Random Edge-Local Complementation With Applications to Iterative Decoding of High-Density Parity-Check Codes

Joakim Grahl Knudsen, Constanza Riera, Lars Eirik Danielsen, Matthew G. Parker, and Eirik Rosnes

***Abstract*—We describe the application of edge-local complementation (ELC) to a Tanner graph associated with a binary linear code, $\mathcal{C}$. Various properties of ELC are described, mainly the special case of *isomorphic* ELC operations and the relationship to the automorphism group of the code, $\mathrm{Aut}(\mathcal{C})$, and the generalization of ELC to *weight-bounding* ELC (WB-ELC) operations under which the number of edges remains upper-bounded. ELC generates all systematic parity-check matrices (the *orbit*) of the code, so WB-ELC facilitates a restriction to low-weight matrices of this orbit. We propose using ELC and WB-ELC as a source of diversity, to improve iterative soft-input soft-output decoding of high-density parity-check (HDPC) codes, with the sum-product algorithm (SPA). A motivation of ELC-enhanced SPA decoding is *locality*; that diversity is achieved by local graph action, and is well-suited to the local actions that constitute the SPA and allows for parallel software implementation. Simulation data on the error-rate performance of the proposed SPA-ELC and SPA-WBELC iterative decoding algorithms is shown for several HDPC codes. A gain is reported over SPA decoding, and over a recently proposed algorithm to decode HDPC codes using permutations from $\mathrm{Aut}(\mathcal{C})$. ELC-enhanced decoding extends the scope of iterative decoding to codes with trivial $\mathrm{Aut}(\mathcal{C})$.**

***Index Terms*—Automorphism group, edge-local complementation (ELC), high-density parity-check (HDPC) codes, iterative decoding, Tanner graph.**

I. Introduction

Iterative soft decision decoding algorithms, applied to suitably designed codes, have been shown to give results which, asymptotically, closely approach the theoretical limits established by Shannon. The advent of turbo codes in 1993 [1] and the rediscovery of low-density parity-check (LDPC) codes at around the same time [2] (LDPC codes were actually invented by Gallager in 1962 [3]) caused much attention to be focused on iterative decoding of large, random or pseudo-random, sparse linear block codes. The sum-product algorithm (SPA) [4] is the standard soft decision iterative algorithm for decoding of LDPC codes on Tanner graphs. The sparse, random nature of these codes makes them well-suited for SPA decoding, using efficient software implementations (factor graphs). Asymptotically, optimum decoding performance is approximated at a complexity linear in code length. However, the large size and random nature of turbo and LDPC codes has negative implications when these are to be used in practice. This inspired researchers to adapt SPA decoding to small-size linear block codes, with blocklengths in the hundreds of bits or below. At small blocklengths, one has the benefit of using strong, nonrandom codes – "classical codes" – for which useful properties are known, such as large minimum distance and nontrivial automorphism group. It is, however, known that many families of codes – e.g., Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes – do not have Tanner graphs without cycles of length 4 [5]. Furthermore, these codes typically do not have sparse duals (i.e., sparse parity-check matrices) [6], so, when such codes are revisited from the context of iterative soft decoding, these are commonly referred to as *high-density parity-check* (HDPC) codes. Recently, the adaptation of iterative soft-input soft-output (SISO) decoding techniques to HDPC codes has received much attention [7]–[13].

This paper describes the pseudorandom use of a simple-graph operation known as edge-local complementation (ELC) [14], [15] to improve the performance of SPA decoding. One advantage of ELC-enhanced SPA decoding is the *locality* argument; diversity is achieved by local graph action, and so is well-suited to the local actions that constitute the SPA. Diversity stems from the change in Tanner graph due to the complementation of edges in a local subgraph. The locality also allows for an efficient parallel software implementation of ELC, in a similar way as for the SPA. The local complementations (which correspond to row-additions on the associated parity-check matrix) which comprise an ELC operation may be performed in parallel. Also, we will show that disjoint ELC operations are independent, and may be performed simultaneously. The effect of ELC on a graph is explored, and we define a subset of ELC operations under which the edge weight of the graph remains upper-bounded (to within some threshold value). We identify and describe all possible occurrences of single and double application of ELC as *weight-bounding* ELC (WB-ELC). We also present a further specialization of WB-ELC to *isomorphic* ELC (iso-ELC), under which the *structure* of the (simple) graph is invariant. We also propose a notion of Tanner graph equivalence, and explore when ELC preserves also the Tanner graph. These properties (weight and structure) are important from an iterative decoding perspective, and are targeted to improve the error-rate performance of a SISO HDPC

The material in this paper was presented in part at the IEEE International Symposium on Information Theory (ISIT), Seoul, Korea, June/July 2009, and the International Zürich Seminar on Communications (IZS), Zürich, Switzerland, March 2010.

J. Knudsen, L. E. Danielsen, C. Riera, and E. Rosnes were with the Selmer Center, Department of Informatics, University of Bergen, N-5020 Bergen, Norway. J. Knudsen and L. E. Danielsen are now with Webstep, Lars Hillesgate 20A, 5008 Bergen, Norway. C. Riera is with Bergen University College, Postboks 7030, 5020 Bergen, Norway. E. Rosnes is with Ceragon Networks, Kokstadveien 23, 5257 Kokstad, Norway. M. G. Parker is with the Selmer Center. E-mail: {joakimk, larsed, eirik, matthew}@ii.uib.no, and csr@hib.no.

decoder based on interleaving SPA iterations with random ELC or WB-ELC operations, giving a novel SPA-ELC and a SPA-WBELC decoding algorithm. ELC-enhanced decoding is a very general technique, and may be applied to a wider range of codes than other algorithms which rely on strong structural properties of the code (e.g., a large automorphism group). For other applications of WB-ELC, e.g., weight reduction, we refer to [16].

A. Outline

This paper is organized as follows. The ELC operation, which is typically defined for a simple graph, is described in Section II. A discussion on the action of ELC, in terms of the resulting graphs, focuses, firstly, on structurally distinct graphs, and, secondly, on isomorphic graphs with a link to the automorphism group of the code. Section III presents WB-ELC, where the action of ELC is discussed in terms of a maximum permitted weight of the resulting graphs. Finally, in Section IV, the use of ELC and WB-ELC as sources of diversity during SPA decoding is described. Two proposed decoding algorithms – SPA-ELC and SPA-WBELC – are described, simulated, and compared against other relevant decoding algorithms on several HDPC codes.

B. Preliminaries

A binary linear code $\mathcal{C}$ of length n, dimension k, and minimum distance $d_{\min}$ is denoted by $[n, k, d_{\min}]$, where $d_{\min}$ is defined as the minimum Hamming weight of any nonzero codeword. The dual code is $\mathcal{C}^{\perp}$, containing the codewords orthogonal to $\mathcal{C}$, and if $\mathcal{C} = \mathcal{C}^{\perp}$ we say the code is self-dual. Permutations are written in cycle notation, where we only specify the indices of the affected positions. For example, given a length-6 vector $\mathbf{v}$ and a permutation $\pi = (0,\ 1,\ 2)(3,\ 4)$, then $\mathbf{u} = \pi(\mathbf{v})$ means $v_0 \to u_1,\ v_1 \to u_2,\ v_2 \to u_0,\ v_3 \to u_4$, and $v_4 \to u_3$, while $v_5 \to u_5$. Similarly, $\pi(H)$ permutes the columns of a matrix, H. The identity permutation, affecting no positions, is, then, $\pi = \emptyset$. The automorphism group of the code, $\mathrm{Aut}(\mathcal{C})$, is the group of permutations, σ, which preserve the code, $\mathrm{Aut}(\mathcal{C}) = \{\sigma : \sigma(\mathcal{C}) = \mathcal{C}\}$. It is well-known that $\mathrm{Aut}(\mathcal{C}) = \mathrm{Aut}(\mathcal{C}^{\perp})$, and permutations are typically applied to H (which generates $\mathcal{C}^{\perp}$) during decoding, or (more conveniently) to the soft-input vector containing the *a posteriori* probability (APP) values [9]. If $\mathrm{Aut}(\mathcal{C})$ consists of the identity permutation alone, we say $\mathrm{Aut}(\mathcal{C})$ is trivial.

Let I_k be the identity matrix of size k, where we use the shorthand notation I when the dimension is not important. The generator matrix, G, generates $\mathcal{C}$, which gives $GH^T = 0$ where $(\,\cdot\,)^T$ denotes the transpose of its argument. H is said to be *systematic* if its columns can be reordered into the *standard form,*

$$\pi(H) = [\, I_{n-k} \mid P \,] \tag{1}$$

by some column permutation π (not necessarily in $\mathrm{Aut}(\mathcal{C})$). The column indices $0,\ 1,\ \ldots,\ n-1$ are referred to as the *coordinates* of the code. An information set, $\mathcal{I}$, of the code corresponds to any set of k columns in G which can be reduced to an identity submatrix by means of Gaussian elimination (GE). The $n-k$ columns at positions $\mathcal{P} := \{0, 1, \ldots, n-1\} \backslash \mathcal{I}$ form a parity set. Note that an information set corresponds to a parity set of the dual code, such that $\mathcal{I}$ refers to the P-part of H. In a systematic parity-check matrix, the columns indexed by $\mathcal{P}$ are referred to as systematic (i.e., weight-1) columns, while the remaining columns (of weight greater than 1) are *nonsystematic.* The (row) index of the single nonzero entry of a systematic column $\mathbf{h}_i$, $i \in \mathcal{P}$, is denoted by $\mathrm{row}(i) \in [0,\ n-k)$. In standard form (1), $\mathrm{row}(i) = i$, $0 \le i < n-k$. The weight of a matrix, H, (i.e., the number of nonzero entries) is denoted by $|H|$.

