code with rows  $\underline{c}_0, \underline{c}_1, \dots, \underline{c}_{m-1}$ ,  $\underline{c}_j \in \mathcal{C}_0$  for  $0 \leq j \leq m - 1$  and, if

$$\hat{s}_i = \sum_{j=i}^{t-1} s_j \quad \text{for } 0 \leq i \leq t - 1, \quad (3)$$

then

$$\bigoplus_{j=0}^{m-1} \alpha^{rj} \underline{c}_j \in \mathcal{C}_i \quad \text{for } 1 \leq i \leq t - 1 \quad \text{and } 0 \leq r \leq \hat{s}_i - 1 \quad (4)$$

$$\bigoplus_{j=0}^{m-1} \alpha^{rj} \underline{c}_j = 0 \quad \text{for } 0 \leq r \leq m - k - 1. \quad (5)$$

Then we say that  $\mathcal{C}(n; k, \underline{u})$  is a  $t$ -level Generalized Product (GPC) code.  $\square$

In reality, it is not necessary that the codes  $\mathcal{C}_i$  in Definition 2.1 are RS with a parity-check matrix as given by (2), or not even MDS, but we make the assumption for simplicity. The codes may even be binary [27].

Before giving the properties of  $t$ -level GPC codes, we present some examples.

**Example 2.1** Assume that  $k = m$  in Definition 2.1, then, there are no conditions (5) and  $\mathcal{C}(n; m, \underline{u})$  is a  $t$ -level Integrated Interleaved (II) [2][25] code.

So,  $t$ -level II codes can be viewed as a special case of  $t$ -level GPC codes.  $\square$

**Example 2.2** Assume that  $t = 1$ , then (1) gives  $\underline{u} = (\overbrace{u_0, u_0, \dots, u_0}^m)$  and, if  $k < m$ ,  $\mathcal{C}(n; k, \overbrace{u_0, u_0, \dots, u_0}^m)$  is a regular product code [14] such that each row is in an  $[n, n - u_0]$  code and each column in an  $[m, k]$  code.

So, product codes can be viewed as a special case of  $t$ -level GPC codes.  $\square$

**Example 2.3** Assume that  $t = 2$ . Then,  $\mathcal{C}_1 \subset \mathcal{C}_0$ ,  $\underline{u} = (\overbrace{u_0, u_0, \dots, u_0}^{s_0}, \overbrace{u_1, u_1, \dots, u_1}^{s_1})$ ,  $s_0 + s_1 = m$ , and consider the 2-level GPC code  $\mathcal{C}(n; k, \underline{u})$  with  $0 \leq m - k < s_1$ . Let  $\underline{c} = (\underline{c}_0, \underline{c}_1, \dots, \underline{c}_{m-1})$  be an  $m \times n$  array in  $\mathcal{C}(n; k, \underline{u})$ . Then,  $\underline{c}_j \in \mathcal{C}_0$  for each  $0 \leq j \leq m - 1$ , and (4) and (5) give

$$\bigoplus_{j=0}^{m-1} \alpha^{rj} \underline{c}_j \in \mathcal{C}_1 \text{ for } 0 \leq r \leq s_1 - 1 \quad (6)$$

$$\bigoplus_{j=0}^{m-1} \alpha^{rj} \underline{c}_j = 0 \text{ for } 0 \leq r \leq m - k - 1. \quad (7)$$

The 2-level II codes presented in [9] correspond to  $\mathcal{C}(n; m, \underline{u})$  in this example, i.e., only equations (6) are taken into account since  $k = m$ .

As another special case, take  $k = m - 1$  and  $\underline{u} = (\overbrace{1, 1, \dots, 1}^{m-2}, 2, 2)$ . The rows  $\underline{c}_0, \underline{c}_1, \dots, \underline{c}_{m-1}$  of  $\mathcal{C}(n; m - 1, (\overbrace{1, 1, \dots, 1}^{m-2}, 2, 2))$  constitute a 2-level II code. Each column is in an  $[m, m - 1, 2]$  code, each row is in an  $[n, n - 1, 2]$  code (single parity). The  $\mathcal{C}_0$  code is the  $[n, n - 1, 2]$  code, and the  $\mathcal{C}_1$  code is an  $[n, n - 2, 3]$  code given by the parity-check matrix

$$H_2 = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \end{pmatrix}.$$

Moreover, (6) and (7) give

$$\bigoplus_{i=0}^{m-1} \alpha^i \underline{c}_i \in \mathcal{C}_1$$

$$\bigoplus_{i=0}^{m-1} \underline{c}_i = 0.$$

It is not hard to prove directly that this code can correct any 5 erasures, but this will be a consequence of Corollary 2.2 below. It consists of a product code (which has minimum distance 4) plus one extra (global) parity. This extra parity brings the minimum distance up from 4 to 6. For instance, if  $m=4$  and  $n=5$ , erasure patterns like the following (vertices of a rectangle)

	X			X
	X			X

are uncorrectable by the product code but not by  $\mathcal{C}(5; 3, (1, 1, 2, 2))$ . An extra erasure in addition to the four depicted above can be corrected by either the horizontal or the vertical code.  $\square$

**Example 2.4** Assume that  $t=3$ . Then,  $\mathcal{C}_2 \subset \mathcal{C}_1 \subset \mathcal{C}_0$ ,

$$\underline{u} = \left( \overbrace{u_0, u_0, \dots, u_0}^{s_0}, \overbrace{u_1, u_1, \dots, u_1}^{s_1}, \overbrace{u_1, u_1, \dots, u_1}^{s_2} \right),$$

$s_0 + s_1 + s_2 = m$ , and consider the 3-level GPC code  $\mathcal{C}(n; k, \underline{u})$  with  $0 \leq m - k \leq s_2$ . Let  $\underline{c} = (\underline{c}_0, \underline{c}_1, \dots, \underline{c}_{m-1})$  be an  $m \times n$  array in  $\mathcal{C}(n; k, \underline{u})$ . Then,  $\underline{c}_j \in \mathcal{C}_0$  for each  $0 \leq j \leq m-1$ , and (4) and (5) give

$$\bigoplus_{j=0}^{m-1} \alpha^{rj} \underline{c}_j \in \mathcal{C}_2 \text{ for } 0 \leq r \leq s_2 - 1 \tag{8}$$

$$\bigoplus_{j=0}^{m-1} \alpha^{rj} \underline{c}_j \in \mathcal{C}_1 \text{ for } 0 \leq r \leq s_1 + s_2 - 1 \tag{9}$$

$$\bigoplus_{j=0}^{m-1} \alpha^{rj} \underline{c}_j = 0 \text{ for } 0 \leq r \leq m - k - 1. \tag{10}$$

$\square$

We are now ready to state the main result regarding GPC codes.

**Theorem 2.1** Consider an  $m \times n$  array corresponding to a  $\mathcal{C}(n; k, \underline{u})$   $t$ -level GPC code as given by Definition 2.1. Then, the code can correct up to  $s_0$  erasures in any row, up to  $u_i$  erasures in any  $s_i$  rows,  $1 \leq i \leq t - 1$ , and up to  $n$  erasures in any  $m - k$  rows.

**Proof:** We may assume that the rows with erasures contain more than  $u_0$  erasures, since each row is in  $\mathcal{C}_0$ , an  $[n, n - u_0, u_0 + 1]$  code, hence, rows with up to  $u_0$  erasures can be corrected. Assume that there are up to  $m - k$  erased rows and a number  $\ell$  of rows with more than  $u_0$  erasures such that there are up to  $u_i$  erasures in any up to  $s_i$  rows,  $1 \leq i \leq t - 1$ . We do induction on  $\ell$ .

Assume first that  $\ell = 0$ , that is, we have up to  $m - k$  erased rows and the rest of the rows are erasure free. We can certainly correct such up to  $m - k$  erased rows by using (5) (which states that each column in the array is in an  $[m, m - k, m - k + 1]$  MDS code).

So, assume that there are  $\ell \geq 1$  rows with more than  $s_0$  erasures each such that there are up to  $u_i$  erasures in any up to  $s_i$  rows,  $1 \leq i \leq t - 1$ . By induction, up to  $\ell - 1$  rows with this property are correctable.

Let  $i_0, i_1, \dots, i_{m-1}$  be an ordering of the rows according to a non-increasing number of erasures such that:

1. Rows  $i_0, i_1, \dots, i_{m-k-1}$  are erased.
2. Row  $i_{m-k+j}$  for  $0 \leq j \leq \ell - 1$  has  $v_j$  erasures, where  $u_{t-1} \geq v_0 \geq v_1 \geq \dots \geq v_{\ell-1} > u_0$ .
3. Rows  $i_{m-k+\ell}, i_{m-k+\ell+1}, \dots, i_{m-1}$  have no erasures.

It suffices to prove that the  $v_{\ell-1}$  erasures in row  $i_{m-k+\ell-1}$  can be corrected. Then we are left with  $\ell - 1$  rows with more than  $s_0$  erasures each such that there are up to  $u_i$  erasures in any up to  $s_i$  rows,  $1 \leq i \leq t - 1$ , and the result follows by induction.

Choose a code  $\mathcal{C}_s$  from the nested set of codes  $\mathcal{C}_i$ ,  $1 \leq i \leq t - 1$ , in Definition 2.1 such that  $\mathcal{C}_s$  can correct  $v_{\ell-1}$  erasures. Rearranging the order of the elements of the sums in (4), and since  $\mathcal{C}_{t-1} \subset \mathcal{C}_{t-2} \subset \dots \subset \mathcal{C}_s$ , from (4) we have

$$\bigoplus_{j=0}^{m-1} \alpha^{r i_j} \underline{c}_{i_j} \in \mathcal{C}_s \text{ for } 0 \leq r \leq m - k + \ell - 1. \quad (11)$$

Since the  $(m - k + \ell) \times m$  matrix corresponding to the coefficients of the  $\underline{c}_{i_j}$ s in (11) is a Vandermonde matrix, it can be triangulated, giving

$$\underline{c}_{i_r} \oplus \left( \bigoplus_{j=r+1}^{m-1} \gamma_{r,j} \underline{c}_{i_j} \right) \in \mathcal{C}_s \text{ for } 0 \leq r \leq m - k + \ell - 1, \quad (12)$$



where the coefficients  $\gamma_{r,j}$  are a result of the triangulation. In particular, taking  $r = m - k + \ell - 1$  in (12), we obtain

$$\underline{c}_{i_{m-k+\ell-1}} \oplus \left( \bigoplus_{j=m-k+\ell}^{m-1} \gamma_{m-k+\ell-1,j} \underline{c}_{i_j} \right) \in \mathcal{C}_s. \quad (13)$$

Since  $\underline{c}_{i_{m-k+\ell-1}}$  has  $v_{\ell-1}$  erasures and  $\underline{c}_{i_j}$  has no erasures for  $m - k + \ell \leq j \leq m - 1$ , then  $\underline{c}_{i_{m-k+\ell-1}} \oplus \left( \bigoplus_{j=m-k+\ell}^{m-1} \gamma_{m-k+\ell-1,j} \underline{c}_{i_j} \right)$  has  $v_{\ell-1}$  erasures. Since the vector is in  $\mathcal{C}_s$ , the erasures can be corrected. Once  $\underline{c}_{i_{m-k+\ell-1}} \oplus \left( \bigoplus_{j=m-k+\ell}^{m-1} \gamma_{m-k+\ell-1,j} \underline{c}_{i_j} \right)$  is corrected,  $\underline{c}_{i_{m-k+\ell-1}}$  is obtained as

$$\underline{c}_{i_{m-k+\ell-1}} = \left( \underline{c}_{i_{m-k+\ell-1}} \oplus \left( \bigoplus_{j=m-k+\ell}^{m-1} \gamma_{m-k+\ell-1,j} \underline{c}_{i_j} \right) \right) \oplus \left( \bigoplus_{j=m-k+\ell}^{m-1} \gamma_{m-k+\ell-1,j} \underline{c}_{i_j} \right)$$

and the result follows by induction on  $\ell$ . □

Theorem 2.1 generalizes Theorem 1 in [2]. The proof of Theorem 2.1 is constructive in the sense that it provides a decoding algorithm. The following example illustrates Theorem 2.1 and the decoding algorithm.

**Example 2.5** Consider the 3-level GPC code  $\mathcal{C}(7; 4, (1, 1, 3, 4, 4, 4))$  according to Definition 2.1 and Example 2.4. We have three codes  $\mathcal{C}_2 \subset \mathcal{C}_1 \subset \mathcal{C}_0$ , where  $\mathcal{C}_0$  is a  $[7, 6, 2]$  code,  $\mathcal{C}_1$  is a  $[7, 4, 4]$  code and  $\mathcal{C}_2$  is a  $[7, 3, 5]$  code. In addition, each column is in a  $[6, 4, 3]$  code. We may assume that the entries of these codes are in  $GF(8)$  and that  $\alpha$  is a primitive element in  $GF(8)$ .

