Generalized Column Distances

Sara D. Cardell, Marcelo Firer and Diego Napp,

***Abstract*—The notion of Generalized Hamming weights of block codes has been investigated since the nineties due to its significant role in coding theory and cryptography. In this paper we extend this concept to the context of convolutional codes. In particular, we focus on column distances and introduce the novel notion of generalized column distances (GCD). We first show that the hierarchy of GCD is strictly increasing. We then provide characterizations of such distances in terms of the truncated parity-check matrix of the code, that will allow us to determine their values. Finally, the case in which the parity-check matrix is in systematic form is treated.**

***Index Terms*—Convolutional codes, column distances, parity-check matrix.**

I. Introduction

The *Generalized Hamming Weights* (GHW) were first introduced by Kløve [19] and later rediscovered by Victor Wei [1]: Let $\mathcal{C}$ be an $[n,k]$-linear code over a Galois field $\mathbb{F}$. We define the support of $\mathcal{C}$ as

$$\text{supp }(\mathcal{C}) = \{i \mid x_i \neq 0 \text{ for some } [x_1, x_2, \ldots, x_n] \in \mathcal{C}\}.$$

The r-th generalized (Hamming) weight of $\mathcal{C}$ is then defined as

$$d_r(\mathcal{C}) = \min\{|\text{supp}(\mathcal{D})| \mid \mathcal{D} \text{ subcode of } \mathcal{C}, \dim(\mathcal{D}) = r\},$$

and the Hamming weight hierarchy is the set of integers $\{d_r(\mathcal{C}) \mid 1 \leq r \leq k\}$.

Although the generalized weights were originally introduced motivated by applications in cryptography in [1], [8], the notion goes far beyond these applications. The concept of generalized weights is a generalization of the minimal distance of linear codes and refines the role of the minimal distance as an indicator of the error probability of a code. The following simple example illustrates this fact. Denote by $\mathbf{1_r}$ and $\mathbf{0_s}$ a sequence of r consecutive ones and s consecutive 0, respectively, and let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two linear codes over $\mathbb{F}_2 = \{0,1\}$ of dimension 2, defined by

$$\mathcal{C}_1 = \langle \mathbf{1_3 0_{n-3}},\ \mathbf{0_3 1_3 0_{n-6}} \rangle,\ \ \mathcal{C}_2 = \langle \mathbf{1_3 0_{n-3}},\ \mathbf{0_3 1_{n-3}} \rangle.$$

Both codes have $d_1 = 3$, hence are capable of correcting every error of length 1, but the second weight of $\mathcal{C}_1$ is 6 while the second generalized weight of $\mathcal{C}_2$ is n. For large n, the first code has $n-6 > 0$ "useless" coordinates, while these coordinates do have a role in $\mathcal{C}_2$, helping to distinguish if a received codeword should be decoded as $\mathbf{1_3 0_{n-3}}$ or $\mathbf{0_3 1_{n-3}}$.

S. D. Cardell and M. Firer are with Imecc, University of Campinas, Brazil e-mail: sdcardell@ime.unicamp.br, mfirer@ime.unicamp.br.

Diego Napp is with Department of Mathematics, University of Alicante, Spain e-mail: diego.napp@ua.es.

This paper was presented in part at IEEE Isit 2019.

A more precise role of the generalized weight hierarchy has been recently investigated to derive error probabilities after decoding of block codes over the erasure channel. This was first approached by Didier [12] and later developed in [11], [33] and finally in [34], were an expression of the unsuccessful decoding probability under list decoding and unambiguous decoding was established in terms of the distribution of generalized weights[1].

On the other hand, David Forney observed in [7] that the results concerning the GHW obtained for block codes admit an interesting extension to convolutional codes and this motivated the work in [17]. In this work the authors tackled the convolutional case focusing on the free distance whereas in this work we consider column distances. Column distance is arguably the most fundamental measure for convolutional codes [20, pag. 109], as the maximal possible growth in the column distances means that these codes have the potential to correct the maximal number of errors per time interval.

The definition of *Generalized Column Distance* of a convolutional code (GCD) is the starting point of this work. We prove the first key result to study the subject: the monotonicity of the GCD hierarchy of a convolutional code $\mathcal{C}$. Moreover, we provide a complete characterization of this new notion in terms of the truncated parity-check matrix of $\mathcal{C}$. This result is general, constructive and its proof only requires elementary notions of linear algebra. An interesting instant is when the convolutional code admits a systematic encoder/parity-check polynomial matrix. If so, say $H(D) = [I_k \ P(D)]$, a simple characterization of the GCD can be given only in terms of the coefficients of the matrix $P(D)$. As $P(D)$ is usually much smaller that the parity-check $H(D)$, this condition is easier to verify. The results presented here on generalized column distances can be seen as a generalization of previous results on generalized Hamming weights for block codes [8], [26], [27], [28], [29], [30], [31]. In this work we need to consider truncated convolutional codes, which are intrinsically nonlinear and therefore very different from the ones treated before and in particular, in [17]. In [24] a more restrictive definition was introduced and later generalized in [25] where some preliminary results were presented.

II. Notation and Definitions

Let $\mathbb{F}$ be a finite field and $\mathbb{F}[D]$, the ring of polynomials with coefficients in $\mathbb{F}$. A **convolutional code** $\mathcal{C}$ of rate k/n is an $\mathbb{F}[D]$-module of $\mathbb{F}[D]^n$ of rank k of the form

$$\mathcal{C} = \text{im } G(D) = \{\boldsymbol{u}(D)G(D) \mid \boldsymbol{u}(D) \in \mathbb{F}[D]^k\}$$

[1]This is a generalization of the weight distribution. We consider the polynomial $p(x,y) = \sum_{i,j} A_i^j x^j y^i$, where $A_i^j := |\{\mathcal{D} \subset \mathcal{C} \mid \dim(\mathcal{D}) = i, |\text{supp}(\mathcal{D})| = j\}|$ and the usual weight distribution polynomial is given by the particular case when $i = 1$, $\sum_{j=0}^{n}(q-1)A_1^j x^j y^1$.

where $G(D) \in \mathbb{F}[D]^{k\times n}$ is a right invertible matrix or *basic*, *i.e.*, there exits a matrix $H(D) \in \mathbb{F}[D]^{(n-k)\times n}$ such that

$$\mathcal{C} = \ker\, H(D) = \{\boldsymbol{w}(D) \in \mathbb{F}[D]^n \mid \boldsymbol{w}(D)H(D)^T = \mathbf{0}\}.$$

We can express the generator matrix as

$$G(D) = \sum_{j=0}^{\mu} G_j D^j, G_j \in \mathbb{F}^{k\times n}, G_\mu \neq 0$$

and the parity-check matrix as

$$H(D) = \sum_{j=0}^{\nu} H_j D^j, H_j \in \mathbb{F}^{(n-k)\times n}, H_\nu \neq 0.$$

Let

$$G_j^c = \begin{bmatrix} G_0 & G_1 & \cdots & G_j \\ \mathbf{0} & G_0 & \cdots & G_{j-1} \\ \vdots & \ddots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & G_0 \end{bmatrix}$$

and

$$H_j^c = \begin{bmatrix} H_0 & \mathbf{0} & \cdots & \mathbf{0} \\ H_1 & H_0 & \ddots & \vdots \\ \vdots & \vdots & & \mathbf{0} \\ H_j & H_{j-1} & \cdots & H_0 \end{bmatrix},$$

be the **truncated sliding generator** and **parity-check matrices** for $j \in \mathbb{N}_0$, respectively, $H_j = 0$ for $j > \nu$ and $G_j = 0$ for $j > \mu$. We define the ***j*-truncated convolutional code** (of $\mathcal{C}$) as

$$\mathcal{C}_j = \left\{ [\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_j]\, G_j^c \mid \boldsymbol{u}_i \in \mathbb{F}^k, \boldsymbol{u}_0 \neq 0 \right\}.$$

We denote $N = (j+1)n$ and $K = (j+1)k$ and $[n] = \{1, 2, \cdots n\}$ and write $\boldsymbol{w} = [\boldsymbol{w}_0, \boldsymbol{w}_1, \cdots, \boldsymbol{w}_j] \in \mathbb{F}^N$ with $\boldsymbol{w}_i \in \mathbb{F}^n$ or $\boldsymbol{w} = [w_1, w_2, \ldots, w_N]$, with $w_i \in \mathbb{F}$. A list of words is denoted by superscripts: $\boldsymbol{w}^1, \cdots, \boldsymbol{w}^r$.