The Tanner graph, $\mathbf{TG}(H)$, associated with H is a $(2n-k)$-node bipartite graph with adjacency matrix,

$$\mathbf{TG}(H) = \left[\begin{smallmatrix} 0 & H \\ H^T & 0 \end{smallmatrix}\right].$$

(At some abuse of notation, we denote both graph and adjacency matrix by $\mathbf{TG}(H)$.) From now on we will assume that H is systematic. We will also assume no pairwise identical columns, i.e., $d_{\min} > 2$. The n "variable" nodes, denoted by v_i, $0 \le i < n$ and corresponding to columns of H, are partitioned into $|\mathcal{P}| = n-k$ systematic and $|\mathcal{I}| = k$ nonsystematic nodes, where the former have degree one. The $n-k$ "check" nodes of $\mathbf{TG}(H)$, denoted by f_j, $0 \le j < n-k$ and corresponding to rows of H, each have an associated (adjacent) systematic variable node. By grouping each check node with its associated systematic (variable) node, an n-node, $(n-k,\ k)$-bipartite, simple (i.e., undirected, with no double edges or loops) graph (BSG) is produced, with adjacency matrix,

$$\mathcal{G} = (\mathcal{U} \cup \mathcal{V},\ \mathcal{E}) = \pi^{-1} \left[\begin{smallmatrix} 0 & P \\ P^T & 0 \end{smallmatrix}\right]$$

where π^{-1} undoes the reordering in (1). $\mathcal{E}$ is the set of edges. The bipartition $(\mathcal{U},\ \mathcal{V})$ contains the $n-k$ grouped check/systematic variable nodes and the nonsystematic variable nodes, respectively. Furthermore, a permutation (here, π^{-1}) acts on both columns and rows of $\mathcal{G}$. By keeping a record of the bipartition at all times, we have a one-to-one mapping between a Tanner graph and a BSG. In summary, given a code represented by some $\mathbf{TG}(H)$, we construct a BSG by ignoring the systematic variable nodes – see Fig. 2. The number of edges in $\mathcal{G}$ is $|\mathcal{G}| = |\mathcal{E}| = |H| - (n-k)$ which we refer to as the weight of $\mathcal{G}$. If nodes in $\mathcal{U}$ and $\mathcal{V}$ have average degree $\bar{\rho}$ and $\bar{\gamma}$, respectively, we have that $|\mathcal{G}| = k\bar{\gamma} = (n-k)\bar{\rho}$. The local neighborhood of a node v is the set of nodes adjacent to v, and is denoted by $\mathcal{N}_v$, while $\mathcal{N}_v^u$ is shorthand notation for $\mathcal{N}_v \setminus \{u\}$. Let $\mathcal{E}_{A,B}$ denote the subgraph induced by the nodes in $A \cup B$ – i.e., a set of $|\mathcal{E}_{A,B}|$ edges. Furthermore, $\mathcal{E}_{u,v}$ is shorthand notation for $\mathcal{E}_{\mathcal{N}_u^v, \mathcal{N}_v^u}$, the local neighborhood of the edge (u, v). We use the notation $\{(u, v),\ \ldots,\ (u', v')\}$ for an ordered list of edges. Define the *distance* between edges (or *nonedges*) (u, v) and (u', v') as the shortest path between the sets of nodes, $\{u, v\}$ and $\{u', v'\}$.

II. Edge-Local Complementation

ELC is defined on an edge of a simple graph, $\mathcal{G}$ [14]. We consider only bipartite graphs in this work, which simplifies the description. ELC on an edge $(u, v) \in \mathcal{G}$ will complement the edges of $\mathcal{E}_{u,v}$ (replacing edges with nonedges and vice versa)

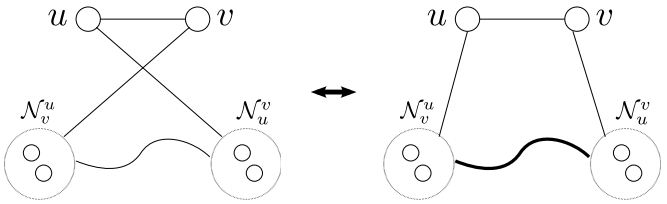


Fig. 1. ELC on edge (u, v) of a BSG. Curved links indicate arbitrary edges. Bold links mean that the edges connecting the two sets have been complemented; edges are replaced by nonedges, and vice versa. This graph may be a subgraph of a larger graph, in which case the rest of the graph remains unchanged.

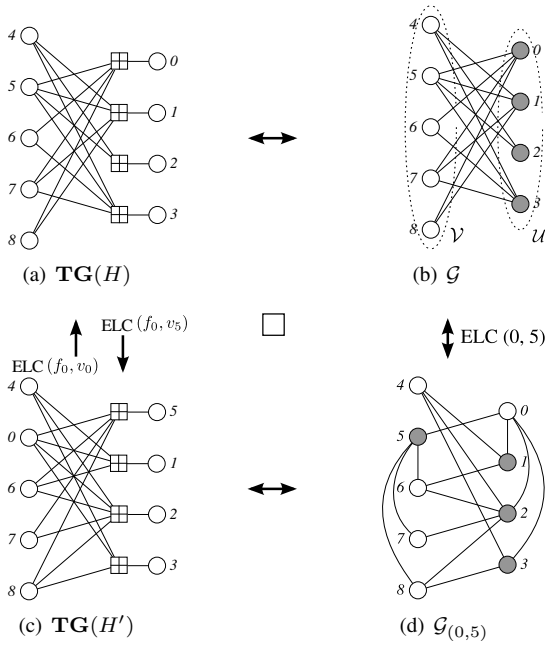


Fig. 2. Example of ELC on a small $[9, 4, 4]$ code, showing also the corresponding Tanner graphs. White and grey nodes correspond to $\mathcal{V}$ and $\mathcal{U}$, respectively.

followed by swapping the nodes u and v – see Fig. 1. In this sense ELC is a local operation as it only affects edges within distance 1 from the ELC edge, (u, v). The resulting graph, after ELC, is denoted by $\mathcal{G}_{(u,v)}$. ELC (on a simple graph) is a self-invertible operation as two ELC operations on the same edge is the identity operation, $\mathcal{G}_{\{(u,v),(u,v)\}} = \mathcal{G}$. The number of edges affected (inserted or removed) by the application of ELC is, on average,

$$|\mathcal{N}_u^v||\mathcal{N}_v^u| \approx (\bar{\gamma} - 1)(\bar{\rho} - 1). \tag{2}$$

For decoding purposes it is convenient to interpret ELC as an operation directly on $\mathbf{TG}(H)$, implicitly considering the corresponding simple graph. From this perspective, it is easily seen that one ELC operation implements the reduction stage of GE (i.e., row additions) on a single column of H. On $\mathbf{TG}(H)$, ELC is invertible but not self-invertible.

Example 1: Consider the optimal (in terms of $d_{\min}$) $[9, 4, 4]$ code, and the Tanner graph shown in Fig. 2(a). Fig. 2(b) shows the corresponding BSG. Fig. 2(d) shows $\mathcal{G}_{(0,5)}$ after ELC on $(0, 5) \in \mathcal{G}$, with the resulting Tanner graph in Fig. 2(c). ELC applied directly to edge $(f_0, v_5) \in \mathbf{TG}(H)$ amounts to adding row 0 to rows 1, 2, and 3 of H, to get H':

$$H = \begin{bmatrix} 1&0&0&0&0&\mathbf{1}&1&1&1\\ 0&1&0&0&1&\mathbf{1}&0&1&1\\ 0&0&1&0&1&\mathbf{1}&1&0&0\\ 0&0&0&1&1&\mathbf{1}&1&1&0 \end{bmatrix}_{+}^{+}_{+}_{+} \quad \rightarrow \quad \begin{bmatrix} \mathbf{1}&0&0&0&0&\mathbf{1}&1&1&1\\ \mathbf{1}&1&0&0&1&\mathbf{0}&1&0&0\\ \mathbf{1}&0&1&0&1&\mathbf{0}&1&1&1\\ \mathbf{1}&0&0&1&1&\mathbf{0}&0&0&1 \end{bmatrix} = H'.$$

Column 5 has been reduced to systematic form, and row additions have effectively swapped columns 0 and 5 between $\mathcal{I}$ and $\mathcal{P}$, giving a new information (and parity) set of the code. The inverse of ELC on (f_0, v_5) is ELC on (f_0, v_0), due to the changed bipartition. □

The link to GE emphasizes that ELC always preserves the code (i.e., the null space of H). Implemented on the Tanner graph, the inverse operation must reflect the changed information set (bipartition), as shown in Fig. 2. In this work, we refer to ELC on $\mathcal{G}$ and on $\mathbf{TG}(H)$ interchangeably, using the simple graph definition to simplify descriptions and proofs, whilst using the Tanner graph version for practical implementation in software. We shall use the shorthand notation (u, v) in the following also when referring to an edge in a Tanner graph (omitting the notation 'f' and 'v'). From a Tanner graph perspective, ELC can be implemented locally and concurrently in software by letting each check node, $u \in \mathcal{N}_v$, complement its subset of $\mathcal{E}_{u,v}$.

A. Minimum-Length ELC Sequence Between Two Structures

The set of structurally distinct graphs which arise by iteratively doing ELC on all edges of a BSG, $\mathcal{G}$, pruning the recursion tree on repeated graphs, is known as the orbit of the graph. This orbit is the same for all graphs corresponding to the same code, so we may refer to it as the orbit of the code. Structural distinctness is with respect to graph isomorphism. By using the software package Nauty [17], we obtain a canonical form of a simple graph, denoted by $N(\mathcal{G})$. Thus, for two simple graphs $\mathcal{G}$ and $\mathcal{G}'$, we have that $\mathcal{G} \overset{\text{iso}}{=} \mathcal{G}' \Leftrightarrow N(\mathcal{G}) = N(\mathcal{G}')$. The one-to-one relationship between a graph and a parity-check matrix (up to node labelling) means that we may also speak of the orbit as a set of parity-check matrices.

If a code has only one graph in its orbit, we say that it is an *ELC-preserved* code (or, equivalently, since this graph is unique, we may say that the graph is ELC-preserved) [18].

Theorem 1 (ELC sequence): A minimum-length ELC sequence $\mathbf{e} = \{(u_0, v_0), \ldots, (u_{l-1}, v_{l-1})\}$ can be found to convert a systematic matrix H into another systematic matrix H' (up to row permutations), where H and H' span the same space (they are in the same orbit), by comparing the corresponding bipartitions as represented by the parity sets $\mathcal{P}$ and $\mathcal{P}'$. The length, l, of $\mathbf{e}$ is $0 \leq l \leq \min(n - k,\ k)$. Depending on H, the sequence $\mathbf{e}$ may not be unique, so equivalent sequences may be derived from $\mathcal{P}$ and $\mathcal{P}'$.

Proof: ELC generates the entire orbit [15], and in particular all systematic parity-check matrices for the corresponding code, so such a sequence $\mathbf{e}$ must exist. Since a systematic basis for a (dual) code is uniquely defined (up to row permutations) by its parity set, the information set (i.e., the P-part of H) is a function of the parity set. Thus, by comparing $\mathcal{P}$ and $\mathcal{P}'$, we determine which coordinates are in opposite partitions, and shall be swapped. Each ELC operation preserves the (dual) code, and

Algorithm 1 MIN_ELC(H, H')

1: $\mathcal{L} := \mathcal{P} \setminus \mathcal{P}'$, $\mathcal{S} := \mathcal{P}' \setminus \mathcal{P}$, $\mathbf{e} := \emptyset$
2: **while** $\mathcal{S} \neq \emptyset$ **do**
3: choose and remove any $s \in \mathcal{S}$, as well as any $r \in \mathcal{L}$ s.t. $(\mathrm{row}(r), s) \in \mathbf{TG}(H)$
4: ELC on $(\mathrm{row}(r), s)$ on $\mathbf{TG}(H)$
5: $\mathbf{e} := \mathbf{e} \cup (\mathrm{row}(r), s)$
6: **end while**

has the effect of swapping a pair of positions between $\mathcal{I}$ and $\mathcal{P}$ (i.e., columns in H), along with some "residual" modifications to H resulting from the row-additions. To modify H into H', we may thus focus on swapping corresponding pairs of columns (via ELC) from $\mathcal{P}$ into $\mathcal{P}'$, to give the I-part of H', and the residual modifications must "resolve" into the required P-part (since the P-part is unique given the I-part). Then, the submatrices I and I' are equal, from which it follows that $P = P'$, such that $H = H'$ (up to row permutations). Alg. 1 is a constructive proof of this theorem, showing how $\mathcal{P}$ and $\mathcal{P}'$ are used to determine a corresponding ELC sequence. ELC has the effect of swapping exactly one pair of positions between $\mathcal{I}$ and $\mathcal{P}$, so the length of $\mathbf{e}$ must be exactly $l = |\mathcal{P} \setminus \mathcal{P}'|$, upper-bounded by $\min(n-k,\ k)$. ∎

The difference (coordinates to swap) corresponds to the sets $\mathcal{L} = \mathcal{P} \setminus \mathcal{P}'$ and $\mathcal{S} = \mathcal{P}' \setminus \mathcal{P}$. As each position in the identity (sub) matrix is unique, $r \in \mathcal{L}$ can be viewed as a row-index, where r is chosen such that $(\mathrm{row}(r), s) \in \mathbf{TG}(H)$, given $s \in \mathcal{S}$. Theorem 1 shows that at least one such (possibly empty) sequence of valid choices must exist, if and only if H and H' are in the same orbit. When several valid choices of r exist, branch points arise in the algorithm which all lead to equivalent ELC sequences; the resulting Tanner graphs are exactly the same (although the matrices may be different, but only in terms of row permutations) – see Section II-B for further discussion.

