



# Construction of LDPC convolutional codes via difference triangle sets

Gianira N. Alfarano<sup>1</sup> · Julia Lieb<sup>1</sup> · Joachim Rosenthal<sup>1</sup>

Received: 27 August 2020 / Revised: 27 August 2020 / Accepted: 3 July 2021 /  
Published online: 22 July 2021  
© The Author(s) 2021

## Abstract

In this paper, a construction of  $(n, k, \delta)$  LDPC convolutional codes over arbitrary finite fields, which generalizes the work of Robinson and Bernstein and the later work of Tong is provided. The sets of integers forming a  $(k, w)$ -(weak) difference triangle set are used as supports of some columns of the sliding parity-check matrix of an  $(n, k, \delta)$  convolutional code, where  $n \in \mathbb{N}$ ,  $n > k$ . The parameters of the convolutional code are related to the parameters of the underlying difference triangle set. In particular, a relation between the free distance of the code and  $w$  is established as well as a relation between the degree of the code and the scope of the difference triangle set. Moreover, we show that some conditions on the weak difference triangle set ensure that the Tanner graph associated to the sliding parity-check matrix of the convolutional code is free from  $2\ell$ -cycles not satisfying the full rank condition over any finite field. Finally, we relax these conditions and provide a lower bound on the field size, depending on the parity of  $\ell$ , that is sufficient to still avoid  $2\ell$ -cycles. This is important for improving the performance of a code and avoiding the presence of low-weight codewords and absorbing sets.

**Keywords** LDPC codes · Difference triangle sets · Convolutional codes

**Mathematics Subject Classification** 11T71 · 68R05 · 94B10

---

Communicated by K. T. Arasu.

---

This work is an extension of the 5 pages conference paper “Construction of Rate  $(n - 1)/n$  Non-Binary LDPC Convolutional Codes via Difference Triangle Sets” [1], which will appear in *Proceedings of IEEE International Symposium on Information Theory* 2020.

---

✉ Gianira N. Alfarano  
gianiranicoletta.alfarano@math.uzh.ch  
  
Julia Lieb  
julia.lieb@math.uzh.ch  
  
Joachim Rosenthal  
rosenthal@math.uzh.ch

<sup>1</sup> Institute of Mathematics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland

## 1 Introduction

In the last three decades, the area of channel coding gained a lot of attention, due to the fact that many researchers were attracted by the practical realization of coding schemes whose performances approach the Shannon limit. This revolution started in 1993 with the invention of turbo codes and their decoding algorithms [6]. Only few years later, researchers investigated also low-density parity-check (LDPC) block codes and their message passing decoding algorithm. These codes were discovered to be also capable of capacity-approaching performances. The class of LDPC block codes was introduced by Gallager [14], in 1962. Their name is due to the fact that they have a parity-check matrix that is sparse. The analysis of LDPC codes attracted many researchers and a lot of work arose in this direction, starting from the papers of Wiberg [38] and Mackay and Neal [21]. Moreover, in [9,26] analytical tools were introduced to investigate the limits of the performance of the message passing iterative decoding algorithm, suggested by Tanner already in 1981, [30].

Similarly to LDPC block codes, one can consider LDPC convolutional codes. These codes are defined as the (right) kernel of a sparse sliding parity-check matrix, which allows to still use iterative message passing decoding algorithms. Moreover, it was proven that LDPC convolutional codes are practical in different communication applications, see for instance [3,4,23].

In the last few years, some attempts to construct binary LDPC convolutional codes were done. Two types of constructions were mainly investigated. The first one exploits the similarity of quasi-cyclic block codes and time-invariant LDPC convolutional codes, [31–33]. The second one regards mostly time varying convolutional codes, see for instance [5,25,39].

The aim of this paper is to give a combinatorial construction of LDPC convolutional codes suitable for iterative decoding. In fact, contrary to LDPC block codes for which a lot of combinatorial constructions have been derived (see for example [16,19,20,29,36,37]), it is rare to use combinatorial tools for constructing LDPC convolutional codes.

In 1967, Robinson and Bernstein [27] used difference triangle sets for the first time to construct binary recurrent codes, which are defined as the (right) kernel of a binary sliding matrix. At that time, the theory of convolutional codes was not developed yet and the polynomial notation was not used, but now, we may regard recurrent codes as a first prototype of convolutional codes. This was the first time that a combinatorial object has been used to construct convolutional codes. Three years later, Tong in [34], used diffuse difference triangle sets to construct self-orthogonal diffuse convolutional codes, defined by Massey [22]. The aim of these authors was to construct codes suitable for iterative decoding and their result was an adapted version of binary LDPC convolutional codes. In [1], the authors constructed  $(n, n - 1)_q$  LDPC convolutional codes, whose sliding parity-check matrix is free from 4 and 6-cycles not satisfying the so called full rank condition, starting from difference triangle sets. This was a generalization of the work of Robinson and Bernstein, in which difference triangle sets were used to construct convolutional codes over the binary field, that can only avoid 4-cycles. In 1971, Tong [35] was the first to generalize their construction over  $\mathbb{F}_q$ , using what we call in this paper weak difference triangle sets. However, his construction is suitable only for limited rate and in a way that the Tanner graph associated to the parity-check matrix of these codes is free only from 4-cycles.

In this paper, we give a construction of LDPC convolutional codes for arbitrary rates over arbitrary fields, using difference triangle sets and weak difference triangle sets. In particular, the use of the weak version of these combinatorial objects allows to relax the assumptions required by Robinson, Bernstein and Tong. Indeed, instead of considering sets of nonnegative

integers where all the pairwise differences are distinct among all the sets, we may require only that the pairwise differences are distinct in each set. Moreover, we show that using difference triangle sets for this construction produces codes with good distance properties and we provide a bound on the field size that is sufficient to have codes with good distance and to avoid the presence of cycles not satisfying the full rank condition.

The paper is structured as follows. In Sect. 2, we start with some background about convolutional codes. Then we define difference triangle sets (DTSs) and weak difference triangle sets (wDTSs) and their scope. Finally, we introduce LDPC block and LDPC convolutional codes over finite fields of arbitrary size. In Sect. 3, we generalize the construction in [1] for LDPC convolutional codes to arbitrary rates, starting from a weak difference triangle set. We show how the parameters of the constructed convolutional code depend on the properties of the weak difference triangle set. We derive some distance properties of the codes and the exact formula for computing their density. Moreover, we show that the free distance and the column distances of convolutional codes constructed from a DTS are in some sense optimal. Finally, we present a construction of LDPC convolutional codes from a wDTS and a lower bound on the corresponding field size such that the free distance is at least 3 and such that the corresponding Tanner graph is free from 4 and 6-cycles not satisfying the FRC. In Sect. 4, we start with some conditions on the underlying wDTS that ensure that the Tanner graph associated to the sliding parity-check matrix is free from any cycle over any finite field. Afterwards, we give a lower bound for the field size sufficient to ensure that our construction provides a Tanner graph free from  $2\ell$ -cycles not satisfying the FRC, for  $\ell$  odd. If  $\ell$  is even, we add some assumptions on the wDTS to be able to derive also in this case a lower bound on the field size. Finally, we modify our construction to be able to relax these conditions on the wDTS, which in turn enlarges the underlying field size.

## 2 Preliminaries

In this section, we provide the background and the known results useful for the rest of the paper.

### 2.1 Convolutional codes

Let  $q$  be a prime power and  $\mathbb{F}_q$  be the finite field with  $q$  elements. Let  $k, n$  be positive integers, with  $k < n$  and consider the polynomial ring  $\mathbb{F}_q[z]$ . An  $(n, k)_q$  convolutional code is defined as a submodule  $\mathcal{C}$  of  $\mathbb{F}_q[z]^n$  of rank  $k$ , such that there exists a polynomial generator matrix  $G(z) \in \mathbb{F}_q[z]^{k \times n}$ , whose rows generate  $\mathcal{C}$ , i.e.

$$\mathcal{C} := \{u(z)G(z) \mid u(z) \in \mathbb{F}_q[z]^k\} \subseteq \mathbb{F}_q[z]^n.$$

If  $G(z)$  is *basic*, i.e., it has a right polynomial inverse, there exists a full row-rank parity-check matrix  $H(z) \in \mathbb{F}_q[z]^{(n-k) \times n}$  such that

$$\mathcal{C} := \{v(z) \in \mathbb{F}_q[z]^n \mid H(z)v(z)^T = 0\}.$$

Finally, we define the *degree*  $\delta$  of the convolutional code  $\mathcal{C}$  as the highest degree among the  $k \times k$  minors in  $G(z)$ . When the degree  $\delta$  of the convolutional code is known, we denote  $\mathcal{C}$  by  $(n, k, \delta)_q$ . We omit to specify the field when it is not needed. If  $G(z)$  is *reduced*, i.e. the sum of the row degrees of  $G(z)$  attains the minimal possible value, then this value is equal to  $\delta$ .