For a set $X \subset \mathbb{F}^N$, $\mathrm{supp}(X) = \bigcup_{\boldsymbol{x}\in X} \mathrm{supp}(\boldsymbol{x})$. For simplicity, given $\boldsymbol{x}^1, \cdots, \boldsymbol{x}^r$, we denote $\mathrm{supp}\left(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^r\right) = \mathrm{supp}\left(\{\boldsymbol{x}^1, \cdots, \boldsymbol{x}^r\}\right)$ (omitting the brackets $\{-\}$).

We can now give the key definitions of this work:

Definition 1: [16], [20] The **j-th column distance** $d_j^c(\mathcal{C}) = d\left(\mathcal{C}_j\right)$ of $\mathcal{C} = \mathrm{im}\ G(D) = \ker\ H(D)$ is given by

$$\begin{aligned} d_j^c(\mathcal{C}) &= \min\left\{\omega\left([\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_j]\, G_j^c\right) \mid \boldsymbol{u}_i \in \mathbb{F}^k, \boldsymbol{u}_0 \neq 0\right\} \\ &= \min\{\omega\left([\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_j]\right) \mid \\ &\qquad \mid H_j^c\, [\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_j]^T = \mathbf{0}, \boldsymbol{w}_0 \neq 0\Big\} \\ &= \min\{\omega(\boldsymbol{w}) \mid \boldsymbol{w} \in \mathcal{C}_j\} \end{aligned}$$

where $\omega\left(\boldsymbol{w}\right)$ is the Hamming weight of the vector $\boldsymbol{w}$.

Definition 2: The r-th **Generalized Column Distance** (or r-th GCD) of the truncated convolutional code $\mathcal{C}_j$ is

$$d_r^g(\mathcal{C}_j) = \min\{|\mathrm{supp}(D)| : D \subset \mathcal{C}_j, |D| = r \text{ and } D \text{ is L.I.}\}, \tag{1}$$

where L.I. stands for a linearly independent set and $|X|$ is the cardinality of a set X.

We remark that the first generalized Hamming weight is actually the column distance, that is, $d_j^c(\mathcal{C}) = d(\mathcal{C}_j) = d_1^g(\mathcal{C}_j)$. The **weight hierarchy** is $\{d_1^g(\mathcal{C}_j), d_2^g(\mathcal{C}_j), \ldots, d_K^g(\mathcal{C}_j)\}$. Since the r-th GCD is defined considering a minimum over subsets $D \subset \mathcal{C}_j$ with r elements, it is not obvious that the weight hierarchy is strictly increasing, that is, $d_1^g(\mathcal{C}_j) < d_2^g(\mathcal{C}_j) < \cdots < d_K^g(\mathcal{C}_j)$. We shall see in Theorem 4 that this is the case. Note that, as opposed to the block code case, the set $\mathcal{C}_j$ in Definition 2 does not have the structure of a vector space, as the sum of two of its elements may not be in $\mathcal{C}_j$.

Example 3: Consider the convolutional code with parameters $n = 4$ and $k = 2$ over $\mathbb{F}_3$ whose generator matrix is

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

and

$$G_0 = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 2 \end{bmatrix}, \quad G_1 = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

For $j = 0$, we have that $d_1^g(\mathcal{C}_0) = 3$ and $d_2^g(\mathcal{C}_0) = 4$. Consider now the truncated sliding generator matrix for $j = 1$:

$$G_1^c = \left[\begin{array}{cccc:cccc} 1 & 0 & 1 & 2 & 0 & 0 & 2 & 1 \\ 0 & 1 & 2 & 2 & 0 & 0 & 1 & 1 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 \end{array}\right]$$

We consider rows of the matrix G_1^c:

$$\begin{aligned} g_1 &= [1\ 0\ 1\ 2\ 0\ 0\ 2\ 1] \\ g_2 &= [0\ 1\ 2\ 2\ 0\ 0\ 1\ 1] \\ g_3 &= [0\ 0\ 0\ 0\ 1\ 0\ 1\ 2] \\ g_4 &= [0\ 0\ 0\ 0\ 0\ 1\ 2\ 2] \end{aligned}$$

It is worth noticing that $g_1, g_2 \in \mathcal{C}_1$ but $g_3, g_4 \notin \mathcal{C}_1$. It is easy to check that

$$\begin{aligned} d_1^g(\mathcal{C}_1) &= 4, \quad \text{realized by span}\{g_2 + g_4\} \text{ and} \\ d_2^g(\mathcal{C}_1) &= 6, \quad \text{realized by span}\{g_1, g_2\}. \end{aligned}$$

III. Monotonicity and Singleton bounds

The first fundamental result on generalized Hamming weights for linear codes obtained by Wei in [1, Theorem 1] states that the generalized weight hierarchy is strictly increasing. The same holds for convolutional codes.

Theorem 4 (Monotonicity): Let $\mathcal{C}_j$ be a truncated convolutional code. Then,

$$d_1^g(\mathcal{C}_j) < d_2^g(\mathcal{C}_j) < \cdots < d_K^g(\mathcal{C}_j) \tag{2}$$

Proof. We shall assume that the weight hierarchy is not strictly increasing and come to a contradiction. So, we assume that there is an r such that $d_r^g\left(\mathcal{C}_j\right) = d_{r+1}^g\left(\mathcal{C}_j\right)$. We assume that r is minimal with such property.

Let $\beta = \left\{\boldsymbol{w}^1, \boldsymbol{w}^2, ..., \boldsymbol{w}^{r+1}\right\}$ be a set of codewords of $\mathcal{C}_j$ that realizes the $(r+1)$-th generalized column distance, that is:

$$\mathrm{rank}\left(\beta\right) = r+1 \quad \text{and} \quad \left|\mathrm{supp}\left(\beta\right)\right| = d_{r+1}^g\left(\mathcal{C}_j\right),$$

where $\mathrm{rank}\left(\beta\right)$ means the rank of the matrix that has $\boldsymbol{w}^1, \boldsymbol{w}^2, ..., \boldsymbol{w}^{r+1}$ as its rows. We denote $\beta^i = \beta \backslash \left\{\boldsymbol{w}^i\right\}$, $\beta^{i,j} = \beta \backslash \left\{\boldsymbol{w}^i, \boldsymbol{w}^j\right\}$. Again we write $\boldsymbol{w} = [\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_j]$ or $\boldsymbol{w} = [w_1, w_2, \ldots, w_N]$.

Note two facts:

1) Since we are assuming that $d_r^g(\mathcal{C}_j) = d_{r+1}^g(\mathcal{C}_j)$ and since $\operatorname{rank}(\beta^i) = r$ we have that
$$\left|\operatorname{supp}\left(\beta^i\right)\right| = |\operatorname{supp}(\beta)| = d_{r+1}^g(\mathcal{C}_j),$$
i.e., $\boldsymbol{w}^i$ does not aggregate to β^i, in the sense that $\operatorname{supp}(\boldsymbol{w}^i) \subset \operatorname{supp}(\beta^i)$, $i \in [r+1]$.
2) The minimality of r ensures that $\left|\operatorname{supp}\left(\beta^{i,j}\right)\right| < \left|\operatorname{supp}\left(\beta^i\right)\right|$, for all $i \neq j$, that is, each $\boldsymbol{w}^i$ aggregates to every $\beta^{i,j}$, for $i \neq j$, in the sense that $\operatorname{supp}(\boldsymbol{w}^i) \not\subset \operatorname{supp}(\beta^{i,j})$.

Based on these facts we shall come to a contradiction. Consider two cases.