Example 2: Consider the $[14, 7, 3]$ doubly circulant quadratic residue (QR) code. The orbit of this code consists of 11 graphs. Choosing two distinct graphs, $\mathcal{G}$ and $\mathcal{G}'$, we must have that $N(\mathcal{G}) \neq N(\mathcal{G}')$. Let H be a parity-check matrix corresponding to $\mathcal{G}$, and H' correspond to $\mathcal{G}'$;

$$H = \begin{bmatrix} 1&0&0&0&1&0&0&1&0&1&0&0&1&1 \\ 1&1&0&0&1&0&0&1&1&0&1&0&0&1 \\ 1&0&1&0&0&0&0&1&1&0&1&0&0&1 \\ 0&0&0&1&1&0&0&1&1&0&1&0&0&1 \\ 0&0&0&0&1&0&0&0&1&0&0&1&1&1 \\ 1&0&0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&0&1&0&0&1 \end{bmatrix}, \quad H' = \begin{bmatrix} 0&1&0&0&0&0&0&0&1&1&1&0&1&0 \\ 1&1&0&0&1&0&1&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&1&1&0&0&1&1&1&0 \\ 0&1&0&0&0&1&1&0&0&0&0&0&0&0 \\ 1&1&0&0&1&0&1&0&0&0&0&0&0&0 \end{bmatrix}.$$

It is easily seen that $\mathcal{G}$ and $\mathcal{G}'$ are indeed nonisomorphic, simply by verifying that $|H| \neq |H'|$. The parity sets are $\mathcal{P} = \{1, 2, 3, 5, 6, 9, 11\}$ and $\mathcal{P}' = \{0, 2, 3, 5, 9, 11, 13\}$, and Alg. 1 computes $\mathcal{L} = \{1, 6\}$ and $\mathcal{S} = \{0, 13\}$. Choosing (and removing) $s = 13$, we find that $r = 1$ gives the edge $(\mathrm{row}(1), 13) = (1, 13) \in \mathbf{TG}(H)$. Let $H_{(1,13)}$ be the resulting matrix after ELC. Finally, the remaining value $s = 0$ gives $r = 6$, where edge $(\mathrm{row}(6), 0) = (6, 0) \in \mathbf{TG}(H_{(1,13)})$;

$$H_{(1,13)} = \begin{bmatrix} 0&1&0&0&0&0&0&0&1&1&1&0&1&0 \\ 1&1&0&0&1&0&0&1&1&0&1&0&0&1 \\ 0&1&1&0&1&0&0&0&0&0&0&0&0&0 \\ 1&1&0&1&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&1&0&0&1&1&1&0 \\ 1&0&0&0&1&1&0&0&0&0&0&0&0&0 \\ 1&1&0&0&1&0&1&0&0&0&0&0&0&0 \end{bmatrix}, \quad H_{\{(1,13),(6,0)\}} = \begin{bmatrix} 0&1&0&0&0&0&0&0&1&1&1&0&1&0 \\ 0&0&0&0&0&0&0&1&1&1&0&1&0&1 \\ 0&1&1&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&1&1&0&0&1&1&1&0 \\ 0&1&0&0&0&1&1&0&0&0&0&0&0&0 \\ 1&1&0&0&1&0&1&0&0&0&0&0&0&0 \end{bmatrix}.$$

By swapping rows 1 and 6, $H_{\{(1,13),(6,0)\}}$ equals H' so these give the same Tanner graph. That the ELC sequence $\mathbf{e} = \{(1, 13), (6, 0)\}$ is not unique is reflected by Alg. 1. Different choices (of s) would result in the sequences $\{(1, 0), (6, 13)\}$ and $\{(6, 13), (1, 0)\}$, which both give the "target" matrix, H' (up to row-equivalence). The sequence $\{(6, 0), (1, 13)\}$, however, is not possible, since the edge $(6, 0) \notin \mathbf{TG}(H)$.[1] □

B. Tanner Graph Invariants

In the context of graph-based, iterative decoding, we are interested in discerning distinct Tanner graphs, when these may correspond to isomorphic BSGs. A linear code is preserved under elementary row operations (i.e., row additions and row permutations) on the associated linear basis (parity-check matrix). However, columns (code coordinates) correspond to variable nodes in the Tanner graph, on which channel inputs are attached. Column permutations which preserve the code, comprise $\mathrm{Aut}(\mathcal{C})$.

We define two Tanner graphs $\mathbf{TG}(H)$ and $\mathbf{TG}(H')$ as isomorphic if and only if the rows of H' can be permuted to give the exact same matrix H. A parity-check matrix, H, can be put in canonical form, denoted by $R(H)$, by sorting its rows in lexicographical order, $\mathbf{TG}(H) = \mathbf{TG}(H') \Leftrightarrow R(H) = R(H')$. Here, we define H and H' as *row-equivalent.* From a decoding perspective, distinct Tanner graphs give increased diversity. In the case where the BSGs are isomorphic, the structural properties (e.g., matrix weight, and number and length of short cycles, etc.) are also preserved. A sequence of ELC operations connecting two parity-check matrices for the same code, $H \neq H'$, with isomorphic BSGs, $N(\mathcal{G}) = N(\mathcal{G}')$, has previously been defined as an *iso-ELC sequence* [20]. (The ELC operation is sometimes referred to as a *pivot* operation.)

Definition 1: A permutation $\theta \in \mathrm{Aut}(\mathcal{C})$ is called *trivial* if and only if $\mathbf{TG}(H) = \mathbf{TG}(\theta(H))$ (i.e., $H = \theta(H)$ up to row permutations).

Theorem 2 (ELC finds entire $\mathrm{Aut}(\mathcal{C})$*):* Each nontrivial permutation in $\mathrm{Aut}(\mathcal{C})$, for a given H, is associated with an iso-ELC sequence of length l, for $1 \leq l \leq \min(n-k,\ k)$. The particular sequence depends on the parity set, $\mathcal{P}$, (i.e., on H), and is not unique.

Proof: For each nontrivial permutation $\sigma \in \mathrm{Aut}(\mathcal{C})$, H and $\sigma(H)$ are two (nonisomorphic) systematic parity-check matrices for $\mathcal{C}$, i.e., they both span the same space, and the result follows from Theorem 1. ∎

We will now explore the algebraic properties of $\mathrm{Aut}(\mathcal{C})$, as a function of a specific parity-check matrix. Keep in mind the relationship between $\mathrm{Aut}(\mathcal{C})$ and ELC operations derived in Theorem 2. The "potential diversity" of a parity-check matrix,

[1] These equivalent ELC sequences also follow from [19].

H, (i.e., number of distinct matrices attainable via permutations or ELC, starting from H) can be used to assess the suitability of $\mathcal{C}$ for diversity decoding, and to search for an optimal starting matrix. We begin by formalizing which permutations do not improve diversity (i.e., the Tanner graph does not change).

Proposition 1 (Trivial permutation): A permutation $\theta \in \mathrm{Aut}(\mathcal{C})$ is trivial if and only if it permutes no positions between $\mathcal{I}$ and $\mathcal{P}$ for the given H. Furthermore, the set of trivial permutations forms a subgroup $\mathcal{D}_H \preceq \mathrm{Aut}(\mathcal{C})$.

Proof: If a permutation θ is trivial for a given parity-check matrix H, then (by definition) H and $\theta(H)$ are row-equivalent, i.e., $R(H) = R(\theta(H))$. Since H and $\theta(H)$ are row-equivalent, θ is constrained to permute the columns from $\mathcal{P}$ (i.e., the I-part of H) to indices from $\mathcal{P}$, and thus permute the columns from $\mathcal{I}$ (i.e., the P-part of H) to indices from $\mathcal{I}$, and the result follows.

Conversely, if a permutation θ permutes no positions between $\mathcal{I}$ and $\mathcal{P}$ for the given H, then the resulting matrix $\theta(H)$ will have weight-1 columns in exactly the same positions as H, i.e., in the positions in $\mathcal{P}$. Permuting the rows of $\theta(H)$ such that the I-parts of H and $\theta(H)$ become identical will also make the P-parts identical (the P-part is a function of the I-part), from which it follows that H and $\theta(H)$ are row-equivalent, and the permutation θ is (by definition) trivial.

Finally, we need to prove that the set of trivial permutations forms a subgroup of $\mathrm{Aut}(\mathcal{C})$. This follows directly from the first result (i.e., that a permutation $\theta \in \mathrm{Aut}(\mathcal{C})$ is trivial if and only if it permutes no positions between $\mathcal{I}$ and $\mathcal{P}$), since the composition of two such permutations obviously permutes no positions between $\mathcal{I}$ and $\mathcal{P}$. ∎

The subgroup $\mathcal{D}_H$ is *not* a code property, but a property of H. Furthermore, since $\mathcal{D}_H$ is a subgroup, we can decompose $\mathrm{Aut}(\mathcal{C})$ into a union of cosets of $\mathcal{D}_H$,

$$\mathrm{Aut}(\mathcal{C}) = \{\mathcal{D}_H \circ \sigma_0\} \cup \{\mathcal{D}_H \circ \sigma_1\} \cup \cdots \cup \{\mathcal{D}_H \circ \sigma_{|\mathrm{Aut}(\mathcal{C})|/|\mathcal{D}_H|-1}\}$$

where $\mathcal{K}_H = \{\sigma_0, \ldots, \sigma_{|\mathrm{Aut}(\mathcal{C})|/|\mathcal{D}_H|-1}\}$ is a set of coset leaders, given H, and σ_0 is the identity permutation. We will sometimes use the shorthand notation $\mathcal{D}$ and $\mathcal{K}$ when the specific matrix, H, is not important.

Alg. 1 can be used to convert any $\sigma \in \mathrm{Aut}(\mathcal{C})$ into an equivalent (iso-)ELC sequence, $\mathbf{e}$, by taking as input H and $H' = \sigma(H)$. The corresponding iso-ELC sequence depends on both σ and H, and we may emphasize this by the notation, $\mathbf{e}_{\sigma,H}$. Then we have that $R(\sigma(H)) = R(\mathbf{e}_{\sigma,H}(H))$. For trivial permutations, $\theta \in \mathcal{D}_H$, $R(H) = R(\theta(H))$ and $\mathbf{e}_{\theta,H} = \emptyset$ (i.e., the same Tanner graph).

