2008

LDPC codes from voltage graphs

Christine A. Kelley
University of Nebraska-Lincoln, ckelley2@unl.edu

Judy L. Walker
University of Nebraska - Lincoln, judy.walker@unl.edu

LDPC codes from voltage graphs

Christine A. Kelley
Department of Mathematics
University of Nebraska-Lincoln
Lincoln, NE 68588, USA.
Email: ckelley2@math.unl.edu

Judy L. Walker
Department of Mathematics
University of Nebraska–Lincoln
Lincoln, NE 68588, USA.
Email: jwalker7@math.unl.edu

Abstract— **Several well-known structure-based constructions of LDPC codes, for example codes based on permutation and circulant matrices and in particular, quasi-cyclic LDPC codes, can be interpreted via algebraic voltage assignments. We explain this connection and show how this idea from topological graph theory can be used to give simple proofs of many known properties of these codes. In addition, the notion of abelian-inevitable cycle is introduced and the subgraphs giving rise to these cycles are classified. We also indicate how, by using more sophisticated voltage assignments, new classes of good LDPC codes may be obtained.**

I. Introduction

Graph-based codes have attracted widespread interest due to their efficient decoding algorithms and remarkable performance on several communication channels. However, theoretical results that give proof of when they are good and how to design them are hard to come by. Much work has focused on understanding the asymptotic performance of ensembles of these codes for block lengths tending to infinity. For practical implementation, the design of short to moderate length codes with algebraic structure is desired. Several researchers have proposed structure-based constructions of these codes and each of these constructions has aimed to optimize one or more properties in the resulting graph that intuitively improve the resulting code's performance, such as girth, expansion, diameter, stopping sets, or pseudocodewords. One area of recent interest is *protograph LDPC codes*, which are codes based on graphs obtained by taking random lifts of a suitably chosen base graph, or *protograph*. However, many of these constructions appear ad hoc and there is a serious lack of a mathematical theory in designing these graph-based codes. In this work, we aim to bridge this gap by unifying several different families of graph-based codes under one common framework—namely, codes on graphs arising as *voltage graphs*.

We consider a voltage graph viewpoint from topological graph theory wherein specific lifts of graphs are determined via "voltage assignments", i.e., assignments of elements of a so-called *voltage group*, to the edges of a base graph, thus making the lifting entirely algebraic. This algebraic characterization of lifts is a powerful tool for analyzing several graph properties of the resulting lifts using the properties of the base graph. In this paper, this tool is applied to codes that are amenable to voltage graph interpretation, and consequently, their graph properties are better understood.

The paper is organized as follows. We introduce some preliminary definitions and notation in Section II. In particular, we review the terminology of voltage graphs. In Section III, we show how graph-based codes can be obtained algebraically from voltage graphs, and illustrate this using the Sridhara-Fuja-Tanner (SFT) LDPC codes and array-based LDPC codes as examples. In Section IV, the notion of *abelian-inevitable* cycles is introduced and the isomorphism classes of subgraphs (or equivalently, matrix substructures) that give rise to these cycles in the LDPC Tanner graph are identified and classified. The results presented here correct a result used in [16], [20], and, subsequently, extend the work from [16]. Ongoing work addressing how this method may be used to construct new families of LDPC and other graph-based codes is outlined in Section V.

II. Preliminaries

A binary low-density parity-check (LDPC) code is defined by a sparse parity-check matrix H or, equivalently, by the incidence graph of H called the *Tanner graph*. The left and right vertices are called *variable* nodes and *check* nodes, respectively. The set of codewords is the set of all binary assignments to the variable nodes such that at each check node, the modulo two sum of the variable node assignments connected to that check node is zero. The notion of *covering* graphs (or *lifts* of graphs) enters into the analysis of the graph-based iterative decoder in the explanation of pseudocodewords [11], [14], [17], [25].

In an entirely different context, various researchers have looked at constructing families of LDPC codes by taking random lifts of a specially chosen base graph, or "protograph", yielding the so-called "protograph codes" [24], [6], [7], [19]. The idea exploited in these constructions is that the properties of the base graph may shed light on the properties of the covering graphs, and therefore on the resulting codes. Indeed, random lifts of graphs have been heavily studied (see, for example, [18], [1], [2]). While these codes have exhibited good performance, we believe that constructions using algebraically-designed lifts may outperform these random methods as well as provide a good handle on the code properties such as minimum distance, girth, stopping sets, and pseudocodewords.

