Searching for voltage graph-based LDPC tailbiting codes with large girth

Bocharova, Irina; Hug, Florian; Johannesson, Rolf; Kudryashov, Boris; Satyukov, Roman

Published in:
IEEE Transactions on Information Theory

DOI:
10.1109/TIT.2011.2176717

2012

Searching for Voltage Graph-Based LDPC Tailbiting Codes with Large Girth

Irina E. Bocharova, Florian Hug, *Student Member, IEEE*, Rolf Johannesson, *Life Fellow, IEEE*, Boris D. Kudryashov, and Roman V. Satyukov

***Abstract*—The relation between parity-check matrices of quasi-cyclic (QC) low-density parity-check (LDPC) codes and biadjacency matrices of bipartite graphs supports searching for powerful LDPC block codes. Using the principle of tailbiting, compact representations of bipartite graphs based on convolutional codes can be found.**

Bounds on the girth and the minimum distance of LDPC block codes constructed in such a way are discussed. Algorithms for searching iteratively for LDPC block codes with large girth and for determining their minimum distance are presented. Constructions based on all-one matrices, Steiner Triple Systems, and QC block codes are introduced. Finally, new QC regular LDPC block codes with girth up to 24 are given.

***Index Terms*—biadjacency matrix, convolutional code, girth, LDPC code, minimum distance, tailbiting, Tanner graph**

I. Introduction

Low-density parity-check (LDPC) codes, invented by Gallager [1] in the early 1960s, constitute a hot research topic since they are a main competitor to turbo codes [2]–[5]. Recently, a connection between LDPC codes and codes based on graphs was shown (see, for example, [6]–[10]), which opens new perspectives in searching for powerful LDPC codes. Moreover, coding theory methods can be applied in describing and searching for graphs better than previously known. For example, in [11], [12] compact representations based on convolutional LDPC codes for famous bipartite graphs such as Heawood's, Tutte's, and Balaban's graphs [13] are presented.

Typically, LDPC codes have a minimum distance which is less than that of the best known linear codes, but due to their structure they are suitable for low-complexity iterative decoding, like for example the belief propagation algorithm. An important parameter determining the efficiency of iterative decoding algorithms for LDPC codes is the *girth*, which determines the number of independent iterations [1] and is a parameter of the underlying graph. The minimum distance seems not to play an important role within iterative decoding algorithms, since the error-correcting capabilities of such a suboptimal procedure are often less than those guaranteed by the minimum distance. In fact, it was shown in [14] that the performance of LDPC codes in the high signal-to-noise (SNR) region is predominantly dictated by the structure of the smallest absorbing sets. However, as the size of these absorbing sets is upper-bounded by the minimum distance, LDPC codes with large minimum distance are of particular interest.

LDPC codes can be characterized by having either random/pseudo-random or nonrandom structures, where nonrandom codes can be subdivided into regular or irregular [7], [8], [11], [15]–[28], while random/pseudo-random codes are always irregular [29], [30]. A (J, K)-regular (nonrandom) LDPC code is determined by a parity-check matrix with exactly J ones in each column and exactly K ones in each row.

The class of quasi-cyclic (QC) (J, K)-regular LDPC codes is a subclass of regular LDPC codes with low encoding complexity. Such codes are most suitable for algebraic design and are commonly constructed based on combinatorial approaches using either finite geometries [15] or Steiner Triple Systems [16], [17], having girth $g \geq 6$. Amongst other algebraic constructions leading to QC LDPC codes with larger girth we would like to mention [19], where a class of QC LDPC codes of rate $R = 2/5$ with girth up to 12 based on subgroups of the multiplicative group of the finite field $\mathbb{F}_p$ was obtained. The same method was used for convolutional codes in [21].

Although QC LDPC codes are not asymptotically optimal they can outperform random or pseudorandom LDPC codes (from asymptotically optimal ensembles) for short or moderate block lengths [18]. This motivates searching for good short QC LDPC codes.

The problem of finding QC LDPC codes with large girth and large minimum distance for a wide range of code rates was considered in several papers, for example, [7], [8], [18], [21], [23], [24]. Codes with girth at most 12 are constructed in [18], [19], [21], [23], while [8] gives examples of rather short codes with girth 14. Codes with girth up to 18 with $J \geq 3$ are presented in [24] and it is shown that QC LDPC codes with girth ≥ 14 and block length between 34,000 and 92,000 outperform random codes of the same block length and rate.

Most of the papers devoted to constructing nonrandom LDPC codes with large girth combine some algebraic tech-

Manuscript received February 23, 2011. Current version published October 10, 2011. This work was supported in part by the Swedish Research Council by Grant 621-2007-6281.

The material in this paper was presented in part at the IEEE International Symposiums on Information Theory, St. Petersburg, Russia, 2011.

I. E. Bocharova and B. D. Kudryashov are with the Department of Information Systems, St. Petersburg University of Information Technologies, Mechanics and Optics, St. Petersburg, Russia (e-mail: irina@eit.lth.se; boris@eit.lth.se).

F. Hug and R. Johannesson are with the Department of Electrical and Information Technology, Lund University, Lund, Sweden (e-mail: florian@eit.lth.se; rolf@eit.lth.se).

R. V. Satyukov is with Google Switzerland, Zürich, Switzerland (e-mail: satyukov@gmail.com).

Communicated by A. Ashikhmin, Associate Editor for Coding Techniques.

niques and computer search. Commonly these procedures start by choosing a proper base matrix (also called weight or degree matrix) or the corresponding base graph (also called seed graph [20] or protograph [31]). The references [18], [20] are focused on all-one base matrices, while in [7], [24] base matrices are constructed from Steiner Triple Systems and integer lattices. In both cases, a system of inequalities with integer coefficients describing all cycles of a given length is obtained and suitable labels or degrees are derived. For example, if we replace all nonzero entries in the base matrix by permutation matrices [27], [31], circulant matrices [8], [18], [19], [24], [26], or sums of circulant matrices [25], we obtain the corresponding QC LDPC block codes. On the other hand, if we replace all nonzero entries in the base matrix by either monomials or binomials, we obtain the corresponding (parent) LDPC convolutional codes [21], [25], [28].

Notice that both constructing the inequalities and the labelings require significant computational efforts. Some methods directed towards reducing the computational complexity of these steps can be found in [23], [26].

Parameters of the so far shortest QC LDPC block codes with $J = 3$ and girth $6, 8$, and 10 found via computer search are presented in [26], improving previous results from [18].

As mentioned earlier, it is important that the constructed QC LDPC block codes have large minimum distance for achieving a suitable upper bound on their error-correcting performance at high SNR. It is proved in [2] that the minimum distance of QC LDPC codes whose base parity-check matrices are $J \times K$ all-one matrices is upper-bounded by $(J+1)!$. However, considering base matrices with zeros leads to QC LDPC codes with larger minimum distance. For example, in [25] it is shown that replacing all nonzero entries in the base matrix by sums of circulants and all zero entries by all-zero matrices, increases the minimum distance of the resulting code while preserving its regularity. For LDPC convolutional codes this approach implies that a parity-check matrix contains binomials instead of monomials. The corresponding upper bound on the minimum distance of such LDPC codes is presented in [25]. A particular case of this upper bound, valid only for codes with zeros and monomials is derived in [11].

In this paper, we focus on LDPC block codes constructed from tailbitten convolutional codes with monomial parity-check matrices. Notations for generator and parity-check matrices of convolutional codes and their tailbiting block codes are introduced in Section II. Section III focuses on bipartite graphs, biadjacency matrices, and their relation to parity-check matrices of LDPC block codes. Our constructions of base and voltage matrices, used when searching for LDPC block codes with large girth, are introduced in Section IV. Bounds on the girth and the minimum distance for QC (J, K)-regular LDPC block codes are discussed in Section V. New search algorithms for QC LDPC block codes constructed from all-one matrices, Steiner Triple Systems, and QC regular matrices are presented in Section VI. Moreover, depending on the desired girth, algorithms of different complexity for constructing the set of inequalities and searching for suitable labelings are described. A new algorithm for computing the minimum distance of QC (J, K)-regular LDPC codes is described in Section VII and used to compute the minimum distance of our newly found codes. Moreover, we determined the hitherto unknown minimum distance for some of the shortest known LDPC codes given in [24]. In Section VIII, we present new examples of (J, K)-regular QC LDPC codes in the form of tailbiting LDPC codes with girth between 10 and 24. This representation is compact and it is possible to apply low-complexity encoding, searching, and decoding procedures well developed for convolutional and tailbiting block codes [32], [33]. In particular, the presented codes with girth 10 and 12 are shorter than the ones in [26] and [8], [22], respectively. Moreover, our codes with girth 14 to 18 are shorter than those in [24]. Section IX concludes the paper with some final remarks.

II. Generator and Parity-Check Matrices

Consider a rate $R = b/c$ binary convolutional code $\mathcal{C}$ with the semi-infinite generator matrix

$$G = \begin{pmatrix} G_0 & G_1 & \dots & & G_{m_g} & & \\ & G_0 & G_1 & \dots & & G_{m_g} & \\ & & \ddots & \ddots & & & \ddots \end{pmatrix} \quad (1)$$

of memory m_g where G_i, $i = 0, 1, \dots, m_g$, are $b \times c$ binary matrices. Its semi-infinite syndrome former

$$H^{\mathrm{T}} = \begin{pmatrix} H_0^{\mathrm{T}} & H_1^{\mathrm{T}} & \dots & & H_m^{\mathrm{T}} & & \\ & H_0^{\mathrm{T}} & H_1^{\mathrm{T}} & \dots & & H_m^{\mathrm{T}} & \\ & & \ddots & \ddots & & & \ddots \end{pmatrix} \quad (2)$$

of memory m, where in general $m \neq m_g$; H_j, $j = 0, 1, \dots, m$, are $(c-b) \times c$ binary matrices, and T denotes transpose. Clearly G and H satisfy

$$GH^{\mathrm{T}} = \mathbf{0} \quad (3)$$

and

$$\boldsymbol{v}H^{\mathrm{T}} = \mathbf{0} \quad (4)$$

where

$$\boldsymbol{v} = \boldsymbol{u}G \quad (5)$$

is the code sequence and $\boldsymbol{u}$ is the information sequence.

Next we tailbite the semi-infinite generator matrix (1) to length M c-tuples, where $M > \max\{m, m_g\}$. Then we obtain the $Mb \times Mc$ generator matrix of the quasi-cyclic (QC) block code $\mathcal{B}$ as

$$G_{\mathrm{TB}} = \begin{pmatrix} G_0 & G_1 & \dots & & G_{m_g} & & & & \\ & G_0 & G_1 & \dots & & G_{m_g} & & & \\ & & \ddots & \ddots & & & \ddots & & \\ & & & G_0 & G_1 & \dots & & G_{m_g} & \\ G_{m_g} & & & & G_0 & G_1 & \dots & G_{m_g-1} & \\ G_{m_g-1} & G_{m_g} & & & & \ddots & \ddots & \vdots & \\ \vdots & & \ddots & & & & \ddots & G_1 & \\ G_1 & G_2 & \dots & G_{m_g} & & & & G_0 & \end{pmatrix} \quad (6)$$

Every cyclic shift of a codeword of $\mathcal{B}$ by c places modulo Mc is a codeword. The corresponding tailbiting parity-check matrix is the $M(c-b) \times Mc$ matrix

$$H_{\text{TB}} = \begin{pmatrix} H_0 & & & & H_m & H_{m-1} & \dots & H_1 \\ H_1 & H_0 & & & & H_m & & H_2 \\ \vdots & H_1 & \ddots & & & & \ddots & \vdots \\ & \vdots & \ddots & H_0 & & & & H_m \\ H_m & & & H_1 & H_0 & & & \\ & H_m & & \vdots & H_1 & \ddots & & \\ & & \ddots & & \vdots & \ddots & \ddots & \\ & & & H_m & H_{m-1} & \dots & H_1 & H_0 \end{pmatrix} \tag{7}$$

It is easily shown that G_{TB} and H_{TB} satisfy

$$G_{\text{TB}} H_{\text{TB}}^{\text{T}} = \mathbf{0} \tag{8}$$

given that (3) is fulfilled. We use the terminology *tailbiting* block code when we emphasize that the code is constructed from a convolutional code and *quasi-cyclic* when we emphasize this property.