Proposition 2: Given a parity-check matrix H, $\mathbf{e}_{\sigma,H}$ is an iso-ELC sequence representation of *all* permutations in the coset $\mathcal{D} \circ \sigma$, $\sigma \in \mathrm{Aut}(\mathcal{C})$.

Proof: The coset decomposition is in terms of row equivalence, i.e., $R(\sigma(H) = R(\sigma'(H))$ for any $\sigma' \in \mathcal{D} \circ \sigma$, and the result follows. ∎

Given H, the set $\mathcal{K}_H \setminus \{\sigma_0\}$ contains permutations from $\mathrm{Aut}(\mathcal{C})$ which give a distinct parity-check matrix $\sigma(H)$, where $\sigma \in \mathcal{K}_H \setminus \{\sigma_0\}$. Each coset leader σ corresponds to a matrix $R(\sigma(H))$ representing the $|\mathcal{D}|$ row-equivalent matrices $\theta(\sigma(H)),\ \forall\ \theta \in \mathcal{D}$. In other words, these all correspond to the same Tanner graph. In this sense, the set of coset leaders is not unique (any $\sigma' \in \mathcal{D} \circ \sigma$, where $\sigma \neq \sigma_0$, could be used as a coset leader), which means that $\mathcal{K}_H$ is not unique even for a given H. Since σ_0 is the identity mapping, $\mathcal{K}_H$ can be a group.

The set of (distinct) Tanner graphs resulting from the permutations in $\mathcal{K}_H$ comprise the *iso-orbit* of H,[2] $\{\sigma_0(H), \ldots, \sigma_{|\mathcal{K}|-1}(H)\}$. These Tanner graphs are all distinct, but correspond to isomorphic simple graphs, $R(H) \neq R(\sigma(H))$, but $N(\mathcal{G}) = N(\sigma(\mathcal{G}))$, $\forall\ \sigma \in \mathcal{K}_H \setminus \{\sigma_0\}$. The iso-orbit can be partitioned into disjoint subsets according to the (minimal) length, $0 \leq l \leq \min(n-k,\ k)$, of the corresponding ELC sequences: $\mathcal{K}_H^l = \{\sigma \in \mathcal{K}_H : |\mathcal{P} \setminus \sigma(\mathcal{P})| = l\}$. In particular, $\mathcal{K}^0 = \{\sigma_0\}$. Thus, for $l > 0$, $\mathcal{K}^l$ is *not* a group since it does not contain the identity permutation, σ_0.

We shall now see how $\mathcal{D}_H$ and $\mathcal{K}_H$ relate to H.

Proposition 3: For any permutation α (not necessarily in $\mathrm{Aut}(\mathcal{C})$) the trivial subgroup $\mathcal{D}_{\alpha(H)} = \alpha \circ \mathcal{D}_H \circ \alpha^{-1}$, for a given H. Furthermore, $\mathcal{K}_{\alpha(H)} = \alpha \circ \mathcal{K}_H \circ \alpha^{-1}$ and $\mathcal{K}^l_{\alpha(H)} = \alpha \circ \mathcal{K}^l_H \circ \alpha^{-1}$, for all l, $0 \leq l \leq \min(n-k,\ k)$.

Proof: Let $\sigma = \alpha \circ \theta \circ \alpha^{-1}$ where $\theta \in \mathcal{D}_H$. After applying α^{-1} to $\alpha(H)$, the original matrix H is reconstructed. Then, the effect of applying θ to H is to permute the rows of H. Finally, the columns are permuted according to α, and the resulting matrix will be row-equivalent to $\alpha(H)$. Thus, σ is trivial with respect to $\alpha(H)$, from which it follows that $\alpha \circ \mathcal{D}_H \circ \alpha^{-1}$ is a subset of $\mathcal{D}_{\alpha(H)}$. To prove equality, we use this result with $H' = \alpha(H)$, from which it follows that $\kappa \circ \mathcal{D}_{H'} \circ \kappa^{-1} \subseteq \mathcal{D}_{\kappa(H')}$, where κ is any permutation. Choosing $\kappa = \alpha^{-1}$, we get $\alpha^{-1} \circ \mathcal{D}_{\alpha(H)} \circ \alpha \subseteq \mathcal{D}_H$, from which it follows that $\mathcal{D}_{\alpha(H)} \subseteq \alpha \circ \mathcal{D}_H \circ \alpha^{-1}$. Since $\mathcal{D}_{\alpha(H)}$ is both a subset and a super-set of $\alpha \circ \mathcal{D}_H \circ \alpha^{-1}$, we have equality.

To prove the second part, i.e., to show that $\mathcal{K}_{\alpha(H)} = \alpha \circ \mathcal{K}_H \circ \alpha^{-1}$, we use the fact that for any two permutations $\sigma_1 = \theta_1 \circ \sigma \in \mathcal{D}_{\alpha(H)} \circ \sigma$ and $\sigma_2 = \theta_2 \circ \sigma \in \mathcal{D}_{\alpha(H)} \circ \sigma$ from the same coset (based on $\mathcal{D}_{\alpha(H)}$), where σ denotes the coset leader and $\theta_1, \theta_2 \in \mathcal{D}_{\alpha(H)}$, we must have that $\sigma_1 \circ \sigma_2^{-1} = \theta_1 \circ \sigma \circ \sigma^{-1} \circ \theta_2^{-1} = \theta_1 \circ \theta_2^{-1} \in \mathcal{D}_{\alpha(H)}$. Thus, if for any two permutations σ_1 and σ_2 from $\mathrm{Aut}(\mathcal{C})$, the composition $\sigma_1 \circ \sigma_2^{-1} \notin \mathcal{D}_{\alpha(H)}$, then σ_1 and σ_2 belong to two different cosets (based on $\mathcal{D}_{\alpha(H)}$). Now, let $\sigma_1 = \alpha \circ \kappa_1 \circ \alpha^{-1}$ and $\sigma_2 = \alpha \circ \kappa_2 \circ \alpha^{-1}$, where $\kappa_1, \kappa_2 \in \mathcal{K}_H$, from which it follows that $\sigma_1 \circ \sigma_2^{-1} = \alpha \circ \kappa_1 \circ \alpha^{-1} \circ \alpha \circ \kappa_2^{-1} \circ \alpha^{-1} = \alpha \circ (\kappa_1 \circ \kappa_2^{-1}) \circ \alpha^{-1}$. Since $\kappa_1 \circ \kappa_2^{-1} \notin \mathcal{D}_H$ (κ_1 and κ_2 are from different cosets based on $\mathcal{D}_H$), we must have that $\sigma_1 \circ \sigma_2^{-1} \notin \mathcal{D}_{\alpha(H)}$, and it follows that σ_1 and σ_2 are from two different cosets based on $\mathcal{D}_{\alpha(H)}$. The result now follows since $|\mathcal{K}_H| = |\mathrm{Aut}(\mathcal{C})|/|\mathcal{D}_H| = |\mathrm{Aut}(\mathcal{C})|/|\mathcal{D}_{\alpha(H)}| = |\mathcal{K}_{\alpha(H)}|$.

To prove the third part, i.e., to show that $\mathcal{K}^l_{\alpha(H)} = \alpha \circ \mathcal{K}^l_H \circ \alpha^{-1}$ for all l, we use the fact that the *depth* of σ (i.e., the length of the corresponding ELC sequence) based on H, is the same as the depth of $\alpha \circ \sigma \circ \alpha^{-1}$ based on $\alpha(H)$, for any

[2]The iso-orbit of H, containing Tanner graphs, should not be confused with the orbit of $\mathcal{C}$, which contains simple graphs.

TABLE I
PAIRS OF PERMUTATIONS FROM $\mathrm{Aut}(\mathcal{C})$ WHICH GENERATE $\mathcal{K}$ FOR THE $[8,4,4]$ EXTENDED HAMMING CODE, SEE EXAMPLE 3. THESE 8 GROUPS ARE ALL ISOMORPHIC TO ONE GROUP, WHICH IS UNIQUE.

$\langle(0,4,2,7,6,3,1),\ (0,6,7,4,5,2,3)\rangle$	$\langle(0,1,3,6,5,7,2),\ (0,6,1,7,4,5,2)\rangle$
$\langle(0,6,4,5,1,2,3),\ (0,7,5,2,1,4,3)\rangle$	$\langle(0,6,7,4,2,3,1),\ (0,4,5,2,7,6,3)\rangle$
$\langle(0,2,1,6,4,5,3),\ (0,6,7,5,4,2,1)\rangle$	$\langle(0,6,2,1,5,7,3),\ (0,7,5,3,4,2,1)\rangle$
$\langle(0,5,7,2,4,3,1),\ (0,2,6,4,7,5,3)\rangle$	$\langle(0,4,5,1,2,7,3),\ (0,6,7,5,2,1,3)\rangle$

σ in $\mathrm{Aut}(\mathcal{C})$. To show this, we write the depth of $\alpha\circ\sigma\circ\alpha^{-1}$ based on $\alpha(H)$ as,

$$\begin{aligned}&|\{\alpha\circ\sigma\circ\alpha^{-1}(\mathcal{P}_{\alpha(H)})\cap\mathcal{I}_{\alpha(H)}\}|\\ &\quad=|\{\alpha\circ\sigma\circ\alpha^{-1}(\alpha(\mathcal{P}_H))\cap\alpha(\mathcal{I}_H)\}|\\ &\quad=|\{\alpha\circ\sigma(\mathcal{P}_H)\cap\alpha(\mathcal{I}_H)\}|\\ &\quad=|\{\alpha(\sigma(\mathcal{P}_H)\cap\mathcal{I}_H)\}|\\ &\quad=|\{\sigma(\mathcal{P}_H)\cap\mathcal{I}_H\}|.\end{aligned}$$

Now, we can conclude that the depth of all coset leaders in $\mathcal{K}^l_{\alpha(H)}$ (based on $\mathcal{D}_{\alpha(H)}$) is the same and equal to the depth of the coset leaders from $\mathcal{K}^l_H$ (based on $\mathcal{D}_H$), from which the result follows. ∎

As discussed above, $\mathcal{D}_H$ depends on H, so the iso-orbit is not a code property. The partitioning of permutations in $\mathcal{K}_H$ into disjoint subsets according to the length of the corresponding iso-ELC sequence may vary for each $H'=\sigma(H),\ \sigma\in\mathrm{Aut}(\mathcal{C})$. Still, from Proposition 3, $|\mathcal{K}^l_H|=|\mathcal{K}^l_{\sigma(H)}|,\ 0\le l\le\min(n-k,\ k)$ and $\sigma\in\mathrm{Aut}(\mathcal{C})$, and we call the set $\{|\mathcal{K}^l_H|\},\ 0\le l\le\min(n-k,\ k)$, the *profile* of the iso-orbit of H. This profile varies with H, but is invariant over the iso-orbit of H (one profile per graph in the orbit). Since the profile varies with H, it may be desirable to search the orbit for a graph that has certain properties with respect to the profile. We illustrate this with some examples.

