Some Voltage Graph-Based LDPC Tailbiting Codes with Large Girth

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

Published in:
[Host publication title missing]

DOI:
10.1109/ISIT.2011.6034230

2011

Some Voltage Graph-Based LDPC Tailbiting Codes with Large Girth

Irina E. Bocharova[1], Florian Hug[2], Rolf Johannesson[2], Boris D. Kudryashov[1], and Roman V. Satyukov[1]

[1] Dept. of Information Systems
St. Petersburg Univ. of Information Technologies, Mechanics and Optics
St. Petersburg 197101, Russia
Email: {irina, boris}@eit.lth.se, satyukov@gmail.com

[2] Dept. of Electrical and Information Technology, Lund University
P. O. Box 118, SE-22100 Lund, Sweden
Email: {florian, rolf}@eit.lth.se

***Abstract*—The relation between the parity-check matrices of quasi-cyclic (QC) low-density parity-check (LDPC) codes and the biadjacency matrices of bipartite graphs supports searching for powerful LDPC block codes. Algorithms for searching iteratively for LDPC block codes with large girth are presented and constructions based on Steiner Triple Systems and short QC block codes are introduced, leading to new QC regular LDPC block codes with girth up to 24.**

I. Introduction

The connection between low-density parity-check (LDPC) codes and codes based on graphs (see, for example, [1]) opens new perspectives in searching for powerful LDPC codes.

Typically, LDPC codes have minimum distances which are less than those for the best known linear codes, but due to their structure they are suitable for low-complexity iterative decoding, like the believe-propagation algorithm. One important parameter determining the efficiency of iterative decoding algorithms for LDPC codes is the *girth*, which is a parameter of the underlying Tanner graph and corresponds to the number of independent decoding iterations [2].

In this paper we shall focus on quasi-cyclic (QC) (J, K)-regular LDPC codes, which can be encoded in linear time and are most suitable for algebraic design. Such codes are commonly constructed based on combinatorial approaches using either finite geometries [3] or Steiner Triple Systems [4]. 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 [5]. This motivates searching for good short QC LDPC codes.

The problem of finding QC LDPC codes with large girth was considered in several papers. For example, codes with girth 14 are constructed in [6] while codes with girth up to 18 are presented in [7]. Most papers combine some algebraic techniques and computer search. Commonly these procedures start by choosing a proper base matrix or base graph (seed graph [8] or protograph [9]). Then a system of inequalities with integer coefficients describing all cycles of a given length is constructed and suitable labels or degrees are derived.

In Section II, we introduce notations for parity-check matrices of convolutional codes and for their corresponding tailbiting block codes. Section III focuses on bipartite graphs, biadjacency matrices, and their relation with parity-check matrices of LDPC block codes. Our construction of base and voltage matrices, used when we search for LDPC block codes with large girth, is introduced in Section IV. New search algorithms are presented in Section V. In Section VI new examples of (J, K)-regular QC LDPC codes with girth between 14 and 24 based on Steiner Triple Systems and small QC regular matrices are tabulated. Section VII concludes the paper with some final remarks.

II. Parity-Check Matrices

A rate $R = b/c$ binary convolutional code $\mathcal{C}$ is determined by its parity-check matrix of memory m

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

with parity-check polynomials $h_{ij}(D)$. In the sequel we consider parity-check matrices with either zero or monomial entries $h_{ij}(D) = D^{w_{ij}}$ of degree w_{ij}, where w_{ij} are nonnegative integers. If each column and each row contain exactly J and K nonzero elements, respectively, we call $\mathcal{C}$ a (J, K)-*regular* LDPC convolutional code.

Expressing the $(c - b) \times c$ parity-check matrix $H(D)$ in terms of its binary matrices H_i, $i = 0, 1, \dots, m$, that is,

$$H(D) = H_0 + H_1 D + H_2 D^2 + \cdots + H_m D^m \quad (2)$$

we obtain its semi-infinite syndrome former

$$H^T = \begin{pmatrix} H_0^T & H_1^T & \dots & & H_m^T & \\ & H_0^T & H_1^T & \dots & & H_m^T \\ & & \ddots & \ddots & & & \ddots \end{pmatrix} \quad (3)$$

where T denotes transpose.