Consider the  $6 \times 7$  array with erasures denoted by  $X$

$\underline{c}_0$			$X$				
$\underline{c}_1$	$X$	$X$	$X$	$X$	$X$	$X$	$X$
$\underline{c}_2$		$X$	$X$		$X$		$X$
$\underline{c}_3$	$X$			$X$		$X$	
$\underline{c}_4$	$X$	$X$	$X$	$X$	$X$	$X$	$X$
$\underline{c}_5$					$X$		

The first step is correcting the single erasures in  $\underline{c}_0$  and in  $\underline{c}_5$ . An ordering of the remaining rows in non-increasing number of erasures is  $\{i_0, i_1, i_2, i_3\} = \{1, 4, 2, 3\}$ . In particular,  $\underline{c}_3$  has three erasures. Following the proof of Theorem 2.1, there are  $m - k = 2$  erased rows (rows  $\underline{c}_1$  and  $\underline{c}_4$ ) and  $\ell = 2$  rows with erasures, but not totally erased (rows  $\underline{c}_2$  and  $\underline{c}_3$ ). According to (8) and (9),

$$\begin{aligned}
\underline{c}_0 \oplus \underline{c}_1 \oplus \underline{c}_2 \oplus \underline{c}_3 \oplus \underline{c}_4 \oplus \underline{c}_5 &\in \mathcal{C}_2 \\
\underline{c}_0 \oplus \alpha \underline{c}_1 \oplus \alpha^2 \underline{c}_2 \oplus \alpha^3 \underline{c}_3 \oplus \alpha^4 \underline{c}_4 \oplus \alpha^5 \underline{c}_5 &\in \mathcal{C}_2 \\
\underline{c}_0 \oplus \alpha^2 \underline{c}_1 \oplus \alpha^4 \underline{c}_2 \oplus \alpha^6 \underline{c}_3 \oplus \alpha^8 \underline{c}_4 \oplus \alpha^{10} \underline{c}_5 &\in \mathcal{C}_2 \\
\underline{c}_0 \oplus \alpha^3 \underline{c}_1 \oplus \alpha^6 \underline{c}_2 \oplus \alpha^9 \underline{c}_3 \oplus \alpha^{12} \underline{c}_4 \oplus \alpha^{15} \underline{c}_5 &\in \mathcal{C}_1.
\end{aligned}$$

Notice that  $\mathcal{C}_1$  can correct three erasures (i.e.,  $s=1$  in the proof of Theorem 2.1). Rearranging the  $\underline{c}_i$ s above in non-increasing number of erasures, we obtain

$$\begin{aligned}
\underline{c}_1 \oplus \underline{c}_4 \oplus \underline{c}_2 \oplus \underline{c}_3 \oplus \underline{c}_0 \oplus \underline{c}_5 &\in \mathcal{C}_2 \\
\alpha \underline{c}_1 \oplus \alpha^4 \underline{c}_4 \oplus \alpha^2 \underline{c}_2 \oplus \alpha^3 \underline{c}_3 \oplus \underline{c}_0 \oplus \alpha^5 \underline{c}_5 &\in \mathcal{C}_2 \\
\alpha^2 \underline{c}_1 \oplus \alpha^8 \underline{c}_4 \oplus \alpha^4 \underline{c}_2 \oplus \alpha^6 \underline{c}_3 \oplus \underline{c}_0 \oplus \alpha^{10} \underline{c}_5 &\in \mathcal{C}_2 \\
\alpha^3 \underline{c}_1 \oplus \alpha^{12} \underline{c}_4 \oplus \alpha^6 \underline{c}_2 \oplus \alpha^9 \underline{c}_3 \oplus \underline{c}_0 \oplus \alpha^{15} \underline{c}_5 &\in \mathcal{C}_1,
\end{aligned}$$

which corresponds to (11) in the proof of Theorem 2.1 (notice,  $\mathcal{C}_2 \subset \mathcal{C}_1$ ). The coefficients in the linear system above correspond to the following matrix:

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 \\
\alpha & \alpha^4 & \alpha^2 & \alpha^3 & 1 & \alpha^5 \\
\alpha^2 & \alpha^8 & \alpha^4 & \alpha^6 & 1 & \alpha^{10} \\
\alpha^3 & \alpha^{12} & \alpha^6 & \alpha^9 & 1 & \alpha^{15}
\end{pmatrix}.$$

Triangulating this matrix in  $GF(8)$ , where  $1 \oplus \alpha \oplus \alpha^3 = 0$ , gives

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \\
0 & 0 & 1 & \alpha & \alpha^3 & \alpha \\
0 & 0 & 0 & 1 & \alpha^3 & \alpha^5
\end{pmatrix}.$$

Applying this triangulation to the linear system, and since  $\mathcal{C}_2 \subset \mathcal{C}_1$ , we obtain the following triangulated system:

$$\begin{aligned}
\underline{c}_1 \oplus \underline{c}_4 \oplus \underline{c}_2 \oplus \underline{c}_3 \oplus \underline{c}_0 \oplus \underline{c}_5 &\in \mathcal{C}_2 \\
\underline{c}_4 \oplus \alpha^2 \underline{c}_2 \oplus \alpha^5 \underline{c}_3 \oplus \alpha \underline{c}_0 \oplus \alpha^4 \underline{c}_5 &\in \mathcal{C}_2 \\
\underline{c}_2 \oplus \alpha \underline{c}_3 \oplus \alpha^3 \underline{c}_0 \oplus \alpha \underline{c}_5 &\in \mathcal{C}_2 \\
\underline{c}_3 \oplus \alpha^3 \underline{c}_0 \oplus \alpha^5 \underline{c}_5 &\in \mathcal{C}_1.
\end{aligned}$$

Since  $\underline{c}_3$  has 3 erasures and  $\underline{c}_0$  and  $\underline{c}_5$  have no erasures,  $\underline{c}_3 \oplus \alpha^3 \underline{c}_0 \oplus \alpha^5 \underline{c}_5$  has 3 erasures, which can be corrected in  $\mathcal{C}_1$ . Then,

$$\underline{c}_3 = (\underline{c}_3 \oplus \alpha^3 \underline{c}_0 \oplus \alpha^5 \underline{c}_5) \oplus (\alpha^3 \underline{c}_0 \oplus \alpha^5 \underline{c}_5).$$

Similarly,  $\underline{c}_2 \oplus \alpha \underline{c}_3 \oplus \alpha^3 \underline{c}_0 \oplus \alpha \underline{c}_5$  has 4 erasures, which can be corrected in  $\mathcal{C}_2$ , and

$$\underline{c}_2 = (\underline{c}_2 \oplus \alpha \underline{c}_3 \oplus \alpha^3 \underline{c}_0 \oplus \alpha \underline{c}_5) \oplus (\alpha \underline{c}_3 \oplus \alpha^3 \underline{c}_0 \oplus \alpha \underline{c}_5).$$

Finally,  $\underline{c}_1$  and  $\underline{c}_4$  are obtained from (5). We can apply the triangulation, so we obtain

$$\begin{aligned}\underline{c}_4 &= \alpha^2 \underline{c}_2 \oplus \alpha^5 \underline{c}_3 \oplus \alpha \underline{c}_0 \oplus \alpha^4 \underline{c}_5 \\ \underline{c}_1 &= \underline{c}_4 \oplus \underline{c}_2 \oplus \underline{c}_3 \oplus \underline{c}_0 \oplus \underline{c}_5,\end{aligned}$$

completing the decoding.  $\square$

Before discussing the dimension, the encoding and the minimum distance of the code, let us state and prove the following lemma.

**Lemma 2.1** Consider the  $t$ -level GPC code  $\mathcal{C}(n; k, \underline{u})$  as given by Definition 2.1. Then, if  $\hat{s}_t = m - k$  and  $\hat{s}_j$  is given by (3) for  $0 \leq j \leq t - 1$ , given  $u_j + 1$  fixed locations in  $\hat{s}_{j+1} + 1$  different rows, then there is an array in  $\mathcal{C}(n; k, \underline{u})$  that is non-zero in such  $(\hat{s}_{j+1} + 1)(u_j + 1)$  locations and 0 elsewhere.

**Proof:** Given  $j$  such that  $0 \leq j \leq t - 1$  and  $u_j + 1$  fixed locations in a vector of length  $n$ , since  $\mathcal{C}_j$  is an  $[n, n - u_j, u_j + 1]$  MDS code, there is a codeword  $\underline{w}$  in  $\mathcal{C}_j$  whose non-zero entries are in such  $u_j + 1$  fixed locations. Assume that the  $\hat{s}_{j+1} + 1$  rows selected are  $i_0, i_1, \dots, i_{\hat{s}_{j+1}}$ , where

$$0 \leq i_0 < i_1 < \dots < i_{\hat{s}_{j+1}} \leq m - 1.$$

Let  $\underline{v} = (v_0, v_1, \dots, v_{\hat{s}_{j+1}})$  be a codeword of weight  $\hat{s}_{j+1} + 1$  in the (shortened)  $[\hat{s}_{j+1} + 1, 1, \hat{s}_{j+1} + 1]$  RS code whose parity-check matrix is given by

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha^{i_0} & \alpha^{i_1} & \dots & \alpha^{i_{\hat{s}_{j+1}}} \\ 1 & \alpha^{2i_0} & \alpha^{2i_1} & \dots & \alpha^{2i_{\hat{s}_{j+1}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{(\hat{s}_{j+1}-1)i_0} & \alpha^{(\hat{s}_{j+1}-1)i_1} & \dots & \alpha^{(\hat{s}_{j+1}-1)i_{\hat{s}_{j+1}}} \end{pmatrix}.$$

In particular,

$$\bigoplus_{s=0}^{\hat{s}_{j+1}} \alpha^{r i_s} v_s = 0 \text{ for } 0 \leq r \leq \hat{s}_{j+1} - 1. \quad (14)$$

Consider the  $m \times n$  array of weight  $(\hat{s}_{j+1} + 1)(u_j + 1)$  such that row  $i_s$  equals  $v_s \underline{w}$  for  $0 \leq s \leq \hat{s}_{j+1}$ , and the remaining rows are zero. We will show that this array is in  $\mathcal{C}(n; k, \underline{u})$ . Since each row of the array is in  $\mathcal{C}_j$  by design, in particular, it is in  $\mathcal{C}_0$ . Next, according to (4) and (5), we have to show that

$$\bigoplus_{s=0}^{\hat{s}_{j+1}} \alpha^{r i_s} (v_s \underline{w}) \in \mathcal{C}_i \text{ for } 1 \leq i \leq t-1 \text{ and } 0 \leq r \leq \hat{s}_i - 1 \quad (15)$$

$$\bigoplus_{s=0}^{\hat{s}_{j+1}} \alpha^{r i_s} (v_s \underline{w}) = 0 \text{ for } 0 \leq r \leq m - k - 1. \quad (16)$$

Since  $m - k \leq \hat{s}_{j+1}$  and

$$\bigoplus_{s=0}^{\hat{s}_{j+1}} \alpha^{r i_s} (v_s \underline{w}) = \left( \bigoplus_{s=0}^{\hat{s}_{j+1}} \alpha^{r i_s} v_s \right) \underline{w},$$

(16) follows directly from (14).

If  $i \leq j \leq t-1$ , then  $\mathcal{C}_j \subseteq \mathcal{C}_i$  and  $\underline{w} \in \mathcal{C}_i$ , so (15) follows. If  $0 \leq j \leq i-1$ , then,  $\hat{s}_{j+1} \geq \hat{s}_i$ , and, for  $0 \leq r \leq \hat{s}_i - 1$ , by (14),

$$\bigoplus_{s=0}^{\hat{s}_{j+1}} \alpha^{r i_s} (v_s \underline{w}) = \left( \bigoplus_{s=0}^{\hat{s}_{j+1}} \alpha^{r i_s} v_s \right) \underline{w} = 0,$$

so (15) follows also in this case.  $\square$

**Example 2.6** Consider the 3-level GPC code  $\mathcal{C}(7; 4, (1, 1, 3, 4, 4, 4))$  of Example 2.5. According to Lemma 2.1, the locations denoted by  $X$  in the following arrays correspond to the non-zero entries of arrays in  $\mathcal{C}(7; 4, (1, 1, 3, 4, 4, 4))$ :

	$X$		$X$					$X$	$X$		$X$		$X$							
	$X$		$X$											$X$		$X$		$X$	$X$	$X$
								$X$	$X$		$X$		$X$							
	$X$		$X$					$X$	$X$		$X$		$X$							
	$X$		$X$											$X$		$X$		$X$	$X$	$X$
	$X$		$X$					$X$	$X$		$X$		$X$	$X$		$X$		$X$	$X$	$X$

The arrays with erasures in locations  $X$  above are uncorrectable, since, provided the zero array was stored, the decoding cannot decide between the zero array and the arrays with non-zero entries in the locations  $X$ .  $\square$

Next we compute the dimension  $K$  of a  $t$ -level GPC code  $\mathcal{C}(n; k, \underline{u})$ .