Example 3: For the $[8,4,4]$ extended Hamming code, which is ELC-preserved, the parity-check matrix,

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

has the profile listed in Table II. For this code, there exists only one conjugacy class of subgroups of $\mathrm{Aut}(\mathcal{C})$ of the required size $|\mathcal{K}|=|\mathrm{Aut}(\mathcal{C})|/|\mathcal{D}|=1344/24=56$ (verified in MAGMA). $\mathcal{K}$ can be any of the eight distinct (but isomorphic) subgroups in this class. The eight subgroups may all be generated by two permutations, as listed in Table I. This shows that $\mathcal{K}$ can be a group, and the minimum number of generators is 2 (i.e., $\mathcal{K}$ can not be a cyclic subgroup). $\mathcal{D}_H$ is $\langle(0,2)(6,7),\ (1,3)(4,5),\ (2,3)(5,7)\rangle$.

Example 4: The $[24,12,8]$ extended Golay code, where $|\mathrm{Aut}(\mathcal{C})|=244\,823\,040$ is a rare [18] example of a code with only two graphs in its orbit, corresponding to parity-check matrices,

$$H_0=\begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&0&0&0&1&1\\
0&1&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&1&0&0&1&0&1&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&1&0&1&0&1&1&1\\
0&0&0&1&0&0&0&0&0&0&0&1&1&0&0&0&1&1&1&0&1&1&0&1&0\\
0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&0&0&1&1&1\\
0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&0&1&1&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&1&0&1\\
0&0&0&0&0&0&0&1&0&0&0&1&0&1&1&0&1&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&0&1&1&0&1&1&1&1&0&0&0&1\\
0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&1&1&0&1&1&1&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&1&0&0&0&1&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&1&0&0&0&1&1&1
\end{bmatrix}$$

$$H_1=\begin{bmatrix}
1&0&1&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&0\\
1&0&0&1&0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&1&0&1&0&1\\
1&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&1&0&0&0&0&1&1&0\\
1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&1&1&1&0&1&0&1&0&0\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&1\\
1&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&1&0&1&1&1&1&1\\
1&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&0&0&0&1&1\\
0&1&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&0&0&1\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&1&1&0&1&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&1&0&0&0&1&1&1
\end{bmatrix}.$$

The weight is 96 and 100, and $|\mathcal{D}|$ is 240 and 660, respectively. The two profiles for $\mathcal{K}$ are listed in Table II. No subgroups of $\mathrm{Aut}(\mathcal{C})$ exist of size $|\mathcal{K}|=|\mathrm{Aut}(\mathcal{C})|/|\mathcal{D}|$ for either of the two graphs (verified in MAGMA), so $\mathcal{K}$ can not be a group.

III. WEIGHT-BOUNDING ELC

In the discussion on isomorphic ELC operations, a requirement is that the number of edges in the graph must be preserved [20]. We generalize this, and introduce a notion of *weight-bounding* ELC (WB-ELC) operations, in which the weight of H after ELC, denoted by $|H'|$, is upper-bounded by $|H|+T$, where T is some threshold. We give necessary and sufficient conditions to achieve this bound, both for single ELC and for two consecutive ELCs. In this work, we restrict our focus to depth-1 or 2, with respect to the locality argument of the ELC operation (in the sense that many ELC operations amount to a global graph operation). However, the concept of WB-ELC extends to arbitrary depth. Note that the depth-i iso-ELC sequences described previously are indeed depth-i WB-ELC for $T=0$, where $0<i\le n-k$ is the length of the ELC-sequence. The weight of H greatly affects its suitability for iterative decoding. In the previous section, graph isomorphism (and code automorphism) was discussed as a means for preserving graph properties during decoding. In this section, we relax this requirement, permitting a certain weight change in H under ELC. The main motivation for this is to achieve a tradeoff between graph diversity and weight.

Let $A\sim B$ be a shorthand notation for the edges in the subgraph $\mathcal{E}_{A,B}$, i.e., those connecting nodes in A to nodes in B. Also, $\mathcal{E}^C_{A,B}$ denotes the subgraph after complementing $A\sim B$. The net difference in edges before and after complementation is $\Delta\mathcal{E}_{A,B}\triangleq|\mathcal{E}^C_{A,B}|-|\mathcal{E}_{A,B}|$.

Lemma 1: The number of edges complemented between sets A and B can be expressed as,

$$\Delta\mathcal{E}_{A,B}\triangleq|\mathcal{E}^C_{A,B}|-|\mathcal{E}_{A,B}|=|A||B|-2|\mathcal{E}_{A,B}|.$$

Proof: The complete bipartite graph between A and B has $|A||B|$ edges. This means that, for any graph between A and B, $|\mathcal{E}_{A,B}|+|\mathcal{E}^C_{A,B}|=|A||B|$, so $\Delta\mathcal{E}_{A,B}=|\mathcal{E}^C_{A,B}|-|\mathcal{E}_{A,B}|=|A||B|-|\mathcal{E}_{A,B}|-|\mathcal{E}_{A,B}|$. ∎

A. Depth-1, Single Edge WB-ELC

If the weight change due to the action of a single ELC is upper-bounded, we say that the ELC is WB-ELC.

Theorem 3: The weight change of $\mathcal{G}$ under ELC on (u,v) is upper-bounded by a threshold T *iff,*

$$\Delta\mathcal{E}_{u,v}=|\mathcal{N}^v_u||\mathcal{N}^u_v|-2|\mathcal{E}_{u,v}|\le T.$$

TABLE II
PROFILES OF $\mathcal{K}$ AS SPLIT INTO SUBSETS ACCORDING TO THE LENGTH OF THE CORRESPONDING ELC SEQUENCE.

Code	$\lvert H\rvert$	0	1	2	3	4	5	6	7	8	9	10	11	12
Ext. Hamming	16	1	12	30	12	1	-	-	-	-	-	-	-	-
Ext. Golay	100	1	22	616	6 490	33 935	85 712	117 392	85 712	33 935	6 490	616	22	1
”	96	1	60	1 650	18 140	92 655	236 520	322 044	236 520	92 655	18 140	1 650	60	1

Proof: ELC on (u, v) complements edges between $\mathcal{N}_u^v$ and $\mathcal{N}_v^u$, and the inequality follows from Lemma 1. The weight change of $\mathcal{G}$ under ELC on (u, v) is therefore $\Delta\mathcal{E}_{u,v}$. ■

B. Depth-2, Double Edge WB-ELC

For many graphs, it is difficult (or impossible) to upper-bound the weight change by any reasonable threshold (i.e., small T), using only a single ELC. We now determine the WB-ELC operations which exist for double application of ELC on a graph. Given a graph, $\mathcal{G}$, and a threshold, T, the definition of a depth-2 WB-ELC operation is an ordered sequence of two ELC operations, where the first ELC operation must change the weight of $\mathcal{G}$ by more than T (to a graph $\mathcal{G}^\star$), whereupon the second ELC must compensate by reducing the weight of $\mathcal{G}^\star$ by at least $|\mathcal{G}^\star| - |\mathcal{G}| - T$. This amount is always positive, as $|\mathcal{G}^\star| > T + |\mathcal{G}|$; otherwise the first ELC would change the weight by an amount less than or equal to T. We emphasize that if the first ELC did *not* exceed the weight-bounding threshold, then it would, by itself, be a (depth-1) WB-ELC operation.

An important observation is that the search space for depth-2 WB-ELC can be significantly reduced from that of checking all pairs of edges in $\mathcal{G}$. First, ELC on two adjacent edges, i.e., at distance 0, reduces to a single ELC operation.

Lemma 2 (Adjacent edges [21], proof omitted): ELC on $\{(u, v), (v, v')\}$, where $v' \in \mathcal{N}_u^v$, gives the same graph as ELC on (u, v').

From Lemma 2, we see that ELC on adjacent edges reduces to a single ELC, which has already been covered by the discussion of depth-1 WB-ELC. So, in order to find additional WB-ELC instances at depth-2, we need not consider adjacent pairs of edges. We now present an important novel result regarding depth-2 WB-ELC; that the distance between a pair of edges can not be greater than two, for $T \geq -1$.[3]

Lemma 3 (Disjoint edges): Let $T \geq -1$. Any depth-2 WB-ELC where the two edges are separated by a distance greater than two will always reduce to either one instance, or two separate instances, of depth-1 WB-ELC.

Proof: Consider two disjoint subgraphs, $\mathcal{E}_{u,v}$ and $\mathcal{E}_{u',v'}$, of the same graph. In this case, ELC on $\{(u, v), (u', v')\}$ gives the same graph as ELC on $\{(u', v'), (u, v)\}$, since the neighborhoods do not interact. Let $T \geq -1$. The only possibilities for WB-ELC are: Both ELC operations classify as depth-1 WB-ELC operations (change weight by no more than T), or one ELC operation changes the weight by w, where $w > T$, while the other ELC reduces weight by at least $w - T$. Since they commute, we can assume without loss of generality that ELC on (u, v) is the operation which reduces weight, but then this, by itself, classifies as a (depth-1) WB-ELC operation. ■

Theorem 4 (Reduced search space): Let $T \geq -1$. All depth-2 WB-ELC can be found by considering pairs of edges spaced by a distance one or two.

Proof: The proof follows from Lemmas 2 and 3. ■

In this sense, we define WB-ELC (both depth-1 and depth-2) as a *local* graph operation, in that its effect is confined to a subgraph of diameter at most 4. The corresponding subgraphs are shown in Figs. 3 and 4. We have restricted the search space considerably, and shall now cover all possible cases for depth-2 WB-ELC, for $T \geq -1$.

Let us first consider the case where the pair of edges are at a distance of exactly two edges apart, see Fig. 3. Given an edge (u, v), let $u', v' \notin \mathcal{N}_u \cup \mathcal{N}_v$ be such that $(u', v') \in \mathcal{G}$, $Q = \mathcal{N}_u^v \cap \mathcal{N}_{u'}^{v'} \neq \emptyset$, and, similarily, $Q' = \mathcal{N}_{v'}^{u'} \cap \mathcal{N}_v^u \neq \emptyset$.

Theorem 5 (Distance 2): The weight change of $\mathcal{G}$ under ELC on $\{(u, v), (u', v')\}$ is upper-bounded by a threshold T *iff,*

$$\Delta\mathcal{E}_{u,v} + \Delta\mathcal{E}_{u',v'} - 2\Delta\mathcal{E}_{Q',Q} \leq T.$$

This case covers all instances of depth-2 WB-ELC where the edges are at a distance two apart.

Proof: See Fig. 3, and [16] for a detailed proof. ■

We now consider distance one. Given an edge (u, v) and two nodes u' and v', we denote by $B = \mathcal{N}_v^{u,u'} \cap \mathcal{N}_{v'}^{u,u'}$, $A = \mathcal{N}_v^{u,u'} \setminus B$, $C = \mathcal{N}_{v'}^{u,u'} \setminus B$, $E = \mathcal{N}_u^{v,v'} \cap \mathcal{N}_{u'}^{v,v'}$, $D = \mathcal{N}_u^{v,v'} \setminus E$, and $F = \mathcal{N}_{u'}^{v,v'} \setminus E$, see Fig. 4. We consider the case where both u' and v' are in the neighborhood of (u, v), and where $(u', v') \notin \mathcal{G}$ is created by the first ELC.