The parity-check matrix for the convolutional code $\mathcal{C}$ can also be written as the $(c-b) \times c$ polynomial matrix

$$H(D) = H_0 + H_1 D + H_2 D^2 + \cdots + H_m D^m \tag{9}$$

or, equivalently, as

$$H(D) = \begin{pmatrix} h_{11}(D) & h_{12}(D) & \dots & h_{1c}(D) \\ h_{21}(D) & h_{22}(D) & \dots & h_{2c}(D) \\ \vdots & \vdots & \ddots & \\ h_{(c-b)1}(D) & h_{(c-b)2}(D) & \dots & h_{(c-b)c}(D) \end{pmatrix} \tag{10}$$

In the sequel we mostly consider parity-check matrices with only monomial entries $h_{ij}(D) = D^{w_{ij}}$ of degree w_{ij}, where w_{ij} are nonnegative integers. Clearly, such a parity-check matrix $H(D)$ can be represented by its *degree matrix* $W = (w_{ij})$, $i = 1, 2, \dots, c-b$ and $j = 1, 2, \dots, c$. Note that starting from Section VIII we will relax the restriction to only monomial entries and also include zero entries.

Example 1: Consider the rate $R = 1/4$ convolutional code $\mathcal{C}$ with parity-check matrix

$$H(D) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & D & D \\ 1 & D & 1 & D \end{pmatrix} \tag{11}$$

whose degree matrix is

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Tailbiting (11) to length $M = 2$, we obtain the tailbitten 6×8 parity-check matrix of a QC LDPC block code

$$H_{\text{TB}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}\right) \end{array} \tag{12}$$

In particular, every cyclic shift of a codeword by $c = 4$ places modulo $Mc = 8$ is a codeword.

Due to the restriction to monomial entries in $H(D)$, H_{TB} is (J, K)-*regular*, that is, it has exactly J and K ones in each column and row, respectively. Moreover, to fulfill the *low* density criterion, M has to be much larger than J and K, and thus the matrix H_{TB} is sparse.

Note that the first c columns of H_{TB} are repeated throughout the whole matrix in a cyclicly shifted manner. By reordering the columns as $1, c+1, 2c+1, \dots, (M-1)c+1,\ 2, c+2, 2c+2, \dots, (M-1)c+2$, *etc.* and the rows as $1, (c-b)+1, 2(c-b)+1, \dots, (M-1)(c-b)+1, 2, (c-b)+2, 2(c-b)+2, \dots, (M-1)(c-b)+2$, *etc.* we obtain a parity-check matrix of an equivalent (J, K)-regular LDPC block code constructed from circulant matrices

$$H_{\text{C}} = \begin{pmatrix} I_{w_{11}} & I_{w_{12}} & \cdots & I_{w_{1c}} \\ I_{w_{21}} & I_{w_{22}} & \cdots & I_{w_{2c}} \\ \cdots & \cdots & \cdots & \cdots \\ I_{w_{(c-b)1}} & I_{w_{(c-b)2}} & \cdots & I_{w_{(c-b)c}} \end{pmatrix} \tag{13}$$

where w_{ij} are the entries of the degree matrix W and $I_{w_{ij}}$ denotes an $M \times M$ *circulant matrix*, that is, an identity matrix with its rows shifted cyclically to the left by w_{ij} positions. Note, that the (J, K)-regular LDPC block code determined by H_{C} is not quasi-cyclic, although equivalent to the QC block code determined by H_{TB}.

Example 1 (Cont'd): We return to (12) in Example 1 and reorder the columns as $1, 5, 2, 6, 3, 7, 4, 8$ and the rows as $1, 4, 2, 5, 3, 6$. Then we obtain the equivalent rate $R = 1 - 6/8$ $(3, 4)$-regular LDPC block code with parity-check matrix

$$H_{\text{C}} = \begin{array}{c} \\ 1 \\ 4 \\ 2 \\ 5 \\ 3 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 5 & 2 & 6 & 3 & 7 & 4 & 8 \end{array} \\ \left(\begin{array}{cc|cc|cc|cc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right) \end{array} \tag{14}$$

III. Graphs and Biadjacency Matrices

A graph $\mathcal{G}$ is determined by a set of *vertices* $\mathcal{V} = \{v_i\}$ and a set of *edges* $\mathcal{E} = \{e_i\}$, where each edge connects exactly two vertices. The *degree of a vertex* denotes the number of edges that are connected to it. If all vertices have the same degree c, the *degree of the graph* is c, or, in other words, the graph is c-*regular*.

Consider the set of vertices $\mathcal{V}$ of a graph partitioned into t disjoint subsets $\mathcal{V}_k$, $k = 0, 1, \dots, t-1$. Such a graph is said to be t-*partite*, if no edge connects two vertices from the same set $\mathcal{V}_k$, $k = 0, 1, \dots, t-1$.

A *path* of length L in a graph is an alternating sequence of $L+1$ vertices v_i, $i = 1, 2, \dots, L+1$, and L edges e_i, $i = 1, 2, \dots, L$, with $e_i \neq e_{i+1}$. If the first and the final vertex coincide, that is, if $v_1 = v_{L+1}$, we obtain a *cycle*. A cycle is called *simple* if all its vertices and edges are distinct, except for the first vertex and final vertex which coincide. The length of the shortest simple cycle is denoted the *girth*

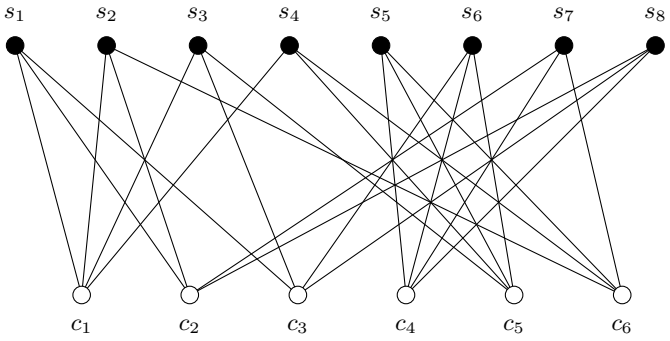

Fig. 1. Tanner graph with 8 symbol nodes (s_i, $i = 1, 2, \ldots, 8$) and 6 constraint nodes (c_i, $i = 1, 2, \ldots, 6$).

of the graph. Consider the block code, whose parity-check matrix is the incidence matrix of a bipartite graph. In [34], which is an extended version of [35], it was proven that the girth of this graph is equal to the minimum distance of such a block code. Moreover, the girth determines the number of independent iterations in belief propagation decoding [1].

Every parity-check matrix H of a rate $R = k/n$ LDPC block code can be interpreted as the *biadjacency matrix* [36] of a bipartite graph, the so-called *Tanner graph* [37], having two disjoint subsets $\mathcal{V}_0$ and $\mathcal{V}_1$ containing n and $n-k$ vertices, respectively. The n vertices in $\mathcal{V}_0$ are called *symbol nodes*, while the $n-k$ vertices in $\mathcal{V}_1$ are called *constraint nodes*. If the underlying LDPC block code is (J, K)-regular, the symbol and constraint nodes have degree J and K, respectively. The incidence matrix of such a Tanner graph coincides with the parity-check matrix of a $(Jn, (J-2)n+k)$ linear block code.

Consider the Tanner graph with the biadjacency matrix H_{TB}, corresponding to a QC (J, K)-regular LDPC code, obtained from the parity-check matrix of a tailbiting LDPC block code. Clearly, by letting the tailbiting length M tend to infinity, we obtain a convolutional parity-check matrix $H(D)$ (10) of the parent convolutional code $\mathcal{C}$. In terms of Tanner graph representations, this procedure corresponds to unwrapping the underlying graph and extending it in the time domain towards infinity. Hereinafter, we will call the girth of this infinite Tanner graph the *free girth* and denote it g_{free}.

Example 1 (Cont'd): Interpreting (14) as a biadjacency matrix, we obtain the corresponding Tanner graph $\mathcal{G}$ as illustrated in Fig. 1 with 8 symbol nodes and 6 constraint nodes, having girth $g = 4$. Its incidence matrix has size 14×24.

IV. Base Matrices, Voltages, and their Graphs

A binary matrix B is called *base matrix* for a tailbiting LDPC block code if its parent convolutional code with parity-check matrix $H(D)$ has only monomial or zero entries and satisfies

$$B = H(D)\big|_{D=1} \tag{15}$$

that is, all nonzero entries in $H(D)$ are replaced by $D^0 = 1$. Different tailbiting LDPC block codes can have the same base matrix B.

The *base graph* $\mathcal{G}_{\text{B}}$ follows as the bipartite graph, whose biadjacency matrix is given by the base matrix B. Denote the girth of such a base graph g_{B}. The terminology "base graph" originates from graph theory and is used, for example, in [38]. It differs from the terminology used in [8], [31], where *protograph* or *seed graph* are used.

Consider the additive group $(\Gamma, +)$, where $\Gamma = \{\gamma\}$. From the base graph $\mathcal{G}_{\text{B}} = \{\mathcal{E}_{\text{B}}, \mathcal{V}_{\text{B}}\}$ we obtain the *voltage graph* [39], [40] $\mathcal{G}_{\text{V}} = \{\mathcal{E}_{\text{B}}, \mathcal{V}_{\text{B}}, \Gamma\}$ by assigning a voltage value $\gamma(e, v, v')$ to the edge e connecting the vertices v and v', satisfying the property $\gamma(e, v, v') = -\gamma(e, v', v)$. Although the graph is not directed, the voltage of the edge depends on the direction in which the edge is passed. Finally, define the *voltage of the path* to be the sum of the voltages of its edges.

Let $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$ be a *lifted graph* obtained from a voltage graph $\mathcal{G}_{\text{V}}$, where $\mathcal{E} = \mathcal{E}_{\text{B}} \times \Gamma$ and $\mathcal{V} = \mathcal{V}_{\text{B}} \times \Gamma$. Two vertices (v, γ) and (v', γ') are connected in the lifted graph by an edge if and only if v and v' are connected in the voltage graph $\mathcal{G}_{\text{V}}$ with the voltage value of the corresponding edge given by $\gamma(e, v, v') = \gamma - \gamma'$. It is easy to see that cycles in the lifted graph correspond to cycles in the voltage graph with zero voltage. Consequently, the girth g_{V} of a voltage graph follows as the length of its shortest cycle with voltage zero, which is equal to the free girth g_{free} [8], [18], [23]. A voltage assignment corresponds directly to selecting the degrees of the parity-check monomials in $H(D)$.

In the following we start from a base graph $\mathcal{G}_{\text{B}}$ and use a voltage assignment based on the monomial degrees w_{ij} of the degree matrix W to determine the corresponding voltage graph $\mathcal{G}_{\text{V}}$. The edge voltage from the constraint node c_i to the symbol node s_j is denoted by μ_{ij} and from the symbol node s_j to the constraint node c_i by $\bar{\mu}_{ji}$, $i = 1, 2, \ldots, (c-b)$ and $j = 1, 2, \ldots, c$, where

$$\begin{cases} \mu_{ij} = w_{ij} \\ \bar{\mu}_{ji} = -w_{ij} \end{cases} \tag{16}$$

When searching for LDPC convolutional codes with given free girth g_{free}, we use integer edge voltages, that is, we deal with an infinite additive group. However, when searching for QC LDPC block codes with given girth g, obtained by tailbiting a parent convolutional code to length M, we use a group of modulo M residues, that is, (16) is replaced by

$$\begin{cases} \mu_{ij} = w_{ij} & \mod M \\ \bar{\mu}_{ji} = -w_{ij} & \mod M \end{cases} \tag{17}$$

The definitions of path and cycle in a voltage graph coincide with those in a regular graph, except for the additional restriction that two neighboring edges may not connect the same nodes in reversed order. The *voltage* of a path or cycle within a voltage graph, follows as the sum of all edge voltages involved.