Case 1: $\boldsymbol{w}_0^{r+1} \neq \boldsymbol{w}_0^i, \forall i \neq r+1$

Take an $\ell \leq n$ such that $w_\ell^{r+1} = \alpha \neq 0$. Define for $i \leq r$,
$$\boldsymbol{w}_*^i = \begin{cases} \boldsymbol{w}^i & \text{if} \quad w_\ell^i = 0, \\ \boldsymbol{w}^i - \frac{\gamma}{\alpha}\boldsymbol{w}^{r+1} & \text{if} \quad w_\ell^i = \gamma \neq 0, \end{cases}$$
and $\beta^* = \left\{\boldsymbol{w}_*^1, \boldsymbol{w}_*^2, ..., \boldsymbol{w}_*^r, \boldsymbol{w}^{r+1}\right\}$. Obviously, $\operatorname{rank}(\beta^*) = r+1$ and $\operatorname{supp}(\beta^*) = \operatorname{supp}(\beta)$, since coordinates that may be annihilated by exchanging some $\boldsymbol{w}^i$ by $\boldsymbol{w}_*^i$ are positions that correspond to entries in $\operatorname{supp}\left(\boldsymbol{w}^{r+1}\right)$. Moreover, $w_{*_\ell}^i = 0$ for every $i = 1, 2, ..., r$. It follows that $\boldsymbol{w}^{r+1}$ aggregates to $\beta^* \backslash \left\{\boldsymbol{w}^{r+1}\right\}$ hence $\left|\operatorname{supp}\left(\beta^* \backslash \left\{\boldsymbol{w}^{r+1}\right\}\right)\right| < |\operatorname{supp}(\beta^*)| = |\operatorname{supp}(\beta)|$ which is a contradiction to the hypothesis $d_r^g(\mathcal{C}_j) = d_{r+1}^g(\mathcal{C}_j)$.

Case 2: $\exists\ t \in [r+1], \boldsymbol{w}_0^{r+1} = \boldsymbol{w}_0^t$

We define $\boldsymbol{u}^t = \boldsymbol{w}^t - \boldsymbol{w}^{r+1}$. Since β is L.I., we have $\boldsymbol{u}_0^t = \boldsymbol{0}$ and $u_\ell^t = \alpha \neq 0$ for some $\ell > n$. For $i \leq r$ we define
$$\boldsymbol{w}_*^i = \begin{cases} \boldsymbol{w}^i & \text{if} \quad w_\ell^i = 0 \\ \boldsymbol{w}^i - \frac{\gamma}{\alpha}\boldsymbol{u}^t & \text{if} \quad w_\ell^i = \gamma \neq 0 \end{cases}.$$
Then we consider $\beta^* = \left\{\boldsymbol{w}_*^1, \boldsymbol{w}_*^2, ..., \boldsymbol{w}_*^r, \boldsymbol{w}^{r+1}\right\}$ and we have that $\operatorname{rank}(\beta^*) = r+1$ and, since $\boldsymbol{w}^{r+1}$ aggregates to $\beta^* \backslash \left\{\boldsymbol{w}^{r+1}\right\}$, we find again a contradiction. ■

As a result let us establish an upper-bound on the GCD of a truncated convolutional code. This result generalizes [1, Corollary 1] (on generalized weights of block codes) and [16, Proposition 2.2] (on column distances of convolutional codes).

Corollary 5 (Singleton bound): Let $\mathcal{C}_j$ be a truncated convolutional code. Then,
$$d_r^g(\mathcal{C}_j) \leq (j+1)(n-k) + r. \tag{3}$$

Proof. It is known (see [16], [22]) that $d_1^g(\mathcal{C}_j) = d_j^c(\mathcal{C}) \leq N - K + 1$. Thus, the result follows from the monotonicity theorem, with a reasoning similar to the original proof of Wei in [1]. ■

Remark 6: Taking into account the monotonicity of the GCD of a convolutional code and Corollary 5, it is easy to see that if $d_r^g(\mathcal{C}_j)$ attains the bound given in (3) for some r, all following generalized column distances $d_s^g(\mathcal{C}_j)$ will also attain the bound for $r \leq s \leq K$.

Example 7: Consider again Example 3. It is easy to check that $d_1^g(\mathcal{C}_0) = 3$, $d_2^g(\mathcal{C}_0) = 4$ and $d_2^g(\mathcal{C}_1) = 6$ attain the bound given in Corollary 5 and $d_1^g(\mathcal{C}_1) = 4$ does not. From the monotonicity theorem and Corollary 5 we get that $d_3^g(\mathcal{C}_1) = 7$ and $d_4^g(\mathcal{C}_1) = 8$.

IV. A CHARACTERIZATION IN TERMS OF PARITY-CHECK MATRICES

In this section we study the structure of the truncated convolutional code $\mathcal{C}_j$ and provide a full algebraic characterization of the GCD in terms of their parity-check matrices.

From now on, we will use the following notations and definitions:

- Given $I = \{i_1 < i_2 < \cdots < i_r\} \subset [N]$ we denote as H_I the sub-matrix of H consisting of the columns indexed by I. The same notation may be applied to every matrix.
- We say that the set of columns of H_I, say $\{h_i \mid i \in I\}$, satisfies **Condition (∗)** if there is $r \in I \cap [n]$ such that $h_r = \sum_{j \in I, j \neq r} \alpha_j h_j$, with $\alpha_j \in \mathbb{F}$.

Theorem 8: Let $\mathcal{C}_j$ be a truncated convolutional code and for convenience we denote here by H the truncated sliding parity-check matrix H_j^c. Then, it holds that:

1) If there is a set of d columns of H whose rank is equal to $d - r$ and satisfies Condition $(*)$, then $d_r^g(\mathcal{C}_j) \leq d$.
2) If every subset $\{h_{j_1}, h_{j_2}, \cdots, h_{j_d - 1}\}$ of $d-1$ columns of H with $j_1 \leq n$ has rank at least equal to $d - r$, then $d_r^g(\mathcal{C}_j) \geq d$.

Proof.

Part 1: Let $I = \{j_1, j_2, \cdots, j_d\} \subset [N]$ and define
$$D_I = \left\{ \boldsymbol{x} \in \mathbb{F}^N \mid x_i = 0 \text{ for } i \notin I \text{ and } \sum_{i=1}^N x_i h_i = 0 \right\},$$
where h_i are the columns of H. D_I is an extension of the kernel of H_I and a subspace of $\mathbb{F}^N$. We have that $|\operatorname{supp}(D_I)| \leq |I| = d$ and $\dim(D_I) = r$. So, we shall prove that $d_r^g(\mathcal{C}_j) \leq d$ by constructing a basis $\beta = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r\}$ of D_I with $\boldsymbol{w}^i \in \mathcal{C}_j$.

Condition $(*)$ ensures there is $j_a \leq n$ such that $h_{j_a} = \sum_{\ell \leq d, \ell \neq a} \alpha_\ell h_{j_\ell}$. We consider now the vector $\boldsymbol{w}^1$ that has 1 in the position j_a, $-\alpha_\ell$ in the position j_ℓ and zero otherwise. This vector belongs to D_I and $\mathcal{C}_j$. Now, we can construct a basis β^* of D_I that contains $\boldsymbol{w}^1$:
$$\beta^* = \{\boldsymbol{w}^1, \boldsymbol{w}_*^2, \ldots, \boldsymbol{w}_*^r\}$$
We know that $\boldsymbol{w}^1 \in \mathcal{C}_j$, but $\boldsymbol{w}_*^i$, $i = 2, \ldots, r$, might not be in $\mathcal{C}_j$. We define
$$\boldsymbol{w}^i = \begin{cases} \boldsymbol{w}_*^i & \text{if } \boldsymbol{w}_*^i \in \mathcal{C}_j \\ \boldsymbol{w}^1 + \boldsymbol{w}_*^i & \text{if } \boldsymbol{w}_*^i \notin \mathcal{C}_j \end{cases}$$
and now $\boldsymbol{w}^i \in \mathcal{C}_j$ and $\beta = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r\}$.

By construction we have that $\beta \subset D_I$ hence its support contains at most d positions. Since the matrix determined by β was obtained from the matrix determined by β^* by the means of elementary operations, we have that β is a L.I. set as well.