**Corollary 2.1** Consider the  $t$ -level GPC code  $\mathcal{C}(n; k, \underline{u})$  as given by Definition 2.1. Then,  $\mathcal{C}(n; k, \underline{u})$  is an  $[N, K]$  code, where  $N = mn$  and

$$K = kn - \left( \sum_{i=0}^{t-2} s_i u_i \right) - (s_{t-1} - m + k)u_{t-1}. \quad (17)$$

**Proof:** Let  $\hat{s}_t = m - k$  and  $\hat{s}_j$  be given by (3) for  $0 \leq j \leq t - 1$ . Assume that the zero array is stored, and a received array has erasures in the last  $u_i$  entries of rows  $m - \hat{s}_i$  to  $m - \hat{s}_{i+1} - 1$  for  $0 \leq i \leq t - 1$ , and all the entries of rows  $k$  to  $m - 1$ . Thus, we have a total of

$$n(m - k) + \left( \sum_{i=0}^{t-2} s_i u_i \right) + (s_{t-1} - m + k)u_{t-1}$$

erasures. By Theorem 2.1, the array with erasures in the locations described above will be correctly decoded as the zero codeword, thus,

$$N - K \geq n(m - k) + \left( \sum_{i=0}^{t-2} s_i u_i \right) + (s_{t-1} + m + k)u_{t-1}. \quad (18)$$

Next, assume that an extra erasure has occurred in the array above, say, in location  $(u, v)$ , where  $u$  satisfies  $m - \hat{s}_i \leq u \leq m - \hat{s}_{i+1} - 1$  for some  $0 \leq i \leq t - 1$ , and, defining  $u_t = n$ ,  $v$  satisfies  $n - u_j \leq v \leq n - u_{j-1} - 1$  for some  $0 \leq j \leq t$ . We will show that the array with the extra erasure is uncorrectable for any such  $(u, v)$ , rendering

$$N - K \leq n(m - k) + \left( \sum_{i=0}^{t-2} s_i u_i \right) + (s_{t-1} + m + k)u_{t-1}, \quad (19)$$

thus (18) and (19) will prove (17).

So, given the extra erasure in  $(u, v)$  such that  $i$  and  $j$  are defined above, consider an array  $Y^{(u,v)} = \left( y_{a,b}^{(u,v)} \right)_{\substack{0 \leq a \leq m-1 \\ 0 \leq b \leq n-1}}$ , whose non-zero coordinates are in the intersection of rows

$$u, u + 1, \dots, u + \hat{s}_{i+1},$$

and columns

$$v, n - u_i, n - u_i + 1, \dots, n - 1.$$

Such a non-zero array exists due to Lemma 2.1. Consider the arrays  $Y^{(u',v)}$ , where  $u + 1 \leq u' \leq m - \hat{s}_j - 1$ .

Since  $y_{u',v}^{(u',v)}$  can be arbitrarily chosen, we require, for  $u' > u$ ,

$$\bigoplus_{z=u}^{u'} y_{z,v}^{(z,v)} = 0.$$

Then, defining

$$Y = \bigoplus_{z=u}^{m-\hat{s}_j-1} Y^{(z,v)},$$

we can see that  $Y$  has a non-zero entry in  $(u, v)$ , while the remaining non-zero entries are contained in the locations of the original erasures. So, the array  $Y^{(u,v)}$  is uncorrectable, since it can be decoded either as the zero array or as  $Y$ .  $\square$

Theorem II.1 in [25], which corresponds to Corollary 2 in [2], is a special case of Corollary 2.1.

The encoding is a special case of the decoding. For example, we may place the parities at the end of the array in increasing order of parities, as shown in Corollary 2.1. The parities are considered as erasures and may be obtained using the triangulation method described in Theorem 2.1. The fact that the locations of the erasures are known allows for a simplification of the decoding algorithm. For example, the triangulated matrix corresponding to the coefficients of (12) may be precomputed. We omit the implementation details.

**Example 2.7** We illustrate the proof of Corollary 2.1 with the 3-level GPC code  $\mathcal{C}(7; 4, (1, 1, 3, 4, 4, 4))$  of Examples 2.5 and 2.6. By Corollary 2.1, this code is a  $[42, 19]$  code. Following the proof of Corollary 2.1, denote by  $X$  the erased locations in an array  $W$ :

$$W = \begin{array}{|c|c|c|c|c|c|c|} \hline & & & & & & X \\ \hline & & & & & & X \\ \hline & & & & X & X & X \\ \hline & & & X & X & X & X \\ \hline X & X & X & X & X & X & X \\ \hline X & X & X & X & X & X & X \\ \hline \end{array}$$

If the non-erased locations are zero, by Theorem 2.1, the array will be decoded as the zero array. Now, assume that the array has an extra erasure in location  $(u, v) = (0, 1)$ , rendering

$$V = \begin{array}{|c|c|c|c|c|c|c|} \hline & X & & & & & X \\ \hline & & & & & & X \\ \hline & & & & X & X & X \\ \hline & & & X & X & X & X \\ \hline X & X & X & X & X & X & X \\ \hline X & X & X & X & X & X & X \\ \hline \end{array}$$

Consider the following arrays  $Y^{(u',1)}$ ,  $0 \leq u' \leq 3$ , defined as in Corollary 2.1, whose non-zero entries are denoted  $y_{a,b}^{(u',1)}$  below:

$$Y^{(0,1)} = \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & y_{0,1}^{(0,1)} & 0 & 0 & 0 & 0 & y_{0,6}^{(0,1)} \\ \hline 0 & y_{1,1}^{(0,1)} & 0 & 0 & 0 & 0 & y_{1,6}^{(0,1)} \\ \hline 0 & y_{2,1}^{(0,1)} & 0 & 0 & 0 & 0 & y_{2,6}^{(0,1)} \\ \hline 0 & y_{3,1}^{(0,1)} & 0 & 0 & 0 & 0 & y_{3,6}^{(0,1)} \\ \hline 0 & y_{4,1}^{(0,1)} & 0 & 0 & 0 & 0 & y_{4,6}^{(0,1)} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}.$$

$$Y^{(1,1)} = \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & y_{1,1}^{(1,1)} & 0 & 0 & 0 & 0 & y_{1,6}^{(1,1)} \\ \hline 0 & y_{2,1}^{(1,1)} & 0 & 0 & 0 & 0 & y_{2,6}^{(1,1)} \\ \hline 0 & y_{3,1}^{(1,1)} & 0 & 0 & 0 & 0 & y_{3,6}^{(1,1)} \\ \hline 0 & y_{4,1}^{(1,1)} & 0 & 0 & 0 & 0 & y_{4,6}^{(1,1)} \\ \hline 0 & y_{5,1}^{(1,1)} & 0 & 0 & 0 & 0 & y_{5,6}^{(1,1)} \\ \hline \end{array}.$$

$$Y^{(2,1)} = \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & y_{2,1}^{(2,1)} & 0 & 0 & y_{2,4}^{(2,1)} & y_{2,5}^{(2,1)} & y_{2,6}^{(2,1)} \\ \hline 0 & y_{3,1}^{(2,1)} & 0 & 0 & y_{3,4}^{(2,1)} & y_{3,5}^{(2,1)} & y_{3,6}^{(2,1)} \\ \hline 0 & y_{4,1}^{(2,1)} & 0 & 0 & y_{4,4}^{(2,1)} & y_{4,5}^{(2,1)} & y_{4,6}^{(2,1)} \\ \hline 0 & y_{5,1}^{(2,1)} & 0 & 0 & y_{5,4}^{(2,1)} & y_{5,5}^{(2,1)} & y_{5,6}^{(2,1)} \\ \hline \end{array}.$$

$$Y^{(3,1)} = \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & y_{3,1}^{(3,1)} & 0 & y_{3,2}^{(3,1)} & y_{3,4}^{(3,1)} & y_{3,5}^{(3,1)} & y_{3,6}^{(3,1)} \\ \hline 0 & y_{4,1}^{(3,1)} & 0 & y_{4,2}^{(3,1)} & y_{4,4}^{(3,1)} & y_{4,5}^{(3,1)} & y_{4,6}^{(3,1)} \\ \hline 0 & y_{5,1}^{(3,1)} & 0 & y_{5,2}^{(3,1)} & y_{5,4}^{(3,1)} & y_{5,5}^{(3,1)} & y_{5,6}^{(3,1)} \\ \hline \end{array}.$$

Such arrays with non-zero entries exist by Lemma 2.1 (see also Example 2.6). We choose  $y_{1,1}^{(1,1)}$ ,  $y_{2,1}^{(2,1)}$  and  $y_{3,1}^{(3,1)}$  such that

$$\begin{aligned} y_{1,1}^{(0,1)} \oplus y_{1,1}^{(1,1)} &= 0 \\ y_{2,1}^{(0,1)} \oplus y_{2,1}^{(1,1)} \oplus y_{2,1}^{(2,1)} &= 0 \\ y_{3,1}^{(0,1)} \oplus y_{3,1}^{(1,1)} \oplus y_{3,1}^{(2,1)} \oplus y_{3,1}^{(3,1)} &= 0 \end{aligned}$$

Then, defining  $Y = Y^{(0,1)} \oplus Y^{(1,1)} \oplus Y^{(2,1)} \oplus Y^{(3,1)}$ , we see that

$$Y = \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & y_{0,1}^{(0,1)} & 0 & 0 & 0 & 0 & X \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & X \\ \hline 0 & 0 & 0 & 0 & X & X & X \\ \hline 0 & 0 & 0 & X & X & X & X \\ \hline 0 & X & 0 & X & X & X & X \\ \hline 0 & X & 0 & X & X & X & X \\ \hline \end{array}$$

Array  $Y$  is non-zero since  $y_{0,1}^{(0,1)} \neq 0$  (entries denoted by  $X$  may be non-zero or not). Array  $V$  may be decoded either as the zero array or as  $Y$ , so it is uncorrectable. Since we can make the same argument for any entry  $(u, v)$  not contained in the erasures of  $W$ , the number of parity symbols is exactly 23 and the dimension of the code is 19.  $\square$

The following corollary extends Theorem II.2 on  $t$ -level II codes as stated in [25] and proven as Corollary 3 in [2]. It also generalizes the well known result that the minimum distance of a product code is the product of the minimum distances of the two component codes.

**Corollary 2.2** Consider the  $t$ -level GPC code  $\mathcal{C}(n; k, \underline{u})$  as given by Definition 2.1. Then, if  $\hat{s}_t = m - k$  and  $\hat{s}_i$  is given by (3) for  $0 \leq i \leq t - 1$ , the minimum distance of  $\mathcal{C}(n; k, \underline{u})$  is

$$d = \min \{ (\hat{s}_{i+1} + 1)(u_i + 1) , 0 \leq i \leq t - 1 \}. \quad (20)$$

**Proof:** For each  $i$  such that  $0 \leq i \leq t - 1$ , consider an array in  $\mathcal{C}(n; k, \underline{u})$  that has  $\hat{s}_{i+1}$  rows with  $u_i + 1$  erasures each, one row with  $u_i$  erasures, and all the other entries are zero. By Theorem 2.1, such arrays will be corrected by the code  $\mathcal{C}(n; k, \underline{u})$  as the zero codeword, thus

$$d \leq \min \{ (\hat{s}_{i+1} + 1)(u_i + 1) , 0 \leq i \leq t - 1 \}.$$

On the other hand, by Lemma 2.1, for each  $0 \leq i \leq t - 1$ , there is an array in  $\mathcal{C}(n; k, \underline{u})$  of weight  $(\hat{s}_{i+1} + 1)(u_i + 1)$ , so

$$d \geq \min \{ (\hat{s}_{i+1} + 1)(u_i + 1) , 0 \leq i \leq t - 1 \}$$

and (20) follows.  $\square$



**Example 2.8** Consider the 3-level GPC code  $\mathcal{C}(7; 4, (1, 1, 3, 4, 4, 4))$  of Example 2.5. According to Corollary 2.2, since  $m = 6$ ,  $k = 4$ ,  $u_0 = 1$ ,  $u_1 = 3$ ,  $u_2 = 4$ ,  $s_0 = 2$ ,  $s_1 = 1$ ,  $s_2 = 3$  (and hence,  $\hat{s}_3 = m - k = 2$ ,  $\hat{s}_2 = s_2 = 3$ ,  $\hat{s}_1 = s_1 + s_2 = 4$ ), according to (20), the minimum distance of this code is

$$d = \min \{(5)(2); (4)(4); (3)(5)\} = 10.$$

□

Consider next a product code, such that the vertical code is an  $[m, k_0, m - k_0 + 1]$  code, and the horizontal code is an  $[n, k_1, n - k_1 + 1]$  code. In the notation of GPC codes, this is a 1-level  $(n; k_0, \overbrace{(n - k_1 + 1, n - k_1 + 1, \dots, n - k_1 + 1)}^m)$  GPC code. However, we can look at it also from the perspective of columns, and the code is also a 1-level  $(m; k_1, \overbrace{(m - k_0 + 1, m - k_0 + 1, \dots, m - k_0 + 1)}^n)$  GPC code. The following theorem generalizes this argument for a  $t$ -level GPC code.