If we tailbite the convolutional code $\mathcal{C}$ to length M c-tuples, where $M > m$, we obtain the $M(c - b) \times Mc$ parity-check matrix of the quasi-cyclic (QC) block code $\mathcal{B}$ as

$$H_{\text{TB}}^T = \begin{pmatrix} H_0^T & H_1^T & \cdots & H_{m-1}^T & H_m^T & \mathbf{0} \\ \mathbf{0} & H_0^T & H_1^T & \cdots & H_{m-1}^T & H_m^T \\ H_m^T & \mathbf{0} & H_0^T & H_1^T & \cdots & H_{m-1}^T \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ H_1^T & \cdots & H_{m-1}^T & H_m^T & \mathbf{0} & H_0^T \end{pmatrix} \quad (4)$$

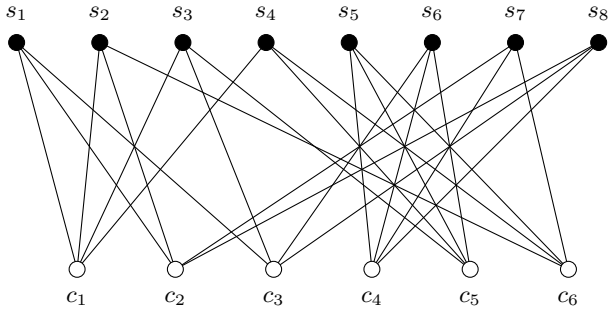

Fig. 1. Tanner graph with 8 symbol nodes (s_i, $i = 1, 2, \ldots, 8$) and 6 constraint nodes (c_i, $i = 1, 2, \ldots, 6$).

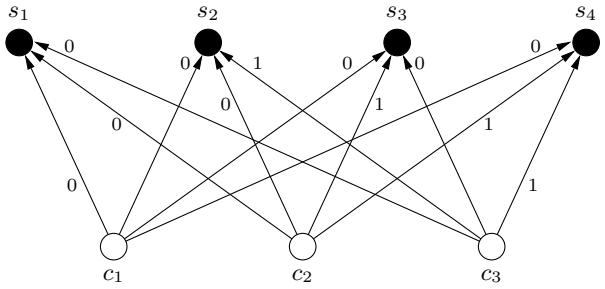

Fig. 2. Bipartite voltage graph with 4 symbol nodes (s_i, $i = 1, 2, 3, 4$) and 3 constraint nodes (c_i, $i = 1, 2, 3$).

Note that every cyclic shift of a codeword of $\mathcal{B}$ by c places modulo Mc is again a codeword.

The parity-check matrix H_{TB} is also (J, K)-regular, that is, there are exactly J ones in every column and exactly K ones in every row. Moreover, with J and K being much smaller than M, the matrix H_{TB} is sparse.

III. Graphs & 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.

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

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

Every full-rank parity-check matrix H of a rate $R = k/n$ LDPC block code can be interpreted as the *biadjacency matrix* [10] of a bipartite graph, the so-called *Tanner graph*, 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*. Note that, if the underlying LDPC block code is (J, K)-regular, all symbol and constraint nodes have degree J and K, respectively.

Consider the Tanner graph of 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. By letting the tailbiting length M tend to infinity, we obtain a convolutional parity-check matrix $H(D)$ as given in (1) of the parent convolutional code $\mathcal{C}$. In terms of Tanner graph representations, this corresponds to unwrapping the underlying graph and extending it in the time domain towards infinity. Hereinafter, we will denote the girth of this infinite Tanner graph as the *free girth* g_{free}.

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

$$H(D) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & D & D \\ 1 & D & 1 & D \end{pmatrix} \tag{5}$$

Tailbiting (5) to length $M = 2$, we obtain the tailbitten 6×8 parity-check matrix of a QC $(3, 4)$-regular LDPC block code

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

In particular, every cyclic shift of a codeword by $c = 4$ places modulo $Mc = 8$ is again a codeword. Interpreting (6) 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 $g = 4$. In this case, the free girth coincides with the girth, that is, $g_{\text{free}} = g = 4$.