An algebraic construction of specific covering spaces for graphs was introduced by Gross and Tucker in the 1970s [13]. Given a graph $X = (V_X, E_X)$ where each edge in X has a positive and negative orientation, a function α, called an *ordinary voltage assignment*, maps the positively oriented edges to elements from a chosen finite group G, called
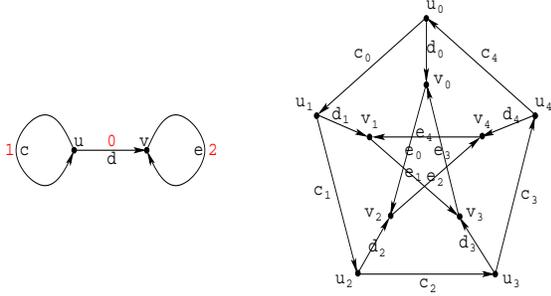
Fig. 1. A right-derived voltage graph with voltage group $A = \mathbb{Z}/5\mathbb{Z}$.

the *voltage group*. The negative orientation of each edge is assigned a voltage that is the inverse element of the voltage assigned to its positive orientation. The base graph X, together with the function α, is called an *ordinary voltage graph*. The values of α on the edges are referred to as *voltages*. A new graph X^α, called the *(right) derived graph*, is a degree $|G|$ lift of X and has vertex set $V_X \times G$ and edge set $E_X \times G$, where if (u, v) is a positively oriented edge in X with voltage $h \in G$, then there is an edge from (u, g) to (v, gh) in X^α for each $g \in G$. Figure 1, which is taken from [13], shows a base graph X with voltages assigned to its edges from the additive group of integers modulo 5 (i.e., $G = \mathbb{Z}/5\mathbb{Z}$), and the corresponding right-derived graph obtained from this assignment.

In the case that the voltage group is the symmetric group S_n on n elements, one can also view the pair (X, α) as a *permutation voltage graph*. The *permutation derived graph* X^α has vertex set $V_X \times \{1, \ldots, n\}$ and edge set $E_X \times \{1, \ldots, n\}$. If $\pi \in S_n$ is a permutation voltage on the edge $e = (u, v)$ of X, then there is an edge from (u, i) to $(v, \pi(i))$ in X^α for $i = 1, 2, \ldots, n$. Note that X^α is a degree n lift of X rather than a degree $n!$ lift as it would be if viewed as an ordinary derived graph as discussed above.

For an edge e, let e^- and e^+ denote the negative and positive orientations, respectively, of e. A walk in the voltage graph X with voltage assignment α may be represented by the sequence of oriented edges as they are traversed, e.g. $W = e_1^{\sigma_1} e_2^{\sigma_2} \ldots e_n^{\sigma_n}$ where each σ_i is $+$ or $-$ and $e_1, \ldots, e_n$ are edges in G. In this setting, the *net voltage* of the walk W is defined as the voltage group product

$$\alpha(e_1^{\sigma_1})\alpha(e_2^{\sigma_2}) \ldots \alpha(e_n^{\sigma_n})$$

of the voltages on the edges of W in the order and direction of the walk.

For example, the walk $W = d^+ e^- d^- c^+$ in the voltage graph of Figure 1 has net voltage $0 + (-2) + (-0) + 1 = -1 = 4 \in \mathbb{Z}/5\mathbb{Z}$.

The following theorem from [13] will be useful to us.

Theorem 2.1: Let W be a walk in a voltage graph X with initial vertex v. Then for each vertex (v, g) in X^α for $g \in G$, there is a unique walk W_g in X^α that starts at (v, g) and projects down to W. Assume $W = e_1^{\sigma_1} e_2^{\sigma_2} \cdots e_n^{\sigma_n}$ is closed, backtrackless, and tailless. Then W_g, for any $g \in G$, is a cycle on X^α if and only if the net voltage of W is the identity of G.