Part 2: We shall prove this part by contradiction. Assuming that $d_r^g(\mathcal{C}_j) < d$, we will prove the existence of a set $I \subset [N]$ with $|I| = d - 1$ such that $\operatorname{rank}(H_I) < d - r$ and $I \cap [n] \neq \emptyset$.

Assuming that $d_r^g(\mathcal{C}_j) < d$ means there is a set $\beta = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots \boldsymbol{w}^r\} \subset \mathcal{C}_j$ such that β is L.I. and $|\operatorname{supp}(\beta)| = d_r^g(\mathcal{C}_j) < d$. We denote $I = \operatorname{supp}(\beta)$ and consider H^I the matrix obtained from H by turning into zero the columns corresponding to entries in $[N] \setminus I$. Then, for each $\boldsymbol{w}^i \in \beta$

we have that $H^I(\boldsymbol{w}^i)^T = \mathbf{0}$. Since β is assumed to be L.I., we have that $\text{rank}(H^I) \leq d_r^g(\mathcal{C}_j) - r$. If we denote by $\widehat{H^I}$ the matrix formed by the non-zero columns of H^I, we have that $\text{rank}(\widehat{H^I}) = \text{rank}(H^I) \leq d_r^g(\mathcal{C}_j) - r < d - r$. Note that $I \cap [n] \neq \emptyset$ as $\boldsymbol{w}^i \in \mathcal{C}_j$. Therefore, we have $d_r^g(\mathcal{C}_j)$ columns with rank $< d - r$. Now, if $d_r^g(\mathcal{C}_j) = d - 1$, there is nothing else to prove. Otherwise, if $d_r^g(\mathcal{C}_j) = d - t$, with $t > 1$, we can choose $t-1$ columns of H whose indices are not included in I. Thus, we have a set of $d - 1$ columns of H with rank $< d - r$ as claimed. ■

One should remark that Condition $(*)$ is essential for the validity of the first part of the theorem. This difficulty is due to Definitions 1 and 2, *i.e.*, if $\boldsymbol{w} = [\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_j] \in \mathcal{C}_j$ then $\boldsymbol{w}_0 \neq 0$.

Indeed, let us consider Example 3 again. Direct calculations showed that $d_1^g(\mathcal{C}_1) = 4$, hence $d_2^g(\mathcal{C}_1) \geq 5$ (actually $d_2^g(\mathcal{C}_1) = 6$). A sliding parity-check matrix is

$$H_1^c = \left[\begin{array}{cccc:cccc} 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right].$$

It is immediate to note that the last $d = 4$ columns of H_1^c have rank $d - 2 = 2$, but $d < d_2$. This would contradict the first part of the previous theorem, except for the fact that the set of columns $\{h_5, h_6, h_7, h_8\}$ does not satisfy Condition $(*)$.

Theorem 9 (reciprocal of Theorem 8): Let $\mathcal{C}_j$ be a truncated convolutional code and for clarity we denote here by H the truncated sliding parity-check matrix H_j^c. If we denote $d_r^g(\mathcal{C}_j) = d_r$ then:

1) There exists a set of d_r columns of H with $j_1 \leq n$ with rank equal to $d_r - r$
2) Every set of $d_r - 1$ columns of H satisfying Condition $(*)$ has rank $\geq d_r - r$.

Proof.

Part 1: According to the definition of d_r, there exists $\beta = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^r\} \subset \mathcal{C}_j$ a L.I. set with $|\text{supp}(\beta)| = d_r$. Let $I = \text{supp}(\beta) = \{j_1, j_2, \cdots, j_{d_r}\}$. We assume without loss of generality that $j_r < j_{r+1}$, for all $1 \leq r \leq d_r - 1$. Since each $\boldsymbol{w}^r \in \beta$ is a codeword of $\mathcal{C}_j$, we have that $\text{supp}(\boldsymbol{w}^r) \cap [n] \neq \emptyset$ and this implies that $I \cap [n] \neq \emptyset$. It follows that $j_1 \leq n$. We may assume without loss of generality that $j_1 = 1$.

The rank-nullity theorem ensures that $\text{rank}(H_I) + \dim(\ker(H_I)) = d_r$. Since $\dim(\ker(H_I)) \geq r$ we get that

$$\text{rank}(H_I) = d_r - \dim(\ker(H_I)) \leq d_r - r.$$

Let us assume that $\text{rank}(H_I) < d_r - r$. This means that there is another element $\boldsymbol{w}^{r+1} \in \ker(H)$ such that $\text{supp}(\boldsymbol{w}^{r+1}) \subset I$ and $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r, \boldsymbol{w}^{r+1}\}$ is still a L.I. set. As a consequence, we obtain that $d_{r+1} = d_r$, which is a contradiction (see Theorem 4). Then, $\text{rank}(H_I) = d_r - r$.

Part 2: Suppose there is a set $\{h_{j_1}, h_{j_2}, \cdots, h_{j_{d_r-1}}\}$ of $d_r - 1$ columns of H satisfying Condition $(*)$ with rank $< d_r - r$. Let $I = \{j_1, j_2, \cdots j_{d_r-1}\}$. Then, the rank-nullity theorem ensures that $\dim(\ker H_I) > r - 1$. So, there is a L.I. set with r elements contained in $\ker(H_I)$. Condition $(*)$ ensures that we may assume, without lost of generality, that $h_{j_1} = \sum_{k=2}^{d_r-1} \alpha_k h_{j_k}$ and hence we have that the vector $\boldsymbol{w}^1 = [1, -\alpha_2, \cdots - \alpha_{d-1}] \in \ker(H_I)$. We complete it to a L.I. set $\beta^* = \{\boldsymbol{w}^1, \boldsymbol{w}_*^2, \cdots, \boldsymbol{w}_*^r\} \subset \ker(H_I)$ and, in the same manner we proceeded in the proof of part 1 of Theorem 8, out of β^* we get a set $\beta = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^r\} \subset \mathcal{C}_j$ that is L.I. Finally, from the definition of the generalized column distance we get

$$d_r \leq |\text{supp}(\beta)| = |\text{supp}(\beta^*)| \leq d_r - 1,$$

(the last inequality follows from the fact that $\beta^* \subset \ker(H_I)$), which is a contradiction. ■

Corollary 10: Let $\mathcal{C}_j$ be a truncated convolutional code with sliding truncated parity-check matrix $H = H_j^c$. Then, $d_r^g(\mathcal{C}_j) = d$ if, and only if, the two following conditions hold:

1) There exists a set $\{h_{j_1}, h_{j_2}, \cdots, h_{j_d}\}$ of d columns of H with $j_1 \leq n$ with rank equals to $d - r$.
2) Every set of $d - 1$ columns of H satisfying Condition $(*)$ has rank $\geq d - r$.

Proof. Theorem 8 ensures the *if* part and Theorem 9 the *only if* part. ■

Example 11: Consider again Example 3. A truncated sliding parity-check matrix H_1^c is given by:

$$H_1^c = \left[\begin{array}{cccc:cccc} 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 0 & 0 & 0 & 0 \\ \hdashline 2 & 0 & 0 & 0 & 1 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 & 0 & 1 & 2 & 2 \end{array}\right].$$

Now we shall derive the weight hierarchy using Corollary 10. First of all, we denote the columns of H_1^c as $h_1, h_2, \ldots, h_8$.

For establishing $d_1^g(\mathcal{C}_1) = 4$ we need to show that:

- There are 4 columns that determine a matrix of rank 3 and one of them, let us say h_i satisfies $i \leq 4$. Indeed, columns h_1, h_3, h_4 and h_5 determine a matrix of rank 3.
- Since there is no set of 3 columns satisfying Condition $(*)$, there is nothing else to check.

For establishing $d_2^g(\mathcal{C}_1) = 6$ we need to show that:

- There are 6 columns that determine a matrix of rank $4 (= 6-2)$ one of them, let us say h_i, satisfying $i \leq 4$. Indeed, columns h_1, h_3, h_4, h_5 and h_6 determine a matrix with rank 4.
- Every set of 5 columns satisfying Condition $(*)$ has rank ≥ 4. Indeed, a set of columns satisfying Condition $(*)$ must contain at least 3 columns corresponding to the first 4 positions. Every set of such three columns has rank 3. It easy to check that, by adding two more columns the rank will increase and hence will be equal to 4.