IV. Base Matrices, Voltages, & 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

$$B = H(D)\big|_{D=1} \tag{7}$$

which corresponds to all nonzero entries in $H(D)$ being replaced by $D^0 = 1$. Note, that different 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 by g_{B}. The terminology "base graph" originates from graph theory and is used, for example, in [11]. It differs from the terminology used in [6], [9], where *protograph* or *seed graph* are used.

Let $\Gamma = \{\gamma\}$ be an additive group. From the base graph $\mathcal{G}_{\text{B}} = \{\mathcal{E}_{\text{B}}, \mathcal{V}_{\text{B}}\}$ we obtain the *voltage graph* [12], [13] $\mathcal{G}_{\text{V}} = \{\mathcal{E}_{\text{B}}, \mathcal{V}_{\text{B}}, \Gamma\}$ by assigning a voltage value $\gamma(e, v, v')$ to the edge e connecting the vertices v and v', satisfying the property $\gamma(e, v, v') = -\gamma(e, v', v)$. Note that, 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* as the sum of the voltages of its edges.

Let $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$ be a *lifted graph* obtained from a voltage graph, where $\mathcal{E} \subset \mathcal{E}_{\mathrm{B}} \times \Gamma$ and $\mathcal{V} = \mathcal{V}_{\mathrm{B}} \times \Gamma$. 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 $\mathcal{G}_{\mathrm{V}}$ 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. Note that a voltage assignment corresponds directly to selecting the degrees of the parity-check monomials in $H(D)$.

We describe LDPC convolutional codes using integer edge voltages, that is, an infinite additive voltage group, whereas QC LDPC are described using a voltage group of modulo M residues. The edge voltage from the constraint node c_i to the symbol node s_j is denoted by μ_{ij} while the corresponding edge voltage for the opposite passing direction from symbol node s_j to constraint node c_i is denoted by $\bar{\mu}_{ji}$, that is,

$$\mu_{ij} = -\bar{\mu}_{ji} = w_{ij} \mod M \tag{8}$$

where w_{ij} is the degree of the parity-check monomial $h_{ij}(D)$. Thus, using voltage graphs allows a compact description of LDPC codes and finding their (free) girth g_{V} (g_{free}) is reduced to finding their shortest cycle with voltage zero.

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 (8), Fig. 2 corresponds to a voltage graph with girth $g_{\mathrm{V}} = 4$ (for example, $s_1 \to c_1 \to s_2 \to c_2 \to s_1$). The edge from, for example, constraint node c_2 to symbol node s_3 is labeled according to

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

As the free girth of the infinite Tanner graph is equal to the girth of the voltage graph, we can conclude that $g_{\mathrm{free}} = g_{\mathrm{V}} = 4$. If we neglect all edge labels, we would obtain the corresponding base graph.

V. New Search Algorithms

When searching for QC (J, K)-regular LDPC block codes with large girth, we start from a base graph and determine a suitable voltage assignment based on nonnegative integers, such that the girth of this voltage graph is greater than or equal to a given girth g. Next we replace all edge labels by their modulo M residuals, where we try to minimize M while preserving the girth g. Using the duality between the edge voltages and the degree of the monomial entries in $H(D)$, we obtain the corresponding parity-check matrix of a convolutional code whose bipartite graph has girth $g = g_{\mathrm{free}}$. Tailbiting to lengths M, leads to 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:

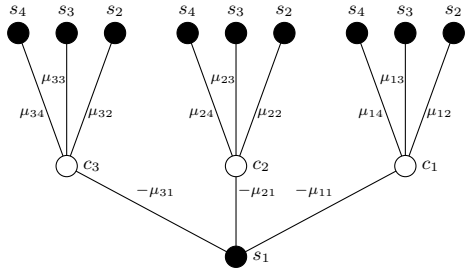

Fig. 3. A tree representation with maximum depth two, starting with symbol node s_1.

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.

In general, when searching for all cycles of length g, roughly $(J-1)^g$ different paths have to be considered. However, using a similar approach as in [14] we can reduce the complexity to roughly $(J-1)^{g/2}$ by using a tree representation.