Example 1 (Cont'd): The bipartite graph whose biadjacency matrix is given by the base matrix B of the rate $R = 1/4$ $(3, 4)$-regular LDPC convolutional code $\mathcal{C}$ is illustrated in Fig. 2. As the edges are labeled according to (16), Fig. 2 corresponds to a voltage graph with girth $g_{\text{V}} = 4$ (for example,

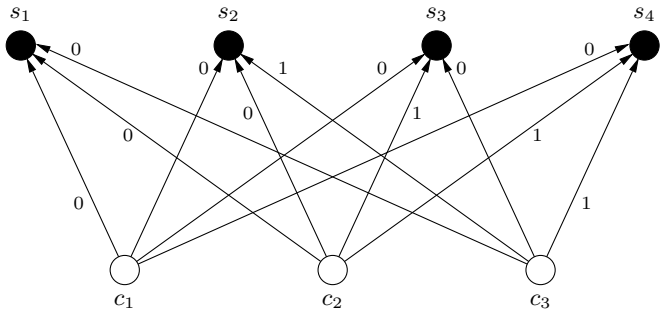

Fig. 2. Bipartite graph with 4 symbol nodes (s_i, $i = 1, 2, 3, 4$) and 3 constraint nodes (c_i, $i = 1, 2, 3$). Since the edges are labeled according to (16), this corresponds to a voltage graph.

$s_1 \to c_1 \to s_2 \to c_2 \to s_1$). The edge from, for example, constraint node c_2 to symbol node s_3 is labeled according to

$$\mu_{23} = -\bar{\mu}_{32} = w_{23} = 1$$

and, hence, the parity-check polynomial $h_{23}(D) = D^{w_{23}} = D$. The free girth of the infinite Tanner graph, corresponding to the parent convolutional code $\mathcal{C}$, is determined by the convolutional parity-check matrix $H(D)$ and is equal to the girth of the voltage graph; hence we can conclude that $g_{\text{free}} = g_{\text{V}} = 4$.

If we neglect all edge labels, we would obtain the corresponding base graph.

V. Bounds on the girth and the minimum distance of $(J \geq 3, K)$ QC LDPC block codes

There are a number of approaches which can be applied to construct and search for QC $(J = 2, K)$-regular LDPC block and convolutional codes [12] or the bipartite graphs constructed by their incidence matrices. Since every LDPC convolutional code can be represented by a bipartite Tanner graph using the biadjacency matrix, these techniques can be also applied to $(J \geq 3, K)$ QC LDPC codes. Moreover, bounds on the girth and the minimum distance of $(J = 2, K)$ QC LDPC codes [12] can be generalized to an arbitrary J.

Theorem 1: The minimum distance $d_{\min}$ and the girth g of an $(n, k, d_{\min})$ QC LDPC block code $\mathcal{B}$ obtained from a rate $R = b/c$ convolutional code $\mathcal{C}$ with free distance d_{free} and girth g_{free} by tailbiting to length M are upper-bounded by the inequalities

$$d_{\min} \leq d_{\text{free}}$$
$$g \leq g_{\text{free}}$$

Proof: The first statement follows directly from the fact that any codeword $\boldsymbol{v}(D)$ of the tailbiting block code $\mathcal{B}$, obtained from the parity-check matrix $H(D)$ of the parent convolutional code $\mathcal{C}$, satisfies

$$\boldsymbol{v}(D)H^{\text{T}}(D) = \mathbf{0} \bmod (D^M - 1) \tag{18}$$

Since the parent convolutional code $\mathcal{C}$ satisfies (18) without reduction modulo $(D^M - 1)$ and reducing modulo $(D^M - 1)$ does not increase the weight of a polynomial, the first statement follows directly.

For the second statement we consider the voltage graph $\mathcal{G}_{\text{V}}$ representation of the parent convolutional code $\mathcal{C}$ with girth $g_{\text{V}} = g_{\text{free}}$ together with the Tanner graph representation of the QC LDPC block code $\mathcal{B}$ with girth g. Similar to the relations between the free distance d_{free} and the minimum distance $d_{\min}$, there exists a relation between each cycle within the voltage graph $\mathcal{G}_{\text{V}}$ of the parent convolutional code and the Tanner graph $\mathcal{G}$ of the corresponding block code obtained by tailbiting to length M. The edge voltages for every cycle in $\mathcal{G}_{\text{V}}$ have to sum up to zero. Similarly, every cycle in $\mathcal{G}$ corresponds to a cycle in $\mathcal{G}_{\text{V}}$ such that its edge voltages have to sum up to zero modulo M. From the same argument as before it follows directly that

$$g \leq g_{\text{V}} = g_{\text{free}}$$ ■

In [12] a lower bound on the girth of a voltage graph g_{V} was found via the girth of the corresponding base graph g_{B} for ordinary graphs. It is straightforward to generalize this bound:

Consider a base graph of a QC $(J \geq 3, K)$-regular LDPC convolutional code with girth g_{B} and let d_s denote the sth generalized minimum Hamming distance of the linear $M((J-2)c + b) \times JMc$ block code $\mathcal{B}_{\text{T}}$ determined by the parity-check matrix which corresponds to the incidence matrix of the Tanner graph. In other words, d_s corresponds to the number of nontrivial (not identically zero) positions of an s-dimensional linear subcode.

Theorem 2: There exist a tailbiting length M and a voltage assignment, such that the girth g of the Tanner graph for the corresponding TB block code of length Mc satisfies the inequality

$$g \geq 2 \max \{g_{\text{B}} + \lceil g_{\text{B}}/2 \rceil, d_2\} \tag{19}$$

where d_2 is the second generalized minimum Hamming distance, that is, the minimum support of a subcode of $\mathcal{B}_{\text{T}}$ having dimension two. We have equality in (19), if the underlying base graph consists of two connected cycles, having at least one common vertex.

Proof: According to Theorem 1, any cycle in the Tanner graph of a QC LDPC block code corresponds to a cycle of the same length in the voltage graph. As the labels of the voltage Tanner graph can be freely chosen, it is enough to prove that there is no zero cycle shorter than $2(g_{\text{B}} + \lceil g_{\text{B}}/2 \rceil)$, that is, no such cycle whose voltage is zero regardless of the labeling of the base graph. In particular, such a cycle is also known as an *inevitable cycle* [7] or *balanced cycle* [8]. The number of times each edge in such a cycle of the voltage graph is passed in different directions has to be even. This cycle cannot be simple, since in a simple cycle each edge is passed in one direction only. Hence, the cycle passes through the vertices of a subgraph which contains at least two different cycles, corresponding to two different nonzero codewords. The minimum distance of the code determined by the parity-check matrix which is the incident matrix to the base graph is equal to girth g_{B} (see [12]). According to the Griesmer bound, the smallest length of a linear code with two nonzero codewords of minimum distance d is $d + \lceil d/2 \rceil$, and, hence, the first lower bound of inequality (19) follows.

Consider the second lower bound. The definition of the second generalized minimum Hamming distance implies that the smallest subgraph with two cycles has to have at least d_2 edges. Thus, the second of the two lower bounds gives the precise value of the girth of a subgraph containing two connected cycles, having at least one common vertex. Otherwise, d_2 is a lower bound. ∎

The bounds are tighter than the $3g_{\mathrm{B}}$ bound [7], [38] but not tight if the shortest non-simple cycle consists of two simple cycles connected by a path.

Finally, we want to recall an upper bound on the achievable girth and minimum distance. We start by reformulating the theorem on the achievable girth by Fossorier [18] and thereby generalize it to include base matrices with zero elements.

Theorem 3: Consider the parity-check matrix $H(D)$ of a rate $R = b/c$ convolutional code with base matrix B. Denote the corresponding base graph $\mathcal{G}_{\mathrm{B}}$ and let B' be the 2×3 submatrix

$$B' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{20}$$

If the base matrix B, after possibly reordering its rows and columns, contains the submatrix B', then the girth g_{V} of the corresponding voltage graph $\mathcal{G}_{\mathrm{V}}$ is upper-bounded by

$$g_{\mathrm{V}} \leq 12 \tag{21}$$

regardless of the voltage assignment.

Proof: The subgraph determined by the 2×3 submatrix B' contains 3 symbol nodes, 2 constraint nodes, and 6 edges. Moreover, there exist 3 shortest cycles of length 4. Thus, the base graph $\mathcal{G}_{\mathrm{B}}$ has girth $g_{\mathrm{B}} = 4$ and its second generalized Hamming distance is $d_2 = 6$. Applying Theorem 2, we obtain the precise value of the achievable girth as $2d_2 = 12$, which completes the proof. ∎

Let $H(D)$ be the parity-check matrix of a rate $R = b/c$ (J, K)-regular LDPC convolutional code with free distance d_{free}. By tailbiting to length M we obtain a QC LDPC block code of block length Mc and minimum distance d_{min}. As proven in [2] for parity-check matrices without zero elements and reformulated in [11] for parity-check matrices with zero elements, the corresponding minimum distance d_{min} can be upper-bounded by

$$d_{\text{min}} \leq d_{\text{free}} \leq (c - b + 1)! \tag{22}$$

For parity-check matrices with only nonzero monomials, the inequality simplifies to $(J + 1)!$.

VI. Searching for QC LDPC block codes with large girth

When searching for QC LDPC block codes with large girth, we start from a base graph of a rate $R = b/c$ (J, K)-regular LDPC convolutional code. Using the following algorithm, we determine a suitable voltage assignment based on the group of nonnegative integers, such that the girth of this voltage graph is greater than or equal to a given girth g. Afterwards we replace all edge labels by their corresponding modulo M residuals, where we try to minimize M while preserving the girth g. Using the concept of biadjacency matrices we obtain the corresponding degree matrix W and hence the parity-check matrix of a convolutional code whose bipartite graph has girth $g = g_{\text{free}}$. Tailbiting to lengths M, yields the rate $R = Mb/Mc$ QC LDPC block code whose parity-check matrix is equal to the biadjacency matrix of a bipartite graph with girth g.

The algorithm for determining a suitable voltage assignment for a base graph consists of the following two main steps:

1) Construct a list containing all inequalities describing cycles of length smaller than g within the base graph.
2) Search for such a voltage assignment of the base graph that all inequalities are satisfied.

The efficiency of the second step, searching for a suitable voltage assignment, depends on the chosen representation for the list of inequalities determined during the first step. In general, when searching for all cycles of length g roughly $(J-1)^g$ different paths have to be considered. However, by using a similar idea as in [33] when searching for a path within a trellis, we create a tree of maximum depth $g/2$ and search only for identical nodes within the tree and thereby reduce the complexity to roughly $(J-1)^{g/2}$.

Creating a tree structure

Utilizing the base graph of a rate $R = b/c$ (J, K)-regular LDPC convolutional code, with c symbol and $c - b$ constraint nodes, we construct a separate subtree starting with each of the c symbol nodes.

Before describing the algorithm, we have to introduce some notations. A node in the tree will be denoted by ξ and has a unique parent node ξ^{p}. The underlying base graph is bipartite, that is, every node ξ in the tree with $\xi \in \mathcal{V}_i$ is only connected to nodes $\xi' \in \mathcal{V}_j$ with $i, j \in \{0, 1\}$, $i \neq j$, where $\mathcal{V}_0$ and $\mathcal{V}_1$ are the sets of symbol and constraint nodes, respectively. In other words, a symbol node is only connected to constraint nodes and vice versa. Moreover, every node ξ is characterized by its depth $\ell(\xi)$ and its number $n(\xi)$, where $n(\xi) = i$ follows directly from $\xi = s_i$ or $\xi = c_i$ depending on whether its depth $\ell(\xi)$ is even or odd. In particular, every node satisfies $\xi \in \mathcal{V}_i$ where $i = \ell(\xi) \bmod 2$.

Having introduced those basic notations, we can grow c separate subtrees, with the root node $\xi_{i,\text{root}}$ of the ith subtree being initialized with $\xi \in \mathcal{V}_0$ and depth $\ell(\xi) = 0$. Extend every node $\xi \in \mathcal{V}_i$ at depth $\ell(\xi) = n$ with $i = n \bmod 2$ by connecting it with the nodes $\xi' \in \mathcal{V}_{i+1 \bmod 2}$ at depth $n + 1$ according to the underlying base graph, except ξ^{p} which is already connected to ξ at depth $n - 1$.