**Theorem 2.2** Consider an  $m \times n$  array corresponding to a  $\mathcal{C}(n; k, \underline{u})$   $t$ -level GPC code as given by Definition 2.1. Then, viewed as an  $n \times m$  array on columns, the code is a  $\mathcal{C}(m; n - u_0, \underline{u}')$   $t$ -level GPC code, where  $\hat{s}_t = m - k$ ,  $\hat{s}_i$  is given by (3) for  $1 \leq i \leq t - 1$ ,  $u_t = n$ ,

$$\underline{u}' = \left( \overbrace{u'_0, u'_0, \dots, u'_0}^{s'_0}, \overbrace{u'_1, u'_1, \dots, u'_1}^{s'_1}, \dots, \overbrace{u'_{t-1}, u'_{t-1}, \dots, u'_{t-1}}^{s'_{t-1}} \right), \quad (21)$$

$$u'_{t-i} = \hat{s}_i \quad \text{for } 1 \leq i \leq t, \quad s'_i = u_{t-i} - u_{t-i-1} \quad \text{for } 0 \leq i \leq t - 2 \quad \text{and} \quad s'_{t-1} = u_1. \quad (22)$$

**Proof:** Denote by  $\underline{c}_i^{(\mathbf{H})}$ ,  $0 \leq i \leq m - 1$ , the rows of the array, and by  $\underline{c}_j^{(\mathbf{V})}$ ,  $0 \leq j \leq n - 1$ , the columns. Specifically, if the array consists of symbols  $(c_{i,j})_{\substack{0 \leq i \leq m-1 \\ 0 \leq j \leq n-1}}$ , then

$$\underline{c}_i^{(\mathbf{H})} = (c_{i,0}, c_{i,1}, \dots, c_{i,n-1}) \quad \text{for } 0 \leq i \leq m - 1$$

and

$$\underline{c}_j^{(\mathbf{V})} = (c_{0,j}, c_{1,j}, \dots, c_{m-1,j}) \quad \text{for } 0 \leq j \leq n - 1.$$

Consider the  $t$  nested codes (on columns)  $\mathcal{C}'_{t-1} \subset \mathcal{C}'_{t-2} \subset \dots \subset \mathcal{C}'_0$ , where  $\mathcal{C}'_i$  is an  $[m, m - u'_i, u'_i + 1]$  code. A parity-check matrix of  $\mathcal{C}'_i$  is

$$H'_i = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{m-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(m-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{u'_i-1} & \alpha^{2(u'_i-1)} & \dots & \alpha^{(m-1)(u'_i-1)} \end{pmatrix} \quad (23)$$

In order to prove the theorem, according to Definition 2.1, we have to prove that each  $\underline{c}_j^{(\mathbf{V})} \in \mathcal{C}'_0$ ,  $0 \leq j \leq n-1$ , and

$$\bigoplus_{j=0}^{n-1} \alpha^{rj} \underline{c}_j^{(\mathbf{V})} \in \mathcal{C}'_i \text{ for } 1 \leq i \leq t-1 \text{ and } 0 \leq r \leq \hat{s}'_i - 1 \quad (24)$$

$$\bigoplus_{j=0}^{n-1} \alpha^{ij} \underline{c}_j^{(\mathbf{V})} = 0 \text{ for } 0 \leq i \leq u_0 - 1. \quad (25)$$

$\mathcal{C}'_0$  is an  $[m, m - u'_0, u'_0 + 1]$  code and by (22),  $u'_0 = m - k$ , so from (5),  $\underline{c}_j^{(\mathbf{V})} \in \mathcal{C}'_0$ .

Notice also that since each  $\underline{c}_i^{(\mathbf{H})} \in \mathcal{C}_0$  for  $0 \leq i \leq m-1$ , (25) follows.

Next we have to prove (24). In effect, (24) holds if and only if, by (23),

$$\bigoplus_{v=0}^{m-1} \alpha^{uv} \bigoplus_{j=0}^{n-1} \alpha^{rj} c_{v,j} = 0 \text{ for } 1 \leq i \leq t-1, 0 \leq u \leq u'_i - 1 \text{ and } 0 \leq r \leq \hat{s}'_i - 1,$$

if and only if, changing the summation order,

$$\bigoplus_{j=0}^{n-1} \alpha^{rj} \bigoplus_{v=0}^{m-1} \alpha^{uv} c_{v,j} = 0 \text{ for } 1 \leq i \leq t-1, 0 \leq u \leq u'_i - 1 \text{ and } 0 \leq r \leq \hat{s}'_i - 1,$$

if and only if

$$\bigoplus_{v=0}^{m-1} \alpha^{uv} \underline{c}_v^{(\mathbf{H})} \in \hat{\mathcal{C}}_i \text{ for } 1 \leq i \leq t-1, 0 \leq u \leq u'_i - 1, \quad (26)$$

where a parity-check matrix for  $\hat{\mathcal{C}}_i$  is given by

$$\hat{H}_i = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{\hat{s}'_i-1} & \alpha^{2(\hat{s}'_i-1)} & \dots & \alpha^{(n-1)(\hat{s}'_i-1)} \end{pmatrix}. \quad (27)$$

By (3) and (22) ,

$$\hat{s}'_i = \sum_{z=i}^{t-1} s'_z = \left( \sum_{z=i}^{t-2} u_{t-z} - u_{t-z-1} \right) + u_1 = u_{t-i},$$

so, by (27) and (2),  $\hat{H}_i = H_{t-i}$  and hence  $\hat{\mathcal{C}}_i = \mathcal{C}_{t-i}$ . By (22),  $u'_i = \hat{s}_{t-i}$ , so (26) becomes

$$\bigoplus_{v=0}^{m-1} \alpha^{uv} \mathcal{C}_v^{(\mathbf{H})} \in \mathcal{C}_{t-i} \text{ for } 1 \leq i \leq t-1 \text{ and } 0 \leq u \leq \hat{s}_{t-i} - 1,$$

which is equivalent to (4) and thus (24) and (4) are equivalent, completing the proof.  $\square$

**Example 2.9** Consider the 3-level GPC code  $\mathcal{C}(7; 5, (1, 1, 3, 3, 5, 5))$ . According to Theorem 2.2, this code is also a 3-level GPC code  $\mathcal{C}(6; 6, (1, 1, 2, 2, 4, 4, 4))$  consisting of  $7 \times 6$  arrays, so, according to Theorem 2.1, it can correct any column with one erasure, up to two columns with 2 erasures, up to 2 columns with 4 erasures and up to one erased column. This allows for correction of erasures that cannot be handled by the correction on rows. For example, consider the following array, where the erasures are denoted by  $X$ :

$X$				$X$		
$X$	$X$					
	$X$	$X$				
		$X$	$X$			
			$X$	$X$		

The erasures cannot be decoded by the horizontal code  $\mathcal{C}(7; 5, (1, 1, 3, 3, 5, 5))$ , but they can certainly be handled by the vertical code  $\mathcal{C}(6; 6, (1, 1, 2, 2, 4, 4, 4))$ .  $\square$

Example 2.9 suggests an expansion of the decoding algorithm as given in the proof by triangulation of Theorem 2.1: given a  $t$ -level GPC code  $\mathcal{C}(n; k, \underline{u})$ , each time there are erasures we apply the decoding algorithm on rows as described in Theorem 2.1. If after this process there are still erasures remaining, we apply the decoding algorithm on columns for the  $\mathcal{C}(m; m - u_0, \underline{u}')$   $t$ -level GPC code as determined by Theorem 2.2. The method extends the decoding method of product codes, in which erasures are iteratively corrected by both codes, until they are either corrected or an uncorrectable pattern remains.

### 3 Extended Product Codes and Optimality Issues

The  $t$ -level GPC codes  $\mathcal{C}(n; k, \underline{u})$  described in Section 2 are a special case of product codes with some extra (global) parities. Let us call an extended product (EPC) code such a code, and denote it by  $EP(m, v; n, h; g)$ , where  $v$  is the number of vertical parities,  $h$  the number of horizontal parities, and  $g$  the number of global parities. We further assume

$$g < \min\{m - v; n - h\}.$$

For example, the 3-level GPC code  $\mathcal{C}(7; 4, (1, 1, 3, 4, 4, 4))$  of Examples 2.5, 2.6 and 2.7 is an  $EP(6, 2; 7, 1; 5)$ , while the 3-level GPC code  $\mathcal{C}(7; 5, (1, 1, 3, 3, 5, 5))$  of Example 2.9 is an  $EP(6, 1; 7, 1; 8)$ .

The next lemma gives an upper bound for the minimum distance of an  $EP(m, v; n, h; g)$  code.

**Lemma 3.1** Let  $d(v, h, g)$  be the minimum distance of an  $EP(m, v; n, h; g)$  code. Then,

$$d(v, h, g) \leq \min\{(v + 1)(h + g + 1), (h + 1)(v + g + 1)\} \quad (28)$$

**Proof:** Assume that the zero array is stored, and the received array has the locations  $(i, j)$  erased, where  $0 \leq i \leq v$  and  $0 \leq j \leq h + g$  (in particular, we have  $(v + 1)(h + g + 1)$  erasures). We argue that such a received array is uncorrectable. Notice that we have  $h(v + 1)$  horizontal parities and  $v(h + g + 1)$  vertical parities corresponding to the product code in order to correct such patterns, but  $hv$  of such parities are dependent, so that leaves us with a total of  $h(v + 1) + v(h + g + 1) - hv = (h + g + 1)v + h$  parities corresponding to the product code. In addition, an extra  $g$  global parities can be used, giving a total of  $(h + g + 1)v + h + g = (v + 1)(h + g + 1) - 1$  parities, insufficient to correct  $(v + 1)(h + g + 1)$  erasures. A similar pattern of  $(h + 1)(v + g + 1)$  erasures is also uncorrectable, proving (28).  $\square$

Consider the special case of a 2-level GPC code  $\mathcal{C}(n; k_0, \underline{u})$ , where

$$\underline{u} = \overbrace{(k_1, k_1, \dots, k_1)}^{m-k_0-1}, \overbrace{(k_1+1, k_1+1, \dots, k_1+1)}^{m-k_0+1}$$

(the even more special case in which  $k_0 = m - 1$  and  $k_1 = n - 1$  was examined in Example 2.3). In particular,  $\mathcal{C}(n; k_0, \underline{u})$  is an  $EP(m, m - k_0; n, n - k_1; 1)$  code. By Corollary 2.2, the minimum distance of this code is

$$d = \min\{(m - k_0 + 1)(n - k_1 + 2), (n - k_1 + 1)(m - k_0 + 2)\}. \quad (29)$$

Notice that (29) meets bound (28) with equality, showing that when  $g = 1$ , the bound is tight.

Notice also that if  $m - k_0 = n - k_1 = 1$  (single parity horizontal and vertical codes), then  $d = 6$ , as claimed in Example 2.3.

Let us examine now the case of  $EP(m, 1; n, 1; 2)$  codes. In this case, bound (28) gives

$$d(1, 1, 2) \leq 8. \quad (30)$$

Consider for example a 2-level GPC code  $\mathcal{C}(n; m - 1, \overbrace{(1, 1, \dots, 1)}^{m-2}, 3, 3)$  or a 2-level GPC code  $\mathcal{C}(n; m - 1, \overbrace{(1, 1, \dots, 1)}^{m-3}, 2, 2, 2)$ . Those are the only cases of GPC codes that are  $EP(m, 1; n, 1; 2)$  codes. In both cases, according to Corollary 2.2, the minimum distance is 6, so bound (30) is not met.