Theorem 6 (Distance 1): The weight change of $\mathcal{G}$ under ELC on $\{(u, v), (u', v')\}$ is upper-bounded by a threshold T *iff,*

$$\Delta\mathcal{E}_{A,E\cup F} + \Delta\mathcal{E}_{B,D\cup E} + \Delta\mathcal{E}_{C,D\cup F} + |C| + |F| - |B| - |E| \leq T.$$

This case covers all instances of depth-2 WB-ELC where the edges are at distance one apart.

Proof: See Fig. 4, and [16] for a detailed proof. ■

We have shown that, for $T \geq -1$, the depth-2 WB-ELC cases must occur on pairs of edges spaced by distance at most two. Let us now for completeness consider $T < -1$.

Proposition 4: Let $T < -1$. In this case a pair of edges spaced by a distance of more than two may give depth-2 WB-ELC that does not reduce to (neither a single, nor a double instance of) depth-1 WB-ELC.

Proof: A small example proves the proposition. For $T = -2$, two independent ELC operations may each reduce the weight by -1. ■

[3] A special case exists for $T < -1$, which is accounted for in Proposition 4.

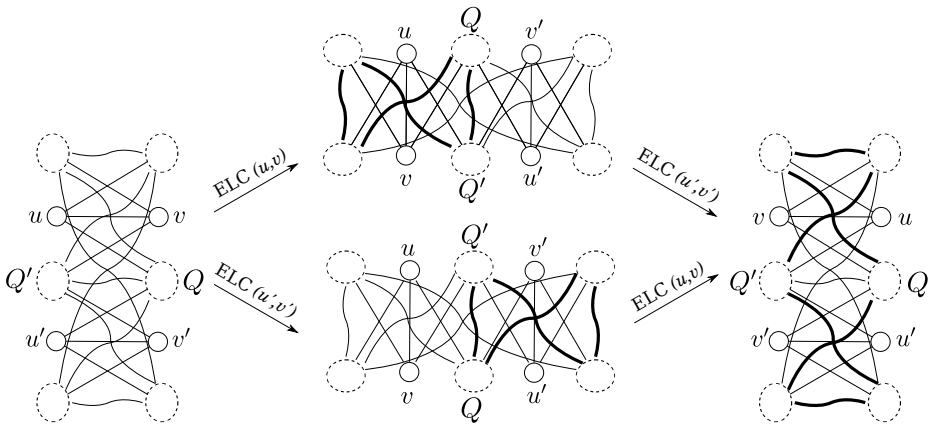


Fig. 3. Proof of Theorem 5. Curved links indicate arbitrary edges; bold links indicate complemented edges. A special case of commutativity gives the equivalent sequence, $(u', v'), (u, v)$; although the local subgraphs $\mathcal{E}_{u,v}, \mathcal{E}_{u',v'}$ are *not* independent, the overlap is confined to Q, Q' (which is complemented twice) [19].

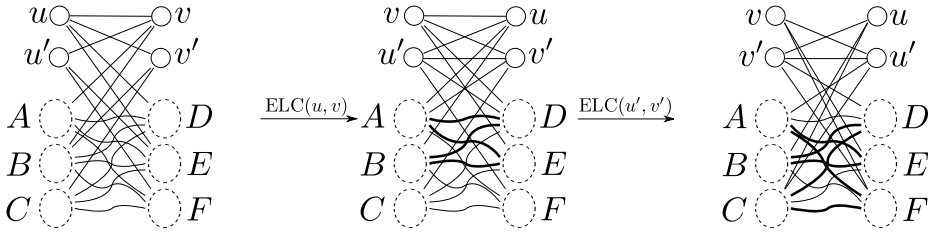


Fig. 4. Proof of Theorem 6 (using one of the three equivalent cases described in [16]) showing the complementations which give the expression.

IV. ELC-Enhanced SISO HDPC Decoding

For this work, the most important application is the use of WB-ELC operations during SISO HDPC decoding, where the aim is to have increased diversity, i.e., more distinct Tanner graphs for the same code which are all well-suited for use in iterative decoding. Other applications are discussed in [16]. Several parameters of a parity-check matrix affect its suitability for decoding, where one of these is the weight, or density, of the matrix. Let the received noisy channel vector be $\mathbf{y} = (-1)^{\mathbf{x}} + \mathbf{n}$, where $\mathbf{x}$ is a codeword and $\mathbf{n}$ is additive white Gaussian noise (AWGN). In the log-likelihood ratio (LLR) domain, the initial LLR at position v is $L_0^v \triangleq \frac{2}{\eta^2} y_v$, where η is the standard deviation of the AWGN.

A. Generalized SISO HDPC Decoder

The idea of using permutations (from a cyclic subgroup of $\mathrm{Aut}(\mathcal{C})$) to gain diversity during iterative decoding originates from [10]. This was recently generalized to using the full $\mathrm{Aut}(\mathcal{C})$ (and for noncyclic codes) with the random redundant iterative decoder (SPA-PD) in [9]. It consists of three nested loops. After I_1 SPA (flooding) iterations, a random permutation from $\mathrm{Aut}(\mathcal{C})$ is applied (to the input vector, $\mathbf{L}$) followed by a damping stage [10]. This is repeated I_2 times, before the damping coefficient, α, is incremented and the decoder restarts from $\mathbf{y}$. This can be thought of as making I_2 new attempts at decoding $\mathbf{y}$, with increased damping coefficient. This is all repeated I_3 times, and unless SPA converges to a valid codeword within $\tau = I_1 I_2 I_3$ iterations, the decoder outputs a failure.

Generalizing this algorithm, we propose a generalized SISO HDPC decoder. The permutation may be replaced by any operation to achieve diversity (e.g., random ELC or WB-ELC), and we do p such operations at a time. Using this framework, we propose the novel SPA-ELC and SPA-WBELC decoders. While the SPA-ELC decoder may do ELC on any edge in $\mathcal{G}$, the SPA-WBELC decoder must search the graph during decoding for a WB-ELC operation (which is either one or two ELC operations). As we have discussed, the search space is significantly reduced from searching all pairs of edges in the graph. Further heuristics are used to improve search time, and the search stops as soon as the first (random) WB-ELC operation is found. We refer to [16] for a detailed description and theoretical analysis of a search algorithm.

The most important difference between SPA-PD and ELC-enhanced decoding (SPA-ELC and SPA-WBELC) is that ELC does not require any specific structural properties of the code. As n increases, the probability of a randomly chosen code of blocklength n to have a nontrivial $\mathrm{Aut}(\mathcal{C})$ goes to zero (when the rate is not too high, or too low) [22]. So among the main contributions of this work is in this sense to extend the range of SISO HDPC decoding to codes for which SPA-PD does not work ("reduces" to SPA). Compared to other decoding algorithms, we emphasize how SPA-ELC does not require any preprocessing – not counting the search for a reduced or minimum-weight initial graph/matrix, as this is a common preprocessing stage and not part of the decoding algorithm. For SPA-WBELC, an initial Tanner graph should also be verified to have a sufficiently large (in terms of diversity) "sparse

sub-orbit." This sub-orbit is understood as an initial Tanner graph, $\mathbf{TG}(H)$, and all distinct Tanner graphs reachable via (repeated action of) WB-ELC, all within some threshold, T, i.e., graphs of weight upper-bounded by $|H| + T$. The size of this sub-orbit will depend on $\mathbf{TG}(H)$ and T, so these are determined in a preprocessing stage [16]. Both SPA-ELC and SPA-WBELC are online algorithms, based on local decisions, and no memory overhead is incurred (the graph is modified in-place, as opposed to storing multiple redundant matrices, e.g., as in multiple-bases belief propagation (MBBP) decoding [23]).

B. Edge-Local Damping Rule

The purpose of damping is to scale down the extrinsic contribution (i.e., messages on edges), typically to moderate the impact of some global change to the graph [7], [9], [10]. Every I_1 iterations, a diversity stage is executed in which the extrinsic contribution of the LLRs, Γ_j^v, of each variable node v at iteration j, is scaled down by a damping coefficient, α, $0 < \alpha < 1$, and accumulated on the input to the next iteration according to the *damping rule* $L_{j+1}^v = L_j^v + \alpha\Gamma_j^v$. The extrinsic contribution to variable node v (the sum of all incoming messages, $\mu_j^{v\leftarrow u}$) in iteration j is,

$$\Gamma_j^v = \sum_{u\in\mathcal{N}_v} \mu_j^{v\leftarrow u} \tag{3}$$

where we define $\Gamma_0^v \triangleq 0$. The initial contribution from the received noisy channel vector is never damped, which is apparent if we rewrite $L_{j+1}^v = L_0^v + \alpha\Sigma_{j'=1}^{j}\Gamma_{j'}^v$. These new, damped LLRs are then used to re-initialize the decoder. So, after *resetting* all messages, $\mu_j^{v\leftarrow u} := 0\ \forall\ (u,v) \in \mathcal{G}$, iteration $j+1$ begins by forwarding the new, damped input towards the check nodes. This "global reset stage" is necessary when the operation used in the SISO HDPC decoder acts on the variable node level, e.g., as in SPA-PD, which permutes $\mathbf{L}$ [9]. After this, the relationship (3) between extrinsic information (on edges) and LLRs (in nodes) no longer holds. The global stage of accumulating the input followed by re-initializing all edges, is referred to as *global damping* (GD). In contrast to GD, we have previously proposed edge-local damping schemes more suited to the edge-local action of ELC [12], [24]. The damping rule can be generalized to include and take advantage of extrinsic information on an edge (u,v), $\mu_j^{v\leftarrow u}$, in iteration j;

$$\mu_{j+1}^{v\to u} = L_j^v + \alpha(\Gamma_j^v - \mu_j^{v\leftarrow u}). \tag{4}$$

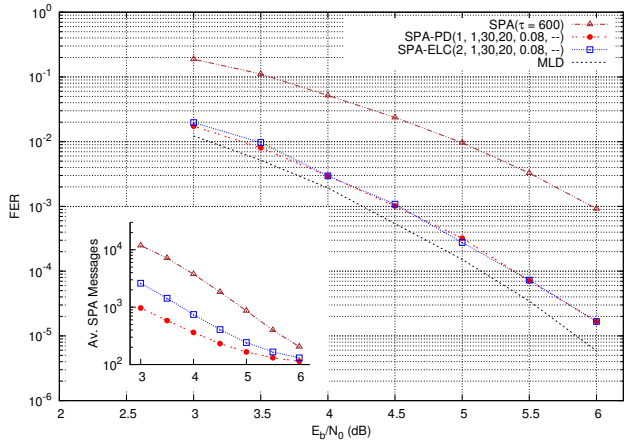
Each edge adjacent to v is damped individually. Note how $\mu_j^{v\leftarrow u}$ is subtracted, to adhere to the extrinsic principle of the SPA. Thus, less information is lost than is the case with GD.