**Lemma 1** ([13,17]) *Let  $H(z) = [h_{i,j}(z)] \in \mathbb{F}[z]^{(n-k) \times n}$  with row degrees  $v_1, v_2, \dots, v_{n-k}$  and  $[H]_{hr}$  be the highest row degree coefficient matrix defined as the matrix with the  $i$ th row consisting of the coefficients of  $z^{v_i}$  in the  $i$ th row of  $H(z)$ . Then  $H(z)$  is reduced if and only if  $[H]_{hr}$  is full row-rank.*

It is well-known that if  $H(z) \in \mathbb{F}[z]^{(n-k) \times n}$  is a basic and reduced parity-check matrix of an  $(n, k, \delta)$  convolutional code  $\mathcal{C}$ , then the sum of the row degrees of  $H(z)$  is equal to  $\delta$  (see [28]) which is also equal to the maximal (polynomial) degree of the full-size minors of  $H(z)$ . If  $H(z)$  is basic but not reduced, then the sum of its row degrees is larger than  $\delta$ .

There is a natural isomorphism between  $\mathbb{F}_q[z]^n$  and  $\mathbb{F}_q^n[z]$ , which extends to the space of matrices and allows to consider a generator and a parity-check matrix of a convolutional code as polynomials whose coefficients are matrices. In particular, we will consider  $H(z) \in \mathbb{F}_q^{(n-k) \times n}[z]$ , such that  $H(z) = H_0 + H_1z + \dots + H_\mu z^\mu$ , with  $\mu > 0$ . With this notation, we can expand the kernel representation  $H(z)v(z)^\top$  in the following way:

$$Hv^\top = \begin{bmatrix} H_0 & & & & & & \\ \vdots & \ddots & & & & & \\ H_\mu & \cdots & H_0 & & & & \\ & \ddots & & \ddots & & & \\ & & & & H_\mu & \cdots & H_0 \\ & & & & & \ddots & \\ & & & & & & \vdots \\ & & & & & & H_\mu \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_r \end{bmatrix} = 0, \tag{1}$$

where  $r = \deg(v)$ . We will refer to the representation of the parity-check matrix of  $\mathcal{C}$  in Eq. (1) as *sliding parity-check matrix*.

Let  $v(z) = \sum_{i=0}^r v_i z^i \in \mathbb{F}_q^n[z]$  be a polynomial vector. We define the *weight* of  $v(z)$  as the sum of the Hamming weights of its coefficients, i.e.  $\text{wt}(v(z)) := \sum_{i=0}^r \text{wt}_H(v_i) \in \mathbb{N}_0$ , where  $\text{wt}_H(v_i)$  denotes the Hamming weight of  $v_i \in \mathbb{F}_q^n$ . This definition allows to endow a convolutional code  $\mathcal{C} \subseteq \mathbb{F}_q[z]^n$  with a distance. We define the *free distance* of  $\mathcal{C}$ , denoted by  $d_{\text{free}}(\mathcal{C})$ , as the minimum of the nonzero weights of the codewords in  $\mathcal{C}$ . The parameters  $\delta$  and  $d_{\text{free}}$  are needed to determine respectively the decoding complexity and the error correction capability of a convolutional code with respect to some decoding algorithm. For this reason, for any given  $k$  and  $n$  and field size  $q$ , the aim is to construct convolutional codes with ‘‘small’’ degree  $\delta$  and ‘‘large’’ free distance  $d_{\text{free}}$ .

For any  $j \in \mathbb{N}_0$  we define the  $j$ th column distance of  $\mathcal{C}$  as

$$\begin{aligned} d_j^c(\mathcal{C}) &:= \min \left\{ \text{wt} \left( v_0 + v_1z + \dots + v_jz^j \right) \mid v(z) \in \mathcal{C}, v_0 \neq 0 \right\} \\ &= \min \left\{ \text{wt} \left( v_0 + \dots + v_jz^j \right) \mid H_j^c[v_0 \cdots v_j]^\top = 0, v_0 \neq 0 \right\}, \end{aligned}$$

where  $H_j^c$  is called  $j$ -th truncated parity-check matrix and it is defined for any  $j \in \mathbb{N}_0$  as

$$H_j^c := \begin{bmatrix} H_0 & & & & \\ H_1 & H_0 & & & \\ \vdots & \vdots & \ddots & & \\ H_j & H_{j-1} & \cdots & H_0 & \end{bmatrix} \in \mathbb{F}_q^{(j+1)(n-k) \times (j+1)n}.$$

We also recall the following well-known result.

**Theorem 1** ([15, Proposition 2.1]) *Let  $\mathcal{C} \subseteq \mathbb{F}_q[z]^n$  be an  $(n, k)_q$  convolutional code. Let  $d \in \mathbb{N}$ . Then the following properties are equivalent.*

1.  $d_j^c = d$ .
2. *None of the first  $n$  columns of  $H_j^c$  is contained in the span of any other  $d - 2$  columns and one of the first  $n$  columns of  $H_j^c$  is in the span of some other  $d - 1$  columns of that matrix.*

### 2.2 Difference triangle sets

A difference triangle set is a collection of sets of integers such that any integer can be written in at most one way as difference of two elements in the same set. Difference triangle sets find application in combinatorics, radio systems, optical orthogonal codes and other areas of mathematics [7,8,18]. We refer to [10] for a more detailed treatment. More formally, we define them in the following way, by distinguishing between weak difference triangle sets and difference triangle sets.

**Definition 1** Let  $N, M$  be positive integers. An  $(N, M)$ -weak difference triangle set (wDTS) is a collection of sets  $\mathcal{T} := \{T_1, T_2, \dots, T_N\}$ , where for any  $1 \leq i \leq N, T_i := \{a_{i,j} \mid 1 \leq j \leq M\}$  is a set of nonnegative integers such that  $a_{i,1} < a_{i,2} < \dots < a_{i,M}$  and for  $1 \leq i \leq N$  the differences  $a_{i,j} - a_{i,k}$ , with  $1 \leq k < j \leq M$  are distinct. If all the differences in all the sets are distinct, we call  $\mathcal{T}$  a  $(N, M)$ -difference triangle set (DTS).

An important parameter characterizing an  $(N, M)$ -(w)DTS  $\mathcal{T}$  is the *scope*  $m(\mathcal{T})$ , which is defined as

$$m(\mathcal{T}) := \max\{a_{i,M} \mid 1 \leq i \leq N\}.$$

A very well-studied problem in combinatorics is finding families of  $(N, M)$ -DTSs with minimum scope. In this work, we will use the sets in a (w)DTS as supports of some columns of a sliding parity-check matrix of a convolutional code. We will then relate the scope of the (w)DTS with the degree of the code. Since we want to minimize the degree of the code, it is evident that the mentioned combinatorial problem plays a crucial role also here.

The name “difference triangle” is derived from a way of writing the differences inside the sets composing  $\mathcal{T}$  in a triangular form .

**Example 1** (wDTS) Let  $\mathcal{T} = \{\{1, 2, 4, 8\}, \{1, 3, 7, 15\}, \{1, 5, 10, 16\}\}$ . Then  $\mathcal{T}$  is a  $(3, 4)$ -wDTS.

The “triangles” associated to  $\mathcal{T}$  are the following:

$$\begin{array}{cccc} 1 & 2 & 4 & \\ & 3 & 6 & \\ & & 7 & \end{array} \quad \begin{array}{cccc} 2 & 4 & 8 & \\ & 6 & 12 & \\ & & 14 & \end{array} \quad \begin{array}{cccc} 4 & 5 & 6 & \\ & 9 & 11 & \\ & & 15 & \end{array}$$

**Example 2** (DTS) Let  $\mathcal{T} = \{\{1, 4, 16, 20\}, \{1, 7, 12, 14\}, \{1, 9, 18, 19\}\}$ . Then  $\mathcal{T}$  is a  $(3, 4)$ -DTS.

The “triangles” associated to  $\mathcal{T}$  are the following:

$$\begin{array}{cccc} 3 & 12 & 4 & \\ & 15 & 16 & \\ & & 19 & \end{array} \quad \begin{array}{cccc} 6 & 5 & 2 & \\ & 11 & 7 & \\ & & 13 & \end{array} \quad \begin{array}{cccc} 8 & 9 & 1 & \\ & 17 & 10 & \\ & & 18 & \end{array}$$

### 2.3 LDPC codes over arbitrary finite fields

LDPC codes are known for their performance near the Shannon-limit over the additive white Gaussian noise channel [21]. Shortly after they were rediscovered, binary LDPC codes were generalized over arbitrary finite fields. This new construction was first investigated by Davey and Mackay in 1998 in [11]. In [12], it was observed that LDPC codes defined over a finite field with  $q$  elements can have better performances than the binary ones. An LDPC code is defined as the kernel of an  $N \times M$  sparse matrix  $H$  with entries in  $\mathbb{F}_q$ . We can associate to  $H$  a bipartite graph  $\mathcal{G} = (V, E)$ , called *Tanner graph*, where  $V = V_s \cup V_c$  is the set of vertices. In particular,  $V_s = \{v_1, \dots, v_N\}$  is the set of *variable nodes* and  $V_c = \{c_1, \dots, c_M\}$  is the set of *check nodes*.  $E \subseteq V_s \times V_c$  is the set of edges, with  $e_{n,m} = (v_n, c_m) \in E$  if and only if  $h_{n,m} \neq 0$ . The edge  $e_{n,m}$  connecting a check node and a variable node is labelled by  $h_{n,m}$ , that is the corresponding *permutation node*. For an even integer  $m = 2\ell$ , we call a simple closed path consisting of  $\ell$  check nodes and  $\ell$  variable nodes in  $\mathcal{G}$  an *m-cycle*. The length of the shortest cycle is called the *girth* of  $\mathcal{G}$  or *girth* of  $H$ . It is proved that higher the girth is, the lower the decoding failure of the bit flipping algorithm is. Moreover, in [24] the authors showed that short cycles in an LDPC code may be harmful if they do not satisfy the so called full rank condition (FRC). This is because if the FRC is not satisfied, the short cycles produce low-weight codewords or they form absorbing sets, [2].