However, bound (30) is tight, and to show this we present an  $EP(m, 1; n, 1; 2)$  code with minimum distance 8. The construction is related to the PMDS constructions in [3], and we pay the price of requiring a larger finite field than for GPC codes.

Let  $GF(2^b)$  be a finite field such that  $mn < 2^b$ . Let  $\alpha$  be a primitive element in  $GF(2^b)$ . Consider the parity-check matrix  $\mathcal{H}_2$  of the code given by

$$\mathcal{H}_2 = \left( \begin{array}{c} I_m \otimes \overbrace{(1, 1, \dots, 1)}^n \\ \overbrace{(1, 1, \dots, 1)}^m \otimes I_n \\ \hline 1 \quad \alpha \quad \alpha^2 \quad \dots \quad \alpha^{mn-1} \\ 1 \quad \alpha^{-1} \quad \alpha^{-2} \quad \dots \quad \alpha^{-(mn-1)} \end{array} \right), \quad (31)$$

where  $I_m$  denotes the  $m \times m$  identity matrix and  $\otimes$  the Kronecker product [14] of two matrices. Notice that

$$\left( \begin{array}{c} I_m \otimes \overbrace{(1, 1, \dots, 1)}^n \\ \overbrace{(1, 1, \dots, 1)}^m \otimes I_n \end{array} \right)$$

corresponds to the parity-check matrix of the product code with single parity in rows and columns. We denote the matrix in (31) as  $\mathcal{H}_2$  to indicate that two extra parities are added to the product code.

The following lemma gives the minimum distance of the code whose parity-check matrix is given by  $\mathcal{H}_2$ .

**Lemma 3.2** Consider the  $EP(m, 1; n, 1; 2)$  code whose parity-check matrix  $\mathcal{H}_2$  is given by (31). Then, the code has minimum distance 8.

**Proof:** We have to prove that any 7 erasures can be corrected.

First assume that there are six erasures in locations  $(i_0, j_0), (i_0, j_1), (i_0, j_2), (i_1, j_0), (i_1, j_1)$  and  $(i_1, j_2)$ , where  $0 \leq i_0 < i_1 \leq m - 1$  and  $0 \leq j_0 < j_1 < j_2 \leq n - 1$  or  $(i_0, j_0), (i_0, j_1), (i_1, j_0), (i_1, j_1), (i_2, j_0)$  and  $(i_2, j_1)$ , where  $0 \leq i_0 < i_1 < i_2 \leq m - 1$  and  $0 \leq j_0 < j_1 \leq n - 1$ , and a seventh erasure in any other location. This seventh erasure can be corrected using either horizontal or vertical parities, thus, it is enough to prove that the two situations of six erasures described above are correctable. For example, using  $5 \times 5$  arrays, these two situations are illustrated below:

E	E		E	
E	E		E	

	E		E	
	E		E	
	E		E	

Consider the first case, as illustrated by the array in the left. It suffices to prove, using the parity-check matrix as given by (31), that the  $6 \times 6$  matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ \alpha^{i_0n+j_0} & \alpha^{i_0n+j_1} & \alpha^{i_0n+j_2} & \alpha^{i_1n+j_0} & \alpha^{i_1n+j_1} & \alpha^{i_1n+j_2} \\ \alpha^{-i_0n-j_0} & \alpha^{-i_0n-j_1} & \alpha^{-i_0n-j_2} & \alpha^{-i_1n-j_0} & \alpha^{-i_1n-j_1} & \alpha^{-i_1n-j_2} \end{pmatrix}$$

is invertible. Redefining  $i \leftarrow i_1 - i_0$ ,  $j_1 \leftarrow j_1 - j_0$  and  $j_2 \leftarrow j_2 - j_0$ , where now  $1 \leq i \leq m - 1$  and  $1 \leq j_1 < j_2 \leq n - 1$ , this matrix is invertible if and only if matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & \alpha^{j_1} & \alpha^{j_2} & \alpha^{in} & \alpha^{in+j_1} & \alpha^{in+j_2} \\ 1 & \alpha^{-j_1} & \alpha^{-j_2} & \alpha^{-in} & \alpha^{-in-j_1} & \alpha^{-in-j_2} \end{pmatrix}$$

is invertible. This matrix is invertible if and only if the  $5 \times 5$  matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 \oplus \alpha^{j_1} & 1 \oplus \alpha^{j_2} & \alpha^{in} & \alpha^{in+j_1} & \alpha^{in+j_2} \\ 1 \oplus \alpha^{-j_1} & 1 \oplus \alpha^{-j_2} & \alpha^{-in} & \alpha^{-in-j_1} & \alpha^{-in-j_2} \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the  $4 \times 4$  matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 \oplus \alpha^{j_2} & \alpha^{in} & 1 \oplus \alpha^{j_1} \oplus \alpha^{in+j_1} & \alpha^{in+j_2} \\ 1 \oplus \alpha^{-j_2} & \alpha^{-in} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-in-j_1} & \alpha^{-in-j_2} \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the  $3 \times 3$  matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha^{in} & 1 \oplus \alpha^{j_1} \oplus \alpha^{in+j_1} & 1 \oplus \alpha^{j_2} \oplus \alpha^{in+j_2} \\ \alpha^{-in} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-in-j_1} & 1 \oplus \alpha^{-j_2} \oplus \alpha^{-in-j_2} \end{pmatrix}$$

is invertible, if and only if the  $2 \times 2$  matrix

$$\begin{pmatrix} (1 \oplus \alpha^{j_1})(1 \oplus \alpha^{in}) & (1 \oplus \alpha^{j_2})(1 \oplus \alpha^{in}) \\ (1 \oplus \alpha^{-j_1})(1 \oplus \alpha^{-in}) & (1 \oplus \alpha^{-j_2})(1 \oplus \alpha^{-in}) \end{pmatrix}$$

is invertible, if and only if, since  $1 \oplus \alpha^{in} \neq 0$ ,

$$\begin{pmatrix} 1 \oplus \alpha^{j_1} & 1 \oplus \alpha^{j_2} \\ 1 \oplus \alpha^{-j_1} & 1 \oplus \alpha^{-j_2} \end{pmatrix} = \begin{pmatrix} 1 \oplus \alpha^{j_1} & 1 \oplus \alpha^{j_2} \\ \alpha^{-j_1}(1 \oplus \alpha^{j_1}) & \alpha^{-j_2}(1 \oplus \alpha^{j_2}) \end{pmatrix}$$

is invertible, if and only if, since  $1 \oplus \alpha^{j_1} \neq 0$  and  $1 \oplus \alpha^{j_2} \neq 0$ ,

$$\begin{pmatrix} 1 & 1 \\ \alpha^{-j_1} & \alpha^{-j_2} \end{pmatrix}$$

is invertible, if and only if  $\alpha^{j_1} \neq \alpha^{j_2}$ , which is the case since  $1 \leq j_1 < j_2 \leq n-1 < 2^b-1$ .

Consider now the second case, then, again by (31), we have to prove that the  $6 \times 6$  matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \alpha^{i_0n+j_0} & \alpha^{i_0n+j_1} & \alpha^{i_1n+j_0} & \alpha^{i_1n+j_1} & \alpha^{i_2n+j_0} & \alpha^{i_2n+j_1} \\ \alpha^{-i_0n-j_0} & \alpha^{-i_0n-j_1} & \alpha^{-i_1n-j_0} & \alpha^{-i_1n-j_1} & \alpha^{-i_2n-j_0} & \alpha^{-i_2n-j_1} \end{pmatrix}$$

is invertible.

Redefining  $i_1 \leftarrow i_1 - i_0$ ,  $i_2 \leftarrow i_2 - i_0$  and  $j \leftarrow j_1 - j_0$ , where  $1 \leq i_1 < i_2 \leq m-1$  and  $1 \leq j \leq n-1$ , the matrix above is invertible if and only if the matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \alpha^j & \alpha^{i_1n} & \alpha^{i_1n+j} & \alpha^{i_2n} & \alpha^{i_2n+j} \\ 1 & \alpha^{-j} & \alpha^{-i_1n} & \alpha^{-i_1n-j} & \alpha^{-i_2n} & \alpha^{-i_2n-j} \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

is invertible. Proceeding with Gaussian elimination like in the previous case, this matrix is invertible if and only if the 2 matrix

$$\begin{pmatrix} (1 \oplus \alpha^j)(1 \oplus \alpha^{i_1 n}) & (1 \oplus \alpha^j)(1 \oplus \alpha^{i_2 n}) \\ \alpha^{-j-i_1 n}(1 \oplus \alpha^j)(1 \oplus \alpha^{i_1 n}) & \alpha^{-j-i_2 n}(1 \oplus \alpha^j)(1 \oplus \alpha^{i_2 n}) \end{pmatrix}$$

is invertible, if and only if  $\alpha^{i_1 n} \neq \alpha^{i_2 n}$ , which is the case since

$$1 \leq i_1 n < i_2 n \leq (m-1)n < 2^b - 1.$$

Next, assume that there are seven erasures, such that each row and column has at least two erasures. This can only happen if one row (column) has three erasures and two rows (columns) have two erasures. The situation is illustrated by the two cases below:

E	E			
E	E		E	
	E		E	

	E	E		
	E	E	E	
	E		E	

Let  $i_0$  be the row with three erasures, and  $j_0$  the column with three erasures, while  $j_1 < j_2$  and  $i_1$  is such that erasures are in  $(i_1, j_0)$  and  $(i_1, j_1)$  so the remaining two erasures are in  $(i_2, j_0)$  and  $(i_2, j_2)$ . It suffices to prove, using the parity-check matrix as given by (31), that the  $7 \times 7$  matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \alpha^{i_0 n + j_0} & \alpha^{i_0 n + j_1} & \alpha^{i_0 n + j_2} & \alpha^{i_1 n + j_0} & \alpha^{i_1 n + j_1} & \alpha^{i_2 n + j_0} & \alpha^{i_2 n + j_2} \\ \alpha^{-i_0 n - j_0} & \alpha^{-i_0 n - j_1} & \alpha^{-i_0 n - j_2} & \alpha^{-i_1 n - j_0} & \alpha^{-i_1 n - j_1} & \alpha^{-i_2 n - j_0} & \alpha^{-i_2 n - j_2} \end{pmatrix}$$

is invertible.

Redefining  $i_1 \leftarrow i_1 - i_0$ ,  $i_2 \leftarrow i_2 - i_0$ ,  $j_1 \leftarrow j_1 - j_0$  and  $j_2 \leftarrow j_2 - j_0$ , the matrix above is invertible if and only if the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & \alpha^{j_1} & \alpha^{j_2} & \alpha^{i_1 n} & \alpha^{i_1 n + j_1} & \alpha^{i_2 n} & \alpha^{i_2 n + j_2} \\ 1 & \alpha^{-j_1} & \alpha^{-j_2} & \alpha^{-i_1 n} & \alpha^{-i_1 n - j_1} & \alpha^{-i_2 n} & \alpha^{-i_2 n - j_2} \end{pmatrix}$$



is invertible, if and only if, doing Gaussian elimination like in the other two cases, the  $2 \times 2$  matrix

$$\begin{pmatrix} (1 \oplus \alpha^{j_1})(1 \oplus \alpha^{i_1 n}) & (1 \oplus \alpha^{j_2})(1 \oplus \alpha^{i_2 n}) \\ (1 \oplus \alpha^{-j_1})(1 \oplus \alpha^{-i_1 n}) & (1 \oplus \alpha^{-j_2})(1 \oplus \alpha^{-i_2 n}) \end{pmatrix}$$

is invertible, if and only if, since  $1 \oplus \alpha^{j_1}$ ,  $1 \oplus \alpha^{i_1 n}$ ,  $1 \oplus \alpha^{j_2}$  and  $1 \oplus \alpha^{i_2 n}$  are non-zero,

$$\begin{pmatrix} 1 & 1 \\ \alpha^{-i_1 n - j_1} & \alpha^{-i_2 n - j_2} \end{pmatrix}$$

is invertible, if and only if, computing the determinant,

$$\alpha^{i_1 n + j_1} \neq \alpha^{i_2 n + j_2}$$

which is the case since, being  $\alpha$  a primitive element in  $GF(2^b)$  and  $mn < 2^b - 1$ ,

$$(i_2 - i_1)n + j_2 - j_1 \not\equiv 0 \pmod{2^b - 1}.$$

□

Lemma 3.2 shows that the code given by parity-check matrix  $\mathcal{H}_2$  meets bound (30) with equality, something that could not be achieved by GPC codes with two global parities. However, there is a tradeoff. Consider the 3-level GPC code  $\mathcal{C}(n; m-1, \underline{u})$ , where

$$\underline{u} = (\overbrace{1, 1, \dots, 1}^{m-3}, 2, 3, 3).$$

This is an  $EP(n, 1; m, 1; 3)$  code. According to Corollary 2.2,  $\mathcal{C}(n; m-1, \underline{u})$  has minimum distance 8, the same as the code given by parity-check matrix  $\mathcal{H}_2$ , at the price of an extra parity. However, the size of the field required by  $\mathcal{C}(n; m-1, \underline{u})$  is much smaller, greater than  $\max\{m; n\}$ , as opposed to greater than  $mn$ . Also, by Theorem 2.1,  $\mathcal{C}(n; m-1, \underline{u})$  can correct 8 erasures involving two rows with 3 erasures and one row with two erasures, like for example

E	E		E	
E	E			
E	E		E	

The code generated by  $\mathcal{H}_2$  is unable to correct such pattern since it does not have enough parities, so even if both codes have the same minimum distance,  $\mathcal{C}(n; m-1, \underline{u})$  can correct more erasure patterns. These tradeoffs need to be evaluated when implementation is considered.