Voltage graphs have been successfully used to obtain many instances of graphs with extremal properties; see [9], [4], [5], [3].

III. CODES DESCRIBED USING VOLTAGE TERMINOLOGY

We now describe three popular families of quasi-cyclic LDPC codes proposed in [22], [23], [10] and [8], respectively, and show how the codes can be interpreted via voltage graphs.

A. SFT codes [22], [23]

For a prime q, the integers $\{0, 1, \ldots, q-1\}$ form a field under addition and multiplication modulo q, i.e., the Galois field $\mathbb{F}_q$. The non-zero elements of $\mathbb{F}_q$ form a cyclic multiplicative group $\mathbb{F}_q^\times$ of order $q - 1$. Let j and k be distinct divisors of $q - 1$ and let a and b be elements of $\mathbb{F}_q^\times$ with orders $o(a) = k$ and $o(b) = j$, respectively. Form the $j \times k$ matrix P over $\mathbb{F}_q$ that has as its (s, t) entry $P_{s,t} = b^{(s-1)}a^{(t-1)}$, for $1 \le s \le j$ and $1 \le t \le k$.

The LDPC code is constructed by specifying its parity check matrix H. Specifically, H is made up of a $j \times k$ array of circulant sub-matrices as shown below:

$$H = \begin{bmatrix} I_1 & I_a & I_{a^2} & \ldots & I_{a^{k-1}} \\ I_b & I_{ab} & I_{a^2 b} & \ldots & I_{a^{k-1}b} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ I_{b^{j-1}} & I_{ab^{j-1}} & I_{a^2 b^{j-1}} & \ldots & I_{a^{k-1}b^{j-1}} \end{bmatrix}. \quad (1)$$

where I_x is a $q \times q$ identity matrix with rows cyclically shifted to the left by x positions. The circulant sub-matrix in position (s, t) within H is obtained by cyclically shifting the rows of the identity matrix to the left by $P_{s,t}$ places. The resulting binary parity check matrix is of size $jq \times kq$, which means the associated code has a rate $R \ge 1 - (j/k)$. The codes constructed using this technique are quasi-cyclic with period k, i.e., cyclically shifting a codeword by one position within each of the k blocks of circulant sub-matrices (where each block consists of q code bits) results in another codeword.

Example 3.1: A [155,64,20] SFT code ($q = 31$): Set $q = 31$, $k = 5$ and $j = 3$. Then we may take $a = 2$, $b = 5$ in $\mathbb{F}_{31}^\times$ to have $o(a) = k$ and $o(b) = j$, and the parity check matrix is given by

$$H = \begin{bmatrix} I_1 & I_2 & I_4 & I_8 & I_{16} \\ I_5 & I_{10} & I_{20} & I_9 & I_{18} \\ I_{25} & I_{19} & I_7 & I_{14} & I_{28} \end{bmatrix}_{(93 \times 155)},$$

where I_x is a 31×31 identity matrix with rows shifted cyclically to the left by x positions.

As shown in Figure 2 and in [15], the Tanner graph of this code may be viewed as the derived graph arising from a permutation voltage assignment on the complete bipartite graph $K_{3,5}$ on 3 right and 5 left nodes, where the voltage assignments come from the symmetric group S_{31} on 31 elements, and the voltages are permutation elements that yield the shifts as given in the construction. For example, the entry $I_{2^s 5^t}$ in the parity-check matrix corresponds to the

edge between the sth left node, $s = 0, \ldots, 4$, and the tth right node, $t = 0, 1, 2$ in the base graph $K_{3,5}$ and a voltage equal to a permutation element that yields a circulant shift of $2^s 5^t \pmod{31}$. Thus, the circulant matrix I_1 corresponds to the voltage element $(1\ 2\ 3\ \ldots\ 31) \in S_{31}$ assigned to the edge from the 0th left node to the 0th right node in $K_{3,5}$. The circulant I_2 corresponds to the element (voltage) $(1\ 3\ 5\ 7\ 9\ \ldots\ 31\ 2\ 4\ 6\ \ldots\ 30) \in S_{31}$ assigned to the edge from the 1st left node to the 0th right node in $K_{3,5}$ and so on.