Finally we label the edges according to (16) and obtain the voltage for node ξ in the ith subtree as the sum of the edge voltages along the path $\xi_{i,\text{root}} \to \xi$.

Clearly, all subtrees together contain all paths of a given length in the voltage graph. Moreover, taking into account that the girth g of bipartite graphs is always even, we can conclude that in order to check all possible cycles of length at most $g - 2$ in the voltage graph, it is sufficient to grow the corresponding c subtrees up to depth $(g - 2)/2$. Next we construct voltage inequalities for all node pairs (ξ, ξ') in the

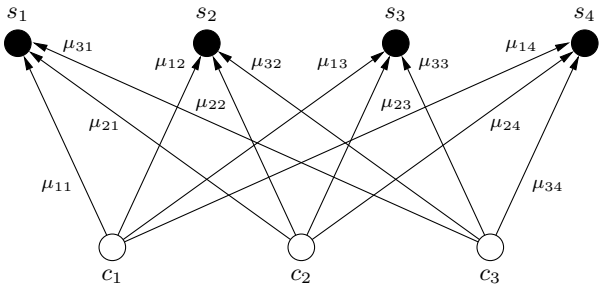

Fig. 3. A bipartite voltage graph with 4 symbol nodes (s_i, $i = 1, 2, 3, 8$) and 3 constraint nodes (c_i, $i = 1, 2, 3$) with its edges labeled according to (16).

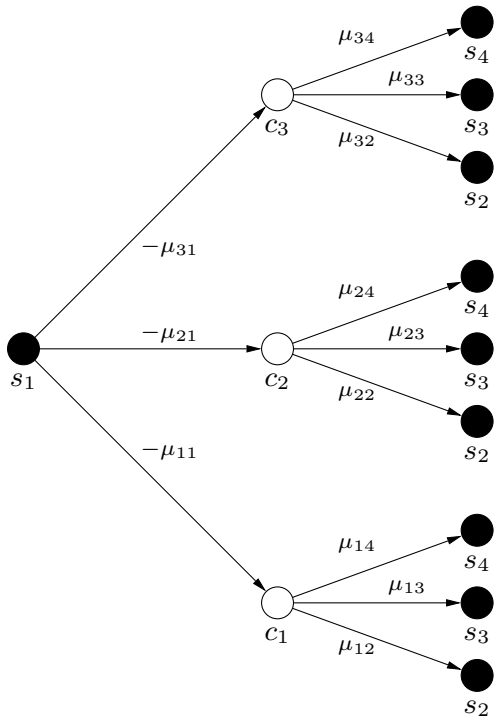

Fig. 4. A tree representation with maximum depth two, starting with symbol node s_1.

same subtree i with the same number $n(\xi) = n(\xi')$ and depth $\ell(\xi) = \ell(\xi')$ but different parent nodes $\xi^{\mathrm{p}} \neq \xi'^{\mathrm{p}}$.

The corresponding voltage inequality for the node pair (ξ, ξ') follows directly as the difference between the voltages for the paths from $\xi_{i,\text{root}}$ to ξ and ξ', respectively, that is, $(\xi_{i,\text{root}} \to \xi) - (\xi_{i,\text{root}} \to \xi')$.

If there exists a cycle $g' < g$, then at depth $g'/2$ there exists at least one pair of nodes (ξ, ξ'), whose corresponding voltage inequality is not satisfied, that is, is equal to zero. If there is no cycle shorter than g in the voltage graph, then there are no such pairs, and all voltage inequalities are satisfied.

Example 2: Consider the rate $R = 1/4$ $(3, 4)$-regular LDPC convolutional code given by (11). The corresponding base graph, with four symbol nodes $s_i \in \mathcal{V}_0$, $i = 1, 2, 3, 4$, and three constraint nodes $c_i \in \mathcal{V}_1$, $i = 1, 2, 3$, is illustrated in Fig. 3. In the following, we shall search for a set of edge labels, that is, monomial degrees in W, such that the corresponding voltage graph has at least a given girth g. Thus, we label the branches by the general edge voltages according to (16) and obtain a bipartite voltage graph.

In order to find a suitable labeling for the edge voltages from the ith constraint node c_i to the jth symbol node s_i, that is, μ_{ij}, $i = 1, 2, 3$, $j = 1, 2, 3, 4$, whose underlying voltage graph has at least girth $g = 6$, we have to grow 4 subtrees up to length $(g-2)/2 = 2$, with their root nodes being initialized by s_i, $i = 1, 2, 3, 4$.

The subtree with root node s_1 is illustrated in Fig. 4. Clearly, at depth $\ell(\xi) = 1$ there are no identical nodes, while at depth $\ell(\xi) = 2$ there are $3 \times \binom{3}{2} = 9$ pairs of identical nodes ($n(\xi) = n(\xi')$) with different parents. Taking into account that a similar subtree is constructed using the remaining three symbol nodes s_2, s_3 and s_4 as root nodes, we obtain in total $36 = 4 \times 9$ node pairs, which all correspond to a voltage inequality.

For example, the voltage inequality obtained by checking all node pairs (ξ, ξ') with $\xi = s_2$, that is, $n(\xi) = 2$, at depth $\ell(\xi) = 2$ in the subtree starting with symbol node s_1, are

$$-\mu_{11} + \mu_{12} - \mu_{22} + \mu_{21} \neq 0$$
$$-\mu_{11} + \mu_{12} - \mu_{32} + \mu_{31} \neq 0$$
$$-\mu_{21} + \mu_{22} - \mu_{32} + \mu_{31} \neq 0$$

Note that amongst all 36 voltage inequalities, there are only 18 unique ones.

Algorithm TR: Constructing a tree representation.

1) Grow c separate subtrees according to the underlying base graph up to depth $g/2 - 1$, with the root node $\xi_{i,\text{root}}$ of the ith subtree being initialized with $\xi \in \mathcal{V}_0$ and depth $\ell(\xi) = 0$.
2) Extend every node $\xi \in \mathcal{V}_i$ at depth $\ell(\xi) = n$ with $i = n \mod 2$ by connecting it with the nodes $\xi' \in \mathcal{V}_{i+1 \bmod 2}$ at depth $n + 1$ according to the underlying base graph, except ξ^{p} which is already connected to ξ at depth $n-1$. Denote the set of all nodes within the ith subtree by $\mathcal{T}_i$.

Searching for a suitable voltage assigment

Using the subtrees $\mathcal{T}_i$, $i = 1, 2, \ldots, c$, with maximum depth $g/2 - 1$, we have found all cycles of length smaller than or equal to $g - 2$ as well as their corresponding voltage inequalities.

Note that the same cycle might be found several times within the c subtrees. Moreover, two different cycles can correspond to the same voltage inequality.

We continue by creating a reduced list $\mathcal{L}$ of node pairs (ξ, ξ') of all c subtrees $\mathcal{T}_i$, $i = 1, 2, \ldots, c$, containing all unique voltage inequalities. Thereby, we remove all duplicate cycles, as well as different cycles corresponding to the same voltage inequality. Using the reduced list $\mathcal{L}$ we can reduce the c subtrees $\mathcal{T}_i$, $i = 1, 2, \ldots, c$, in a similar way by removing all nodes, not participating in any of the cycles in $\mathcal{L}$, and denote the reduced subtree by $\mathcal{T}_{i,\text{min}}$. In other words, we remove all nodes in $\mathcal{T}_i$, $i = 1, 2, \ldots, c$, which only participate in already known cycles or new cycles with already known voltage inequalities.

In the following we present two different approaches for finding suitable edge labels (edge voltages), which we shall denote as Algorithm A and Algorithm B.

In Algorithm A, we label the edges of the reduced subtrees $\mathcal{T}_{i,\min}$, $i = 1, 2, \ldots, c$, with a set of randomly chosen voltages. For every node pair (ξ, ξ') in the list $\mathcal{L}$, we calculate the voltage of the corresponding cycle as the difference of the path voltages $\xi_{i,\text{root}} \to \xi$ and $\xi_{i,\text{root}} \to \xi'$. If none of these cycle voltages is equal to zero, the girth of the underlying base graph with such a voltage assignment is greater than or equal to g.

In Algorithm B, we discard the list $\mathcal{L}$ and only focus on the c reduced subtrees $\mathcal{T}_{i,\min}$. After labeling their edges with a set of randomly chosen voltages, we sort the nodes ξ of each subtree according to their path voltage $\xi_{i,\text{root}} \to \xi$. If there exists no pair of nodes (ξ, ξ') with the same path voltage, number $n(\xi) = n(\xi')$, and depth $\ell(\xi) = \ell(\xi')$, but different parent nodes $\xi^{\text{p}} \neq \xi'^{\text{p}}$, the girth of the underlying base graph with such a voltage assignment is greater than or equal to g.

A formal description of those two algorithms is given below.

Algorithm A: Constructing a system of voltage inequalities and searching for an optimum voltage assignment using a list.

1) Create a reduced list $\mathcal{L}$ of node pairs (ξ, ξ') for all c subtrees $\mathcal{T}_i$, $i = 1, 2, \ldots, c$, containing all node pairs (ξ, ξ') with a unique voltage inequality, having the same number $n(\xi) = n(\xi')$, depth $\ell(\xi) = \ell(\xi')$, but different parent nodes $\xi^{\text{p}} \neq \xi'^{\text{p}}$.
2) Reduce each of the c subtrees $\mathcal{T}_i$ by removing all nodes, which do not participate in any of the found cycles corresponding to the voltage inequalities in $\mathcal{L}$, and denote the reduced subtree structure by $\mathcal{T}_{i,\min}$.
3) Assign randomly chosen voltages to the edges of all trees and perform the following steps:
   a) Find the voltages for all paths leading from the root node $\xi_{i,\text{root}}$ of the ith reduced subtree $\mathcal{T}_{i,\min}$ to all nodes $\xi \in \mathcal{T}_{i,\min}$, $i = 1, 2, \ldots, c$.
   b) Determine the voltage inequality for all cycles $(\xi, \xi') \in \mathcal{L}$, as the difference of the corresponding path voltages in $\mathcal{T}_{i,\min}$, $i = 1, 2, \ldots, c$, computed previously.
   c) If all voltage inequalities are satisfied, the girth of the underlying base graph with such a voltage assignment is greater than or equal to g. Otherwise, assign new random voltages to all edges and go to step a).

Algorithm B: Constructing a system of voltage inequalities and searching for an optimum voltage assignment using a tree.

1) Construct the reduced list $\mathcal{L}$ and the reduced subtrees $\mathcal{T}_{i,\min}$, $i = 1, 2, \ldots, c$, as in Algorithm A without storing the corresponding list $\mathcal{L}$.
2) Assign randomly chosen voltages to the edges of all trees and perform the following steps:
   a) Find the voltages for all paths from the root node $\xi_{i,\text{root}}$ to all nodes within $\mathcal{T}_{i,\min}$, $i = 1, 2, \ldots, c$, and sort all elements within $\mathcal{T}_{i,\min}$ according to their voltages.
   b) Search for a pair of nodes (ξ, ξ') in the sorted list with the same path voltage, number $n(\xi) = n(\xi')$, and depth $\ell(\xi) = \ell(\xi')$, but different parent nodes $\xi^{\text{p}} \neq \xi'^{\text{p}}$.
   c) If no such pair exists, then the girth of the corresponding voltage graph with such a voltage assignment is greater than or equal to g. Otherwise, assign new random voltages to all edges and go to step a).

TABLE I
COMPLEXITY OF A SEARCH FOR SUITABLE VOLTAGE ASSIGNMENT FOR QC $(J = 3, K)$-REGULAR LDPC BLOCK CODES WITH GIRTH $g \leq 12$

K	$g = 8$		$g = 10$		$g = 12$	
	N_{T}	N_{L}	N_{T}	N_{L}	N_{T}	N_{L}
4	53	42	150	231	269	519
5	93	90	286	645	581	1905
6	142	165	485	1470	1060	5430
7	200	273	759	2919	1742	12999
8	267	420	1120	5250	2663	27426
9	343	612	1580	8766	3859	52614
10	428	855	2151	13815	5358	93735
11	522	1155	2845	20790	7210	157410
12	625	1518	3674	30129	9446	251889

Complexity

Denote the sum of all nodes in the reduced tree $\mathcal{T}_{i,\min}$, $i = 1, 2, \ldots, c$, and the number of unique inequalities in the list $\mathcal{L}$ by N_{T} and N_{L}, respectively, that is,

$$N_{\text{T}} = \sum_{i=1}^{c} |\mathcal{T}_{i,\min}| \quad \text{and} \quad N_{\text{L}} = |\mathcal{L}|$$

where $|\mathcal{X}|$ denotes the number of entries in the set $\mathcal{X}$.