Moreover, in [24] and in [2] it is shown that an  $m$ -cycle, with  $m = 2\ell$  in an LDPC code with parity-check matrix  $H$  can be represented, up to permutations, by an  $\ell \times \ell$  submatrix of  $H$  of the form

$$A = \begin{bmatrix} a_1 & a_2 & 0 & \cdots & \cdots & 0 \\ 0 & a_3 & a_4 & \cdots & \cdots & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ 0 & & & & a_{2\ell-3} & a_{2\ell-2} \\ a_{2\ell} & 0 & \cdots & \cdots & 0 & a_{2\ell-1} \end{bmatrix}, \tag{2}$$

where  $a_i \in \mathbb{F}_q^*$ . The cycle does not satisfy the FRC if the determinant of  $A$  is equal to 0. In this case, the cycle gives an absorbing set. Hence, it is a common problem to construct LDPC codes in which the shortest cycles satisfy the FRC.

In this work, we are interested in the convolutional counterpart of LDPC block codes, which is given by convolutional codes defined over a finite field  $\mathbb{F}_q$  as kernel of a sparse sliding parity-check matrix (here with sparse we mean that in particular each  $H_i$  is sparse).

### 3 Construction of LDPC convolutional codes

In this section, we use difference triangle sets to construct LDPC convolutional codes over  $\mathbb{F}_q$ . The construction was provided for  $(n, n - 1)_q$  convolutional codes in [1]. Here, we generalize it for arbitrary  $n$  and  $k$ .

We will construct a sliding parity-check matrix  $H$  as in (1), whose kernel defines a convolutional code. Due to the block structure of  $H$ , it is enough to consider

$$\mathcal{H} := H_\mu^c = \begin{bmatrix} H_0 & & & \\ H_1 & H_0 & & \\ \vdots & \vdots & \ddots & \\ H_\mu & H_{\mu-1} & \cdots & H_0 \end{bmatrix}, \tag{3}$$

since  $H$  is then constructed by sliding it. It is easy to see that  $H$  does contain a cycle of length  $2\ell$  not satisfying the FRC if and only if  $\mathcal{H}$  does. Assuming that  $H_0$  is full rank, we can perform Gaussian elimination on the matrix

$$\begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_\mu \end{bmatrix},$$

which results in the block matrix

$$\tilde{H} = \left[ \begin{array}{c|c} A_0 & I_{n-k} \\ A_1 & 0 \\ \vdots & \vdots \\ A_\mu & 0 \end{array} \right], \tag{4}$$

with  $A_i \in \mathbb{F}_q^{(n-k) \times k}$  for  $i = 1, \dots, \mu$ . With abuse of notation, we write  $H_0$  for  $[A_0|I_{n-k}]$ , and  $H_i$  for the matrices  $[A_i|0]$ .

**Remark 1** If we define the matrix  $\tilde{H}(z) = \sum_{i=0}^\mu A_i z^i \in \mathbb{F}_q[z]^{(n-k) \times k}$ , then we obtain that  $H(z) = [\tilde{H}(z) \ I_{n-k}]$  and hence  $H(z)$  has a polynomial right inverse, i.e.  $H(z)$  is basic.

Given  $n \in \mathbb{N}$ , with the following definition we describe how we construct the above mentioned matrix  $\tilde{H}$  from a  $(k, w)$ -wDTS, which then will define an  $(n, k)_q$  convolutional code.

**Definition 2** Let  $k, n$  be positive integers with  $n > k$  and  $\mathcal{T} := \{T_1, \dots, T_k\}$  be a  $(k, w)$ -wDTS with scope  $m(\mathcal{T})$ . Set  $\mu = \left\lceil \frac{m(\mathcal{T})}{n-k} \right\rceil - 1$  and define the matrix  $\tilde{H} \in \mathbb{F}_q^{(\mu+1)(n-k) \times n}$ , in which the  $l$ th column has weight  $w$  and support  $T_l$ , i.e. for any  $1 \leq i \leq (\mu + 1)(n - k)$  and  $1 \leq l \leq k$ ,  $\tilde{H}_{i,l} \neq 0$  if and only if  $i \in T_l$ . We say that  $\tilde{H}$  has support  $\mathcal{T}$ . The last  $n - k$  columns of  $\tilde{H}$  are given by  $[I_{n-k}, 0_{n-k}, \dots, 0_{n-k}]^\top$ . Derive the matrix  $\mathcal{H}$  by “shifting” the columns of  $\tilde{H}$  by multiples of  $n - k$  and then a sliding matrix  $H$  of the form of Eq. (1). Finally, define  $\mathcal{C} := \ker(\mathcal{H})$  over  $\mathbb{F}_q$ .

Observe that if  $k = n - 1$ , we simply get the construction provided in [1, Definition 4].

**Proposition 1** Let  $n, k, w$  be positive integers with  $n > k$ ,  $\mathcal{T}$  be a  $(k, w)$ -wDTS with scope  $m(\mathcal{T})$  and set  $\mu = \left\lceil \frac{m(\mathcal{T})}{n-k} \right\rceil - 1$ . If  $\tilde{H}$  has support  $\mathcal{T}$ , then the corresponding code is an  $(n, k, \delta)$  convolutional code with  $\mu \leq \delta \leq \mu(n - k)$ . Moreover  $H_\mu$  is full rank if and only if  $\delta = \mu(n - k)$ .

**Proof** As the matrix  $H(z)$  defined in Remark 1 is basic,  $\delta$  is the maximal degree of the full-size minors of  $H$ , which is clearly upper bounded by  $\mu(n - k)$ . Moreover, any minor formed by a column with degree  $\mu$  and suitable columns of the systematic part of  $H$  has degree  $\mu$ , which proves the lower bound.

If  $H_\mu$  is full rank, it is equal to  $[H]_{hr}$ , and  $H$  is reduced. Hence,  $\delta$  is equal to the sum of the  $n - k$  row degrees that are all equal to  $\mu$ , i.e.  $\delta = \mu(n - k)$ . If  $H_\mu$  is not full rank, there are two possible cases. First, if  $H_\mu$  contains no all-zero row, then  $[H]_{hr} = H_\mu$  is not full rank, and hence  $\delta$  is strictly smaller than the sum of the row degrees which is  $\mu(n - k)$ . Second, if  $H_\mu$  contains a row of zeros, then the sum of the row degrees of  $H$  is strictly smaller than  $\mu(n - k)$  and thus, also  $\delta$  is strictly smaller than  $\mu(n - k)$ .  $\square$

**Remark 2** If  $k < n - k$ , i.e. the rate of the code is smaller than  $1/2$ , then (4) implies that  $H_\mu$  cannot be full rank. Moreover, in this case,  $[H]_{hr}$  can only be full rank if at least  $n - 2k$  row degrees of  $H$  are zero.

**Proposition 2** Let  $n, k, w$  be positive integers with  $n > k$  and  $\mathcal{T}$  be a  $(k, w)$ -wDTS. Assume  $\bar{H}$  has support  $\mathcal{T}$  and consider the convolutional code  $\mathcal{C}$  constructed as kernel of the sliding parity-check matrix corresponding to  $\bar{H}$ . If  $N$  is the maximal codeword length, i.e. for any codeword  $v(z) \in \mathcal{C}$ ,  $\deg(v) + 1 \leq N/n$ , then the sliding parity-check matrix corresponding to  $\bar{H}$  has density

$$\frac{wk + n - k}{(n - k)(\mu n + N)}$$

**Proof** To compute the density of a matrix, one has to divide the number of nonzero entries by the total number of entries. The result follows immediately.  $\square$

**Theorem 2** Let  $\mathcal{C}$  be an  $(n, k)$  convolutional code with parity-check matrix  $H$ . Assume that all the columns of  $[A_0^\top \cdots A_\mu^\top]^\top$  defined as in (4) have weight  $w$  and denote by  $w_j$  the minimal column weight of  $[A_0^\top \cdots A_j^\top]^\top$ . For  $I \subset \{1, \dots, (n - k)(\mu + 1)\}$  and  $J \subset \{1, \dots, n(\mu + 1)\}$  we define  $[\mathcal{H}]_{I;J}$  as the submatrix of  $\mathcal{H}$  with row indices  $I$  and column indices  $J$ . Assume that for some  $\tilde{w} \leq w$  all  $I, J$  with  $|J| \leq |I| \leq \tilde{w}$  and  $j_1 := \min(J) \leq k$  and  $I$  containing the indices where column  $j_1$  is nonzero, we have that the first column of  $[\mathcal{H}]_{I;J}$  is not contained in the span of the other columns of  $[\mathcal{H}]_{I;J}$ . Then

- (i)  $\tilde{w} + 1 \leq d_{\text{free}}(\mathcal{C}) \leq w + 1$ ,
- (ii)  $\min(w_j, \tilde{w}) + 1 \leq d_j^c(\mathcal{C}) \leq w_j + 1$ .