$K_{3,5}.$
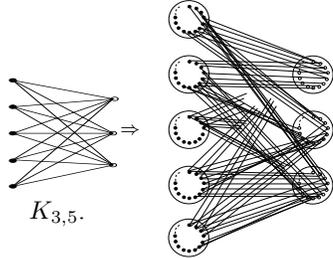
Tanner graph of an SFT code.

Fig. 2. An SFT code viewed as a voltage graph. Note that the graph on the right is a schematic, as a true representation would have 155 variable nodes, 93 check nodes and 465 edges.

B. Array-based LDPC codes

Array-based LDPC codes were introduced in [10]. We present the construction and show how these codes can be interpreted as permutation voltage derived graphs in a straightforward manner.

For a prime q and positive integer $j \leq q$, the parity-check matrix of the array code is defined by

$$H(q,j) = \begin{bmatrix} I & I & I & \ldots & I \\ I & P & P^2 & \ldots & P^{(q-1)} \\ I & P^2 & P^4 & \ldots & P^{2(q-1)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ I & P^{(j-1)} & P^{(j-1)2} & \ldots & P^{(j-1)(q-1)} \end{bmatrix},$$

where I is the $q \times q$ identity matrix and $P = I_{q-1}$ is the $q \times q$ identity matrix cyclically shifted to the right by one position. The array code is quasi-cyclic with block length $N = q^2$ and rate $R \geq 1 - \frac{j}{q}$.

The Tanner graph of the above code may be obtained via a voltage assignment on the permutation voltage graph $K_{j,q}$ which is the complete bipartite graph on q left vertices and j right vertices. The edge that connects the sth left node to the tth right node in $K_{j,q}$, for $0 \leq s \leq q-1$ and $0 \leq t \leq j-1$, is the permutation π^{ts}, where $\pi = (1\ 2\ 3\ \ldots\ q)$ is the permutation in the symmetric group S_q and π^{ts} is the permutation π applied recursively ts times.

Eleftheriou, et al., proposed [8] a modification to the above construction in order to obtain an efficient encoding. The modified array code is defined by designing a parity-check matrix $H(q, j, k)$, for $q \leq j < k$, as follows:

$$H(q,j,k) = \begin{bmatrix} I & I & I & \ldots & I & \ldots & I \\ 0 & I & P & \ldots & P^{(j-2)} & \ldots & P^{(k-2)} \\ 0 & 0 & I & \ldots & P^{2(j-3)} & \ldots & p^{2(k-3)} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & 0 & I & \ldots & P^{(j-1)(k-j)} \end{bmatrix},$$

where I is the $q \times q$ identity matrix and P is the $q \times q$ identity matrix cyclically shifted to the right by one position. The above code is an irregular quasi-cyclic code with block length $N = qk$ and rate $R = 1 - \frac{j}{k}$.

The Tanner graph of the above code may be obtained via a voltage assignment on the permutation voltage graph $Q_{j,k}$ which is the bipartite graph on k left vertices X and j right vertices Y and having edges (s,t), $0 \leq s \leq k - 1$, $0 \leq t \leq j - 1$ and satisfying $s \in X$, $t \in Y$, $s \geq t$. The edge that connects the sth left node to the tth right node in $Q_{j,k}$, for $0 \leq s \leq k - 1$, $0 \leq t \leq j - 1$ and $s \geq t$, is the permutation $\pi^{t(s-t)}$, where $\pi = (1\ 2\ 3\ \ldots\ q)$ is the permutation in the symmetric group S_q.

Remark 3.2: The permutation voltages assigned to the edges of the base graph in the above constructions belong to an abelian subgroup of the symmetric group S_q. More general assignments of these voltages may yield codes that have girth and minimum distance not limited by the upper bounds (see Section IV) imposed by using abelian voltage groups.

C. Other constructions

One other notable construction of quasi-cyclic LDPC codes is by Song, et al., [21] where the parity-check matrix is composed of blocks of circulant matrices that are not necessarily shifted identity matrices, but rather matrices obtained by superimposing shifted identity matrices. These are also amenable to the voltage graph interpretation where we allow multiple edges between pairs of vertices in the base graph.