Algorithm A requires N_{T} summations for computing the path voltages and N_{L} comparisons for finding cycles, leading to the complexity estimate $N_{\text{T}} + N_{\text{L}}$. Algorithm B requires the same number of N_{T} summations for computing the path voltages, roughly $N_{\text{T}} \log_2 N_{\text{T}}$ operations for sorting the set, and N_{T} comparisons, leading to a total complexity estimate of $N_{\text{T}} \log_2 N_{\text{T}}$.

In Table I the values of N_{T} and N_{L} are given when searching for suitable voltage assignment for (J, K)-regular rate $R = 1 - J/K$ QC LDPC convolutional codes with $J = 3$ and arbitrary $K \geq 4$ and girth g constructed from all-one base matrices. In this case, up to $g = 10$, Algorithm A is preferable, while when searching for voltage assignment with girth $g \geq 12$, Algorithm B should be used.

In the general case we have to consider all node pairs, and as N_{L} is roughly N_{T}^2 we conclude that Algorithm B performs asymptotically better (when $N_{\text{T}} \to \infty$).

VII. MINIMUM DISTANCE OF QC LDPC CODES

Usually the girth of the Tanner graph of a QC LDPC block code is considered to be the most important parameter that affects the performance of belief propagation decoding, as it determines the number of independent iterations [1]. Therefore, most research is focused on finding QC LDPC block

codes with large girth, while their corresponding minimum distance is mostly unknown. In [14] it was shown, that the performance of belief propagation decoding algorithms at high SNRs depends on the structure and the size of the smallest absorbing sets, which, however, can be upper-bounded by the minimum distance. This is the rationale for computing the minimum distance of the shortest known QC LDPC block codes as well as our search for QC LDPC codes with both large girth and large minimum distance.

Our method of calculating the minimum distance is based on the well-known fact that the minimum distance of a linear block code $\mathcal{B}$ with parity-check matrix H is equal to the minimum number of columns of H which sum up to zero. While the proposed method is not limited to the presented code construction, it exploits the sparsity of the parity-check matrix and is as such infeasible for general block codes.

Consider the $M(c-b)\times Mc$ parity-check matrix H_{TB} of the (J,K)-regular rate $R=Mb/Mc$ tailbiting block code $\mathcal{B}$ with block-length Mc (7). Starting with each of the first c columns of H_{TB} as a root, we will construct c separate trees, where each node ξ is characterized by its depth $\ell(\xi)$ and partial syndrome state column vector $\boldsymbol{\sigma}(\xi)$.

Initialize the partial syndrome state of the root $\xi_{i,\text{root}}$ of the ith tree with column i of the corresponding parity-check matrix, that is, $\boldsymbol{\sigma}(\xi_{i,\text{root}})=\boldsymbol{h}_i$, $i=1,2,\ldots,c$. Then grow each tree in such a way, that every branch between any two nodes ξ and ξ' is labeled by a column vector $\boldsymbol{h}_j$, $j\neq i$, such that $\boldsymbol{\sigma}(\xi')=\boldsymbol{\sigma}(\xi)+\boldsymbol{h}_j$, where every branch label on the path $\xi_{i,\text{root}}\rightarrow\xi'$ does not occur more than once.

Consider now a certain node ξ with nonzero state $\boldsymbol{\sigma}(\xi)=(\boldsymbol{\sigma}_1(\xi)\,\boldsymbol{\sigma}_2(\xi)\ldots\boldsymbol{\sigma}_{(c-b)}(\xi))^{\text{T}}$, where $\boldsymbol{\sigma}_j(\xi)$, $j=1,2,\ldots,c-b$ is an $M\times 1$ column vector. If we assume that the kth position of the column vector $\boldsymbol{\sigma}(\xi)$ is nonzero, then there are at most $K-1$ columns which can cancel this nonzero position and have not been considered previously. Therefore, every node ξ has at most $K-1$ children nodes per nonzero position.

Such a tree would grow until all possible linear combinations have been found. Therefore, we assume that the minimum distance is restricted by $d_{\text{min}}<t$, that is, the maximum depth of the tree is $t-1$. Consequently, a node ξ at depth $\ell(\xi)$ will not be extended, if the number of nonzero positions in its partial syndrome state column vector $\boldsymbol{\sigma}(\xi)$ exceeds $J(t-\ell(\xi)-1)$, since at most J ones can be canceled by each branch.

By initially reordering the columns of the parity-check matrix H_{TB} in such a way that each of the $c-b$ nonoverlapping blocks of M rows contains not more than a single one per column, we can strengthen the stopping criterion as follows: A node ξ at depth $\ell(\xi)$ will not be extended, if the number of nonzero positions in each of its partial syndrome state column vectors $\boldsymbol{\sigma}_j(\xi)$, $j=1,2,\ldots,c-b$ exceeds $(t-\ell(\xi)-1)$, since at most one 1 in each block can be canceled by each branch. In particular, such a reordering of the parity-check matrix H_{TB} corresponds directly to the parity-check matrix H_{C} (13) of the equivalent (J,K)-regular LDPC block code constructed from circulant matrices and is consequently only feasible for such LDPC block codes.

Algorithm MD: Determine the minimum distance of a rate $R=b/c$ (J,K)-regular LDPC block code.

1) Assume a suitable restriction t on the minimum distance $d_{\text{min}}<t$.
2) Grow c separate trees as follows:
   a) Initialize the root node of the ith tree by $\boldsymbol{\sigma}(\xi_{\text{root},i})=\boldsymbol{h_i}$ with depth $\ell(\xi)=0$.
   b) Extend all nodes ξ as long as the Hamming weights of their partial syndrome states $w_{\text{H}}(\boldsymbol{\sigma}(\xi))\leq J(t-\ell(\xi)-1)$ (Note, for codes with blocks of M rows containing only a single one, this criterion can be strengthen to $w_{\text{H}}(\boldsymbol{\sigma}_j(\xi))\leq(t-\ell(\xi)-1)$, $j=1,2,\ldots,c-b$).
   c) The minimum distance d_{min} follows directly as
   $$d_{\text{min}}=\min_{\xi}\{\ell(\xi)\mid\boldsymbol{\sigma}(\xi)=\mathbf{0}\}$$
   If there is no node ξ whose partial syndrome state $\boldsymbol{\sigma}(\xi)=\mathbf{0}$, then the minimum distance is lower-bounded by $d_{\text{min}}>t$.

VIII. Search results

When presenting our search results for QC $(J=3,K)$-regular LDPC block codes with different girth we will distinguish two cases.

We started by searching for QC $(J=3,K)$-regular LDPC block codes using an all-one base matrix B, applied the algorithms as described above, and obtained QC $(3,K)$-regular LDPC block codes with girth $g=6,8,10$, and 12 as given in Tables II – V. These codes correspond to parity-check matrices $H(D)$ of convolutional codes with only monomial entries as given, for example, in (10).

However, according to Theorem 3 the achievable girth g of a QC (J,K)-regular LDPC code constructed in such a way is limited by $g\leq 12$. Thus, in order to find QC $(J=3,K)$-regular LDPC block codes with girth $g>12$ as presented in Tables VI and VII, we allow zero entries in our base matrix B. This is a straight-forward generalization of the restriction to only monomial entries in the parity-check matrix $H(D)$ of the underlying convolutional code.

Case I: monomial entries

In Tables II – V, parity-check matrices of short known QC $(J=3,K)$-regular LDPC block codes with girth $g=6$, 8, 10, and 12 together with those of large minimum distance are presented. When searching for such codes, we applied the following restrictions to reduce the number of possible voltage assignments:

- As the girth of a voltage graph is defined as the shortest cycle with voltage zero, and the sign of the voltage depends on the direction of the edge, we can add the same arbitrary offset to the voltage of all edges being connected to the same node. Thus, without loss of generality, we set the voltage of all edges connected to one specific symbol node as well as all edges connected to one specific constraint node to voltage zero. (For consistency with codes constructed from Steiner Triple System, that will

TABLE II
DEGREE MATRICES FOR QC $(J = 3, K)$-REGULAR LDPC CODES WITH GIRTH $g = 6$

K	(n, k)	$d_{\min}$	M	W'
short codes				
4	$(20, 7)$	6	5	$1, 2, 4$ $3, 1, 2$
5	$(25, 12)$	6	5	$1, 2, 3, 4$ $3, 1, 4, 2$
6	$(42, 23)$	4	7	$1, 2, 3, 4, 6$ $3, 5, 2, 1, 4$
7	$(49, 30)$	4	7	$1, 2, 3, 4, 5, 6$ $3, 5, 2, 1, 6, 4$
8	$(72, 47)$	4	9	$1, 2, 3, 4, 5, 7, 8$ $3, 6, 2, 1, 8, 5, 4$
9	$(81, 56)$	4	9	$1, 2, 3, 4, 5, 6, 7, 8$ $3, 6, 2, 1, 8, 7, 5, 4$
10	$(110, 79)$	6	11	$1, 2, 3, 4, 5, 6, 8, 9, 10$ $3, 1, 7, 2, 10, 9, 4, 6, 5$
11	$(121, 90)$	4	11	$1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ $3, 1, 7, 2, 10, 9, 8, 4, 6, 5$
12	$(156, 119)$	6	13	$1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12$ $3, 1, 8, 2, 9, 12, 4, 11, 5, 7, 6,$
large distance codes				
4	$(92, 25)$	22	23	$1, 2, 4$ $5, 3, 12$
5	$(245, 100)$	22	49	$1, 3, 10, 14$ $40, 31, 33, 30$
6	$(414, 209)$	22	69	$3, 4, 21, 26, 67$ $34, 15, 64, 33, 44$
7	$(763, 438)$	22	109	$1, 3, 11, 15, 45, 93$ $101, 34, 18, 9, 1, 4$
8	$(1224, 767)$	22	153	$2, 10, 26, 57, 89, 4, 49$ $22, 19, 5, 23, 61, 90, 123$

TABLE III
DEGREE MATRICES FOR QC $(J = 3, K)$-REGULAR LDPC CODES WITH GIRTH $g = 8$

K	(n, k)	$d_{\min}$	M	W'
short codes				
4	$(36, 11)$	6	9	$1, 4, 6$ $5, 2, 3$
5	$(65, 28)$ $\big((75, 30)$ [24]$\big)$	10	13	$1, 3, 7, 11$ $10, 4, 5, 6$
6	$(108, 56)$ $\big((156, 78)$ [24]$\big)$	10	18	$2, 3, 5, 7, 9$ $4, 6, 13, 1, 16$
7	$(147, 86)$	10	21	$2, 3, 8, 15, 17, 20$ $4, 6, 7, 9, 12, 13$
8	$(200, 127)$	8	25	$1, 3, 4, 10, 14, 15, 19$ $5, 6, 11, 24, 2, 9, 12$
9	$(270, 182)$	8	30	$1, 3, 10, 16, 23, 25, 26, 28$ $2, 6, 5, 9, 8, 12, 14, 22$
10	$(350, 247)$	8	35	$2, 6, 7, 18, 19, 26, 29, 31, 34$ $4, 5, 3, 13, 10, 16, 12, 11, 23$
11	$(451, 330)$	8	41	$1, 4, 8, 20, 27, 28, 29, 33, 39, 40$ $5, 7, 6, 9, 10, 19, 13, 21, 14, 35$
12	$(564, 425)$	8	47	$3, 7, 8, 22, 24, 27, 29, 35, 40, 41, 43$ $6, 2, 4, 5, 14, 16, 1, 21, 28, 9, 34$
large distance codes				
4	$(116, 31)$	24	29	$3, 14, 21$ $7, 1, 17$
5	$(225, 92)$	24	45	$1, 3, 10, 14$ $40, 31, 33, 30$
6	$(431, 218)$	24	72	$3, 4, 21, 26, 67$ $34, 15, 64, 33, 44$
7	$(777, 446)$	24	111	$3, 11, 15, 45, 93, 110$ $34, 18, 9, 1, 4, 101$
8	$(1280, 802)$	24	160	$2, 4, 10, 26, 49, 57, 89$ $22, 90, 19, 5, 123, 23, 61$
9	$(1386, 926)$	20	154	$6, 9, 26, 65, 79, 99, 124, 153$ $24, 16, 14, 1, 46, 62, 137, 84$

be introduced later, we choose the first symbol node and the last constraint node. This corresponds directly to a degree matrix with zeros in its first column and last row.) From a graph theoretical point of view, this procedure is equivalent to creating a spanning tree $\mathcal{S}$ of the base graph $\mathcal{G}_{\text{B}}$ and assigning the edge voltage zero to all edges within $\mathcal{S}$. In particular, the spanning tree above is created such that all edges connected to the first symbol node as well as to the last constraint node are included.