A. Creating a tree structure

Consider the bipartite base graph of a $(c-b) \times c$ base matrix and denote the set of c symbol nodes s_i, $i = 1, 2, \ldots, c$, by $\mathcal{V}_0$ and the set of $c-b$ constraint nodes c_i, $i = 1, 2, \ldots, c-b$, by $\mathcal{V}_1$. A node in the tree will be referred to as ξ and has a unique parent node ξ^{p}. 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.

Next we grow c separate subtrees with the root node $\xi = \xi_{i,\mathrm{root}}$ of the ith subtree being initialized with $\xi \in \mathcal{V}_0$ and depth $\ell(\xi) = 0$. We extend every node $\xi \in \mathcal{V}_i$ at depth $\ell(\xi) = n$ with $i = n \mod 2$ by connecting it with the nodes $\xi' \in \mathcal{V}_{i+1 \mod 2}$ 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 (8) and obtain the voltage for node ξ in the ith subtree as the sum of the edge voltages of the path $\xi_{i,\mathrm{root}} \to \xi$.

All c subtrees contain all paths of a given length of the voltage graph. As the girth g is always even, we 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 $(g-2)/2$ and to construct voltage inequalities for all node pairs (ξ, ξ') in the same subtree with the same number $n(\xi) = n(\xi')$ and depth $\ell(\xi) = \ell(\xi')$ but different parent nodes $\xi^{\mathrm{p}} \neq \xi'^{\mathrm{p}}$.

Consider the node pair (ξ, ξ') and let $f_{\xi_{i,\mathrm{root}},\xi,\xi'}$ denote the difference between the voltages for the path from $\xi_{i,\mathrm{root}}$ to ξ and the path from $\xi_{i,\mathrm{root}}$ to ξ', that is, $f_{\xi_{i,\mathrm{root}},\xi,\xi'} = (\xi_{i,\mathrm{root}} \to \xi) - (\xi_{i,\mathrm{root}} \to \xi')$. If there exists a cycle of length $g' < g$, then at depth $g'/2$ there exists at least one node pair (ξ, ξ'), whose corresponding path voltages are equal, that is, their voltage difference is $f_{\xi_{i,\mathrm{root}},\xi,\xi'} = 0$. Otherwise there is no cycle shorter than g.

Example 2: Consider the rate $R = 1/4$ $(3, 4)$-regular LDPC convolutional code given by (5). The voltage 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. 2. By neglecting all labels, we obtain the corresponding base graph.

Starting from such a base graph, we will find suitable edge voltages for μ_{ij}, $i = 1, 2, 3$, $j = 1, 2, 3, 4$, such that the resulting voltage graph has at least girth $g = 6$. As a first step we grow 4 subtrees up to length $(g - 2)/2 = 2$, with their root nodes being initialized by s_i, $i = 1, 2, 3, 4$. For example, the subtree with root node s_1 is illustrated in Fig. 3.

While there are no identical nodes at depth $\ell(\xi) = 1$, we find $3 \times \binom{3}{2} = 9$ pairs of identical nodes with different parents at depth $\ell(\xi) = 2$. In all four subtrees, there are in total 36 identical node pairs, but only 18 unique ones.

B. Searching for a suitable voltage assignment

Using the c obtained subtrees $\mathcal{T}_i$, $i = 1, 2, \ldots, c$, with depth $g/2 - 1$, we will present hereinafter two different algorithms to determine a suitable voltage assignment, such that all corresponding inequalities are satisfied.

For both algorithms, we create 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. Note that even different cycles can correspond to the same voltage inequality. In a similar manner we remove those nodes from each of the c subtrees $\mathcal{T}_i$ which do not participate in any cycle listed in $\mathcal{L}$ and denote the reduced subtree by $\mathcal{T}_{i,\min}$.

In *Algorithm A*, we label the edges of the reduced subtrees $\mathcal{T}_{i,\min}$, $i = 1, 2, \ldots, c$, with a set of predetermined voltages. For every node pair (ξ, ξ') in $\mathcal{L}$, we determine 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 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 focus on the reduced subtrees $\mathcal{T}_{i,\min}$. After labeling their edges with a set of predetermined voltages, we sort all 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 $\xi^{\mathsf{p}} \neq \xi'^{\mathsf{p}}$, the girth of the underlying base graph with such a voltage assignment is greater than or equal to g.