ELC on an edge (u',v') complements the edges of $\mathcal{E}_{u',v'}$ – the "internal" edges with both endpoints in $\mathcal{N}_{u'}^{v'} \cup \mathcal{N}_{v'}^{u'}$. By defining a flooding SPA iteration as the update of all check nodes followed by all variable nodes, we ensure that all soft information (on edges) is stored in Γ^v, for all variable nodes v, before ELC. Thus, the information loss due to edges *removed* by ELC is reduced, and we need only focus on edges *inserted* by ELC; precisely $(u,v) \in \mathcal{E}_{u',v'}$. These new edges must be initialized with some outgoing message, $\mu_{j+1}^{v\to u}$, before the next SPA iteration (iteration $j+1$, which begins with check nodes, u), so (4) implements a damping-and-initialization rule. However, since $\mu_j^{v\leftarrow u} = 0$ for new edges, (4) reduces to $\mu_{j+1}^{v\to u} = L_j^v + \alpha\Gamma_j^v = L_{j+1}^v$ (GD). We emphasize that edges connected to $\mathcal{N}_v \backslash \mathcal{N}_{v'}$, i.e., those unaffected by ELC on (u',v'), are *not* damped and retain their extrinsic messages for the next iteration. Restricting damping to the edges affected by ELC is referred to as *edge-local damping* (LD) [12].

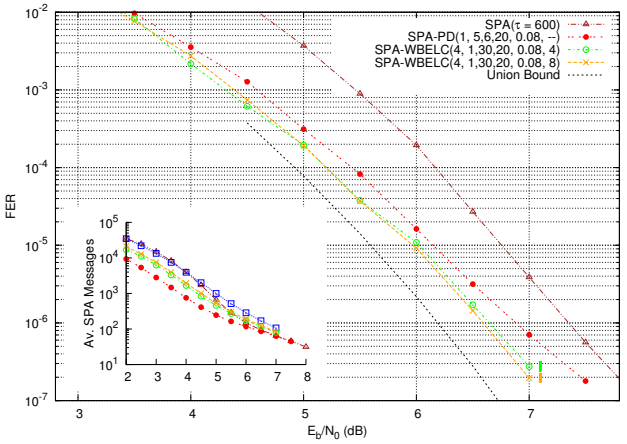
C. Error-Rate Observations

We will show the effectiveness of the proposed ELC-based decoding algorithms, SPA-ELC and SPA-WBELC, by comparing against the benchmark SPA-PD algorithm. These are all implemented using the generalized SISO HDPC decoder. For all algorithms, we ensure the same maximum number of SPA iterations, $\tau = I_1 I_2 I_3$. In the diversity stage (every I_1 iterations), p random operations are applied. These can be permutations from $\mathrm{Aut}(\mathcal{C})$ (as in SPA-PD), or ELC operations (recall that one WB-ELC operation consists of one or two ELC operations). The values of p, I_1, I_2, and I_3 are chosen empirically, based on frame error-rate (FER) simulations. As discussed in [16], the performance is most sensitive to p and I_1, and optimal performance appears to be when these are both low. To emphasize the effect of various operations, we also compare against the standard SPA decoder. The most general observation is that SPA decoding of HDPC codes benefits from increased diversity, see Fig. 5. For all codes and decoders simulated, we observe a significant gain in FER over SPA decoding, especially in the high signal-to-noise ratio (SNR) region (for a lowered "error floor").

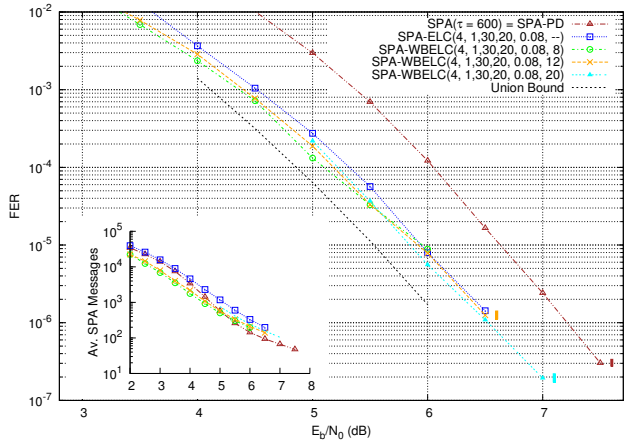
Since any ELC operation must either preserve the graph (up to isomorphism) or give a different graph from the orbit, a small orbit must imply a large (relative to code size) $\mathrm{Aut}(\mathcal{C})$. Generally, SPA-ELC can be described as a combination of SPA-PD (when ELC is iso-ELC) and MBBP (otherwise). The extended Hamming and Golay codes are famous examples of codes with very strong structure (orbit size 1 and 2, respectively), and we see that the performance of SPA-ELC matches closely that of SPA-PD (see Fig. 5(a) and also [20]). From a decoding perspective, diversity is in terms of Tanner graphs. We can express the probability of gaining diversity when using SPA-PD by $1 - |\mathcal{D}|/|\mathrm{Aut}(\mathcal{C})|$; i.e., the probability of not drawing a trivial permutation from $\mathrm{Aut}(\mathcal{C})$. For such strongly structured codes, the size of $\mathrm{Aut}(\mathcal{C})$ will ensure good diversity. Using SPA-ELC, any ELC operation will swap a pair of columns between $\mathcal{I}$ and $\mathcal{P}$ (in H), and will thus necessarily give a distinct Tanner graph. However, a sequence of $p > 1$ ELC operations may cancel, and give the same Tanner graph (no diversity); $p-1$ ELC operations work to "undo" the swap induced by the first ELC. This is to restore the I-part of H, and thus (by extension) restoring the initial H (see Alg. 1 and Example 2). The probability of diversity is $1 - D(p)/S(p)$, where $D(p)$ is the number of such "redundant" (i.e., nonminimal) length-p ELC-sequences, and $S(p)$ is the total number of (possibly redundant) length-p ELC-sequences encountered in a depth-first search (on some graph). Recall



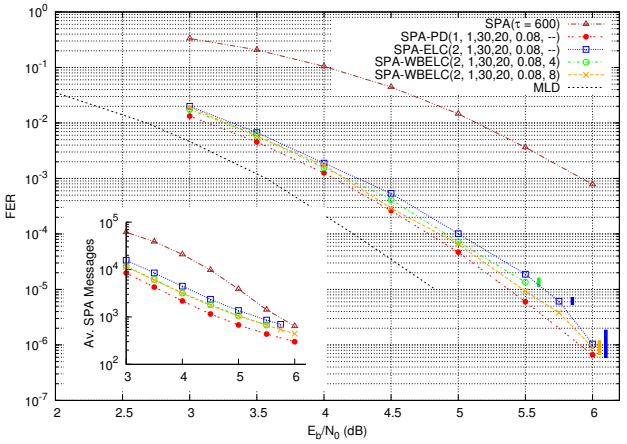
(a) Ext. Golay $= [24, 12, 8]$, with $|\mathrm{Aut}(\mathcal{C})| = 244\,823\,040$



(b) $R2 = [36, 18, 8]$, with $|\mathrm{Aut}(\mathcal{C})| = 32$



(c) $\mathcal{C}_{38,2} = [38, 19, 8]$, with $|\mathrm{Aut}(\mathcal{C})| = 1$



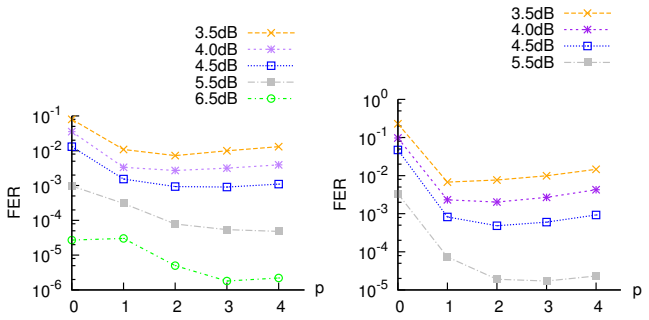
(d) EQR48 $= [48, 24, 12]$, with $|\mathrm{Aut}(\mathcal{C})| = 51\,888$. MLD data provided by Alban Goupil.

Fig. 5. Simulation results. Each SNR point is simulated until at least 100 frame-error events are observed (otherwise, error bars indicate a 95% confidence interval [25]). The union bound is calculated based on the full weight enumerator of the code. Parameters are listed as $(p,\ I_1,I_2,I_3,\ \alpha,\ T)$.

that the extended Golay code has only two graphs in its orbit (Table II). For $\mathcal{G}_0$ we count $D(1) = 0$, $S(1) = 84$; $D(2) = 84$, $S(2) = 7152$; and $D(3) = 1008$, $S(3) = 608\,640$. Similarily, for $\mathcal{G}_1$, $D(1) = 0$, $S(1) = 88$; $D(2) = 88$, $S(2) = 7480$; and $D(3) = 1144$, $S(3) = 636\,592$. Using SPA-ELC, the probability of diversity remains quite high, also as we increase p, which gives us the additional benefit of (implicitly) running SPA-PD on both graphs. For this code, no additional gain can be achieved by SPA-WBELC ($|\mathcal{G}|$ is either 96 or 100).

For such strongly structured codes (where $\mathrm{Aut}(\mathcal{C})$ is large), SPA-PD is known to perform well [9]. ELC-enhanced decoding, however, acting on the entire orbit of the code, can be made effective on a greater range of codes. When the orbit is large, the probability of iso-ELC becomes negligible, and SPA-ELC is reminiscent of an online, local-action MBBP. Consider the extremal (in terms of $d_{\min}$) self-dual $[36, 18, 8]$ "$R2$" code [26] in Fig. 5(b). This code has a small $\mathrm{Aut}(\mathcal{C})$, of size $|\mathrm{Aut}(\mathcal{C})| \approx n$, which thus hampers the performance of SPA-PD. For this code, we observe a consistent gain (over the entire SNR range simulated) of SPA-ELC, especially by removing a floor effect. The optimal value of p is seen in Fig. 6(a) to be 3. The gain due to improved diversity depends on the quality of the resulting Tanner graphs. SPA decoding is sensitive to short cycles and, more generally, an increase in graph weight (number of edges). This code has a large orbit, so we can not expect all graphs to be well-suited for SPA decoding – and it is easily verifiable that they are not [16]. Especially at the low-SNR range we observe a gain by using SPA-WBELC over SPA-ELC. This demonstrates the benefit of restricting decoding to a sparse sub-orbit of the code.

As an example of a nonrandom, constructed HDPC code with a trivial $\mathrm{Aut}(\mathcal{C})$ we consider the $[38, 19, 8]$ "$\mathcal{C}_{38,2}$" code [26] in Fig. 5(c). This code is related to the "$R2$" code, and has otherwise very similar parameters and properties. The most important practical result of this paper is that we find the same (large) gain over SPA as for "$R2$" – despite the trivial $\mathrm{Aut}(\mathcal{C})$. This verifies the benefits of ELC-based decoding on codes less suited for SPA-PD. For this code we observe a more consistent gain for SPA-WBELC over SPA-ELC, especially at the low-SNR range. The break-off point where SPA-WBELC converges with SPA-ELC depends on the WB-ELC threshold, T. As we increase T, we allow graphs of higher weight to participate in the decoding process. Yet the search complexity of WB-ELC is obviously lower for less restrictive (i.e., higher)



(a) "$R2$" (b) "EQR48"

Fig. 6. Details for SPA-ELC with $I_1 = 1$, $I_2 = 30$, and $I_3 = 20$. Here, $p = 0$ denotes SPA decoding (with no damping). p may be increased to slightly reduce flooring effects.

thresholds [16]. Eventually, as we increase T sufficiently, the weight is no longer bounded (compared to "unbounded" SPA-ELC). So, with increasing T, the break-off point is shifted to a higher SNR but the low-SNR gain is reduced, as SPA-WBELC "reduces" to SPA-ELC.