Let us finish this section with the case of  $EP(n, 1; m, 1; 3)$  codes. Bound (28) gives

$$d(1, 1, 3) \leq 10.$$

The next lemma shows that this bound can be improved.

**Lemma 3.3** Let  $d(1, 1, 3)$  be the minimum distance of an  $EP(m, 1; n, 1; 3)$  code. Then,

$$d(1, 1, 3) \leq 9. \quad (32)$$

**Proof:** Assume that the zero array is stored, and the received array has the locations  $(i, j)$  erased, where  $0 \leq i \leq 2$  and  $0 \leq j \leq 2$  (in particular, we have 9 erasures). We claim that such a received array is uncorrectable. Notice that we have 3 horizontal parities and 3 vertical parities corresponding to the product code in order to correct such erasures, but one of such parities is dependent, so that leaves us with a total of 5 parities corresponding to the product code. In addition, 3 extra global parities can be used to correct these 9 erasures, giving a total of 8 parities, which are insufficient and proving (32).  $\square$

The next question is if bound (32) is tight. The answer is yes.

As in the case of two global parities, let  $GF(2^b)$  be a finite field and let  $\alpha$  be an element in  $GF(2^b)$  such that  $\mathcal{O}(\alpha) > mn$ . Consider the parity-check matrix  $\mathcal{H}_3$  given by

$$\mathcal{H}_3 = \begin{pmatrix} I_m \otimes \overbrace{(1, 1, \dots, 1)}^n \\ \overbrace{(1, 1, \dots, 1)}^m \otimes I_n \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{mn-1} \\ 1 & \alpha^{-1} & \alpha^{-2} & \dots & \alpha^{-(mn-1)} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(mn-1)} \end{pmatrix}. \quad (33)$$

Notice that  $\mathcal{H}_2$  as given by (31) consists of the first  $m + n + 2$  rows of  $\mathcal{H}_3$ . The following lemma gives the minimum distance of these codes under a certain condition.

**Lemma 3.4** The  $EP(m, 1; n, 1; 3)$  code whose parity-check matrix  $\mathcal{H}_3$  is given by (33) has minimum distance 9, where  $mn < \mathcal{O}(\alpha)$ , if and only if, for any  $i_1, i_2 \neq 0$ ,  $1 \leq i_1 \leq m - 1$ ,  $1 \leq |i_2| \leq m - 1$  and  $j_1, j_2 \neq 0$ ,  $1 \leq j_1 \leq n - 1$ ,  $1 \leq |j_2| \leq n - 1$ ,

$$1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_2 n + j_2} \oplus \alpha^{-(i_2 - i_1)n + j_2} \neq 0. \quad (34)$$

**Proof:** We have to prove that any 8 erasures are going to be corrected under condition (34).

Assume that there are 8 erasures, such that each row and column has at least two erasures. There are three situations under which this can happen:

1. Two rows have four erasures and four columns have two erasures.
2. Four rows have two erasures and two columns have four erasures.
3. Two rows (columns) have three erasures and one row (column) has two erasures.

The situation is illustrated by the four cases below. The first array illustrates case 1, the second array illustrates case 2, and the third and fourth arrays illustrate case 3.

E	E		E	E
E	E		E	E

	E	E		
	E	E		
	E	E		
	E	E		

E	E		E	
E	E		E	
	E		E	

	E	E		
	E	E	E	
	E	E	E	

Consider case 1 and assume that the erasures occurred in locations  $(i_0, j_0)$ ,  $(i_0, j_1)$ ,  $(i_0, j_2)$ ,  $(i_0, j_3)$ ,  $(i_1, j_0)$ ,  $(i_1, j_1)$ ,  $(i_1, j_2)$  and  $(i_1, j_3)$ , where  $0 \leq i_0 < i_1 \leq m - 1$  and  $0 \leq j_0 < j_1 < j_2 < j_3 \leq n - 1$ . We need to prove, using the parity-check matrix as given by (33), that the  $8 \times 8$  matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \alpha^{i_0n+j_0} & \alpha^{i_0n+j_1} & \alpha^{i_0n+j_2} & \alpha^{i_0n+j_3} & \alpha^{i_1n+j_0} & \alpha^{i_1n+j_1} & \alpha^{i_1n+j_2} & \alpha^{i_1n+j_3} \\ \alpha^{-i_0n-j_0} & \alpha^{-i_0n-j_1} & \alpha^{-i_0n-j_2} & \alpha^{-i_0n-j_3} & \alpha^{-i_1n-j_0} & \alpha^{-i_1n-j_1} & \alpha^{-i_1n-j_2} & \alpha^{-i_1n-j_3} \\ \alpha^{2(i_0n+j_0)} & \alpha^{2(i_0n+j_1)} & \alpha^{2(i_0n+j_2)} & \alpha^{2(i_0n+j_3)} & \alpha^{2(i_1n+j_0)} & \alpha^{2(i_1n+j_1)} & \alpha^{2(i_1n+j_2)} & \alpha^{2(i_1n+j_3)} \end{pmatrix}$$

is invertible. Redefining  $i \leftarrow i_1 - i_0$ ,  $j_1 \leftarrow j_1 - j_0$ ,  $j_2 \leftarrow j_2 - j_0$  and  $j_3 \leftarrow j_3 - j_0$ , where now  $1 \leq i \leq m - 1$  and  $1 \leq j_1 < j_2 < j_3 \leq n - 1$ , this matrix is invertible if and only if matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & \alpha^{j_1} & \alpha^{j_2} & \alpha^{j_3} & \alpha^{in} & \alpha^{in+j_1} & \alpha^{in+j_2} & \alpha^{in+j_3} \\ 1 & \alpha^{-j_1} & \alpha^{-j_2} & \alpha^{-j_3} & \alpha^{-in} & \alpha^{-in-j_1} & \alpha^{-in-j_2} & \alpha^{-in-j_3} \\ 1 & \alpha^{2j_1} & \alpha^{2j_2} & \alpha^{2j_3} & \alpha^{2in} & \alpha^{2(in+j_1)} & \alpha^{2(in+j_2)} & \alpha^{2(in+j_3)} \end{pmatrix}$$

is invertible. This matrix is invertible if and only if the  $7 \times 7$  matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 \oplus \alpha^{j_1} & 1 \oplus \alpha^{j_2} & 1 \oplus \alpha^{j_3} & \alpha^{in} & \alpha^{in+j_1} & \alpha^{in+j_2} & \alpha^{in+j_3} \\ 1 \oplus \alpha^{-j_1} & 1 \oplus \alpha^{-j_2} & 1 \oplus \alpha^{-j_3} & \alpha^{-in} & \alpha^{-in-j_1} & \alpha^{-in-j_2} & \alpha^{-in-j_3} \\ 1 \oplus \alpha^{2j_1} & 1 \oplus \alpha^{2j_2} & 1 \oplus \alpha^{2j_3} & \alpha^{2in} & \alpha^{2(in+j_1)} & \alpha^{2(in+j_2)} & \alpha^{2(in+j_3)} \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the  $6 \times 6$  matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 \oplus \alpha^{j_2} & 1 \oplus \alpha^{j_3} & \alpha^{in} & 1 \oplus \alpha^{j_1} \oplus \alpha^{in+j_1} & \alpha^{in+j_2} & \alpha^{in+j_3} \\ 1 \oplus \alpha^{-j_2} & 1 \oplus \alpha^{-j_3} & \alpha^{-in} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-in-j_1} & \alpha^{-in-j_2} & \alpha^{-in-j_3} \\ 1 \oplus \alpha^{2j_2} & 1 \oplus \alpha^{2j_3} & \alpha^{2in} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2(in+j_1)} & \alpha^{2(in+j_2)} & \alpha^{2(in+j_3)} \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the  $5 \times 5$  matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 \oplus \alpha^{j_3} & \alpha^{in} & 1 \oplus \alpha^{j_1} \oplus \alpha^{in+j_1} & 1 \oplus \alpha^{j_2} \oplus \alpha^{in+j_2} & \alpha^{in+j_3} \\ 1 \oplus \alpha^{-j_3} & \alpha^{-in} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-in-j_1} & 1 \oplus \alpha^{-j_2} \oplus \alpha^{-in-j_2} & \alpha^{-in-j_3} \\ 1 \oplus \alpha^{2j_3} & \alpha^{2in} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2(in+j_1)} & 1 \oplus \alpha^{2j_2} \oplus \alpha^{2(in+j_2)} & \alpha^{2(in+j_3)} \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the  $4 \times 4$  matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ \alpha^{in} & 1 \oplus \alpha^{j_1} \oplus \alpha^{in+j_1} & 1 \oplus \alpha^{j_2} \oplus \alpha^{in+j_2} & 1 \oplus \alpha^{j_3} \oplus \alpha^{in+j_3} \\ \alpha^{-in} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-in-j_1} & 1 \oplus \alpha^{-j_2} \oplus \alpha^{-in-j_2} & 1 \oplus \alpha^{-j_3} \oplus \alpha^{-in-j_3} \\ \alpha^{2in} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2(in+j_1)} & 1 \oplus \alpha^{2j_2} \oplus \alpha^{2(in+j_2)} & 1 \oplus \alpha^{2j_3} \oplus \alpha^{2(in+j_3)} \end{pmatrix}$$

is invertible, if and only if the  $3 \times 3$  matrix

$$\begin{pmatrix} (1 \oplus \alpha^{j_1})(1 \oplus \alpha^{in}) & (1 \oplus \alpha^{j_2})(1 \oplus \alpha^{in}) & (1 \oplus \alpha^{j_3})(1 \oplus \alpha^{in}) \\ (1 \oplus \alpha^{-j_1})(1 \oplus \alpha^{-in}) & (1 \oplus \alpha^{-j_2})(1 \oplus \alpha^{-in}) & (1 \oplus \alpha^{-j_3})(1 \oplus \alpha^{-in}) \\ (1 \oplus \alpha^{2j_1})(1 \oplus \alpha^{2in}) & (1 \oplus \alpha^{2j_2})(1 \oplus \alpha^{2in}) & (1 \oplus \alpha^{2j_3})(1 \oplus \alpha^{2in}) \end{pmatrix}$$

is invertible, if and only if the  $3 \times 3$  matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha^{-j_1} & \alpha^{-j_2} & \alpha^{-j_3} \\ 1 \oplus \alpha^{j_1} & 1 \oplus \alpha^{j_2} & 1 \oplus \alpha^{j_3} \end{pmatrix}$$

is invertible, if and only if the  $2 \times 2$  matrix

$$\begin{pmatrix} \alpha^{-j_1} \oplus \alpha^{-j_2} & \alpha^{-j_1} \oplus \alpha^{-j_3} \\ \alpha^{j_1} \oplus \alpha^{j_2} & \alpha^{j_1} \oplus \alpha^{j_3} \end{pmatrix}$$

is invertible, if and only if the  $2 \times 2$  matrix

$$\begin{pmatrix} 1 & 1 \\ \alpha^{-j_1-j_2} & \alpha^{-j_1-j_3} \end{pmatrix}$$

is invertible, if and only if, computing the determinant of the matrix above,

$$\alpha^{-j_1}(\alpha^{-j_2} \oplus \alpha^{-j_3}) \neq 0,$$

which is true since  $1 \leq j_2 < j_3 \leq n-1$ . So all these cases of 8 erasures are correctable.