IV. ABELIAN-INEVITABLE CYCLES

In [16], the authors identify matrix substructures that, in quasi-cyclic LDPC codes, give rise to so-called inevitable cycles. However, they use brute force methods and only generate a list of sub-matrices in the base graph parity-check matrix having up to ten ones that yield these inevitable cycles. In the following, we completely classify all such submatrices, or equivalently, subgraphs, that generate these cycles. We start by formalizing the notion of an inevitable cycle suggested in those papers by introducing the term *abelian-forcing walk*.

A sequence of vertices and edges $v_0 e_1 v_1 \ldots v_{n-1} e_n v_n$ on a graph is called a *closed walk* if the vertices v_{i-1} and v_i are the endpoints of the edge e_i and $v_n = v_0$. A closed walk $v_0 e_1 v_1 \ldots v_{n-1} e_n v_n$ is *backtrackless* if $e_i \neq e_{i+1}$ for $1 \leq i \leq n - 1$. A backtrackless closed walk $v_0 e_1 v_1 \ldots v_{n-1} e_n v_n$ is said to be *tailless* if $e_n \neq e_1$. A backtrackless, tailless closed walk W is *abelian-forcing* if for each edge in W, the number of traversals of that edge in the positive direction is the same as that in the negative direction.

Lemma 4.1: An abelian-forcing walk W on X has net voltage 0 for any voltage assignment α to any abelian voltage group G. Hence for each $g \in G$, the lift W_g of W in X^α is a cycle of length $|W|$.

   Proof: Since the voltage assigned to a negatively-oriented edge is the inverse of the voltage of the corresponding positively-oriented edge, the first statement is clear. The second statement is immediate from Theorem 2.1. ∎

We define U to be an *abelian-forcing graph* if there is an abelian-forcing walk on U which uses every edge of U.

Definition 4.2: Let X be a graph. A positive integer n is an *abelian-inevitable cycle length* for X if, for every abelian group G and every voltage assignment α of G on X, the derived graph X^α must have a simple cycle of length n.

The relationship between abelian-forcing walks and abelian-inevitable cycle lengths is given by the next lemma, the proof of which is immediate from definitions and Lemma 4.1 above.

Lemma 4.3: If X has an abelian-forcing walk of length n, then n is an abelian-inevitable cycle length for X.
For the classification, we will need terminology for two main types of subgraphs. Define an (a,b,c)-*theta graph*, denoted by $T(a,b,c)$, to be a graph consisting of two vertices v and w, each of degree three, that are connected to each other via three disjoint paths A, B, C of (edge) lengths $a \geq 1$, $b \geq 1$, and $c \geq 1$, respectively, and define a $(a_1, a_2; b)$-*dumbbell graph*, denoted $D(a_1, a_2; b)$ to be a connected graph consisting of two edge-disjoint cycles A_1 and A_2 of lengths $a_1 \geq 1$ and $a_2 \geq 1$, respectively, that are connected by a path B of length $b \geq 0$. In the case that $b = 0$, we get a bouquet of two circles, which we refer to as a *degenerate* dumbbell graph.

Note that if $T(a,b,c)$ is a subgraph of a simple bipartite graph with $a \geq b \geq c \geq 1$ then $b \geq 2$ and $a \equiv b \equiv c \pmod 2$. That is, a, b and c have the same parity. If $D(a_1, a_2; b)$ is a subgraph of a simple bipartite graph with $a_1, a_2 \geq 1$ and $b \geq 0$, then $a_1, a_2 \geq 4$ and are even. The next proposition is a generalization of Exoo's "Observation 1" [9].

Proposition 4.4:
   1) If X contains an (a,b,c)-theta graph, then $2(a + b + c)$ is an abelian-inevitable cycle length for X.
   2) If X contains an $(a_1, a_2; b)$-dumbbell graph, then $2(a_1 + a_2) + 4b$ is an abelian-inevitable cycle length for X.