For example, the degree matrix of the $(J = 3, K = 4)$ QC LDPC block code with girth $g = 8$ from Table III follows directly as

$$W = \begin{pmatrix} 0 & 1 & 4 & 6 \\ 0 & 5 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- Furthermore, to reduce the number of only permuted degree matrices, we assume that
  - The first row is sorted in ascending order.
  - When sorting the first and the second row in ascending order, the second row is lexicographically less than the first row.
  - The maximum degree is less than the tailbiting length M for which there exists a QC $(J = 3, K)$-regular LDPC block code with given girth g.
- QC $(J = 3, K = 4)$-regular LDPC block codes were found by exhaustive search over the previously defined set of restricted edge voltages.
- QC $(J = 3, K = N)$-regular LDPC block codes with $N > 4$ were obtained by adding one additionally, randomly chosen column to the best degree matrices of codes with $K = N - 1$ having the same girth g. The maximum degree in this additional column is limited by twice the maximum degree of the previous code.

Using these restrictions, the obtained QC $(J = 3, K)$-regular LDPC block codes with girth $g = 6$, 8, 10, and 12 are presented in Tables II – V.

The first column K denotes the number of nonzero elements per row, which corresponds to the number of columns in $H(D)$ and W. As all entries in the first column and the last row of the degree matrix W are zero, they are omitted in the submatrix W'.

Consider now the parity-check matrix $H(D)$ of the rate $R = 1-J/K$ convolutional code $\mathcal{C}$ with only monomial entries corresponding to the degree matrix W. By tailbiting the semi-infinite parity-check matrix H to length M (given in the fourth column), we obtain the parity-check matrix H_{TB} of an (n, k) block code $\mathcal{B}$ with minimum distance $d_{\min}$, where (n, k) and

TABLE IV
DEGREE MATRICES FOR QC $(J = 3, K)$-REGULAR LDPC CODES WITH GIRTH $g = 10$

K	(n, k)	$d_{\min}$	M	W'
short codes				
4	$(148, 39)$ $\big((144, 36)$ [24]$\big)$	14	37 $\big(39$ [26]$\big)$	$1, 14, 17$ $11, 6, 2$
5	$(305, 124)$ $\big((550, 220)$ [24]$\big)$	24	61 $\big(61$ [19]$\big)$	$2, 20, 54, 60$ $26, 16, 31, 48$
6	$(606, 305)$ $\big((780, 390)$ [24]$\big)$	24	101 $\big(103$ [26]$\big)$	$2, 24, 25, 54, 85$ $21, 15, 11, 8, 59$
7	$(1113, 638)$	24	159 $\big(160$ [26]$\big)$	$2, 14, 27, 67, 97, 130$ $21, 24, 1, 6, 75, 58$
8	$(1752, 1097)$	24	219 $\big(233$ [26]$\big)$	$3, 14, 26, 63, 96, 128, 183$ $24, 6, 19, 46, 4, 77, 107$
9	$(2871, 1916)$	24	319 $\big(329$ [26]$\big)$	$6, 9, 26, 65, 99, 153, 233, 278$ $24, 16, 14, 1, 62, 84, 200, 137$
10	$(4300, 2912)$	24	430 $\big(439$ [26]$\big)$	$9, 11, 26, 67, 101, 161, \ldots$ $233, 302, 395$ $23, 5, 1, 54, 33, 96, \ldots$ $120, 104, 244$
11	$(6160, 4482)$	24	560 $\big(577$ [26]$\big)$	$2, 11, 25, 62, 101, 162, 225, \ldots$ $268, 421, 492$ $24, 21, 5, 55, 6, 59, 178, \ldots$ $132, 204, 311$
12	$(8844, 6635)$	22	737 $\big(758$ [26]$\big)$	$2, 22, 23, 63, 101, 147, 219, \ldots$ $322, 412, 569, 601$ $16, 9, 6, 58, 34, 91, 126, \ldots$ $155, 185, 298, 232$
large distance codes				
4	$(176, 46)$	24	44	$1, 14, 17$ $11, 6, 2$

TABLE V
DEGREE MATRICES FOR QC $(J = 3, K)$-REGULAR LDPC CODES WITH GIRTH $g = 12$

K	(n, k)	$d_{\min}$	M	W'
short codes				
4	$(292, 75)$ $\big((444, 111)$ [24]$\big)$	24	73 $\big(97$ [8]$\big)$	$2, 25, 33$ $18, 6, 5$
5	$(815, 328)$ $\big((1700, 680)$ [24]$\big)$	24	163 $\big(181$ [19]$\big)$	$5, 33, 42, 117$ $36, 35, 25, 57$
6	$(1860, 932)$ $\big((4680, 2340)$ [24]$\big)$	24	310 $\big(393$ [22]$\big)$	$1, 24, 38, 145, 246$ $16, 36, 5, 82, 110$
6	$(1836, 920)$	24	306 $\big(393$ [22]$\big)$	$9, 36, 38, 154, 204$ $33, 1, 13, 54, 123$
7	$(3962, 2266)$	24	566 $\big(881$ [8]$\big)$	$3, 10, 33, 147, 297, 442$ $31, 22, 4, 93, 133, 219$
8	$(6784, 4242)$	24	848 $\big(1493$ [8]$\big)$	$4, 24, 31, 143, 303, 498, 652$ $32, 9, 6, 70, 130, 193, 222$
9	$(12384, 8258)$	24	1376 $\big(2087$ [8]$\big)$	$4, 20, 32, 160, 284, \ldots$ $569, 794, 1133$ $30, 7, 1, 92, 169, \ldots$ $350, 437, 645$
10	$(21030, 14723)$	24	2103	$6, 13, 28, 150, 291, 565, \ldots$ $678, 1258, 1600$ $30, 16, 5, 64, 225, 207, \ldots$ $491, 838, 746$
11	$(34507, 25098)$	24	3137	$9, 11, 24, 150, 306, 508, \ldots$ $666, 1279, 1765, 1958$ $31, 28, 1, 83, 131, 160, \ldots$ $429, 550, 956, 1391$
12	$(56760, 42572)$	24	4730	$3, 15, 22, 140, 286, 537, \ldots$ $811, 1113, 1878, 2524, 3349$ $31, 26, 1, 66, 95, 210, 373, \ldots$ $729, 878, 1365, 1644$

$d_{\min}$ follow from the second and third column, respectively. Note that due to linear dependent rows in H_{TB} the rate of $\mathcal{B}$ might be larger than $1 - J/K$.

The codes presented in Tables II and III coincide with the QC LDPC block codes found by the "hill-climbing" algorithm [26], but we determined their minimum distance with our algorithm described in Section VII. Tables IV and V contain new QC $(J = 3, K)$-regular LDPC block codes, which, to the best of our knowledge, are shorter than the previously known codes obtained from an all-one base matrix [8], [19], [22], [26]. In particular, these codes are significantly shorter than those presented in [24], which are obtained from base matrices with zeros. However, due to the zeros in their base matrix, the minimum distance of the LDPC block codes in [24] can exceed $(J + 1)!$. For example, we determined the minimum distance of the $(444, 111)$ QC $(3, 4)$-regular LDPC block code with girth $g = 12$ in [24] to be $d_{\min} = 28$, while the corresponding code in Table V, that is, the $(292, 75)$ QC $(3, 4)$-regular LDPC block code, has only minimum distance $d_{\min} = 24$, but shorter block length. Using the BEAST [33], we calculated the free distance of the corresponding parent convolutional code for the code in [24] to be $d_{\text{free}} = 46$. Therefore, using our approach and a larger tailbiting length it would be possible to construct corresponding QC $(3, 4)$-regular LDPC block codes with minimum distance up to 46.

Case 2: monomial or zero entries

In order to find QC $(J = 3, K)$-regular LDPC block codes with girth $g \geq 14$, we have to allow zero entries in our base matrix B; that is, relax the restriction from only monomial entries in $H(D)$ to include also zero entries. According to Theorem 2, a code with girth g exists if the corresponding base graph has girth g_{B} satisfying (19). Additionally, as we are searching for codes with short block length, we consider the shortest possible base matrices B.

Case 2-I: Steiner Triple Systems

When searching for QC $(J = 3, K)$-regular LDPC block codes with girth $g = 14, 16$, and 18, we started with a (shortened) base graph constructed by using Steiner triple systems of order n, that is, STS(n) [16], [17], [31].

For all n, where $n \bmod 6$ is equal to 1 or 3, there exists a Steiner triple system of order n. Then we construct a (J, K)-regular, $(c - b) \times c$ base matrix B with entries b_{ij}, $i = 1, 2, \ldots, c - b$ and $j = 1, 2, \ldots, c$, in such a way that the positions of the nonzero entries in each column correspond to a Steiner triple system of order $(c-b)$. Denote such a $(c-b)\times c$ base matrix $B_{\text{STS}(c-b)}$.

Using the obtained (J, K)-regular $(c-b)\times c$ base matrix B, we search for a set of edge labels, such that the corresponding voltage graph has at least girth g.

TABLE VI
DEGREE MATRICES FOR QC LDPC CODES WITH GIRTH $g = 14$ TO 18

K	g	(n, k)	M	Base graph	W'
4	14	$(1812, 453)$ $\big((2208, 732)$ [24]$\big)$	151 $\big(184$ [24]$\big)$	STS(9) (9×12)	$0, 123, 36, 3, 2, 79, 4, 7, 52, 4, 1$ $0, 96, 23, 11, 1, 37, 12, 2, 61, 1, 4$
5	14	$(9720, 3888)$ $\big((11525, 4612)$ [24]$\big)$	486	S-STS(13) (12×20)	$423, 0, 437, 5, 237, 235, 170, 333, 260, 109, 241, 2, 114, 5, 2, 428, 92, 228, 299$ $0, 0, 0, 445, 465, 51, 440, 22, 111, 307, 433, 4, 285, 2, 1, 4, 113, 282, 5$
6	14	$(29978, 14989)$ $\big((37154, 18577)$ [24]$\big)$	1153 $\big(1429$ [24]$\big)$	STS(13) (13×26)	$1037, 0, 1051, 1105, 933, 1027, 962, 1000, 665, 805, 646, 2, \ldots$ $906, 5, 2, 1095, 788, 633, 913, 264, 51, 772, 672, 686, 737$ $0, 0, 0, 1112, 1132, 51, 1107, 22, 807, 921, 1100, 4, 952, 2, \ldots$ $1, 4, 905, 949, 5, 0, 1111, 922, 620, 351, 140$
12	14	$n = 80000000$	800000	STS(25)	*available at [41]*
4	16	$(7980, 1995)$	665	STS(9) (9×12)	$0, 468, 99, 3, 2, 305, 43, 9, 251, 3, 2$ $0, 351, 41, 6, 8, 215, 18, 1, 79, 1, 8$
5	16	$(51240, 20496)$ $\big((62500, 25000)$ [24]$\big)$	2562	S-STS(13) (12×20)	$937, 0, 1551, 1264, 1670, 2119, 1973, 1960, 1848, 1223, 1806, \ldots$ $15, 1761, 1, 2, 2175, 1169, 1768, 548$ $0, 0, 0, 2367, 2491, 126, 2296, 66, 1197, 582, 2200, 9, \ldots$ $1836, 2, 1, 0, 1757, 1833, 4$
6	16	$(227032, 113516)$ $\big((229476, 114738)$ [24]$\big)$	8732 $\big(8826$ [24]$\big)$	STS(13) (13×26)	$8328, 0, 8393, 8106, 7840, 8289, 8143, 8130, 6821, 7393, 6779, 15, 7931, \ldots$ $1, 2, 8345, 7339, 6741, 7390, 1557, 498, 6357, 5666, 5001, 1684$ $0, 0, 0, 8537, 8661, 126, 8466, 66, 7367, 7424, 8370, 9, 8006, 2, 1, 0, \ldots$ $7927, 8003, 4, 0, 8412, 5799, 4553, 2142, 6293$
4	18	$(32676, 8169)$	2723 $\big(2855$ [24]$\big)$	STS(9) (9×12)	$0, 853, 217, 6, 2, 1108, 75, 20, 586, 1, 5$ $0, 1797, 97, 3, 4, 485, 33, 37, 246, 1, 5$
5	18	$(271760, 108704)$ $\big((371100, 148440)$ [24]$\big)$	13588	S-STS(13) (12×20)	$10484, 0, 12275, 10611, 9703, 10786, 10227, 11122, 3263, 7933, \ldots$ $3129, 21, 9554, 1, 2, 12183, 7837, 3084, 8297$ $0, 0, 0, 12012, 13041, 498, 12534, 223, 7947, 8356, \ldots$ $12213, 13, 10701, 2, 1, 0, 9550, 10698, 4$