**Proof** (i) Without loss of generality, we can assume that the first entry in the first row of  $H_0$  is nonzero. Denote the first column of  $\mathcal{H}$  by  $[h_{1,1}, \dots, h_{1,(n-k)\mu}]^\top$ . Then,  $v(z) = \sum_{i=0}^r v_i z^i$  with

$$\begin{aligned} v_0 &= [1 \ 0 \cdots 0 \ -h_{1,1} \cdots -h_{1,(n-k)}] \quad \text{and} \\ v_i &= [0 \ 0 \cdots 0 \ -h_{1,(n-k)i+1} \cdots -h_{1,(n-k)(i-1)}], \end{aligned}$$

for  $i \geq 1$  is a codeword with  $\text{wt}(v(z)) = w + 1$  as the weight of the first column of  $\mathcal{H}$  is equal to  $w$ . Hence  $d_{\text{free}} \leq w + 1$ .

Assume by contradiction that there exists a codeword  $v(z) \neq 0$  with weight  $d \leq \tilde{w}$ . We can assume that  $v_0 \neq 0$ , i.e. there exists  $i \in \{1, \dots, n\}$  with  $v_{0,i} \neq 0$ . We know that  $\mathcal{H}v^\top = 0$  and from (4) we obtain that there exists  $j \in \{1, \dots, n\}$  with  $j \neq i$  and  $v_{0,j} \neq 0$  and we can assume that  $i \leq k$ .

Now, we consider the homogeneous system of linear equations given by  $\mathcal{H}v^\top = 0$  and we only take the rows, i.e. equations, where column  $i$  of  $\mathcal{H}$  has nonzero entries. Moreover, we define  $\tilde{v} \in \mathbb{F}^d$  as the vector consisting of the nonzero components of  $v_0, v_1, \dots, v_{\deg(v)}$ . We end up with a system of equations of the form  $[\mathcal{H}]_{I;J} \tilde{v}^\top = 0$  where  $[\mathcal{H}]_{I;J}$  fulfills



the assumptions stated in the theorem. But this is a contradiction as  $\tilde{v}^\top$  has all components nonzero and therefore  $[\mathcal{H}]_{I,J} \tilde{v}^\top = 0$  implies that the first column of  $[\mathcal{H}]_{I,J}$  is contained in the span of the other columns of this matrix.

(ii) The result follows from Theorem 1 with an analogue reasoning as in part (i). □

**Remark 3** With the assumptions of Theorem 2, if  $\tilde{w} = w$ , one has  $d_j^c = d_{\text{free}}$  for  $j \geq \mu$ . Moreover, if  $\tilde{H}$  has support  $\mathcal{T}$ , one achieves higher column distances (especially for small  $j$ ) if the elements of  $\mathcal{T}$  are small.

**Corollary 1** *If  $\mathcal{T}$  is a  $(k, w)$ -DTS and  $\mathcal{C}$  is an  $(n, k)$  convolutional code constructed from  $\mathcal{T}$  as in Definition 2, then one has that:*

- (i)  $d_{\text{free}}(\mathcal{C}) = w + 1$ ,
- (ii)  $d_j^c(\mathcal{C}) = w_j + 1$ .

**Proof** As already mentioned in [27], matrices  $\mathcal{H}$  constructed from a DTS have the property that for every pair of columns, their supports intersect at most once. Since  $[\mathcal{H}]_{I,J}$  as defined in Theorem 2 has the property that all entries in the first column are non-zero, all other columns have at most one non-zero entry. But this implies that the first column cannot be in the span of the other columns and thus, the requirements of Theorem 2 are fulfilled for  $\tilde{w} = w$ , which proves the corollary. □

**Remark 4** If  $n - k > 1$ , it is not necessary to have a DTS to obtain that all columns of  $\mathcal{H}$  intersect at most once since one only has to consider shifts of columns by multiples of  $n - k$ . Therefore, we still need to consider a set  $\mathcal{T} = \{T_1, \dots, T_k\}$  such that all the differences  $a_{i_1, j_1} - a_{i_1, s_1}$  and  $a_{i_2, j_2} - a_{i_2, s_2}$  for  $i_1 \neq i_2$  are different, i.e. two differences coming from different triangles of  $\mathcal{T}$  have always to be different, but  $a_{i, j_1} - a_{i, s_1}$  and  $a_{i, j_2} - a_{i, s_2}$ , i.e. differences coming from the same triangle, only have to be different if  $(n - k) \mid (a_{i, j_1} - a_{i, j_2})$ .

**Example 3** Consider  $n = 3, k = 1$  and  $T_1 = \{1, 2, 3\}$ . It holds  $2 - 1 = 3 - 2$  but since  $3 - 2$  is not divisible by  $n - k = 2$ , this does not matter and we still get that all columns of  $\mathcal{H}$  intersect at most once. For example for  $\mu = 1$ , we get

$$\mathcal{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

From Corollary 1 we know that if we use a DTS to construct the parity-check matrix of the code, then the values of the nonzero entries are not important to achieve good distance properties. In the following, we present a construction that achieves also quite large distances if one takes the sets in a wDTS as support sets for the columns of the non-systematic part of  $\tilde{H}$ . Moreover, in Sect. 4, we show that this construction ensures that the Tanner graph associated to  $H$  is free from cycles of arbitrary length not satisfying the FRC if the size of the underlying field is sufficiently large and the wDTS fulfills some additional properties.

**Definition 3** Let  $k, n$  be positive integers with  $n > k$  and  $\mathcal{T} := \{T_1, \dots, T_k\}$  be a  $(k, w)$ -wDTS with scope  $m(\mathcal{T})$ . Set  $\mu = \left\lceil \frac{m(\mathcal{T})}{n-k} \right\rceil - 1$  and let  $\alpha$  be a primitive element for  $\mathbb{F}_q$ , so that every non-zero element of  $\mathbb{F}_q$  can be written as power of  $\alpha$ . For any  $1 \leq i \leq (\mu + 1)(n - k)$ ,  $1 \leq l \leq k$ , define

$$\tilde{H}_{i,l}^\mathcal{T} := \begin{cases} \alpha^{il} & \text{if } i \in T_l \\ 0 & \text{otherwise} \end{cases}.$$



with  $\mathcal{H}^T$  as defined in (3) and assume that  $(\mu + 1)(n - k) > 2$ . Assume also that  $q = p^N$ , where  $p > 2$  and

$$N > (\mu + 1)(n - k)(k - 1) = \left\lceil \frac{m(T)}{n - k} \right\rceil (n - k)(k - 1).$$

Then, all the  $3 \times 3$  minors in  $\mathcal{H}^T$  that are non-trivially zero are non-zero.

**Proof** We need to distinguish different cases.

*Case I* The  $3 \times 3$  minors are of the form

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{vmatrix},$$

with  $a_i \neq 0$  for any  $i$ . As we observed in Theorem 3, in this case all the columns are shifts of three different columns from  $\bar{H}^T$ , since each column can intersect any of its shifts at most once. Observe that we can write a minor of this form as

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{vmatrix} = \begin{vmatrix} \alpha^{ij} & \alpha^{lu} & \alpha^{tm} \\ \alpha^{(i+r)j} & \alpha^{(l+r)u} & \alpha^{(t+r)m} \\ \alpha^{(i+r+s)j} & \alpha^{(l+r+s)u} & \alpha^{(t+r+s)m} \end{vmatrix},$$

where  $1 \leq i, l, t \leq (\mu + 1)(n - k)$ ,  $r, s \in \mathbb{Z}$  are possibly negative, with  $r \neq s$ , and  $1 \leq j, u, m \leq k$  representing the index of the column from which the selected element comes from (or if the selected elements belongs to the shift of some column,  $j, u, m$  are still the indexes of the original column). Due to symmetry in this case we can assume  $r, s \in \mathbb{N}$  and  $1 \leq i, l, t \leq (\mu + 1)(n - k) - 3$ . Moreover,  $-(\mu + 1)(n - k) + 1 \leq i + r, l + r, t + r \leq (\mu + 1)(n - k) - 1$  and  $-(\mu + 1)(n - k) \leq i + r + s, l + r + s, t + r + s \leq (\mu + 1)(n - k)$ . This determinant is 0 if and only if

$$\alpha^{ru+rm+sm} + \alpha^{rm+rj+s} + \alpha^{rj+ru+sk} = \alpha^{ru+rj+s} + \alpha^{rj+rm+sm} + \alpha^{ru+rm+sk}. \tag{5}$$

Without loss of generality we can assume that  $j < u < m$  and it turns out that the maximum exponent in Eq. (5) is  $ru + rm + sm$  while the minimum is  $ru + rj + sj$ . Let  $M := ru + rm + sm - (ru + rj + sj)$ . It is not difficult to see that the maximum value for  $M$  is  $((\mu + 1)(n - k) - 1)(k - 1)$  hence this determinant can not be zero because  $\alpha$  is a primitive element for  $\mathbb{F}_q$  and, by assumption,  $q = p^N$ , where  $N > M$ .