   Proof: Suppose X contains an (a,b,c)-theta graph with paths A, B and C. Then $AB^{-1}CA^{-1}BC^{-1}$ is an abelian-forcing walk of length $2(a+b+c)$. Similarly, if X contains an $(a_1, a_2; b)$-dumbbell graph with cycles A_1 and A_2 connected by the path B, then $A_1 B A_2 B^{-1} A_1^{-1} B A_2^{-1} B^{-1}$ is an abelian-forcing walk of length $2(a_1 + a_2) + 4b$. The result now follows from Lemma 4.3. ∎

The utility of this voltage graph viewpoint may been seen when one analyzes the *girth* of the Tanner graph of the SFT codes. The girth is the length of the smallest cycle in the graph, and is important as it measures the number of iterations of decoding for which the messages passed along the graph remain independent. Indeed, iterative decoding is optimal on cycle-free graphs. It was shown in [23] that the $[155, 64, 20]$
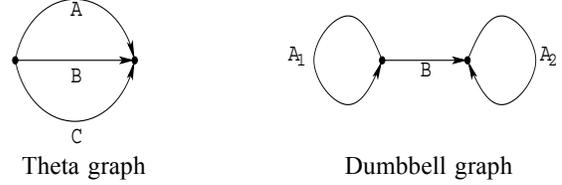

Fig. 3.   A theta graph and a dumbbell graph.

SFT code in Section III has girth 8 and, more generally, all codes in the family have girth at most 12. The base graphs in the SFT construction all contain the complete bipartite graph $K_{2,3}$ as a subgraph, and $K_{2,3}$ is the theta graph $T(2,2,2)$. Thus, Proposition 4.4 immediately gives that the girth of the Tanner graphs of the SFT codes is at most 12. This argument on girth is much more concise than other proof methods such as in [23], [12]. The next result classifies the subgraphs that give rise to abelian-inevitable cycles, namely the abelian-forcing graphs, and gives upper bounds on the girth of an abelian voltage graph in terms of these subgraphs.

Theorem 4.5: Suppose X contains an abelian-forcing subgraph and let α be any abelian voltage assignment for X. Then one of the following holds:

   1) X contains an (a,b,c)-theta graph for some a, b, $c \geq 1$, in which case the girth of X^α is at most $2(a+b+c)$.
   2) X contains an $(a_1, a_2; b)$-dumbbell graph for some a_1, $a_2 \geq 1$ and $b \geq 0$, in which case the girth of X^α is at most $2(a_1 + a_2) + 4b$.

   Proof: By Proposition 4.4, it is enough to prove that X must contain either a theta graph or a dumbbell graph. Without loss of generality, we may assume X is abelian-forcing. Let Γ and Δ be distinct simple cycles on X of lengths m and n respectively. Let I be the set of vertices in $\Gamma \cap \Delta$. We consider three cases: $|I| = 0$, $|I| = 1$ and $|I| \geq 2$. If $|I| = 0$, let v be any vertex on Γ and let w be any vertex on Δ. Since X is connected, there is a path P in X from v to w. Write $P = P_1 P_2 P_3$, where $P_1 \subset \Gamma$, $P_3 \subset \Delta$ and P_2 shares edges with neither Γ nor Δ; note that one or both of P_1 and P_3 may be empty but P_2 is certainly nonempty. Then $\Gamma \cup P_2 \cup \Delta$ is the nondegenerate dumbbell graph $D(m, n; b)$, where $b \geq 1$ is the length of P_2. Next, assume $|I| = 1$. Then $\Gamma \cup \Delta$ is clearly the degenerate dumbbell graph $D(m, n; 0)$. Finally, assume $|I| \geq 2$, and write Γ and Δ in terms of their vertices $\Gamma = v_1, v_2, \ldots, v_m, v_1$ and $\Delta = w_1, w_2, \ldots, w_n, w_1$. Without loss of generality, we may assume $v_1 = w_1$ and there is some $j \geq 1$ such that $v_i = w_i$ for $i \leq j$ and $v_{j+1} \neq w_{j+1}$. If $j = |I|$, then $\Gamma \cup \Delta = T(a,b,c)$ with $a = j-1$, $b = m-j$ and $c = n-j$. If $j < |I|$, then let t be minimal such that $v_{j+t} = w_k$ for some $k \geq j+1$. Let A be the path $v_j, v_{j+1}, \ldots, v_{j+t}$; let B be the path $w_j, w_{j+1}, \ldots, w_k$; and, noting that $j < k$, let C be the path $w_j, w_{j-1}, \ldots, w_1, w_n, w_{n-1}, \ldots w_k$. Then $A \cup B \cup C$ is a $(t, k-j, j-1+n-k)$-theta graph. ∎