For the cases $d_3^g(\mathcal{C}_1) = 7$ and $d_4^g(\mathcal{C}_1) = 8$ it is enough to note the existence of 7 or 8 columns with rank 4 and the fact that every 6 or 7 columns satisfying Condition $(*)$ has rank ≥ 4.

V. Systematic parity-check matrix and GHW

In this last section we consider the case when the sliding parity-check matrix is in systematic form. This case was also considered in [31] in the context of block codes and therefore we extend here their results to the context of column distances.

We show that the GCDs may be obtained with smaller effort out of this systematic form, considering only the parity part of the matrix.

Using elementary row operations (which keeps the code invariant) and possible column permutations (what gives rise to an equivalent code), the sliding truncated parity-check matrix of $\mathcal{C}_j$ can be expressed in the form:

$$H_j^c = \begin{bmatrix} I & P_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ I & P_1 & I & P_0 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ I & P_{j-1} & I & P_{j-2} & I & P_{j-3} & \dots & \mathbf{0} & \mathbf{0} \\ I & P_j & I & P_{j-1} & I & P_{j-2} & \dots & I & P_0 \end{bmatrix}$$

where I stands for the identity matrix of size $(n-k)\times(n-k)$. As the permutation of columns does not change the distance properties of the code, we can consider the matrix H_j^c as $[I_{N-K} \;\; P_j^c]$ where

$$P_j^c = \begin{bmatrix} P_0 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ P_1 & P_0 & \mathbf{0} & \dots & \mathbf{0} \\ P_2 & P_1 & P_0 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ P_j & P_{j-1} & P_{j-2} & \dots & P_0 \end{bmatrix},$$

and I_{N-K} is the identity matrix of size $N-K=(n-k)(j+1)$. For simplicity, in this section, we will denote the matrix P_j^c by P and $[I_{N-K} \; P_j^c]$ by H.

We say that the set of columns of P_I, say $\{p_i \mid i \in I\}$, satisfies Condition $(*)$ if there is $d \in I \cap [k]$ such that $p_d = \sum_{j\in I, j\neq d} \alpha_j p_j$, with $\alpha_j \in \mathbb{F}$.

Theorem 12: Let $\mathcal{C}_j$ be a truncated convolutional code with weight hierarchy $[d_1^g(\mathcal{C}_j), d_2^g(\mathcal{C}_j), \dots, d_K^g(\mathcal{C}_j)]$. Let $H = [I_{N-K} \; P]$ and $P = P_j^c$ as above and consider $d_r = d_r^g(\mathcal{C}_j)$, then:

1) For every $r \leq g \leq \min\{d_r - 1, K\}$, every submatrix consisting of g columns of P satisfying Condition $(*)$ has $\text{rank} \geq g - r + 1$
2) There is g satisfying $r < g \leq \min\{d_r, K\}$ such that there is a sub-matrix of P of size $N - K - d_r + g \times g$ and $\text{rank} = g - r$.

Proof.

1) We suppose that (1) fails to hold, *i.e.*, there exits a non trivial submatrix P_I consisting of the columns $I = \{j_1, j_2, \dots, j_g\}$ of P with rank less than $g-r+1$, for some g with $r \leq g \leq \min\{d_r - 1, k\}$.
Since P_I has g columns and $\text{rank}(P_I) < g$, we know that one of the columns is a linear combination of the others. By Condition $(*)$, we can assume, without loss of generality, that this column is p_{j_1}. Consider then, the linear combination $p_{j_1} = \sum_{i=2}^{g} \alpha_i p_{j_i}$. We set $\boldsymbol{w}^1 = [1, -\alpha_2, \dots, -\alpha_g]$. As $\text{rank}(P_I) < g - r + 1$, by the rank-nullity theorem there exists a set of L.I. vectors $\{\boldsymbol{w}^1, \boldsymbol{w}_*^2, \dots, \boldsymbol{w}_*^r\} \subset \mathbb{F}^g$ such that $P_I(\boldsymbol{w}^i)^T = \mathbf{0}$, $i = 1, 2, \dots, r$. Now, for each $i \leq r$,

$$\boldsymbol{w}^i = \begin{cases} \boldsymbol{w}_*^i & \text{if } \boldsymbol{w}_{*_0}^i \neq \mathbf{0} \\ \boldsymbol{w}_*^i + \boldsymbol{w}^1 & \text{if } \boldsymbol{w}_{*_0}^i = \mathbf{0} \end{cases}$$

and we get that:

- For each $i \leq r$, $\boldsymbol{w}^i \in \ker(P_I)$.
- Each $\boldsymbol{w}^i$ has the first coordinate different from zero.
- The set $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\}$ is L.I..

Without loss of generality, we assume that $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\} \subset \mathbb{F}^g$ is a set of vectors satisfying these three properties.
Given $\boldsymbol{w} = [w_1, w_2, \dots, w_g] \in \mathbb{F}^g$, consider $\hat{\boldsymbol{w}} = [\hat{w}_1, \hat{w}_1, \dots, \hat{w}_N] \in \mathbb{F}^N$ defined as follows:

$$\hat{w}_s = \begin{cases} w_s \text{ if } s \in \text{supp}(P_I) \\ 0 \text{ otherwise} \end{cases}$$

From the definition we have that $|\text{supp}(\boldsymbol{w})| = |\text{supp}(\hat{\boldsymbol{w}})|$ hence we have that $\left|\text{supp}\left(\widehat{\boldsymbol{w}^1}, \widehat{\boldsymbol{w}^2}, \dots, \widehat{\boldsymbol{w}^r}\right)\right| = \left|\text{supp}\left(\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\right)\right| \leq g < d_r - 1$.
Moreover, as $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\}$ are L.I., so are $\{\widehat{\boldsymbol{w}^1}, \widehat{\boldsymbol{w}^2}, \dots, \widehat{\boldsymbol{w}^r}\} \in \mathcal{C}_j$ and it follows that

$$d_r \leq \left|\text{supp}\left(\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\right)\right| \leq d_r - 1 < d_r,$$

which is a contradiction.

2) Let $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\} \subset \mathcal{C}_j$ be a L.I. set such that

$$\left|\text{supp}\left(\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\right)\right| = d_r.$$

Let I and J be the set of corresponding indices of $\text{supp}\left(\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^r\right)$ according to I_{N-K} and P, from $H = [I_{N-K} \; P]$, respectively. Analogously, let us divide $\boldsymbol{w}^i = [\boldsymbol{w}_I^i \;\; \boldsymbol{w}_J^i]$ and denote by $H_{I,J} = [H_I \; H_J]$ the columns of H with indices in I and J.
Furthermore, let $g = |J|$ and $|I| = d_r - g$. Since $|I| + |J| = d_r$ we have that $g \leq d_r$ and $g \leq K$ because P has K columns, so that $g \leq \min\{d_r, K\}$.
Consider $\widehat{H}_I$, the submatrix of $H_I \in \mathbb{F}^{d_r - g \times d_r - g}$ constructed by deleting the zero rows and consider $\widehat{H}_J \in \mathbb{F}^{N-K-d_r+g\times g}$, the submatrix of H_J constructed by deleting the rows of H_J corresponding to non-zero rows in H_I.
We remark that

$$\text{rank}(\widehat{H}_I) = \text{rank}\,(H_I)\,,$$

since we obtained one from the other by deleting all-zero rows. Moreover, up to a permutation of the rows, we may write

$$H_{I,J} = [H_I \; H_J] = \begin{bmatrix} \widehat{H}_I & H_J^+ \\ 0 & \widehat{H}_J \end{bmatrix}$$

where H_J^+ is a matrix obtained from H_J by considering the rows not included in $\widehat{H}_J$. Obviously, since $\widehat{H}_I$ is submatrix of the identity with one 1 entry in each row, we have that $H_{I,J}$ and the matrix

$$\begin{bmatrix} \widehat{H}_I & 0 \\ 0 & \widehat{H}_J \end{bmatrix}$$

are column equivalent so

$$\text{rank}(H_{I,J}) = \text{rank}(\widehat{H}_I) + \text{rank}(\widehat{H}_J) = d_r - g + \text{rank}(\widehat{H}_J).$$

As $\{\boldsymbol{w}^1, \ldots, \boldsymbol{w}^r\}$, are L.I. and according to the rank-nullity theorem, we have that $\text{rank}(H_{I,J}) \leq d_r - r$ and therefore $\text{rank}(\widehat{H}_J) \leq g - r$.
Finally, if $\text{rank}(\widehat{H}_J) < g - r$, according to the rank-nullity theorem again, one can find a $\boldsymbol{w}^{r+1} \in \mathcal{C}_j$ such that $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r, \boldsymbol{w}^{r+1}\}$ is L.I. and also with $\text{supp}(\boldsymbol{w}^{r+1}) \subset \text{supp}(\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r)$. But this implies that $d_{r+1} \leq d_r$, which is a contradiction, hence $\text{rank}(\widehat{H}_J) = g - r$, as desired.