*Case II* The  $3 \times 3$  minors are of the form

$$\begin{vmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & a_5 \\ a_6 & 0 & a_7 \end{vmatrix}.$$

As in the first case, we can assume that the minor is given by

$$\begin{vmatrix} \alpha^{ij} & \alpha^{lu} & 0 \\ \alpha^{(i+r)j} & \alpha^{(l+r)u} & \alpha^{(t+r)m} \\ \alpha^{(i+r+s)j} & 0 & \alpha^{(t+r+s)m} \end{vmatrix},$$

with the same bounds on the variables as before. But, in this case  $j \neq u, m$  but  $u$  can be equal to  $m$ . Indeed, the first column intersects the other two in two places, which means that they are not all shifts of the same column. However, the second and third ones can belong to the same column. This determinant is 0 when  $\alpha^{ru+sm} + \alpha^{rj+s} - \alpha^{rm+sm} = 0$ . In this case, according to the different possibilities for  $j, u, m$  and  $r, s$  we check the maximum and the

minimum exponent. We present here only the worst case for the field size, which is obtained when  $j < u < m, r < 0$ . We see that the minimum exponent is  $rj + sj$  and the maximum is  $rj + sm$ . We consider  $M := rj + sm - rj - sj$  and we check what is the maximum value that  $M$  can reach. It is not difficult to see that this is  $(\mu + 1)(n - k)(k - 1)$ . When  $p = p^N$ , with  $N > M$ , the considered determinant is never 0.

*Case III* The  $3 \times 3$  minors are of the form

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 0 \end{vmatrix},$$

with  $a_i \neq 0$  for any  $i$ . We can assume that, the minor is given by

$$\begin{vmatrix} \alpha^{ij} & \alpha^{lu} & \alpha^{tm} \\ \alpha^{(i+r)j} & \alpha^{(l+r)u} & \alpha^{(t+r)m} \\ \alpha^{(i+r+s)j} & \alpha^{(l+r+s)u} & 0 \end{vmatrix},$$

with the same bounds on the variables as in previous cases. However, this time  $1 \leq j < u < m \leq k$ . After some straightforward computations, we get that this determinant is 0 if and only if

$$\alpha^{rm+rj+sj} + \alpha^{rj+ru+su} = \alpha^{ru+rj+sj} + \alpha^{ru+rm+su}. \tag{6}$$

In the worst case, consider  $M := ru + rj + su - (rm + rj + sj) = r(u - m) + s(u - j)$  with  $r < 0$ . We immediately see that the maximum value that  $M$  can reach is  $(\mu + 1)(n - k)(k - 2) + 1$ , hence this determinant can not be zero because  $\alpha$  is a primitive element for  $\mathbb{F}_q$  and, by assumption,  $q = p^N$ , where  $N > M$ .

*Case IV* The  $3 \times 3$  minors are of the form

$$\begin{vmatrix} a_1 & a_2 & 0 \\ 0 & a_3 & a_4 \\ a_6 & 0 & a_5 \end{vmatrix}.$$

In this case, we can have that the three considered columns come from different shifts of the same one, hence we allow that some (or all) among  $j, u, m$  are equal. Arguing as before, we notice that these minors are given by

$$\begin{vmatrix} \alpha^{ij} & \alpha^{lu} & 0 \\ 0 & \alpha^{(l+r)u} & \alpha^{(t+r)m} \\ \alpha^{(i+r+s)j} & 0 & \alpha^{(t+r+s)m} \end{vmatrix} = \alpha^{ij+lu+tm+rm} (\alpha^{ru+sm} + \alpha^{rj+sj}).$$

This determinant is 0 whenever  $r(u - j) + s(m - j) - (q - 1)/2 = 0 \pmod{q - 1}$ . Analyzing all the possibilities we can have according to  $r, s$  being negative or positive and  $j, u, m$  being equal or different, after some computations, we obtain that, whenever  $q > 2(k - 1)((\mu + 1)(n - k) - 1) + 1$ , the considered determinant is never 0. And this is the case for our field size assumption.  $\square$

Observe that Case IV of Theorem 4 corresponds to the lower bound for the field size sufficient to avoid the presence of 6-cycles not satisfying the FRC. Hence, we have the following result.

**Corollary 2** *Let  $\mathcal{C}^T$  be an  $(n, k)$  convolutional code constructed from a  $(k, w)$  wDTS  $\mathcal{T}$  and satisfying the conditions of Theorems 3 and 4. Then,  $d_{free}(\mathcal{C}^T) \geq 3$  and the code is free from 4 and 6-cycles not satisfying the FRC.*

**Remark 5** If  $C^T$  is an  $(n, k)$  convolutional code constructed from a  $(k, w)$  wDTS  $\mathcal{T}$  and satisfying the conditions of Theorems 3 and 4, such that  $H_\mu$  has no zero row and  $n - k \leq \min\{3, k\}$ , then, it follows from Proposition 1 that  $\delta = \mu(n - k)$ .

**Example 5** Consider the  $(3, 2)_q$  code constructed in Example 4. Note that  $\mu = 5$ , hence, for  $q > 11$  we can avoid all the 6-cycles not satisfying the FRC (Case IV of Theorem 4).

### 4 Excluding $2\ell$ -cycles not satisfying the FRC

In this section, we give some conditions that ensure that the Tanner graph associated to the sliding parity-check matrix of a convolutional code constructed via a difference triangle set is free of  $2\ell$ -cycles not satisfying the FRC.

First of all we recall from Sect. 2.3 that a  $2\ell$ -cycle can be represented by an  $\ell \times \ell$  submatrix of  $\mathcal{H}$  that up to column and row permutations is of the form

$$A = \begin{bmatrix} a_1 & a_2 & 0 & \cdots & \cdots & 0 \\ 0 & a_3 & a_4 & \cdots & \cdots & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ 0 & & & & a_{2\ell-3} & a_{2\ell-2} \\ a_{2\ell} & 0 & \cdots & \cdots & 0 & a_{2\ell-1} \end{bmatrix}, \tag{7}$$

where  $a_i \in \mathbb{F}_q^*$ .

**Remark 6** Observe that

$$\begin{bmatrix} A_0 \\ \vdots \\ A_\mu \end{bmatrix} \in \mathbb{F}^{(\mu+1)(n-k) \times (\mu+1)k},$$

hence it is clear that the Tanner graph associated to  $H$  can only contain  $2\ell$ -cycles for

$$\ell \leq \min\{(\mu + 1)(n - k), (\mu + 1)k\}.$$

At first, we will investigate conditions on the wDTS used to construct the convolutional code that ensure that the associated Tanner graph contains no cycles at all independently of the nonzero values of the sliding parity-check matrix and hence also independently of the underlying finite field.