We note that Theorem 4 in [20] is incorrect; the proof assumes the overlaps are consecutive although the statement does not. Since Theorems 1 and 3 in [16] rely on this result,

they are incorrect as well. However, Theorem 1 in [16] is correct when rephrased as

Theorem 4.6: Let Γ and Δ be simple cycles with $r \geq 0$ consecutive edge overlaps and lengths $2k$ and 2ℓ, respectively. In the case that $r = 0$, assume further that Γ and Δ share at least one vertex. Then there is an inevitable cycle of length $2(2\ell + 2k - r)$ in the protograph code.

We now give a correct version of Theorem 3 of [16]:

Theorem 4.7: Let X be a graph of girth g. Then every abelian-inevitable cycle length for X is at least $3g$.

Proof: If X contains no abelian-forcing walks, there is nothing to prove. So suppose n is an abelian-inevitable cycle length for X. Then X, by Theorem 4.5, X has either a subgraph $T(a, b, c)$ with $2a + 2b + 2c \leq n$ or a subgraph $D(a_1, a_2; b)$ with $2(a_1 + a_2) + 4b \leq n$. If X contains a $T(a, b, c)$ subgraph, then $a + b \geq g$, $b + c \geq g$, and $a + c \geq g$. We have

$$g \leq a + c = (a + b) + (b + c) - 2b \qquad \text{and so}$$
$$-b \leq \frac{g}{2} - \frac{a+b}{2} - \frac{b+c}{2}.$$

Therefore,

$$n \geq 2(a + b + c) = 2\left((a + b) + (b + c) - b\right)$$
$$\geq 2\left((a + b) + (b + c) + \left(\frac{g}{2} - \frac{a+b}{2} - \frac{b+c}{2}\right)\right)$$
$$= (a + b) + (b + c) + g$$
$$\geq 3g.$$

On the other hand, if X contains $D(a_1, a_2; b)$, then $a_1 \geq g$, $a_2 \geq g$, and so $n \geq 4g + 4b > 3g$. Hence, in either case, we have $n \geq 3g$, proving the theorem. ∎

V. ONGOING WORK

We are currently applying this voltage-graph analysis to understand other properties of the derived graphs and their implications for the resulting graph-based codes. Simultaneously, we are investigating constructions of LDPC codes by specific voltage assignments. We are considering both the application of one voltage group to a sequence of base graphs and also the use of a tower of groups as voltage groups applied to a specific base graph to generate these families of LDPC codes. The techniques may yield new codes as well as improve the existing constructions such as the SFT codes, array-based codes, and other families. Our preliminary results suggest that using appropriate non-abelian groups for the voltage assignments may yield superior codes. This novel voltage graph approach is not limited to LDPC codes; rather, it can be applied in the algebraic design of other graph-based codes such as turbo codes, repeat-accumulate codes, serial-concatenated codes, etc. Indeed, some constructions of repeat accumulate codes have offset functions that can be related to the voltage assignment function.

REFERENCES

[1] A. Amit and N. Linial, "Random lifts of graphs II: Edge expansion", *Combinatorics Probability and Computing*, vol. 15, pp. 317-332, 2006.

[2] A. Amit, N. Linial, and J. Matousek, "Random lifts of graphs: independence and chromatic number", *Random Structures and Algorithms*, vol. 20, no. 1 , pp. 1-22, Jan. 2002.

[3] D. Archdeacon, J-H. Kwak, J. Lee, and Y. Sohn, "Bipartite covering graphs", *Discrete Mathematics*, 214 (2000) pp. 51-63.

[4] L. Brankovic, M. Miller, J. Plesnik, J. Ryan, and J. Siran, "Large graphs with small degree and diameter: A voltage assignment approach", *Jrnl. of Combinatorial Math. and Combinatorial Computing*, vol. 24, pp. 161-176, 1997.