■

Example 13: We consider now the sliding truncated parity-check matrix $H_1^{\mathcal{C}}$ in Example 11. Recalling that we are working on $\mathbb{F}_3$, to obtain the systematic form we just need to add the first and second row to the third row. By permuting the appropriate columns we get that

$$P = P_1^C = \begin{bmatrix} 1 & 2 & 0 & 0 \\ \frac{2}{1} & \frac{2}{2} & \frac{0}{1} & \frac{0}{2} \\ 2 & 2 & 2 & 2 \end{bmatrix}$$

We already know from Example 11 that $d_1 = 4$. We shall see how to check it from the parity check matrix in a systematic form using Theorem 12. For every $1 \leq g \leq \min\{3, 4\}$, we have g columns of P with rank $\geq g$.

- $g = 1$: Since there are no zero columns, every column has rank$= 1$.
- $g = 2$: It is possible to check that every pair of columns of P is a L.I. set, therefore they determine a matrix with rank$= 2$.
- $g = 3$: It is also possible to check that every three columns of P are a L.I. set, therefore they determine a matrix with rank$= 3$.

On the other hand, there exists a matrix of size $g \times g$ with $1 < g \leq \min\{4, 4\}$ with rank $= g - r = g - 1$. For $g = 3$ the submatrix

$$\begin{bmatrix} 2 & 0 & 0 \\ 2 & 0 & 0 \\ 2 & 1 & 2 \end{bmatrix}$$

has rank$= 2$.

Theorem 14: Let C_j be a truncated convolutional code with $H = [I_{N-K} \;\; P]$ the sliding parity-check matrix in systematic form.

1) If for all $g \leq \min\{d - 1, K\}$, every submatrix of P of size $N - K - d + g \times g$ has rank $\geq g - r + 1$, then $d_r \geq d$.
2) If there exists g, with $r < g < \min\{d, K\}$, such that g columns of the matrix P have rank $g - r$ and satisfy Condition $(*)$, then $d_r \leq d$.

Proof.

1) Suppose $d > d_r$. We shall show that for some $g \leq \min\{d - 1, K\}$ there is a submatrix A of P of size $N - K - d + g \times g$ such that $\text{rank}(A) \leq g - r$. According to the definition of d_r, we have that there exists a L.I. set $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r\} \in \mathcal{C}_j$ such that $|\text{supp}(\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r)| = d_r$. Let I and J be the set of indices of $\text{supp}(\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r)$ corresponding to I_{N-K} and P (from $H = [I_{N-K} \;\; P]$), respectively, and denote by $H_{I,J} = [H_I \;\; H_J]$, the columns of H with indices in I and J.
It is easy to check that $\{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^r\} \subset \ker(H)$ and as a consequence $\text{rank}(H_{I,J}) \leq d_r - r$. On the other hand:

$$\text{rank}(H_{I,J}) \leq \text{rank}(H_I) + \text{rank}(H_J)$$

Since H_I is part of the identity matrix, we have that the matrix $H_{I,J}$ is column equivalent, up to permutation of the rows, to the matrix

$$H^*_{I,J} = \begin{bmatrix} \widehat{H_I} & 0 \\ 0 & \widehat{H_J} \end{bmatrix}$$

and clearly $\text{rank}(H_{I,J}) = \text{rank}(H^*_{I,J}) = \text{rank}(\widehat{H_I}) + \text{rank}(\widehat{H_J})$.
Let $|J| = g$ and $|I| = d_r - g$. We clearly have that $g \leq \min\{d - 1, K\}$.
We know that the matrix H_I has rank equal to $d_r - g$, so that

$$\text{rank}(\widehat{H_J}) = \text{rank}(H_{I,J}) - (d_r - g) = d_r - r - (d_r - g) = g - r.$$

Notice that $\widehat{H_J}$ is a matrix of size $N - K - d_r + g \times g$. By removing any $d - d_r$ rows of $\widehat{H_J}$ we get a submatrix A of P of size $N - K - d + g \times g$. Moreover,

$$\text{rank}(A) \leq \text{rank}(\widehat{H_J}) \leq g - r,$$

which is a contradiction.

2) It follows using the same reasoning used in previous proofs.

■

VI. Algorithm and Example

In this section we introduce an algorithm that computes the GCD hierarchy of a truncated convolutional code $\mathcal{C}_j$ and a detailed example of how this algorithm works on a practically used convolutional code.

Let H be the sliding parity-check matrix of $\mathcal{C}_j$ and consider a distance d. First, we introduce the function **Cond**(r, d, H) in Algorithm 1, which checks whether d satisfies the conditions of Corollary 10 (and thus whether d is the r-generalized column distance of $\mathcal{C}_j$). Next, Algorithm 2 shows a pseudo-code that runs over possible values of r and d, avoiding impossible values, to compute the GCD hierarchy of $\mathcal{C}_j$. This algorithm starts assuming the minimum possible value of each distance, according to the monotonicity given in equation (2) and the bound given in (3). If the distance does not satisfy the conditions given Corollary 10 (that is **Cond**$(r, d, H) = 0$), then the distance is increased by one unit until the theorem is satisfied. Notice that, according to the monotonicity given in equation (2), if one of the distances attains the bound given in (3), the posterior distances automatically attain the corresponding bounds as well.

Algorithm 1. Cond: Check if d satisfies Corollary 10.

Input: r, d and H.
function Sol =**Cond**(r, d, H)
01: Store in $[R, C]$ the size of H;
02: Store $T = \binom{C}{d}$;
03: Store $i = 1$ and $Sol = 0$;
04: **while** $i \leq T$ and $Sol = 0$;
05: Choose d columns of H;
06: **if** rank$= d - r$
07: Sol=1;
08: **else**
09: i=i+1;
10: **end if**;
11: **end while**;
12: Store $T = \binom{C}{d-1}$;
13: Store $i = 1$;
14: **while** $i \leq T$ and Sol=1;
15: Choose $d - 1$ columns satisfying Condition(*);
16: **if** rank$< d - r$
17: $Sol = 0$;
18: **else**
19: $i = i + 1$;
20: **end if**;
21: **end while**;
end function
Output:
$Sol = 1$ if the d satisfies Corollary 10,
$Sol = 0$ otherwise.

Algorithm 2. Computation of the GCD hierarchy of a truncated code C_j.

Input: H, n, k and j.
01: Initialize vector d of length $k(j + 1)$;
02: Set $d(0) = 0$;
03: **for** $r = 1$ to $k(j + 1)$
04: $d(r) = d(r - 1) + 1$;
05: **while Cond**$(r, d(r), H) = 0$
06: $d(r) = d(r) + 1$;
07: **end while**
08: **if** $d(r) = (j + 1)(n - k) + r$
09: **for** $t = r$ to $k(j + 1)$
10: $d(t) = (j + 1)(n - k) + t$;
11: **end for**
12: break;
13: **end if**
14: **end for**
Output:
d: GCD hierarchy.