In general, it is possible, without loss of generality, to label a certain subset of edges of the voltage graph simultaneously with zero voltage and thereby decreasing the number of possible labelings. The following algorithm constructs a $(c-b) \times c$ base matrix B based on STS$(c-b)$ and reorders the matrix to maximize the number of zero entries in its lower left corner. Using such a base matrix, it is always possible to label the last nonzero entry in each column with degree zero. Moreover, in each of the remaining rows at the top of the base matrix, we can label at least one nonzero entry with degree zero. (Hereinafter we will always choose at least the first element in the remaining rows to be labeled with zero voltage).

Algorithm STS: Construction of a (J, K)-regular $(c - b) \times c$ base graph B obtained from STS$(c - b)$.

1) Initialize a counter u with the number of nonzero entries per row, that is, $u = K$. Moreover, denote the current row and column by s and t, respectively, starting from the right-most entry in the last row, that is, initialize those variables with $s = c - b$ and $t = c$.
2) Set the u entries in row s and column $t, t-1, \ldots, t-u+1$ to one, that is, set the entries $b_{ij} = 1$ for $i = s$ and $j = t, t - 1, \ldots, t - u + 1$.
3) Choose the remaining $J - 1$ nonzero positions in each of those u columns to fulfill the properties of a Steiner Triple System. If possible, choose the positions b_{ij} to minimize i. In other words, try to avoid using the lowest rows $s - 1, s - 2, \ldots$, if possible, despite of the restrictions imposed by the Steiner Triple System.
4) Finally, decrease t by u. If $t = 0$, then stop as all c columns are used. Otherwise, set s to $s - 1$, denote the number of nonzero elements in the columns $1, 2, \ldots, t$ and row s by u, and go to step 2.

By removing the last row and last K columns of the (J, K)-regular $(c-b) \times c$ base matrix B constructed using STS$(c-b)$, we obtain a shortened $(c-b-1) \times (c-K)$ $(J, K-1)$-regular base matrix B', which we denote $B_{\text{S-STS}(c-b)}$. By deleting additional columns and rows, it is possible to obtain codes with intermediate rates, which are, however, irregular.

Example 3: In the following we shall construct the $(J = 3, K)$ base matrices B of dimension 9×12 $(K = 4)$, dimension 13×26 $(K = 6)$, and dimension 25×100 $(K = 7)$. Using Algorithm STS, we obtain the following Steiner Triple Systems of order 9 (STS(9)), 13 (STS(13)), and 25 (STS(25)).

$$\begin{aligned}&\textbf{STS}(9) = \\ &\Big\{\{2,3,5\},\{1,4,6\},\{1,3,7\},\{2,6,7\},\\ &\ \{4,5,7\},\{1,2,8\},\{5,6,8\},\{3,4,8\},\\ &\ \{1,5,9\},\{2,4,9\},\{3,6,9\},\{7,8,9\}\Big\}\end{aligned}$$

$$\mathbf{STS}(13) =$$
$$\Big\{\{0,3,6\},\ \{0,2,7\},\ \{1,5,7\},\ \{3,4,7\},$$
$$\{3,5,8\},\ \{1,4,8\},\ \{2,6,8\},\ \{2,4,9\},$$
$$\{5,6,9\},\ \{0,1,9\},\ \{1,3,10\},\ \{0,4,10\},$$
$$\{6,7,10\},\ \{2,5,10\},\ \{8,9,10\},\ \{7,8,11\},$$
$$\{4,6,11\},\ \{1,2,11\},\ \{0,5,11\},\ \{3,9,11\},$$
$$\{10,11,12\},\{7,9,12\},\ \{0,8,12\},\ \{1,6,12\},$$
$$\{4,5,12\},\ \{2,3,12\}\Big\}$$

$$\mathbf{STS}(25) =$$
$$\Big\{\{4,5,10\},\ \{1,9,10\},\ \{7,8,11\},\ \{1,6,11\},$$
$$\{2,3,12\},\ \{0,9,12\},\ \{6,8,12\},\ \{8,9,13\},$$
$$\{6,7,13\},\ \{0,5,13\},\ \{2,10,13\},\ \{3,4,14\},$$
$$\{1,12,14\},\ \{0,2,14\},\ \{7,9,14\},\ \{5,11,14\},$$
$$\{5,6,15\},\ \{3,10,15\},\ \{4,12,15\},\ \{1,7,15\},$$
$$\{0,8,15\},\ \{11,13,16\},\{5,7,16\},\ \{6,10,16\},$$
$$\{2,8,16\},\ \{3,9,16\},\ \{0,4,16\},\ \{9,11,17\},$$
$$\{12,13,17\},\{1,3,17\},\ \{4,7,17\},\ \{0,6,17\},$$
$$\{2,5,17\},\ \{8,17,18\},\ \{3,11,18\},\ \{2,4,18\},$$
$$\{13,15,18\},\{0,10,18\},\ \{1,16,18\},\ \{6,14,18\},$$
$$\{9,18,19\},\ \{4,8,19\},\ \{14,15,19\},\{10,11,19\},$$
$$\{0,3,19\},\ \{2,7,19\},\ \{12,16,19\},\{1,5,19\},$$
$$\{17,19,20\},\{9,15,20\},\ \{10,12,20\},\{0,11,20\},$$
$$\{5,8,20\},\ \{1,4,20\},\ \{13,14,20\},\{3,7,20\},$$
$$\{2,6,20\},\ \{5,18,21\},\ \{4,6,21\},\ \{1,13,21\},$$
$$\{16,17,21\},\{10,14,21\},\{2,9,21\},\ \{3,8,21\},$$
$$\{11,15,21\},\{7,12,21\},\ \{19,21,22\},\{18,20,22\},$$
$$\{0,7,22\},\ \{10,17,22\},\{3,5,22\},\ \{6,9,22\},$$
$$\{2,15,22\},\ \{1,8,22\},\ \{11,12,22\},\{4,13,22\},$$
$$\{14,16,22\},\{20,21,23\},\{0,1,23\},\ \{6,19,23\},$$
$$\{15,16,23\},\{2,11,23\},\ \{7,18,23\},\ \{5,12,23\},$$
$$\{14,17,23\},\{4,9,23\},\ \{8,10,23\},\ \{3,13,23\},$$
$$\{0,21,24\},\ \{22,23,24\},\{1,2,24\},\ \{16,20,24\},$$
$$\{7,10,24\},\ \{8,14,24\},\ \{13,19,24\},\{3,6,24\},$$
$$\{12,18,24\},\{15,17,24\},\{5,9,24\},\ \{4,11,24\}\Big\}$$

Each number $1,2,\ldots,c$ occurs K times within the set of Steiner triples. Note that the chosen Steiner triples are not uniquely determined.

The corresponding base matrices of dimension 9×12 $\mathrm{STS}(9)$, dimension 13×26 $\mathrm{STS}(13)$, and dimension 25×100 $\mathrm{STS}(25)$ are sparse matrices with nonzero elements only in column i and row j, where the ith Steiner Triple contains the value j. The 9×12 $(3,4)$-regular base matrix constructed from $\mathrm{STS}(9)$ denoted by $B_{\mathrm{STS}(9)}$ is given, for example, by

$$B_{\mathrm{STS}(9)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\ \begin{pmatrix} 0 & \mathbf{1} & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{1} & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix} \end{array} \tag{23}$$

Entries that correspond to edges in the base graph that can be, according to Algorithm STS, labeled with zero voltage are marked in bold.

By removing the last row and the last $K = 4$ columns, the corresponding shortened 8×8 $(3,3)$-regular base matrix $B_{\text{S-STS}(9)}$ follows directly as

$$B_{\text{S-STS}(9)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & \mathbf{1} & 1 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \mathbf{1} & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & \mathbf{1} & 0 & 0 & 1 & 0 & 0 & 1 \\ \mathbf{1} & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & \mathbf{1} & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix} \end{array} \tag{24}$$

This corresponds to removing the four Steiner Triples of $\mathrm{STS}(9)$ containing the number of the last row. Shortening the 9×12 base matrix $B_{\mathrm{STS}(9)}$ constructed from $\mathrm{STS}(9)$ to obtain a shortened 8×8 base matrix $B_{\text{S-STS}(9)}$ is unpractical as its code rate is $R = 1 - 8/8 = 0$. However, by shortening the 13×25 base matrix $B_{\mathrm{STS}(13)}$ in the same way we obtain a 12×20 base matrix $B_{\text{S-STS}(13)}$ with the feasible code rate $R = 8/20$.

In Table VI the obtained QC $(J = 3, K)$-regular LDPC block codes with girth $g = 14, 16$, and 18 constructed from Steiner Triple Systems are presented. While the number of nonzero elements in each column is fixed to $J = 3$, the number of nonzero elements in each row K is specified in the first column. The second column corresponds to the obtained girth g, while in the third and forth columns we give the dimensions of the block code (n, k) after tailbiting to length M. The fifth column specifies the used Steiner Triple System ($\mathrm{STS}(n)$).

Finally, in the last column W', we give the degrees of the corresponding degree matrix W in a compact way. As we have constructed the base matrices in such a way that the last nonzero entry in each column and the first entry in all other rows of the base matrix is labeled with a zero voltage, these entries are omitted. An entry of W' in column j and row i corresponds to the voltage degree of the $(j+1)$th nonzero entry in the ith row of the corresponding base matrix.

TABLE VII
PROPERTIES OF QC $(J = 3, K)$-REGULAR LDPC CODES WITH GIRTH $g \geq 20$

K	g	(n, k)	M	Base graph (Table III)
4	20	$(1296000, 324002)$	36000	(27×36), $g = 8$
5	20	$(31200000, 12480002)$	480000	(39×65), $g = 8$
6	20	$(518400000, 259200002)$	4800000	(54×108), $g = 8$
4	22	$(7200000, 1800002)$	200000	(27×36), $g = 8$ [24]
5	22	$(325000000, 130000002)$	5000000	(39×65), $g = 8$
4	24	$(39600000, 9900002)$	1100000	(27×36), $g = 8$

Case 2-II: (J, K)-regular LDPC block codes

When searching for QC $(J = 3, K)$-regular LDPC block codes with girth $g = 20$, 22, and 24, we started with previously obtained QC $(J = 3, K)$-regular LDPC block codes of smaller block size and smaller girth, and (re-)applied our algorithms.