**Proposition 3** *If  $C$  is an  $(n, k)$  convolutional code whose parity-check matrix has support  $\mathcal{T}$  where  $\mathcal{T}$  is a  $(k, w)$ -wDTS with the property that none of the differences  $a_{i, j} - a_{i, m}$  for  $1 \leq i \leq k$  and  $1 \leq m < j \leq w$  is divisible by  $n - k$ , then each pair of columns that is next to each other in  $A$  as in (7) consists of shifts of different columns of  $\bar{H}$ . In particular, at most  $\lfloor \frac{\ell}{2} \rfloor$  columns of  $A$  can be shifts of the same column of  $\bar{H}$ .*

**Proof** The fact that none of the differences in the set is divisible by  $n - k$  implies that the support of any column of  $\bar{H}$  does not intersect the support of any of its shifts (by multiples of  $n - k$ ). Since the supports of neighbouring columns of  $A$  intersect, they have to be shifts of different columns of  $\bar{H}$ . □

**Corollary 3** *If  $C$  is an  $(n, k)$  convolutional code whose parity-check matrix has support  $\mathcal{T}$  where  $\mathcal{T}$  is a  $(k, w)$ -wDTS with the property that  $T_1 = \dots = T_k$  and none of the differences  $a_{1,j} - a_{1,m}$  for  $1 \leq m < j \leq w$  is divisible by  $n - k$ , then the Tanner graph associated to the parity-check matrix  $H$  of  $C$  is free from  $2\ell$  cycles (over every base field), with  $\ell > 1$  not satisfying the FRC.*

**Theorem 5** *Assume that  $C$  is an  $(n, k)$  convolutional code constructed from an  $(k, w)$ -DTS  $\mathcal{T}$  with  $a_{i,1} = 1$  for all  $1 \leq i \leq k$ , where  $(n - k)$  does not divide any of the nonzero differences  $a_{i_1,j} - a_{i_2,m}$  for  $1 \leq i_1, i_2 \leq k$  and  $1 \leq m, j \leq w$ . Then, the Tanner graph associated to the parity-check matrix  $H$  of  $C$  is free from cycles of any size (over every base field) not satisfying the FRC.*

**Proof** Assume by contradiction that  $\mathcal{H}$  contains up to permutations a submatrix  $A$  of the form (7). As the supports of the first two columns of  $A$  intersect, they have to be shifts of different columns of  $\bar{H}$ . The supports of such shifts can only intersect once and the entries of this intersection come from the first row of  $\bar{H}$ . Applying the same reasoning to the intersection of the supports of the second and third column of  $A$ , implies that  $a_2$  and  $a_3$  in  $A$  both come from the first row of  $\bar{H}$  which is not possible. This shows the result.  $\square$

**Example 6** Consider the  $(2, 3)$ -DTS  $\mathcal{T} = \{T_1, T_2\}$  with  $T_1 = \{1, 2, 5\}$  and  $T_2 = \{1, 3, 9\}$ . The set of all occurring nonzero differences  $a_{i_1,j} - a_{i_2,m}$  is  $\{1, 2, 3, 4, 6, 7, 8\}$ , i.e. none of them is divisible by 5. Hence the matrix  $H(z) = H_0 + H_1z$  with  $H_0 = [\bar{H}_0 \ I_5]$  and  $H_1 = [\bar{H}_1 \ 0_5]$ , where

$$\bar{H}_0 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad \bar{H}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and

$$[H]_{hr} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

full rank, i.e.  $\delta = 1$ , is the parity-check matrix of an  $(7, 2, 1)_q$  convolutional code that is free of cycles of any size for any prime power  $q$ .

Next, we want to relax the conditions on the wDTS used for construction of the convolutional code but still exclude cycles in the Tanner graph of the sliding parity-check matrix that do not fulfill the FRC by using the construction from Definition 3 and considering sufficiently large field sizes.

To ensure that the considered cycle does not satisfy the FRC, we have to guarantee that  $\det A \neq 0$  as an element of  $\mathbb{F}_q$ . It is easy to check that

$$\det A = \prod_{\substack{i=1 \\ i \text{ odd}}}^{2\ell} a_i \pm \prod_{\substack{i=1 \\ i \text{ even}}}^{2\ell} a_i.$$

Let  $\mathcal{T}$  be a  $(k, w)$ -wDTS and let  $C^{\mathcal{T}}$  be the convolutional code defined from  $\mathcal{T}$ , with  $\mathcal{H}^{\mathcal{T}}$  as defined in (3). Each matrix representation  $A$  of a  $2\ell$ -cycle comes from selecting  $\ell$  rows

and  $\ell$  columns of  $\mathcal{H}^T$ . Moreover, in each column of  $A$ , exactly two positions are non-zero. Let  $\alpha$  be a primitive element for  $\mathbb{F}_q$ , let  $s_1, \dots, s_\ell \in \mathbb{N}$  be the indexes of the columns of  $\mathcal{H}^T$ , selected to form the cycle, (we consider  $s_i$  also if we select the shift of the  $i$ th column) hence we have that  $1 \leq s_h \leq k$ .

We can write  $A$  in the following form:

$$\begin{bmatrix} \alpha^{r_1 s_1} & \alpha^{r_2 s_2} & 0 & 0 & \dots & 0 \\ 0 & \alpha^{(r_2+i_1)s_2} & \alpha^{(r_3+i_1)s_3} & 0 & \dots & 0 \\ 0 & 0 & \alpha^{(r_3+i_1+i_2)s_3} & \alpha^{(r_4+i_1+i_2)s_4} & \dots & 0 \\ \vdots & & & \ddots & \ddots & \\ \alpha^{(r_1+i_1+\dots+i_{\ell-1})s_1} & 0 & 0 & 0 & \dots & \alpha^{(r_\ell+i_1+\dots+i_{\ell-1})s_\ell} \end{bmatrix},$$

where  $i_h \in \mathbb{Z}$  and  $|i_h|$  is equal to a difference from  $T_{s_{h+1}}$  for  $h = 1, \dots, \ell-1$  and  $|i_1+\dots+i_{\ell-1}|$  is equal to a difference from  $T_{s_1}$ . Moreover,  $1 \leq r_h + i_1 + \dots + i_g \leq (\mu + 1)(n - k)$  for  $h = 1, \dots, \ell$  and  $g = 0, \dots, \ell - 1$ .

We want to estimate the sufficient field size to have that this determinant is nonzero and therefore, we distinguish two cases.

**Case I:** Assume that  $\ell$  is odd. In this case, the determinant of a matrix of the form (7) is given by

$$\det A = \prod_{\substack{i=1 \\ i \text{ odd}}}^{2\ell} a_i + \prod_{\substack{i=1 \\ i \text{ even}}}^{2\ell} a_i.$$

Hence, if the characteristic of the field is  $p > 2$ , it is equal to 0 in  $\mathbb{F}_q$  if and only if

$$\alpha^{(i_1+i_2+\dots+i_{\ell-1})s_1} + \alpha^{i_1 s_2+i_2 s_3+\dots+i_{\ell-1} s_\ell} = 0,$$

which is equivalent to

$$(i_1 + i_2 + \dots + i_{\ell-1})s_1 = i_1 s_2 + i_2 s_3 + \dots + i_{\ell-1} s_\ell + \frac{(q-1)}{2} \pmod{q-1},$$

and hence

$$i_1(s_2 - s_1) + i_2(s_3 - s_1) + \dots + i_{\ell-1}(s_\ell - s_1) - \frac{(q-1)}{2} = 0 \pmod{q-1}.$$

It is then enough to consider  $q$  bigger than the maximum value that can be reached by the function

$$1 + 2 \sum_{h=1}^{\ell-1} i_h (s_{h+1} - s_1).$$

Now, note that  $i_h$  can be also negative but in general, we can say that  $|i_h| \leq (\mu + 1)(n - k) - 1$ . Moreover,  $|s_i - s_1| \leq k - 1$ . Hence, if we can ensure that

$$\begin{aligned} q &> 2((\mu + 1)(n - k) - 1)(\ell - 1)(k - 1) + 1 \\ &= 2(\mu + 1)(n - k)(\ell - 1)(k - 1) - 2(\ell - 1)(k - 1) + 1, \end{aligned}$$

with this construction we have a convolutional code whose sliding parity-check matrix is associated to a Tanner graph free from  $2\ell$ -cycles, with  $\ell$  odd, not satisfying the FRC.

**Remark 7** Observe that in Theorem 4, we computed a more accurate estimation of the field size for getting rid of the  $2\ell$  cycles, for  $\ell = 3$ , namely,  $q > 2(\mu + 1)(n - k)(k - 1) - 2(k - 1) + 1$ .

The computation above shows that with  $q > 4(\mu + 1)(n - k)(k - 1) - 4(k - 1) + 1$  we do not have 6-cycles not satisfying the FRC. This difference is due to the possibility of a better estimation of the terms in the above inequality.

With the discussion above we have proved the following result.

**Theorem 6** *Let  $n, k, w$  be positive integers with  $n > k$ ,  $\mathcal{T}$  be a  $(k, w)$ -wDTS and  $\mathcal{C}^{\mathcal{T}}$  be the  $(n, k)_q$  convolutional code constructed from  $\mathcal{T}$  with  $q = p^N$  and  $p > 2$ . A sufficient condition for obtaining a code whose sliding parity-check matrix is free from  $2\ell$ -cycles not satisfying the FRC with  $\ell$  odd is to choose a field size  $q > 2(\mu + 1)(n - k)(\ell - 1)(k - 1) - 2(k - 1)(\ell - 1) + 1$ , where  $\mu = \left\lceil \frac{m(\mathcal{T})}{n - k} \right\rceil - 1$  is the degree of the parity-check matrix of  $\mathcal{C}^{\mathcal{T}}$ .*

**Example 7** Consider again the code constructed in Example 4. From Remark 6, we know that the highest length that we can have for a cycle is  $10 = 2 \cdot 5$ , but for  $q$  odd with  $q > 41$  all the 10-cycles satisfy the FRC.