C. Complexity Comparison

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,

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

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 $N_T + N_L$. Algorithm B requires the same number of N_T summations for computing the path voltages, roughly $N_T \log_2 N_T$ operations for sorting the set, and N_T comparisons, leading to a total complexity estimate of $N_T \log_2 N_T$.

TABLE I
COMPLEXITY OF SEARCHING FOR VOLTAGE ASSIGNMENTS FOR QC LDPC BLOCK CODES WITH GIRTH $g \leq 12$ AND ALL-ONES BASE MATRIX

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

TABLE II
PROPERTIES OF QC LDPC CODES WITH GIRTH $g = 14$–18

K	g	(n, k)	M	Base graph
4	14	$(1812, 453)$ $\big((2208, 552)$ [7]$\big)$	151 $\big(184$ [7]$\big)$	(9×12) STS(9)
5	14	$(9720, 3888)$ $\big((11525, 4610)$ [7]$\big)$	486	(12×20) S-STS(13)
6	14	$(29978, 14989)$ $\big((37154, 18577)$ [7]$\big)$	1153 $\big(1429$ [7]$\big)$	(13×26) STS(13)
4	16	$(7980, 1995)$ $\big((7488, 1872)$ [7]$\big)$	665 $\big(624$ [7]$\big)$	(9×12) STS(9)
5	16	$(51240, 20496)$ $\big((62500, 25000)$ [7]$\big)$	2562	(12×20) S-STS(13)
6	16	$(227032, 113516)$ $\big((229476, 114738)$ [7]$\big)$	8732 $\big(8826$ [7]$\big)$	(13×26) STS(13)
4	18	$(32676, 8169)$ $\big((34260, 8565)$ [7]$\big)$	2723 $\big(2855$ [7]$\big)$	(9×12) STS(9)
5	18	$(271760, 108704)$ $\big((371100, 148440)$ [7]$\big)$	13588	(12×20) S-STS(13)

In Table I values of N_T and N_L are given when searching for a voltage assignment of a rate $R = 1 - J/K$ (J, K)-regular QC LDPC convolutional code with an all-ones base matrices, $J = 3$ and arbitrary $K \geq 4$. Since in general we have to consider all node pairs, N_L is roughly N_T^2, and thus Algorithm B performs asymptotically better (when $N_T \to \infty$). However, when searching for codes with girth $g \leq 10$, Algorithm A is preferable.

VI. SEARCH RESULTS

Utilizing the previously described algorithms, we performed a search for new QC $(J = 3, K)$-regular LDPC block codes with girth $g \geq 14$. Following [15], such codes can be constructed as lifts of base matrices with monomial labelings, having an approximately three times larger girth.

A. Base Matrices constructed from Steiner Triple Systems

We started by searching for QC $(J = 3, K)$-regular LDPC block codes with girth $g = 14, 16$, and 18 and used (shortened) base matrices constructed from Steiner Triple Systems of order n, that is, STS(n) [4], [9], where $n \mod 6$ has to be equal to 1 or 3.

The corresponding (J, K)-regular $(c - b) \times c$ base matrix B with entries b_{ij} is constructed in such a way that the positions of its nonzero entries in each column correspond to a triple within STS$(c - b)$. We denote such a base matrix by

TABLE III
PROPERTIES OF QC LDPC CODES WITH GIRTH $g \geq 20$

K	g	(n,k)	M	Base graph
4	20	$(1296000, 324002)$	36000	(27×36), $g = 8$ [17]
5	20	$(31200000, 12480002)$	480000	(39×65), $g = 8$ [17]
6	20	$(518400000, 259200002)$	4800000	(54×108), $g = 8$ [17]
4	22	$(7200000, 1800002)$	200000	(27×36), $g = 8$ [7]
5	22	$(325000000, 130000002)$	5000000	(39×65), $g = 8$ [17]
4	24	$(39600000, 9900002)$	1100000	(27×36), $g = 8$ [17]

$B_{\text{STS}(c-b)}$. Note that the columns and rows of the base matrix B can be freely permuted. Using the properties of a Steiner Triple System that no triple contains two identical numbers, we obtain a shortened $(c-b-1) \times (c-K)$ $(J, K-1)$-regular base matrix B' by removing one row and the corresponding K columns from the base matrix B. Hereinafter we refer to such a shortened base matrix as $B_{\text{S-STS}(c-b)}$. Note that by deleting different columns and rows, it is also possible to obtain intermediate codes, which are, however, irregular.