The most complex part of our algorithm comes from function **Cond**, where we have to compute the rank of several sets of columns of H. It is well known that the rank of a matrix can be computed in polynomial time [35, page 119]. However, we have to compute the rank of $\binom{n(j+1)}{d} + \binom{n(j+1)}{d-1}$ sets of columns, in the worst case scenario. As a consequence, the complexity of the algorithm achieves exponential levels. This result was expected, since the computation of the distances of a code is an NP-hard problem. For example, just computing the minimum distance of a linear block code is an NP-hard problem [36].

Another way to compute the GCD hierarchy is using the definition given in (1), that is, computing the complete list of codewords of the truncated code $\mathcal{C}_j$ and for each $r = 1, 2, \ldots, k(j + 1)$ consider the support of every L.I. set of r codewords. However, this approach is presumably more expensive in terms of memory and time.

Example 15: Burst-correcting convolutional codes with low delay form a class of codes that are used in many multimedia applications, such as real-time video conference, since the transmission must be performed sequentially and with minimal perceptible delay at the destination [37], [38], [39]. We consider a simple truncated code $\mathcal{C}_3$ of a burst-correcting convolutional code with low delay with systematic encoder (introduced in [37]) whose parameters are $n = 6$, $k = 4$ and whose sliding generator matrix is given by

$$G = \left[\begin{array}{cc|cc|cc|cc} I_k & G_0 & O & G_1 & O & G_2 & O & G_3 \\ O & O & I_k & G_0 & O & G_1 & O & G_2 \\ O & O & O & O & I_k & G_0 & O & G_1 \\ O & O & O & O & O & O & I_k & G_0 \end{array}\right]$$

with

$$G_0 = G_1 = G_2 = O_{4\times 2}, \quad \text{and} \quad G_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

As sliding parity-check matrix, we can consider

$$H = \left[\begin{array}{ccc|ccc|ccc|ccc} O & O & I_2 & O & O & O & O & O & O & O & O & O \\ O & O & O & O & O & I_2 & O & O & O & O & O & O \\ O & O & O & O & O & O & O & O & I_2 & O & O & O \\ I_2 & O & O & O & O & O & O & O & O & O & O & I_2 \end{array}\right],$$

where O stands for the matrix of zeros of size 2×2. Table I shows the values of **Cond**(r, d, H) for $r = 1, \ldots, 16$ and the different guesses for d that Algorithm 2 takes. Next, we explain how Algorithm 1 proceeds to find **Cond**(r, d, H) in each case.

First, note that H has 14 zero columns and 10 non-zero columns. In order to satisfy Condition $(*)$, a set of columns must include one of the first six column and h_1 (or h_2) must be included together with h_{23} (or h_{24}).

Consider $d_r^g = d = r$, for $r = 1, 2, \ldots, 14$, and let us see whether they satisfy **Cond**$(r, d, H) = 1$:

1) There exists a set of r columns of rank $d - r = 0$ (we have up to 14 zero columns in total including columns in the first block of six).
2) For $r = 1$, there is nothing to prove. Let us consider $2 \leq r \leq 14$. We know that h_1 (or h_2) must be included together with h_{23} (or h_{24}). Therefore, our set of $d-1 = r - 1$ columns must have rank $\geq d - r = 0$, which is trivial.

This means, that for $r = 1, 2, \ldots, 14$ the first try in Algorithm 2, namely $d = 1, 2, \ldots, 14$ is the final one, that is, $d_r = r$.

Let us consider now d_{15}^g. We first have to check whether $d_{15}^g = 15$ ($r = 15$ and $d = 15$). It is obvious that all sets

of $d-1=14$ columns have rank $\geq d-r=0$. However, there is no set of $d=15$ columns with rank $=d-r=0$, since we only have 14 zero columns. Therefore, $d_{15}^g \neq 15$ and $\mathbf{Cond}(15,15,H)=0$.

Let us now verify that $d_{15}^g=16$ ($r=15$ and $d=16$) and $\mathbf{Cond}(15,16,H)=1$.

1) There exists a set of 15 columns of rank $d-r=1$ (choose, for instance, the first column and the 14 zero columns).
2) We know that h_1 (or h_2) must be included together with h_{23} (or h_{24}). Therefore, every set of $d-1=14$ columns have rank $\geq d-r=1$

Let us consider now d_{16}^g. We have to check whether $d_{16}^g=17$ ($r=16$ and $d=17$). Since we only have 14 zero columns, it is easy to see that all set of $d-1=16$ columns have rank $\geq d-r=1$. However, since the rank of any set of three or more non-zero columns is ≥ 2, there is not set of $d=17$ columns with rank$= d-r=1$. Therefore $d_{16}^g \neq 17$ and $\mathbf{Cond}(16,17,H)=0$.

Now, we can prove that $d_{16}^g=18$ ($r=16$ and $d=18$) and $\mathbf{Cond}(16,18,H)=1$:

1) There exists a set of 16 columns of rank $d-r=2$ (choose, for instance, the two first columns and the 14 zero columns).
2) We have to consider every set of $d-1=17$ columns including h_1 (or h_2) together with h_{23} (or h_{24}). If we consider any non-zero column different from h_1 (or h_2), we have that the rank is $=2$. Since we only have 14 zero columns, we necessarily have to consider at least one non-zero column and, therefore, the set will have rank ≥ 2.

It is worth noticing that none of the distances attains the upper bound, however, this code is useful in other ways, due to the low delay. This means that the code might not have an optimal recovery rate, however it corrects very fast.

It is worth noticing that at some point the value of the distances "jumps". This gives us some information on the structure of the codewords. Besides, since the length of the codewords is 24 and $d_{16}=18$, this means that the generator matrix must be sparse. We note that in this example, it is easy to determine the GCD directly from the generator matrix. We chose this particular instance since it is a small example that models an application to a "real world" problem (see [39]).

TABLE I
DIFFERENT VALUES FOR $\mathbf{Cond}(r,d,H)=0$ IN EXAMPLE 15

r	d	$\mathbf{Cond}(r,d,H)$
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1
11	11	1
12	12	1
13	13	1
14	14	1
15	15	0
	16	1
16	17	0
	18	1

VII. CONCLUSIONS

The real figure of merit of a code, with respect to its correction capabilities, is the error probability, which depends both on the code and on the channel model. Since computing this measure of performance is, in general, a very hard task, it is very common to use other indicators or functions that are easier to compute or describe instead. Such functions may be called proximal figure of merit. The minimum distance is the most important and famous of such proximal figure of merit. Since its introduction in the context of block codes, the generalized Hamming weights are perceived as a refinement of the minimal distance. This perception was recently re-enforced by the work [34] where it is shown that the error probability of a linear code over the erasure channel under list decoding or maximum likelihood decoding is expressed by its support weight distributions. We think that a challenging work would be to understand these results considering convolutional codes. Besides, there are many concepts related to GHW of block codes that deserve to be studied in the context of convolutional codes, such as Near-MDS and Almost- MDS codes, see for instance [31].

ACKNOWLEDGMENT

This work was supported by grant 2013/25977-7, Sao Paulo Research Foundation (FAPESP). The work of the first author was also supported by FAPESP, grant 2015/07246-0 and CAPES. The second author was partially supported by CNPq. The third author was partially supported by Spanish grants AICO/2017/128 of the Generalitat Valenciana and VIGROB-287 of the Univesitat d'Alacant.