Consider next case 2 and assume that the erasures occurred in locations  $(i_0, j_0)$ ,  $(i_0, j_1)$ ,  $(i_1, j_0)$ ,  $(i_1, j_1)$ ,  $(i_2, j_0)$ ,  $(i_2, j_1)$ ,  $(i_3, j_0)$  and  $(i_3, j_1)$ , where  $0 \leq i_0 < i_1 < i_2 < i_3 \leq m-1$  and  $0 \leq j_0 < j_1 \leq n-1$ . We need to prove, using the parity-check matrix as given by (33), that the  $8 \times 8$  matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \alpha^{i_0n+j_0} & \alpha^{i_0n+j_1} & \alpha^{i_1n+j_0} & \alpha^{i_1n+j_1} & \alpha^{i_2n+j_0} & \alpha^{i_2n+j_1} & \alpha^{i_3n+j_0} & \alpha^{i_3n+j_1} \\ \alpha^{-i_0n-j_0} & \alpha^{-i_0n-j_1} & \alpha^{-i_1n-j_0} & \alpha^{-i_1n-j_1} & \alpha^{-i_2n-j_0} & \alpha^{-i_2n-j_1} & \alpha^{-i_3n-j_0} & \alpha^{-i_3n-j_1} \\ \alpha^{2(i_0n+j_0)} & \alpha^{2(i_0n+j_1)} & \alpha^{2(i_1n+j_0)} & \alpha^{2(i_1n+j_1)} & \alpha^{2(i_2n+j_0)} & \alpha^{2(i_2n+j_1)} & \alpha^{2(i_3n+j_0)} & \alpha^{2(i_3n+j_1)} \end{pmatrix}$$

is invertible. Redefining  $i_1 \leftarrow i_1 - i_0$ ,  $i_2 \leftarrow i_2 - i_0$ ,  $i_3 \leftarrow i_3 - i_0$  and  $j \leftarrow j_1 - j_0$ , where now  $1 \leq i_1 < i_2 < i_3 \leq m-1$  and  $1 \leq j \leq n-1$ , this matrix is invertible if and only if matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & \alpha^j & \alpha^{i_1n} & \alpha^{i_1n+j} & \alpha^{i_2n} & \alpha^{i_2n+j} & \alpha^{i_3n} & \alpha^{i_3n+j} \\ 1 & \alpha^{-j} & \alpha^{-i_1n} & \alpha^{-i_1n-j} & \alpha^{-i_2n} & \alpha^{-i_2n-j} & \alpha^{-i_3n} & \alpha^{-i_3n-j} \\ 1 & \alpha^{2j} & \alpha^{2i_1n} & \alpha^{2(i_1n+j)} & \alpha^{2i_2n} & \alpha^{2(i_2n+j)} & \alpha^{2i_3n} & \alpha^{2(i_3n+j)} \end{pmatrix}$$

is invertible. This matrix is invertible if and only if the  $4 \times 4$  matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 \oplus \alpha^j & \alpha^{i_1n}(1 \oplus \alpha^j) & \alpha^{i_2n}(1 \oplus \alpha^j) & \alpha^{i_3n}(1 \oplus \alpha^j) \\ 1 \oplus \alpha^{-j} & \alpha^{-i_1n}(1 \oplus \alpha^{-j}) & \alpha^{-i_2n}(1 \oplus \alpha^{-j}) & \alpha^{-i_3n}(1 \oplus \alpha^{-j}) \\ 1 \oplus \alpha^{2j} & \alpha^{2i_1n}(1 \oplus \alpha^{2j}) & \alpha^{2i_2n}(1 \oplus \alpha^{2j}) & \alpha^{2i_3n}(1 \oplus \alpha^{2j}) \end{pmatrix}$$

is invertible, if and only if the  $3 \times 3$  matrix

$$\begin{pmatrix} (1 \oplus \alpha^{i_1 n})(1 \oplus \alpha^j) & (1 \oplus \alpha^{i_2 n})(1 \oplus \alpha^j) & (1 \oplus \alpha^{i_3 n})(1 \oplus \alpha^j) \\ (1 \oplus \alpha^{-i_1 n})(1 \oplus \alpha^{-j}) & (1 \oplus \alpha^{-i_2 n})(1 \oplus \alpha^{-j}) & (1 \oplus \alpha^{-i_3 n})(1 \oplus \alpha^{-j}) \\ (1 \oplus \alpha^{2i_1 n})(1 \oplus \alpha^{2j}) & (1 \oplus \alpha^{2i_2 n})(1 \oplus \alpha^{2j}) & (1 \oplus \alpha^{2i_3 n})(1 \oplus \alpha^{2j}) \end{pmatrix}$$

is invertible, if and only if the  $3 \times 3$  matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha^{-i_1 n} & \alpha^{-i_2 n} & \alpha^{-i_3 n} \\ 1 \oplus \alpha^{i_1 n} & 1 \oplus \alpha^{i_2 n} & 1 \oplus \alpha^{i_3 n} \end{pmatrix}$$

is invertible, if and only if the  $2 \times 2$  matrix

$$\begin{pmatrix} \alpha^{i_1 n} \oplus \alpha^{i_2 n} & \alpha^{i_1 n} \oplus \alpha^{i_3 n} \\ \alpha^{-i_1 n} \oplus \alpha^{-i_2 n} & \alpha^{-i_1 n} \oplus \alpha^{-i_3 n} \end{pmatrix} = (\alpha^{i_1 n} \oplus \alpha^{i_2 n})(\alpha^{i_1 n} \oplus \alpha^{i_3 n}) \begin{pmatrix} 1 & 1 \\ \alpha^{(-i_1-i_2)n} & \alpha^{(-i_1-i_2)n} \end{pmatrix}$$

is invertible, if and only if, computing the determinant of this last matrix,

$$\alpha^{(-i_1-i_2)n} \oplus \alpha^{(-i_1-i_2)n} = \alpha^{-i_1 n}(\alpha^{-i_2 n} \oplus \alpha^{-i_3 n}) \neq 0,$$

which is certainly the case since  $1 \leq i_2 n < i_3 n < mn < \mathcal{O}(\alpha)$ .

Consider finally case 3. Let  $i_0 < i_1$  and  $j_0 < j_1$  be the rows and columns respectively with three erasures. It suffices to prove, using the parity-check matrix  $\mathcal{H}_3$  given by (33), that the  $8 \times 8$  matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \alpha^{i_0 n+j_0} & \alpha^{i_0 n+j_1} & \alpha^{i_0 n+j_2} & \alpha^{i_1 n+j_0} & \alpha^{i_1 n+j_1} & \alpha^{i_1 n+j_2} & \alpha^{i_2 n+j_0} & \alpha^{i_2 n+j_1} \\ \alpha^{-i_0 n-j_0} & \alpha^{-i_0 n-j_1} & \alpha^{-i_0 n-j_2} & \alpha^{-i_1 n-j_0} & \alpha^{-i_1 n-j_1} & \alpha^{-i_1 n-j_2} & \alpha^{-i_2 n-j_0} & \alpha^{-i_2 n-j_1} \\ \alpha^{2(i_0 n+j_0)} & \alpha^{2(i_0 n+j_1)} & \alpha^{2(i_0 n+j_2)} & \alpha^{2(i_1 n+j_0)} & \alpha^{2(i_1 n+j_1)} & \alpha^{2(i_1 n+j_2)} & \alpha^{2(i_2 n+j_0)} & \alpha^{2(i_2 n+j_1)} \end{pmatrix}$$

is invertible.

Redefining  $i_1 \leftarrow i_1 - i_0$ ,  $i_2 \leftarrow i_2 - i_0$ ,  $j_1 \leftarrow j_1 - j_0$  and  $j_2 \leftarrow j_2 - j_0$ , the matrix above is invertible if and only if the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & \alpha^{j_1} & \alpha^{j_2} & \alpha^{i_1 n} & \alpha^{i_1 n + j_1} & \alpha^{i_1 n + j_2} & \alpha^{i_2 n} & \alpha^{i_2 n + j_1} \\ 1 & \alpha^{-j_1} & \alpha^{-j_2} & \alpha^{-i_1 n - j_0} & \alpha^{-i_1 n - j_1} & \alpha^{-i_1 n - j_2} & \alpha^{-i_2 n} & \alpha^{-i_2 n - j_1} \\ 1 & \alpha^{2j_1} & \alpha^{2j_2} & \alpha^{2i_1 n} & \alpha^{2(i_1 n + j_1)} & \alpha^{2(i_1 n + j_2)} & \alpha^{2i_2 n} & \alpha^{2(i_2 n + j_1)} \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 \oplus \alpha^{j_1} & 1 \oplus \alpha^{j_2} & \alpha^{i_1 n} & \alpha^{i_1 n + j_1} & \alpha^{i_1 n + j_2} & \alpha^{i_2 n} & \alpha^{i_2 n + j_1} \\ 1 \oplus \alpha^{-j_1} & 1 \oplus \alpha^{-j_2} & \alpha^{-i_1 n - j_0} & \alpha^{-i_1 n - j_1} & \alpha^{-i_1 n - j_2} & \alpha^{-i_2 n} & \alpha^{-i_2 n - j_1} \\ 1 \oplus \alpha^{2j_1} & 1 \oplus \alpha^{2j_2} & \alpha^{2i_1 n} & \alpha^{2(i_1 n + j_1)} & \alpha^{2(i_1 n + j_2)} & \alpha^{2i_2 n} & \alpha^{2(i_2 n + j_1)} \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 \oplus \alpha^{j_2} & \alpha^{i_1 n} & 1 \oplus \alpha^{j_1} \oplus \alpha^{i_1 n + j_1} & \alpha^{i_1 n + j_2} & \alpha^{i_2 n} & 1 \oplus \alpha^{j_1} \oplus \alpha^{i_2 n + j_1} \\ 1 \oplus \alpha^{-j_2} & \alpha^{-i_1 n} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_1 n - j_1} & \alpha^{-i_1 n - j_2} & \alpha^{-i_2 n} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_2 n - j_1} \\ 1 \oplus \alpha^{2j_2} & \alpha^{2i_1 n} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2(i_1 n + j_1)} & \alpha^{2(i_1 n + j_2)} & \alpha^{2i_2 n} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2(i_2 n + j_1)} \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ \alpha^{i_1 n} & 1 \oplus \alpha^{j_1} \oplus \alpha^{i_1 n + j_1} & 1 \oplus \alpha^{j_2} \oplus \alpha^{i_1 n + j_2} & \alpha^{i_2 n} & 1 \oplus \alpha^{j_1} \oplus \alpha^{i_2 n + j_1} \\ \alpha^{-i_1 n} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_1 n - j_1} & 1 \oplus \alpha^{-j_2} \oplus \alpha^{-i_1 n - j_2} & \alpha^{-i_2 n} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_2 n - j_1} \\ \alpha^{2i_1 n} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2(i_1 n + j_1)} & \alpha^{2(i_1 n + j_2)} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2i_2 n} & 1 \oplus \alpha^{2j_2} \oplus \alpha^{2(i_2 n + j_1)} \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} (1 \oplus \alpha^{i_1 n})(1 \oplus \alpha^{j_1}) & (1 \oplus \alpha^{i_1 n})(1 \oplus \alpha^{j_2}) & \alpha^{i_2 n} & 1 \oplus \alpha^{j_1} \oplus \alpha^{i_2 n + j_1} \\ (1 \oplus \alpha^{-i_1 n})(1 \oplus \alpha^{-j_1}) & (1 \oplus \alpha^{-i_1 n})(1 \oplus \alpha^{-j_2}) & \alpha^{-i_2 n} & 1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_2 n - j_1} \\ (1 \oplus \alpha^{2i_1 n})(1 \oplus \alpha^{2j_1}) & (1 \oplus \alpha^{2i_1 n})(1 \oplus \alpha^{2j_2}) & \alpha^{2i_2 n} & 1 \oplus \alpha^{2j_1} \oplus \alpha^{2(i_2 n + j_1)} \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} (1 \oplus \alpha^{i_1 n})(1 \oplus \alpha^{j_1}) & (1 \oplus \alpha^{i_1 n})(1 \oplus \alpha^{j_2}) & (1 \oplus \alpha^{i_2 n})(1 \oplus \alpha^{j_1}) \\ (1 \oplus \alpha^{-i_1 n})(1 \oplus \alpha^{-j_1}) & (1 \oplus \alpha^{-i_1 n})(1 \oplus \alpha^{-j_2}) & (1 \oplus \alpha^{-i_2 n})(1 \oplus \alpha^{-j_1}) \\ (1 \oplus \alpha^{2i_1 n})(1 \oplus \alpha^{2j_1}) & (1 \oplus \alpha^{2i_1 n})(1 \oplus \alpha^{2j_2}) & (1 \oplus \alpha^{2i_2 n})(1 \oplus \alpha^{2j_1}) \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha^{-i_1 n - j_1} & \alpha^{-i_1 n - j_2} & \alpha^{-i_2 n - j_1} \\ (1 \oplus \alpha^{i_1 n})(1 \oplus \alpha^{j_1}) & (1 \oplus \alpha^{i_1 n})(1 \oplus \alpha^{j_2}) & (1 \oplus \alpha^{i_2 n})(1 \oplus \alpha^{j_1}) \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} \alpha^{-i_1 n - j_1}(\alpha^{-j_1} \oplus \alpha^{-j_2}) & \alpha^{-j_1}(\alpha^{-i_1 n} \oplus \alpha^{-i_2 n}) \\ (1 \oplus \alpha^{i_1 n})(\alpha^{j_1} \oplus \alpha^{j_2}) & (1 \oplus \alpha^{j_1})(\alpha^{i_1 n} \oplus \alpha^{i_2 n}) \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} \alpha^{-i_1 n - 2j_1 - j_2} & \alpha^{-(i_1 + i_2)n - j_1} \\ 1 \oplus \alpha^{i_1 n} & 1 \oplus \alpha^{j_1} \end{pmatrix}$$

is invertible, if and only if the matrix

$$\begin{pmatrix} \alpha^{-j_1 - j_2} & \alpha^{-i_2 n} \\ 1 \oplus \alpha^{i_1 n} & 1 \oplus \alpha^{j_1} \end{pmatrix}$$

which is invertible if and only if its determinant, which is given by the left hand side of (34) times a constant, is non-zero.  $\square$