The obtained results for QC $(J = 3, K)$-regular LDPC block codes with girth $g = 20$, 22, and 24 are presented in Table VII. They are all but one based on previously obtained $(J = 3, K)$-regular LDPC block codes with girth $g = 8$ (*cf.* Table III), as specified in the last column in Table VII. As before, the first column K denotes the number on nonzero elements in each column; then we give the obtained girth g and the dimensions of the block code (n, k) after tailbiting to length M. The corresponding degree matrices are too large and are omitted in Table VII, but are available at [41].

These codes are (probably) not practical due to their huge block length. However, the table illustrates that by interpreting QC (J, K)-regular LDPC block codes as base matrices and re-applying our algorithms we can find QC (J, K)-regular LDPC block codes of "any" girth g.

IX. CONCLUSIONS

Using the relation between the parity-check matrix of QC LDPC block codes and the biadjacency matrix of bipartite graphs, new search techniques have been presented. Starting from a base graph, a set of edge voltages is used to construct the corresponding voltage graph with a given girth.

By representing bipartite graphs in different ways, lower and upper bounds on the girth as well as on the minimum distance of the corresponding tailbiting block code have been discussed.

New algorithms for searching iteratively for bipartite graphs with large girth and for determining the minimum distance of the corresponding QC LDPC block code have been presented. Depending on the given girth, the search algorithms are either based on all-one matrices, Steiner Triple Systems, or QC block codes. Amongst others, new QC regular LDPC block codes with girth between 10 and 24 have been presented. In particular, the previously unknown minimum distance, for some known codes with girth 6 and 8, has been determined.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay and M. C. Davey, "Evaluation of Gallager codes for short block length and high rate applications," in *Codes, Systems and Graphical Models*. Springer-Verlag, 1999, pp. 113–130.

[3] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[4] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[5] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.

[6] G. Schmidt, V. V. Zyablov, and M. Bossert, "On expander codes based on hypergraphs," in *Proc. IEEE International Symposium on Information Theory (ISIT'03)*, Yokohama, Japan, Jun. 29 – Jul. 4, 2003, p. 88.

[7] S. Kim, J.-S. No, H. Chung, and D.-J. Shin, "Quasi-cyclic low-density parity-check codes with girth larger than 12," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2885–2891, Aug. 2007.

[8] M. E. O'Sullivan, "Algebraic construction of sparse matrices with large girth," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 718–727, Feb. 2006.

[9] A. Barg and G. Zémor, "Distance properties of expander codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 78–90, Jan. 2006.

[10] I. E. Bocharova, R. Johannesson, B. D. Kudryashov, and V. V. Zyablov, "Woven graph codes: Asymptotic performances and examples," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 121–129, Jan. 2010.

[11] I. E. Bocharova, B. D. Kudryashov, R. V. Satyukov, and S. Stiglmayr, "Short quasi-cyclic LDPC codes from convolutional codes," in *Proc. IEEE International Symposium on Information Theory (ISIT'09)*, Seoul, South-Korea, Jun. 28 –Jul. 3, 2009, pp. 551–555.

[12] I. E. Bocharova, B. D. Kudryashov, and R. V. Satyukov, "Graph-based convolutional and block LDPC codes," *Problems of Information Transmission*, vol. 45, no. 4, pp. 357–377, 2009.

[13] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. New York: North Holland, 1976.

[14] L. Dolecek, P. Lee, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright, "Predicting error floors of structured LDPC codes: Deterministic bounds and estimates," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 239–246, Aug. 2009.

[15] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[16] S. J. Johnson and S. R. Weller, "Regular low-density parity-check codes from combinatorial designs," in *Proc. IEEE Information Theory Workshop (ITW'01)*, Cairns, Australia, Sep. 2–7, 2001, pp. 90–92.

[17] ——, "Construction of low-density parity-check codes from Kirkman triple systems," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM'01)*, vol. 2, San Antonio, USA, Nov. 25–29, 2001, pp. 970–974.

[18] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[19] R. M. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," in *Proc. ISTA*, Ambleside, England, 2001.

[20] R. M. Tanner, "On graph constructions for LDPC codes by quasi-cyclic extension," in *Information, Coding and Mathematics*, M. Blaum, P. G. Farrell, and H. C. A. van Tilborg, Eds. Norwell, MA: Kluwer, 2002, pp. 209–219.

[21] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, Jr., "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

[22] G. Zhang and X. Wang, "Girth-12 quasi-cyclic LDPC codes with consecutive lengths," *arXiv: 1001.3916v1*, Jan.

[23] O. Milenkovic, N. Kashyap, and D. Leyba, "Shortened array codes of large girth," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3707–3722, Aug. 2006.

[24] M. Esmaeili and M. Gholami, "Structured quasi-cyclic LDPC codes with girth 18 and column-weight $J \geq 3$," *International Journal of Electronics and Communications (AEU)*, vol. 64, no. 3, pp. 202–217, Mar. 2010.

[25] R. Smarandache and P. O. Vontobel, "On regular quasi-cyclic LDPC codes from binomials," in *Proc. IEEE International Symposium on*

Information Theory (ISIT'04), Chicago, USA, Jun. 27 – Jul. 2, 2004, p. 274.
[26] Y. Wang, J. S. Yedidia, and S. C. Draper, "Construction of high-girth QC-LDPC codes," in *Proc. 5th International Symposium on Turbo Codes and Related Topics*, Lausanne, Switzerland, Sep. 1–5, 2008, pp. 180–185.
[27] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2th International Symposium on Turbo Codes and Related Topics*, Brest, France, Sep. 4–7, 2000, pp. 543–546.
[28] I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, and R. V. Satyukov, "New low-density parity-check codes with large girth based on hypergraphs," in *Proc. IEEE International Symposium on Information Theory (ISIT'10)*, Austin, Texas, Jun. 13 – 18, 2010, pp. 819–823.
[29] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[30] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
[31] J. Thorpe, K. Andrews, and S. Dolinar, "Methodologies for designing LDPC codes using protographs and circulants," in *Proc. IEEE International Symposium on Information Theory (ISIT'04)*, Chicago, USA, Jun. 27 – Jul. 2, 2004, p. 238.
[32] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of Convolutional Coding*. Piscataway, NJ: IEEE Press, 1999.
[33] I. E. Bocharova, M. Handlery, R. Johannesson, and B. D. Kudryashov, "A BEAST for prowling in trees," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1295–1302, Jun. 2004.
[34] X.-Y. Hu and M. P. C. Fossorier, "On the computation of the minimum distance of low-density parity-check codes." [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/MINDIST/main.pdf
[35] X.-Y. Hu, M. P. C. Fossorier, and E. Eleftheriou, "On the computation of the minimum distance of low-density parity-check codes," in *Proc. IEEE International Conference on Communications (ICC'04)*, vol. 2, Paris, France, Jun. 20–24, 2004, pp. 767–771.
[36] A. S. Asratian, T. M. J. Denley, and R. Haggkvist, *Bipartite Graphs and Their Applications*. Cambridge, U.K: Cambridge University Press, 1998.
[37] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.
[38] C. A. Kelley and J. L. Walker, "LDPC codes from voltage graphs," in *Proc. IEEE International Symposium on Information Theory (ISIT'08)*, Toronto, Canada, Jul. 6–11, 2008.
[39] J. L. Gross, "Voltage graphs," *Discrete Mathematics*, vol. 9, no. 3, pp. 239–246, 1974.
[40] G. Exoo and R. Jajcay, "Dynamic cage survey," *The Electronic Journal of Combinatorics*, vol. 15, Sep. 2008.
[41] "Degree matrices for QC LDPC codes." [Online]. Available: http://www.eit.lth.se/goto/QC_LDPC_Codes

Irina E. Bocharova was born in Leningrad, U.S.S.R., on July 31, 1955. She received the Diploma in Electrical Engineering in 1978 from the Leningrad Electro-Technical Institute and the Ph.D. degree in technical sciences in 1986 from the Leningrad Institute of Aerospace Instrumentation.

During 1986-2007, she has been a Senior Researcher, an Assistant Professor, and then Associate Professor at the Leningrad Institute of Aerospace Instrumentation (now State University of Aerospace Instrumentation, St.-Petersburg, Russia). Since 2007 she has been an Associate Professor at the State University of Information Technologies, Mechanics and Optics. Her research interests include convolutional codes, communication systems, source coding and its applications to speech, audio and image coding. She has published more than 50 papers in journals and proceedings of international conferences, and seven U.S. patents in speech, audio and video coding. She has authored the textbook *Compression for Multimedia* (Cambridge University Press, 2010).

Professor Bocharova was awarded the Lise Meitner Visiting Chair in engineering at Lund University, Lund, Sweden (January–June 2005 and 2011).

Florian Hug (S'08) was born in Memmingen, Germany, on May 21, 1983. He received the Dipl.-Ing. degree from the Faculty of Electrical Engineering, University of Ulm, Ulm, Germany, in 2008. Since then, he has been with the Department of Electrical and Information Technology, Lund University, Lund, Sweden, where he is working towards the Ph.D. degree in information theory. His research interests covers the field of information and coding theory. Currently, he is focusing on codes over graphs.

Rolf Johannesson (M'72, F'98, LF'12) was born in Hässleholm, Sweden, on July 3, 1946. He received the M.S. and Ph.D. degrees in 1970 and 1975, respectively, both from Lund University, Lund, Sweden, and the degree of Professor, *honoris causa*, from the Institute for Problems of Information Transmission, Russian Academy of Sciences, Moscow, in 2000.

Since 1976, he has been a faculty member with Lund University where he has the Chair of Information Theory. From 1976 to 2003, he was department Head and during 1988-1993 Dean of the School of Electrical Engineering and Computer Sciences. During 1990-1995, he served as a member of the Swedish Research Council for Engineering Sciences. His scientific interests include information theory, error-correcting codes, and cryptography. In addition to papers in the area of convolutional codes and cryptography, he has authored two textbooks on switching theory and digital design (both in Swedish) and one on information theory (in both Swedish and German) and coauthored *Fundamentals of Convolutional Coding* (New York: IEEE Press, 1999), and *Understanding Information Transmission* (Hoboken, NJ: IEEE Press/Wiley-Interscience, 2005).

Professor Johannesson has been an Associate Editor for the *International Journal of Electronics and Communications*. During 1983-1995 he co-chaired seven Russian-Swedish Workshops, which were the chief interactions between Russian and Western scientists in information theory and coding during the final years of the Cold War. He became an elected member of the Royal Swedish Academy of Engineering Sciences in 2006.

Boris D. Kudryashov was born in Leningrad, U.S.S.R., on July 9, 1952. He received the Diploma in Electrical Engineering in 1974 and the Ph.D in technical sciences degree in 1978 both from the Leningrad Institute of Aerospace Instrumentation, and the Doctor of Science degree in 2005 from Institute of Problems of Information Transmission, Moscow.

In 1978, he becane an Assistant Professor and then Associate Professor and Professor in the Leningrad Institute of Aerospace Instrumentation (now the State University on Aerospace Instrumentation, St.-Petersburg, Russia). Since November 2007, he has been a Professor with the State University on Information Technologies, Mechanics and Optics, St.-Petersburg, Russia. His research interests include coding theory, information theory and applications to speech, audio and image coding. He has authored a textbook on information theory (in Russian) and has more than 70 papers published in journals and proceedings of international conferences, 15 U.S. patents and published patent applications in image, speech and audio coding.

Professor Kudryashov served as a member of Organizing Committee of ACCT International Workshops.

Roman V. Satyukov received the B.Sc. and M.Sc. degrees in applied mathematics and informatics from the Saint-Petersburg State University of Information Technologies, Mechanics and Optics, Russia, in 2007 and 2009, respectively.

From March 2009 to December 2009 he was with the Special Technical Center, Saint-Petersburg, Russia, as a software engineer. From February 2010 to May 2011 he was with the Saint-Petersburg State University of Information Technologies, Mechanics and Optics, Russia, teaching number theory. From January 2010 to March 2011 he was with the DevExperts, Saint-Petersburg, Russia, as a Java software engineer. Since June 2011 he has been with Google Switzerland, Zürich, Switzerland as a software engineer. His interests has been focused on information theory, number theory, and cryptography.