[5] L. Brankovic, M. Miller, J. Plesnik, J. Ryan, and J. Siran, "A note on constructing large Cayley graphs of given degree and diameter by voltage assignments", *Elec. Jrnl. of Combinatorics*, vol. 5, no. 1, R9, 1998.

[6] D. J. Costello, Jr., A. Pusane, Jones, D. Divsalar, "A comparison of ARA and protograph-based LDPC block and convolutional codes", in *Proceedings of the 2007 Information Theory and Applications Workshop*, San Diego, CA, USA, Jan. 29–Feb. 2, 2007.

[7] D. Divsalar, S. Dolinar, and C. Jones, "Construction of protograph LDPC codes with minimum distance linearly growing with block size",*IEEE Globecom*, St. Louis, MO, Nov. 2005, pp. 11521156 2005.

[8] E. Eleftheriou and S. Olcer, "Low-density parity-check codes for multilevel modulation", in *Proceedings IEEE International Symposium on Information Theory*, Lausanne, Switzerland, 2002, p. 442.

[9] G. Exoo, "Voltage graphs, group presentations, and cages", *The Electronic Journal of Combinatorics*, vol. 11(1), 2004.

[10] J. L. Fan, "Array codes as low-density parity-check codes", in *Proceedings of the 2nd International Symposium on Turbo Codes and their applications*, Brest, France, Sept. 2000, pp. 543–546.

[11] J. Feldman, M. J. Wainwright, and D. R. Karger,"Using linear programming to decode binary linear codes", *IEEE Trans. on Info. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.

[12] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices", *IEEE Transactions on Information Theory*, vol.50, no.8, pp.17881793, 2004.

[13] J.L. Gross and T.W. Tucker, *Topological graph theory*, Wiley, NY, 1987.

[14] C. A. Kelley and D. Sridhara, "Pseudocodewords of Tanner Graphs", *IEEE Trans. on Info. Theory*, vol. 53, no. 11, pp. 4013-4038, Nov. 2007.

[15] C. A. Kelley and J. L. Walker, "On LDPC codes from voltage graphs", in *Special Session on Algebraic Coding Theory, AMS meeting*, Oct. 2007.

[16] S. Kim, J-S. No, H. Chung, and D-J. Shin, "Construction of protographs for QC-LDPC codes with girth larger than 12", *Proc. of IEEE Intl. Symp. on Info. Theory*, pp.2256-2260, June 2007.

[17] R. Koetter, W.-C. W. Li, P. O. Vontobel and J. L. Walker, "Characterizations of pseudo-codewords of (low-density) parity check codes", *Advances in Mathematics*, pp. 205-229, vol. 213, 2007.

[18] N. Linial and E. Rozenman, "Random lifts of graphs: Perfect matchings", *Combinatorica*, vol. 25, pp. 407 - 424, 2005.

[19] X. Ma and E-H. Yang, "Constructing LDPC codes by 2-lifts", in *Proc. of IEEE Intl. Symp. on Info. Theory*, June 2007.

[20] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic LDPC codes for fast encoding", *IEEE Trans. on Info. Theory*, vol. 51, no. 8, pp. 2894-2901, Aug. 2005.

[21] S. Song, L. Lan, S. Lin, and K. A-Ghaffar, "Construction of quasi-cyclic LDPC codes based on the primitive elements of finite fields", in *Proc. of Conf. on Info. Systems and Sciences*, March 22-24, 2006, pp. 835-838.

[22] D. Sridhara, T. E. Fuja, and R. M. Tanner, "Low density parity check codes from permutation matrices", 2001 Conf. on Info. Sci. & Systems, Johns Hopkins University, March 2001.

[23] R. M. Tanner, D. Sridhara, and T. E. Fuja, "A class of group-structured LDPC codes", in *Proc. of Intl. Symp. on Communication Theory and Applications*, Ambleside, U.K., pp. 365–370, July 2001.

[24] J. Thorpe, "LDPC codes constructed from protographs", *IPN progress report*, pp. 42-154, JPL, August 2003.

[25] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes", *IEEE Transactions on Information Theory*, to appear.