REFERENCES

[1] K. Wei, Victor, "Generalized hamming weights for linear codes," *IEEE Transactions on Information Theory*, vol. 37, pp. 1412–1418, 1991.
[2] T. Kasami, T. Takata, S. Fujiwara, and S. Lin, "On the optimum bit order with respect to the state complexity of trellis diagrams for binary linear codes," *IEEE Transactions on Information Theory*, vol. 39, no. 1, pp. 242–245, 1993.
[3] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, 6th ed. Amsterdam: North-Holland, 1988.
[4] S. Roman, *Coding and Information Theory*. New York, NY: Springer, 1992.
[5] S. M. Dodunekov and I. N. Landgev, "On near-MDS codes," *Journal of Geometry*, vol. 54, pp. 30–43, 1995.
[6] M. A. De Boer, "Almost MDS codes," *Designs, Codes and Cryptography*, vol. 9, no. 2, pp. 143–155, 1996.
[7] D. Forney, "Dimension/Length Profiles and Trellis Complexity of Linear Block Codes ," *IEEE Transactions on Information Theory*, vol. 41, no. 2, pp. 1741–1752, 1995.
[8] T. Helleseth, T. Klove, V. I. Levenshtein and O. Ytrehus, "Bounds on the minimum support weights," in *IEEE Transactions on Information Theory*, vol. 41, no. 2, pp. 432-440, Mar 1995.
[9] V. Tomás, "Complete-MDP convolutional codes over the erasure channel," Ph.D. dissertation, Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, Alicante, España, July 2010.

[10] V. Tomás, J. Rosenthal, and R. Smarandache, "Decoding of MDP convolutional codes over the erasure channel," in *Proceedings of the 2009 IEEE International Symposium on Information Theory (ISIT 2009)*. Seoul, Korea: IEEE, June 2009, pp. 556–560.
[11] L. Cruvinel Lemes and M. Firer, "Generalized weights and bounds for error probability over erasure channels," in *Proceedings of 2014 Information Theory and Applications Workshop (ITA)*, 2014, pp. 1–8.
[12] F. Didier, "A new upper bound on the block error probability after decoding over the erasure channel," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4496–4503, Oct 2006.
[13] S. Fashandi, S. O. Gharan, and A. K. Khandani, "Coding over an erasure channel with a large alphabet size," in *2008 IEEE International Symposium on Information Theory*, July 2008, pp. 1053–1057.
[14] P. Almeida, D. Napp, and R. Pinto, "A new class of superregular matrices and MDP convolutional codes," *Linear Algebra and its Applications*, vol. 439, no. 7, pp. 2145–2157, 2013.
[15] P. Almeida, D. Napp, and R. Pinto, "Superregular matrices and applications to convolutional codes," *Linear Algebra and its Applications*, vol. 499, no. 7, pp. 1–25, 2016.
[16] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache, "Strongly MDS convolutional codes," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 584–598, 2006.
[17] J. Rosenthal, and E. York, " On the generalized Hamming weights of convolutional codes," *IEEE Transactions on Information Theory*, vol. 43, no. 1, pp. 330–335, 1997.
[18] R. Dodunekova, and S.M. Dodunekov, and T. Klove, " Almost-MDS and near-MDS codes for error detection," *IEEE Transactions on Information Theory*, vol. 43, no. 1, pp. 285–290, 1997.
[19] T. Klve, The weight distribution of linear codes over $GF(ql)$ having generator matrix over $GF(q)$, Discr. Math. vol. 23, pp. 159-168, 1978.
[20] R. Johannesson and K. Sh. Zigangirov. *Fundamentals of Convolutional Coding*. IEEE Press, New York, 1999.
[21] R. Hutchinson, and R. Smarandache, and J. Trumpf, "On superregular matrices and MDP convolutional codes," *Linear Algebra and its Applications*, vol. 428, pp. 2585–2596, 2008.
[22] D. Napp and R. Smarandache, ""Constructing strongly-MDS convolutional codes with maximum distance profile". *Advances in Mathematics of Communications* 10 (2), 2016.
[23] S. Ball, "On sets of vectors of a finite vector space in which every subset of basis size is a basis." *J. Eur. Math. Soc*, vol 14(3), pp. 733–748, 2010.
[24] S. D. Cardell, M. Firer and D. Napp, "Generalized column distances for convolutional codes," 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, 2017, pp. 21–25.
[25] S. D. Cardell, M. Firer and D. Napp, "Unrestricted Generalized Column Distances: A Wider Definition", 2019 IEEE International Symposium on Information Theory (ISIT), Paris, 2019, pp. 2783–2787 .
[26] G. L. Feng, K. K. Tzeng and V. K. Wei, "On The Generalized Hamming Weights Of Several Classes Of Cyclic Codes," Proceedings. 1991 IEEE International Symposium on Information Theory, 1991, pp. 53-53.
[27] A. I. Barbero and J. G. Tena, "Weight hierarchy of a product code," in IEEE Transactions on Information Theory, vol. 41, no. 5, pp. 1475-1479, Sep 1995.
[28] F. Heijnen and R. Pellikaan, "Generalized Hamming weights of q-ary Reed-Muller codes," Proceedings of IEEE International Symposium on Information Theory, Ulm, 1997, pp. 360-.
[29] M. Delgado, J. I. Farrn, P. A. Garca-Snchez and D. Llena, "On the Weight Hierarchy of Codes Coming From Semigroups With Two Generators," in IEEE Transactions on Information Theory, vol. 60, no. 1, pp. 282-295, Jan. 2014.
[30] G. Cohen, S. Litsyn and G. Zemor, "Upper bounds on generalized distances," in IEEE Transactions on Information Theory, vol. 40, no. 6, pp. 2090-2092, Nov 1994
[31] G. Viswanath and B. Sundar Rajan, "Matrix characterization of linear codes with arbitrary Hamming weight hierarchy", *Linear Algebra and its Applications,* Vol 412, Issues 23, Pages 396-407, 2006.
[32] A. Nooraiepour and T. M. Duman, "Randomized Convolutional Codes for the Wiretap Channel," in IEEE Transactions on Communications, vol. 65, no. 8, pp. 3442-3452, Aug. 2017.
[33] Shen, Lin-Zhi, and Fang-Wei Fu. "The list decoding error probability of linear codes over the erasure channel." 2015 IEEE International Symposium on Information Theory (ISIT). IEEE, 2015.
[34] Shen, Lin-Zhi, and Fang-Wei Fu. "The Decoding Error Probability of Linear Codes over the Erasure Channel." IEEE Transactions on Information Theory (2019).
[35] M. Agrawal, V. Arvind, *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*. Progress in Computer Science and Applied Logic (Book 26). Birkhuser, 2014.
[36] A. Vardy, "The intractability of computing the minimum distance of a code," in IEEE Transactions on Information Theory, vol. 43, no. 6, pp. 1757-1766, Nov. 1997.
[37] J. J. Climent, D. Napp, V. Requena, RACSAM (2020) 114: 38. https://doi.org/10.1007/s13398-019-00744-y
[38] E. Martinian and C. E. W. Sundberg , "Burst erasure correction codes with low decoding delay,", IEEE Transactions on Information Theory (2004), 50(10), 2494-2502.
[39] Badr, A. and Khisti, A. and Tan, Wai-Tian. and Apostolopoulos, J. , "Layered Constructions for Low-Delay Streaming Codes,", IEEE Transactions on Information Theory (2017), 63(1), 111-141.

Sara D. Cardell received her Ph.D. degree from the University of Alicante, Spain, in 2012. From 2013 until 2015, she was a postdoctoral researcher at the University of Alicante and Spanish Higher Council for Scientific Research. In 2015 she joined the coding theory research group at the University of Campinas, Brazil. Her research interests include $\mathbb{F}_q$-linear codes, SPC codes, stream ciphers and cryptanalysis.

Marcelo Firer received the B.Sc. and M.Sc. degrees in 1989 and 1991 respectively, from State University of Campinas, Brazil, and the Ph.D. degree from the Hebrew University of Jerusalem, in 1997, all in Mathematics. He is currently an Associate Professor of the State University of Campinas, where he served as director of the Universitys Science Museum (2008 2013) and Undergraduate Chair for Mathematics. His research interests include coding theory, group actions, and finite metric spaces and their relationship to coding theory.

Diego Napp joined the Department of Mathematics of the University of Groningen (The Netherlands) in 2004. After 4 years, he received a Ph.D. degree in Mathematics from that University. He immediately moved to the University of Aveiro (Portugal) to be a Post-doc researcher. Between 2011 and 2013, he worked in the University of Valladolid and University Jaume I supported by a Spanish research contract called Juan de la Cierva. He moved back to the University of Aveiro where he holded a research position until 2019, when he joined the Department of Mathematics at University of Alicante. He is mainly interested in systems and coding theory.