Applying the previously described algorithms to such a base matrix B, we obtain a suitable voltage assignment, such that the corresponding voltage graph has at least girth g. Note that adding the same offset to all edge voltages connected to the same vertex, does not influence the voltage of any cycles.

In Table II the obtained QC $(J = 3, K)$-regular LDPC block codes with girth $g = 14, 16$, and 18 based on Steiner Triple Systems are presented. If applicable, previous results from [7] are given for comparison. In the first column K we give the number of nonzero elements in each row. The second column corresponds to the girth g, while the third and forth columns give the dimensions of the (n, k) block code after tailbiting to length M. Finally, the fifth column specifies the used base matrix, that is, which (maybe shortened) Steiner Triple System is used. The corresponding voltage assignments are very large and omitted due to space limitations, but are available at [16].

B. Base Matrices constructed from (J, K)-regular LDPC block codes

When searching for QC $(J = 3, K)$-regular LDPC block codes with girth $g = 20 - 24$, we started with previously obtained QC $(J = 3, K)$-regular LDPC block codes of short block length 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 III, based on $(J = 3, K)$-regular LDPC block codes constructed from all-ones matrices with girth $g = 8$ [7], [17]. As before, the first column K denotes the number of nonzero elements in each row; then we give the obtained girth g and the dimensions of the (n, k) block code after tailbiting to length M. The corresponding voltage assignments are very large and omitted, but are available at [16].

Note that these codes are (probably) not practical due to their huge block length. However, they illustrate that by iteratively applying our algorithms we can find QC (J, K)-regular LDPC block codes of "any" girth g.

VII. CONCLUSIONS

Using the relation between the parity-check matrix of QC LDPC block codes and the biadjacency matrix of bipartite graphs, new searching 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.

New algorithms for searching iteratively for bipartite graphs with large girth have been presented. Depending on the given girth, the search algorithms are either based on Steiner Triple Systems or QC block codes. Amongst others, new QC regular LDPC block codes with girth between 14 and 24 have been presented. In particular, these codes improve previous the published results in [7].

ACKNOWLEDGEMENTS

This research was supported in part by the Swedish Research Council under Grant 621-2007-6281.

REFERENCES

[1] G. Schmidt, V. V. Zyablov, and M. Bossert, "On expander codes based on hypergraphs," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT'03)*, Yokohama, Japan, p. 88.

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

[3] 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.

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

[5] 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.

[6] 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.

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

[8] R. M. Tanner, "On Graph Constructions for LDPC Codes by Quasi-Cyclic Extension," in *Information, Coding and Mathematics*, M. Blaum, P. G. Farrel, and H. C. A. van Tilborg, Eds. Norwell, MA: Kluwer, 2002, pp. 209–219.

[9] J. Thorpe, K. Andrews, and S. Dolinar, "Methodologies for designing LDPC codes using protographs and circulants," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT'04)*, Chicago, USA, p. 238.

[10] A. S. Asratian, T. M. J. Denley, and R. Haggkvist, *Bipartite Graphs and Their Applications*. Cambridge University Press, 1998.

[11] C. A. Kelley and J. L. Walker, "LDPC codes from voltage graphs," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT'08)*, Toronto, Canada.

[12] J. L. Gross, "Voltage graphs," *Discrete Mathematics*, vol. 9, no. 3, pp. 239–246, 1974.

[13] G. Exoo and R. Jajcay, "Dynamic cage survey," *The Electronic Journal of Combinatorics*, vol. 15, Sep. 2008.

[14] 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.

[15] 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 Int. Symp. Inform. Theory (ISIT'10)*, Austin, Texas, pp. 819–823.

[16] Base matrices and their voltage assignments. [Online]. Available: http://www.eit.lth.se/goto/QC_LDPC_Codes

[17] I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, and R. V. Satyukov, "Searching for voltage graph-based LDPC tailbiting codes with large girth," *IEEE Trans. Inf. Theory*, submitted for publication.