**Case II:** Assume that  $\ell$  is even. In this case, the determinant of a matrix of the form (7) is given by

$$\det A = \prod_{\substack{i=1 \\ i \text{ odd}}}^{2\ell} a_i - \prod_{\substack{i=1 \\ i \text{ even}}}^{2\ell} a_i.$$

After some straightforward computation, it is easy to see that this determinant is equal to 0 in  $\mathbb{F}_q$  if and only if

$$\alpha^{(i_1+i_2+\dots+i_{\ell-1})s_1} = \alpha^{i_1s_2+i_2s_3+\dots+i_{\ell-1}s_{\ell}},$$

which is equivalent to

$$(i_1 + i_2 + \dots + i_{\ell-1})s_1 = i_1s_2 + i_2s_3 + \dots + i_{\ell-1}s_{\ell} \pmod{q - 1},$$

and hence

$$f(i, s) := i_1(s_2 - s_1) + i_2(s_3 - s_1) + \dots + i_{\ell-1}(s_{\ell} - s_1) = 0 \pmod{q - 1}.$$

for  $i := (i_1, \dots, i_{\ell-1})$  and  $s := (s_1, \dots, s_{\ell})$ .

Moreover, we have the following constraints:

1.  $-(\mu + 1)(n - k) + 1 \leq i_h \leq (\mu + 1)(n - k) - 1$  for  $h = 1, \dots, \ell - 1$
2.  $-k + 1 \leq s_{h+1} - s_1 \leq k - 1$ , for  $h = 1, \dots, \ell - 1$ ;

We have to find conditions on the corresponding wDTS to ensure that  $f(i, s)$  is nonzero when viewed as an element of  $\mathbb{Z}$  and then, we can determine a lower bound for  $q$  in order that it is also nonzero modulo  $q - 1$ .

Using Proposition 3, we know that if none of the differences in the difference triangle set is divisible by  $n - k$ , then not all the values  $s_1, \dots, s_{\ell}$  can be identical. In particular, there is at least one  $h \in \{2, \dots, \ell\}$  such that  $s_h - s_1 \neq 0$ .

**Theorem 7** *Let  $\ell$  be an even integer,  $k, n, w$  be integers such that  $n > k$ ,  $\mathcal{T}$  be a  $(k, w)$ -wDTS and  $\mathcal{C}^{\mathcal{T}}$  be the  $(n, k)_q$  convolutional code constructed from  $\mathcal{T}$ . Assume that  $\mathcal{T}$  fulfills the conditions of Proposition 3 and has the property that  $f(i, s)$  is nonzero in  $\mathbb{Z}$  for all  $s_1, \dots, s_{\ell} \in \{1, \dots, k\}$  not all equal if  $|i_h|$  is equal to a difference from  $T_{s_{h+1}}$  for  $h = 1, \dots, \ell - 1$  and  $|i_1 + \dots + i_{\ell-1}|$  is equal to a difference from  $T_{s_1}$  and  $q > ((\mu + 1)(n - k) - 1) \left( (k - 1) \frac{\ell}{2} + (k - 2) \frac{\ell - 2}{2} \right) + 1$ . Then, the Tanner graph associated to the sliding parity-check matrix of  $\mathcal{C}^{\mathcal{T}}$  is free from  $2\ell$ -cycles that do not satisfy the FRC.*



**Proof** The conditions of the theorem ensure that  $f(i, s)$  is nonzero in  $\mathbb{Z}$ . Moreover, it follows from Proposition 3 that

$$((\mu + 1)(n - k) - 1) \left( (k - 1) \frac{\ell}{2} + (k - 2) \frac{\ell - 2}{2} \right)$$

is an upper bound for  $|f(i, s)|$ . Hence, the result follows. □

Next, we want to give an example for a convolutional code that fulfills the conditions of the preceding theorem.

**Example 8** Let  $n = 7$  and  $k = 2$  and  $T_1 = \{1, 2, 5, 9\}$  and  $T_2 = \{1, 2, 4, 10\}$ , i.e.  $\mu = 1$ . Note that  $T_1$  is no difference triangle in the strict sense as  $9 - 5 = 5 - 1$  but as  $n - k = 5$  does not divide  $9 - 5$ , we can still use it for the construction of our code (see Remark 4). We get

$$\mathcal{H}^T = \begin{bmatrix} \alpha & \alpha^2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha^2 & \alpha^4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha^8 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha^5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha & \alpha^2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & \alpha^4 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \alpha^9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^8 & 0 & 0 & 0 & 1 \\ 0 & \alpha^{20} & 0 & 0 & 0 & 0 & 0 & \alpha^5 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

From Remark 6 one knows that with these parameters it is not possible to have cycles of length  $2\ell$  for  $\ell > 4$ . Moreover, from Theorem 3, we obtain that we can exclude 4-cycles not fulfilling the FRC if  $q > 11$  and from Theorem 4, that we can exclude 6-cycles not fulfilling the FRC if  $q > 19$  and  $q$  is odd. We will show that with the help of the preceding theorem, we can also exclude 8-cycles in  $\mathcal{H}^T$  that do not fulfill the FRC or in other words, all  $2\ell$ -cycles for any  $\ell$  in  $\mathcal{H}^T$  fulfill the FRC for  $q > 19$ . First, from Proposition 3, we know that in the matrix  $A$  representing any 8-cycle we necessarily have  $s_1 = s_3$  and  $s_2 = s_4$  and each column of  $\bar{H}^T$  is involved once unshifted and once shifted by 5. We get  $f(i, s) = \pm(i_1 + i_3)$  and have to exclude that  $i_1 \neq -i_3$ . Considering  $\mathcal{H}^T$ , we realize that 8-cycles are only possible for  $s_1 = s_3 = 1$  and  $s_2 = s_4 = 2$  and  $i_1 \in \{\pm 8, \pm 9\}$  and  $i_3 \in \{\pm 2, \pm 3\}$ . Hence, for  $q > 9 \cdot 2 + 1 = 19$ , the corresponding convolutional code is free from 8-cycles not fulfilling the FRC and hence, free from  $2\ell$ -cycles not fulfilling the FRC for any  $\ell$ .

To conclude this section, we will modify our construction from Definition 3 in order to further relax the conditions on the underlying wDTS and still ensuring that we have no cycles not fulfilling the FRC. However, this will come with the cost of a larger field size.

**Definition 4** Let  $k, n$  be positive integers with  $n > k$  and  $\mathcal{T} := \{T_1, \dots, T_k\}$  an  $(k, w)$ -wDTS with scope  $m(\mathcal{T})$ . Set  $\mu = \left\lceil \frac{m(\mathcal{T})}{n-k} \right\rceil - 1$  and let  $\alpha$  be a primitive element for  $\mathbb{F}_q$ . Moreover, let  $P$  be a prime (with properties that will be determined later). For any  $1 \leq i \leq (\mu + 1)(n - k)$ ,  $1 \leq l \leq k$ , define

$$\bar{H}_{i,l}^{(\mathcal{T})} := \begin{cases} \alpha^{Pi} & \text{if } i \in T_l \\ 0 & \text{otherwise} \end{cases}.$$

**Theorem 8** *Let  $k, n, w$  be positive integers with  $n > k$  and  $\mathcal{T}$  be a DTS with  $a_{i,1} = 1$  for all  $1 \leq i \leq k$  and  $\mathcal{C}$  be an  $(n, k)_q$  convolutional code constructed from  $\tilde{H}^{(\mathcal{T})}$ . If  $P > \ell k$  and  $q > k P^{(\mu+1)(n-k)} \frac{P^{2\ell}-1}{P^{2\ell}-P^{2\ell-1}} + 1$ , then the Tanner graph associated to the sliding parity-check matrix contains no cycles of size  $2\ell$  not fulfilling the FRC.*

**Proof** As with the construction from Definition 3, we obtain that  $\det(A) = 0$  if and only if a certain linear combination  $\tilde{f}(i, s)$  of exponents of  $P$  with  $2\ell$  coefficients from  $\{1, \dots, k\}$  is zero. As the exponents correspond to row indices before a possible shift and the unshifted columns only intersect in the first row, all exponents that are equal to any other exponent are equal to 1. Moreover, as exponents from the same column of  $A$  cannot be the same, at most  $\ell$  exponents can be equal to 1. In summary, we obtain that  $\tilde{f}(i, s)$  is of the form  $\tilde{f}(i, s) = Px + P^{e_1}x_1 + \dots + P^{e_t}x_t$  with natural numbers  $1 < e_1 < \dots < e_t \leq (\mu+1)(n-k)$ ,  $t \in \{\ell, \dots, 2\ell\}$ ,  $x_j \in \{-k, \dots, k\} \setminus \{0\}$  for  $j = 1, \dots, t$  and  $x \in \{-\ell k, \dots, +\ell k\}$ . Since  $m$  was chosen to be a prime larger than  $\ell k$ ,  $\tilde{f}(i, s)$  is nonzero in  $\mathbb{Z}$ . Furthermore,  $|\tilde{f}(i, s)| \leq k \sum_{i=0}^{2\ell-1} P^{(\mu+1)(n-k)-i} = k P^{(\mu+1)(n-k)} \frac{P^{2\ell}-1}{P^{2\ell}-P^{2\ell-1}}$  and hence it cannot be zero modulo  $q - 1$ .  $\square$

Finally, we illustrate our modified construction with an example.