Notice that in Lemma 3.4, 8 erasures following the patterns of cases 1 and 2 will always be corrected, while case 3 will be corrected only when condition (34) is satisfied. So Lemma 3.4 by itself does not prove that there is an  $EP(m, 1; n, 1; 3)$  code with minimum distance 9, but we can find a code satisfying (34) using an argument similar to the one used to show an infinite family of PMDS codes in [1]. In effect, consider the field  $GF(2^p)$ ,  $p$  a prime number, such that  $GF(2^p)$  is generated by the irreducible polynomial  $M_p(x) = 1 + x + x^2 + \dots + x^{p-1}$ . The polynomial  $M_p(x)$  may not be irreducible, for example,  $M_5(x)$  is irreducible but  $M_7(x) = (1 + x + x^3)(1 + x^2 + x^3)$ , so not any prime number can be chosen. If we choose a prime number large enough, condition (34) will hold, as shown in the next corollary:

**Corollary 3.1** Consider the  $EP(m, 1; n, 1; 3)$  code whose parity-check matrix is given by (33) with  $\alpha$  in (33) a zero of  $M_p(x)$ ,  $p$  a prime number,  $M_p(x)$  irreducible and  $mn < p$ . Then the code has minimum distance 9.



**Proof:** We have to show that (34) is satisfied. Given an integer  $z$ , denote by  $\langle z \rangle_p$  the unique integer  $u$ ,  $0 \leq u \leq p-1$ , such that  $u \equiv z \pmod{p}$ . Let  $M_p(\alpha) = 0$ , then  $\mathcal{O}(\alpha) = p$ . Hence,

$$1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_2 n + j_2} \oplus \alpha^{-(i_2 - i_1)n + j_2} = 1 \oplus \alpha^{\langle -j_1 \rangle_p} \oplus \alpha^{\langle -i_2 n + j_2 \rangle_p} \oplus \alpha^{\langle -(i_2 - i_1)n + j_2 \rangle_p}.$$

Take the first three elements in (34), i.e.,

$$1 \oplus \alpha^{-j_1} \oplus \alpha^{-i_2 n + j_2} = 1 \oplus \alpha^{\langle -j_1 \rangle_p} \oplus \alpha^{\langle -i_2 n + j_2 \rangle_p}.$$

Since  $1 \leq j_1 \leq n-1$ ,  $1 \leq |j_2| \leq n-1$ ,  $1 \leq |i_2| \leq m-1$  and  $mn < p$ ,  $\langle -j_1 \rangle_p = p - j_1 \neq 0$ .

Assume that  $\langle -i_2 n + j_2 \rangle_p = 0$ , then  $-i_2 n + j_2 = sp$  for some integer  $s$ . If  $s = 0$ , then  $i_2 n = j_2$ , a contradiction since  $1 \leq |j_2| \leq n-1$  and  $1 \leq |i_2| \leq m-1$ . So  $s \neq 0$  and  $j_2 = sp + i_2 n$ . If  $s \geq 1$ , since  $1 \leq |i_2| \leq m-1$  and  $mn < p$ ,  $j_2 = sp + i_2 n \geq mn - (m-1)n = n$ , a contradiction since  $1 \leq |j_2| \leq n-1$ . If  $s < 0$ ,  $j_2 = sp + i_2 n \leq -mn + (m-1)n = -n$ , also a contradiction.

If  $\langle -j_1 \rangle_p \neq \langle -i_2 n + j_2 \rangle_p$ , then the first three elements in (34) are distinct from each other, so they cannot be canceled by the 4th element and (34) holds. So, assume that  $\langle -j_1 \rangle_p = \langle -i_2 n + j_2 \rangle_p$ . In particular,

$$-i_2 n + j_2 = -j_1 + sp \text{ for some integer } s. \quad (35)$$

Now, in order for the left hand side of (34) to be zero, in addition to (35), we need  $\langle -(i_2 - i_1)n + j_2 \rangle_p = 0$ , giving

$$-i_2 n + j_2 = -i_1 n + s'p \text{ for some integer } s'. \quad (36)$$

Combining (35) and (36), we obtain

$$i_1 n - j_1 = s''p \text{ for some integer } s''. \quad (37)$$

Since  $1 \leq i_1 \leq m-1$  and  $1 \leq j_1 \leq n-1$ ,

$$1 \leq i_1 n - j_1 < mn < p,$$

contradicting (37) and completing the proof.  $\square$

Corollary 3.1 shows that bound (32) is indeed tight.

The construction in Corollary 3.1 depends on, for each  $m \times n$  array, finding a prime  $p$  such that  $M_p(x)$  is irreducible (it is well known that  $M_p(x)$  is irreducible if and only if 2 is primitive in  $GF(p)$  [14]). Strictly speaking, it is not proven that the number of such primes is infinite, but it is believed it is, and from a practical point of view, it is always possible to find such a large enough prime number.

## 4 Conclusions

We have studied extended product (EPC) codes, in which a few global parities are added to a traditional product code in order to enhance its distance properties. We presented a special case of extended product codes: generalized product (GPC) codes. We showed that GPC codes unify two types of codes: product codes and integrated interleaved (II) codes. We studied the distance properties of these type of codes. Although, except for the special case of one global parity, GPC codes do not optimize the minimum distance, they can be implemented with modest field size, and in addition they provide a large variety of possible parameters, making them an attractive alternative for implementation in practical cases. We showed some optimal constructions for the cases of two and three global parities, requiring a larger field size.

## References

- [1] M. Blaum, J. L. Hafner and S. Hetzler, “Partial-MDS Codes and their Application to RAID Type of Architectures,” *IEEE Trans. on Information Theory*, vol. IT-59, pp. 4510–19, July 2013.
- [2] M. Blaum and S. Hetzler, “Integrated Interleaved Codes as Locally Recoverable Codes,” *Int. J. Information and Coding Theory*, Vol. 3, No. 4, pp. 324–344, September 2016.
- [3] M. Blaum, J. S. Plank, M. Schwartz and E. Yaakobi, “Partial MDS (PMDS) and Sector-Disk (SD) Codes that Tolerate the Erasure of Two Random Sectors,” *IEEE Trans. on Information Theory*, vol. IT-62, pp. 2673–81, May 2016.
- [4] E. L. Blokh and V. V. Zyablov, “Coding of Generalized Concatenated Codes,” *Problemy Peredachii Informatsii*, Vol. 10(3), pp. 218–222, 1974.
- [5] G. A. Gibson, “Redundant Disk Arrays,” MIT Press, 1992.
- [6] P. Gopalan, C. Huang, B. Jenkins and S. Yekhanin, “Explicit Maximally Recoverable Codes with Locality,” *IEEE Trans. on Information Theory*, vol. IT-60, pp. 5245–56, September 2014.
- [7] P. Gopalan, C. Huang, H. Simitci and S. Yekhanin, “On the Locality of Codeword Symbols,” *IEEE Trans. on Information Theory*, vol. IT-58, pp. 6925–34, November 2012.
- [8] P. Gopalan, G. Hu, S. Saraf, C. Wang and S. Yekhanin, “Maximally Recoverable Codes for Grid-like Topologies,” *arXiv 1605.05412v1*, May 2016.
- [9] M. Hassner, K. Abdel-Ghaffar, A. Patel, R. Koetter and B. Trager, “Integrated Interleaving – A Novel ECC Architecture,” *IEEE Transactions on Magnetics*, Vol. 37, No. 2, pp. 773–5, March 2001.

- [10] C. Huang, M. Chen, and J. Li, “Pyramid Codes: Flexible Schemes to Trade Space for Access Efficiency in Reliable Data Storage Systems,” Proc. of IEEE NCA, Cambridge, Massachusetts, July 2007.
- [11] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li and S. Yekhanin, “Erasure Coding in Windows Azure Storage,” 2012 USENIX Annual Technical Conference, Boston, Massachusetts, June 2012.
- [12] M. Kuijper and D. Napp, “Erasure Codes with Simplex Locality,” arXiv:1403.2779, March 2014.
- [13] M. Li and P. C. Lee, “STAIR Codes: A General Family of Erasure Codes for Tolerating Device and Sector Failures in Practical Storage Systems,” 12th USENIX Conference on File and Storage Technologies (FAST 14), Santa Clara, CA, February 2014.
- [14] F. J. MacWilliams and N. J. A. Sloane, “The Theory of Error-Correcting Codes,” North Holland, Amsterdam, 1977.
- [15] Micron, “N-29-17: NAND Flash Design and Use Considerations Introduction,” <http://download.micron.com/pdf/technotes/nand/tn2917.pdf>.
- [16] J. S. Plank and M. Blaum, “Sector-Disk (SD) Erasure Codes for Mixed Failure Modes in RAID Systems,” ACM Transactions on Storage, Vol. 10, No. 1, Article 4, January 2014.
- [17] J. S. Plank, M. Blaum and J. L. Hafner, “SD Codes: Erasure Codes Designed for how Storage Systems Really Fail,” 11th USENIX Conference on File and Storage Technologies (FAST 13), Santa Clara, CA, February 2013.
- [18] D. S. Papailiopoulos and A. G. Dimakis, “Locally Repairable Codes,” IEEE Trans. on Information Theory, vol. IT-60, pp. 5843-55, October 2014.
- [19] N. Prakash, G. M. Kamath, V. Lalitha and P. V. Kumar, “Optimal Linear Codes with a Local-Error-Correction Property,” ISIT 2012, IEEE International Symposium on Information Theory, pp.2776–80, July 2012.
- [20] A. S. Rawat, O. O. Koyluoglu, N. Silberstein and S. Vishwanath, “Optimal Locally Repairable and Secure Codes for Distributed Storage Systems,” IEEE Trans. on Information Theory, vol. IT-60, pp. 212-36, January 2014.
- [21] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis and S. Vishwanath, “Locality and Availability in Distributed Storage,” arxiv:1402.2011, February 2014.
- [22] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, A.G., R. Vadali, S. Chen and D. Borthakur, “XORing Elephants: Novel Erasure Codes for Big Data,” Proceedings of VLDB, Vol. 6, No. 5, pp.325–36, 2013.

- [23] W. Song, S. H. Dau, C. Yuen and T. J. Li, “Optimal Locally Repairable Linear Codes,” *IEEE Journal on Selected Areas in Communications*, Vol. 32 , pp. 1019–36, May 2014.
- [24] I. Tamo and A. Barg, “A Family of Optimal Locally Recoverable Codes,” *IEEE Trans. on Information Theory*, vol. IT-60, pp.4661–76, August 2014.
- [25] X. Tang and R. Koetter, “A Novel Method for Combining Algebraic Decoding and Iterative Processing,” *ISIT 2006, IEEE International Symposium on Information Theory*, pp. 474–78, July 2006.
- [26] A. Wang and Z. Zhang, “Repair Locality with Multiple Erasure Tolerance,” *IEEE Trans. on Information Theory*, vol. IT-60, pp. 6979–87, November 2014.
- [27] Y. Wu, “Generalized Integrated Interleaving BCH Codes,” *ISIT 2016, IEEE International Symposium on Information Theory*, pp. 1098–1102, July 2016.
- [28] A. Zeh and E. Yaakobi, “Bounds and Constructions of Codes with Multiple Localities,” *arXiv:1601.02763*, January 2016.
- [29] V. A. Zinoviev, “Generalized cascade codes,” *Problemy Peredachii Informatsii*, vol. 12, no. 1, pp. 5-15, 1976.