We also consider the $[48, 24, 12]$ extended QR (EQR) code, denoted by "EQR48", as a next step from the extended Golay code but for which the orbit size is large. Correspondingly, $\mathrm{Aut}(\mathcal{C})$ is relatively small, containing "only" 51 888 permutations. This is nevertheless more than sufficient to ensure a strong performance of SPA-PD, which is only 0.5dB to 1dB away from optimal maximum likelihood decoding (MLD). Yet, simply by interspersing SPA iterations with ($p = 2$; see Fig. 6(b)) random ELC operations, we achieve a performance only $\sim$0.25dB away from SPA-PD (and even closer below an SNR of 4dB). However, the weight increase due to ELC has an adverse effect on decoding performance, so to close this gap we use WB-ELC. The minimum weight of any graph of this code is 288, and we are able to find many distinct minimum-weight Tanner graphs (including nonisomorphic simple graphs) [16]. Fig. 5(d) shows how the peformance of SPA-WBELC depends on T in a similar way as for "$R2$."

We also compare against some other decoding algorithms (not included in Fig. 5). A simple scheme running SPA on seven distinct minimum-weight matrices for the extended Golay code gives an improvement over SPA [27]. We observe a performance gain of $\sim$0.5dB at bit-error rate 10^{-4} over this scheme (we still observe a gain of $\sim$0.25dB when we limit SPA-ELC to $\tau = 200$ iterations). We also observe an improvement in error-rate on this code over the more advanced MBBP algorithm, which uses 15 $n \times n$ matrices (based on cyclic shifts of minimum-weight codewords in $\mathcal{C}^\perp$) in a parallel (i.e., list) decoding scheme [23]. At FER $3 \cdot 10^{-3}$ we observe a gain of $\sim$0.2dB when using $\tau = 600$ iterations. In addition to this improvement in performance, we also achieve a significant reduction in complexity, by avoiding parallelism, by using fewer iterations (they use a maximum of 1 050 iterations), and by avoiding the storage (in memory) of redundant parity-check matrices.

D. Complexity Observations

We also report on simulations to determine the average complexity of the various decoding algorithms. The SPA-ELC and SPA-WBELC decoders use a systematic matrix and modify the corresponding graph during decoding, whereas the SPA and SPA-PD decoders use a single, optimized (reduced-weight) nonsystematic matrix. Since the weight varies under ELC decoding, the complexity cannot be reported simply in terms of the average number of iterations per codeword. However, the complexity of all stages of SPA decoding and of the ELC operation is proportional to the number of edges involved, so decoding complexity may be measured by the average number of SPA messages [11], [28]. In terms of messages, the complexity of one (flooding) SPA iteration is $2(|\mathcal{G}| + n - k) = 2(k\bar{\gamma} + n - k)$. For the following argument, we assume that $k = n - k$. At "50% weight" the complexity of one SPA iteration is $2k(\bar{\gamma} + 1) = 2k(k/2 + 1) = k^2 + 2k$, which is significantly higher (by at least a factor of 4) than the ELC complexity, $k^2/4 - k + 1$, from (2). As such, we do not take the overhead of applying ELC operations into account in the comparisons.

For complexity, we observe the desired effect of bounding the weight increase due to ELC. Fig. 5 (inset plots) indicates a general trend where the SPA-PD decoder has the lowest complexity, while the SPA is the most complex decoder. As these two algorithms use the exact same graph (for a given code), any difference must be entirely in terms of number of iterations used per codeword. In other words, this shows how the SPA-PD is an important benchmark, as it gives an improvement in both FER and complexity. Similarily, our proposed SPA-WBELC algorithm also gives an improvement in complexity, over SPA and SPA-ELC, and is not far from this benchmark. The complexity improvement over SPA-ELC is a direct benefit obtained from bounding weight.

The complexity of finding WB-ELC operations, given a graph and a threshold, is analyzed theoretically and empiricially in [16]. However, for practical use in the SPA-WBELC decoding algorithm, the search may be terminated upon finding the first occurrence of a WB-ELC operation. Simulations show that finding a random depth-2 WB-ELC operation on the "EQR48" code with $T = 8$ requires only an average of 150 edges checked per iteration (where each "check" corresponds roughly to one ELC operation). This drops to 50 edges for $T = 12$, and 20 edges for $T = 16$. For comparison, Gaussian elimination (as used in [7] and [24]) can be implemented using $n - k = 24$ ELC operations [16]. So this is not an unmanageable overhead, and we also assume better heuristics can be designed.

Conclusion

In this work, we have presented a mapping from a Tanner graph to a bipartite simple graph so as to facilitate the use of a graph operation known as ELC during iterative, graph-based decoding. ELC modifies locally the structure (i.e., the edges) of a graph, without changing the associated code, thus generating the entire orbit (all systematic parity-check matrices) of the code. We have identified and described how ELC may induce graph isomorphism, and how this is linked to code automorphism, i.e., to $\mathrm{Aut}(\mathcal{C})$. We have also defined a notion of Tanner graph isomorphism (row-equivalence of parity-check

matrices), and shown the relationship to the corresponding trivial (in terms of decoding) subgroup of $\mathrm{Aut}(\mathcal{C})$. This gives a natural relationship with SPA-PD (a state-of-the-art decoding algorithm for HDPC codes) which improves decoding by employing random permutations from $\mathrm{Aut}(\mathcal{C})$ during decoding.

The concept of isomorphic ELC operations has been generalized to a weight-bounding application of ELC, WB-ELC. All possible instances of WB-ELC due to single and double application of ELC on a graph are classified, where we show that all double instances occur on adjacent edges. This locality improves search time (to find a random WB-ELC operation on a graph). We described the usage of ELC (and WB-ELC) to improve iterative SISO decoding of HDPC codes. Generally, the orbit of a code contains many matrices which are less suitable for SPA decoding (specifically, non-sparse matrices), so the generalization to WB-ELC is a valuable extension of the scope of SISO HDPC decoding. To facilitate the convergence of the decoder, we also proposed a novel edge-local damping rule, tailored to our graph-local context. Extensive simulation data showed a consistent gain of SPA-ELC and SPA-WBELC over SPA, and that SPA-WBELC competes closely with the performance of SPA-PD when $\mathrm{Aut}(\mathcal{C})$ is large and outperforms SPA-PD when $\mathrm{Aut}(\mathcal{C})$ is small or trivial.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

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

[4] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[5] T. R. Halford, A. J. Grant, and K. M. Chugg, "Which codes have 4-cycle-free Tanner graphs?," *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 4219–4223, Sep. 2006.

[6] J. S. Yedidia, J. Chen, and M. P. C. Fossorier, "Generating code representations suitable for belief propagation decoding," in *Proc. 40th Allerton Conf. Commun., Contr., and Comp.*, Monticello, IL, Oct. 2002, pp. 447–456.

[7] J. Jiang and K. R. Narayanan, "Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix," *IEEE Trans. Inform. Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.

[8] W. Jin and M. P. C. Fossorier, "Reliability-based soft-decision decoding with multiple biases," *IEEE Trans. Inform. Theory*, vol. 53, no. 1, pp. 105–120, Jan. 2007.

[9] T. R. Halford and K. M. Chugg, "Random redundant iterative soft-in soft-out decoding," *IEEE Trans. Commun.*, vol. 56, no. 4, pp. 513–517, Apr. 2008.

[10] J. Jiang and K. R. Narayanan, "Iterative soft decoding of Reed-Solomon codes," *IEEE Commun. Lett.*, vol. 8, no. 4, pp. 244–246, Apr. 2004.

[11] I. Dimnik and Y. Be'ery, "Improved random redundant iterative HDPC decoding," *IEEE Trans. Commun.*, vol. 57, no. 7, pp. 1982–1985, Jul. 2009.

[12] J. G. Knudsen, C. Riera, L. E. Danielsen, M. G. Parker, and E. Rosnes, "Iterative decoding on multiple Tanner graphs using random edge local complementation," in *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, Jun./Jul. 2009, pp. 899–903.

[13] T. Hehn, J. Huber, O. Milenkovic, and S. Laendner, "Multiple-bases belief-propagation decoding of high-density cyclic codes," *IEEE Trans. Commun.*, vol. 58, no. 1, pp. 1–8, Jan. 2010.

[14] A. Bouchet, "Isotropic systems," *European J. Comb.*, vol. 8, pp. 231–244, Jul. 1987.

[15] L. E. Danielsen and M. G. Parker, "Edge local complementation and equivalence of binary linear codes," *Des. Codes Cryptogr.*, vol. 49, no. 1-3, pp. 161–170, Dec. 2008.

[16] J. G. Knudsen, C. Riera, L. E. Danielsen, M. G. Parker, and E. Rosnes, "Random edge-local complementation with applications to iterative decoding of HDPC codes," Department of Informatics, University of Bergen, Norway, Tech. Report no. 395, Aug. 2010.

[17] B. D. McKay, "Nauty; software for computing automorphism groups of graphs and digraphs," Web page, 2007, http://cs.anu.edu.au/people/bdm/nauty/.

[18] L. E. Danielsen, M. G. Parker, C. Riera, and J. G. Knudsen, "On graphs and codes preserved by edge local complementation," 2010, arXiv:1006.5802.

[19] R. Brijder, T. Harju, and H. J. Hoogeboom, "Pivots, determinants, and perfect matchings of graphs," 2008, arXiv:0811.3500.

[20] J. G. Knudsen, C. Riera, M. G. Parker, and E. Rosnes, "Adaptive soft-decision decoding using edge local complementation," in *Proc. Second Int. Castle Meeting on Coding Theory and Applications, LNCS 5228*, Castillo de la Mota, Medina del Campo, Spain, Sep. 2008, pp. 82–94.

[21] R. Arratia, B. Bollobás, and G. B. Sorkin, "The interlace polynomial of a graph," *J. Comb. Theory, Series B*, vol. 92, no. 2, pp. 199–233, Nov. 2004.

[22] H. Lefmann, K. T. Phelps, and V. Rödl, "Rigid linear binary codes," *J. Comb. Theory, Series A*, vol. 63, no. 1, pp. 110–128, May 1993.

[23] T. Hehn, J. B. Huber, S. Laendner, and O. Milenkovic, "Multiple-bases belief-propagation for decoding of short block codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Nice, France, Jun. 2007, pp. 311–315.

[24] J. G. Knudsen, C. Riera, L. E. Danielsen, M. G. Parker, and E. Rosnes, "Improved adaptive belief propagation decoding using edge-local complementation," in *Proc. IEEE Int. Symp. Inform. Theory*, Austin, TX, Jun. 2010, pp. 774–778.

[25] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[26] M. Harada, "New extremal self-dual codes of lengths 36 and 38," *IEEE Trans. Inform. Theory*, vol. 45, no. 7, pp. 2541–2543, Nov. 1999.

[27] K. Andrews, S. Dolinar, and F. Pollara, "LDPC decoding using multiple representations," in *Proc. IEEE Int. Symp. Inform. Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 456.

[28] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.