**Example 9** If we take the DTS  $\mathcal{T} = \{\{1, 2, 5\}, \{1, 3, 8\}\}$  to construct an  $(6, 2)_q$  convolutional code, we have  $m(\mathcal{T}) = 8$  and  $\mu = 1$ . If we want that the girth of the corresponding parity-check matrix is at least 12, we have to choose  $P > 10$ , i.e.  $P = 11$ . To get the desired property it would be sufficient if the field size is larger than  $4.716 \times 10^8$ . If it is sufficient to have a girth of at least 8, it would be enough to choose  $P = 7$  and the sufficient field size decreases to  $1.35 \times 10^7$ .

### 5 Conclusion

In the past years, several authors tried to algebraically construct LDPC convolutional codes with good distance properties. In the present paper, we provided a combinatorial construction of LDPC convolutional codes of arbitrary rate and over arbitrary fields, with the aid of (weak) difference triangle sets. This construction produces codes with good distance properties. Finally, we provided a lower bound on the field size that is sufficient to have codes with good distance and to avoid the presence of cycles in the associated Tanner graph not satisfying the full rank condition, which is desirable for minimizing the decoding failure of this type of codes.

**Acknowledgements** The authors acknowledge the support of Swiss National Science Foundation Grant No. 188430. Julia Lieb acknowledges also the support of the German Research Foundation Grant LI 3101/1-1.

**Funding** Open Access funding provided by University Zurich.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Alfarano G.N., Lieb J., Rosenthal J.: Construction of rate  $(n - 1)/n$  non-binary LDPC convolutional codes via difference triangle sets. In: 2020 IEEE International Symposium on Information Theory (ISIT), 2020, pp. 138–143, <https://doi.org/10.1109/ISIT44484.2020.9174510>.
2. Amiri B., Kliever J., Dolecek L.: Analysis and enumeration of absorbing sets for non-binary graph-based codes. *IEEE Trans. Commun.* **62**(2), 398–409 (2014).
3. Bates S., Chen Z., Dong X.: Low-density parity-check convolutional codes for ethernet networks. In: PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and signal Processing. 2005, pp. 85–88, <https://doi.org/10.1109/PACRIM.2005.1517231>.
4. Bates S., Elliott D.G., Swamy R.: Termination sequence generation circuits for low-density parity-check convolutional codes. *IEEE Trans. Circuits Syst. I* **53**(9), 1909–1917 (2006).
5. Battaglioni M., Baldi M., Chiaraluca F., Lentmaier M.: Girth properties of time-varying SC-LDPC convolutional codes. In: 2019 IEEE International Symposium on Information Theory (ISIT), pp. 2599–2603, <https://doi.org/10.1109/ISIT.2019.8849323>.
6. Berrou C., Glavieux A., Thitimajshima P.: Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In: Proceedings of ICC'93-IEEE International Conference on Communications, vol. 2, pp. 1064–1070, <https://doi.org/10.1109/ICC.1993.397441>.
7. Chee Y.M., Colbourn C.J.: Constructions for difference triangle sets. *IEEE Trans. Inf. Theory* **43**(4), 1346–1349 (1997).
8. Chen Z., Fan P., Jin F.: Disjoint difference sets, difference triangle sets, and related codes. *IEEE Trans. Inf. Theory* **38**(2), 518–522 (1992).
9. Chung S.-Y., Richardson T.J., Urbanke R.L.: Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation. *IEEE Trans. Inf. Theory* **47**(2), 657–670 (2001).
10. Colbourn C.J.: Difference triangle sets. In: Colbourn C.J., Dintz J (eds.) *The CRC Handbook of Combinatorial Designs*, pp. 312–317. CRC Press, Boca Raton (1996).
11. Davey M.C., MacKay D.J.: Low density parity check codes over  $GF(q)$ . In: *IEEE Communications Letters*, vol. 2, no. 6, pp. 165–167, June 1998, <https://doi.org/10.1109/4234.681360>.
12. Davey M.C., MacKay D.J.: Monte Carlo simulations of infinite low density parity check codes over  $GF(q)$ . In: *Proceedings of International Workshop on Optimal Codes and Related Topics*, pp. 9–15. Citeseer (1998).
13. Forney G.: Convolutional codes I: algebraic structure. *IEEE Trans. Inf. Theory* **16**(6), 720–738 (1970).
14. Gallager R.: Low-density parity-check codes. *IRE Trans. Inf. Theory* **8**(1), 21–28 (1962).
15. Gluesing-Luerssen H., Rosenthal J., Smarandache R.: Strongly-MDS convolutional codes. *IEEE Trans. Inf. Theory* **52**(2), 584–598 (2006).
16. Johnson S.J., Weller S.R.: Regular low-density parity-check codes from combinatorial designs. In: *Proceedings 2001 IEEE Information Theory Workshop (Cat. No. 01EX494)*, pp. 90–92. IEEE (2001), <https://doi.org/10.1109/ITW.2001.955146>.
17. Kailath T.: *Linear Systems*, vol. 156. Prentice-Hall, Englewood Cliffs (1980).
18. Klove T.: Bounds and construction for difference triangle sets. *IEEE Trans. Inf. Theory* **35**(4), 879–886 (1989).
19. Kou Y., Lin S., Fossorier M.: Construction of low density parity check codes: a geometric approach. In: *Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics*, pp. 137–140 (2000).
20. Kou Y., Lin S., Fossorier M.P.: Low-density parity-check codes based on finite geometries: a rediscovery and new results. *IEEE Trans. Inf. Theory* **47**(7), 2711–2736 (2001).
21. MacKay D.J., Neal R.M.: Near Shannon limit performance of low density parity check codes. *Electron. Lett.* **32**(18), 1645–1646 (1996).
22. Massey J.L.: *Threshold Decoding*. MIT, Cambridge (1963).
23. Oswald P., Shokrollahi A.: Capacity-achieving sequences for the erasure channel. *IEEE Trans. Inf. Theory* **48**(12), 3017–3028 (2002).
24. Poulliat C., Fossorier M., Declercq D.: Design of regular  $(2, d_c)$ -LDPC codes over  $GF(q)$  using their binary images. *IEEE Trans. Commun.* **56**(10), 1626–1635 (2008).
25. Pusane A.E., Smarandache R., Vontobel P.O., Costello D.J.: Deriving good LDPC convolutional codes from LDPC block codes. *IEEE Trans. Inf. Theory* **57**(2), 835–857 (2011).
26. Richardson T.J., Urbanke R.L.: The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inf. Theory* **47**(2), 599–618 (2001).
27. Robinson J.P., Bernstein A.: A class of binary recurrent codes with limited error propagation. *IEEE Trans. Inf. Theory* **13**(1), 106–113 (1967).

28. Rosenthal J.: Connections between linear systems and convolutional codes. In: Codes, Systems, and Graphical Models, pp. 39–66. Springer, Berlin (2001).
29. Rosenthal J., Vontobel P.O.: Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis. In: Proceedings of the 38th Allerton Conference on Communication, Control, and Computing (2000).
30. Tanner R.: A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory* **27**(5), 533–547 (1981).
31. Tanner R.M.: Error-correcting coding system. US Patent 4,295,218 (1981).
32. Tanner R.M.: Convolutional Codes from Quasi-cyclic codes: A Link Between the Theories of Block and Convolutional Codes. University of California, Santa Cruz, Computer Research Laboratory (1987).
33. Tanner R.M., Sridhara D., Sridharan A., Fuja T.E., Costello D.J.: LDPC block and convolutional codes based on circulant matrices. *IEEE Trans. Inf. Theory* **50**(12), 2966–2984 (2004).
34. Tong S.-Y.: Systematic construction of self-orthogonal diffuse codes. *IEEE Trans. Inf. Theory* **16**(5), 594–604 (1970).
35. Tong S.-Y.: Character-correcting convolutional self-orthogonal codes. *Inf. Control* **18**, 183–202 (1971).
36. Vasic B., Milenkovic O.: Combinatorial constructions of structured low-density parity check codes for iterative decoding. *IEEE Trans. Inf. Theory*, vol 50, no. 6, pp. 1156–1176, June 2004, <https://doi.org/10.1109/TIT.2004.828066>.
37. Vasic B., Milenkovic O.: Combinatorial constructions of low-density parity-check codes for iterative decoding. *IEEE Trans. Inf. Theory* **50**(6), 1156–1176 (2004).
38. Wiberg N.: Codes and decoding on general graphs. PhD thesis (1996).
39. Zhou H., Goertz N.: Cycle analysis of time-invariant LDPC convolutional codes. In: 2010 17th International Conference on Telecommunications, pp. 23–28. IEEE (2010), <https://doi.org/10.1109/ICTEL.2010.5478744>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.