

Tanner Graphs for Group Block Codes and Lattices: Construction and Complexity

Amir H. Banihashemi, *Associate Member, IEEE*, and Frank R. Kschischang, *Senior Member, IEEE*

Abstract—We develop a Tanner graph (TG) construction for an Abelian group block code \mathbf{L} with arbitrary alphabets at different coordinates, an important application of which is the representation of the label code of a lattice. The construction is based on the modular linear constraints imposed on the code symbols by a set of generators for the dual code \mathbf{L}^* .

As a necessary step toward the construction of a TG for \mathbf{L} , we devise an efficient algorithm for finding a generating set for \mathbf{L}^* . In the process, we develop a construction for lattices based on an arbitrary Abelian group block code, called generalized Construction A (GCA), and explore relationships among a group code, its GCA lattice, and their duals.

We also study the problem of finding low-complexity TGs for Abelian group block codes and lattices, and derive tight lower bounds on the label-code complexity of lattices. It is shown that for many important lattices, the minimal label codes which achieve the lower bounds cannot be supported by cycle-free Tanner graphs.

Index Terms—Dual code, generalized Construction A, group codes, lattices, Tanner graphs, Tanner graph complexity, Tanner graph construction.

I. INTRODUCTION

THE study of codes defined on graphs is currently of great interest in coding theory, mainly because of the superb performance which can be achieved with practical decoding complexity. In this paper, we study the construction and complexity of Tanner graphs (TGs) for finite Abelian group block codes (“group codes” for short), an important application of which is the representation of lattice label codes. One of the main contributions of this work is to represent a group code with a set of modular linear equations, that can in turn be used for a graph representation of the code. We also establish relationships among group codes, lattices, and their duals. At the core of these relationships, we introduce “generalized Construction A” for lattices based on an arbitrary group code, and use this to

develop an efficient algorithm for finding a generating set for the dual of the group code. This is a necessary step toward the construction of a TG for the code. Complexity results are also obtained which support the conjecture that “good lattices cannot be represented by cycle-free TGs.”

Bipartite graph representations for codes begin with the work of Tanner [24], who generalized Gallager’s low-density parity-check (LDPC) codes [13] to codes defined by general bipartite graphs, where the two types of nodes represent the symbols and the linear constraints, respectively. Tanner also developed two types of algorithms, here called *min-sum* and *sum-product*, for the decoding of the corresponding code, and proved that they converge on finite cycle-free graphs. These algorithms are also called *two-way algorithms*, since they perform the decoding by passing the information along the edges of the graph in both directions.

The graph representation of codes and the corresponding decoding algorithms have continued to be an active area of research. A major step was taken in [27], [28], where the authors extended TGs to include *hidden* nodes. This established a bridge to the extensive literature on the trellis representation of codes. In this light, many well-known decoding algorithms such as the Viterbi algorithm and the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm can now be seen as special cases of the min-sum and sum-product algorithms that arise when the underlying graph is a trellis. Other soft-output decoding algorithms such as the soft-output Viterbi algorithm are also closely related to this class of algorithms. In [27], it was shown that the standard decoding algorithms for turbo codes, LDPC codes, and tail-biting codes also fit into this category.

Generalizations of graph-based decoding algorithms to the cases where the set of symbol costs form a semiring with two binary operations have been considered in [20], [27]. It has been realized [16], [19], [21] that many of these algorithms may be viewed as versions of “probability propagation” in various graphical models of the code, such as TGs, Bayesian networks, and Markov random fields. Most recently, these graphical representations and the corresponding algorithms, as well as a variety of algorithms developed in artificial intelligence, statistics, and signal processing, have been brought under the same umbrella and are discussed as distributed “message-passing” algorithms in a *factor graph* [17]. For a history of graph representation of codes and decoding algorithms, see [10]. More recent results on the construction and complexity of factor graphs of codes and the performance analysis of iterative decoding can be found in [1], [5], [8], [12], [15], [18], [23], and [25].

The application of the two-way algorithm, in either its min-sum or sum-product form, has proved to be very efficient in

Manuscript received November 29, 1999; revised September 29, 2000. This work was supported in part by an NSERC Postdoctoral Fellowship. The material in this paper was presented in part at the 1998 IEEE International Symposium on Information Theory, MIT, Cambridge, MA, August 16–21, 1998, and in part at the 1999 Canadian Workshop on Information Theory, Queen’s University, Kingston, ON, Canada, June 15–18, 1999.

A. H. Banihashemi was with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 1A4, Canada. He is now with the Department of Systems and Computer Engineering and Broadband Communications and Wireless Systems (BCWS) Centre, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: ahashemi@sce.carleton.ca).

F. R. Kschischang is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: frank@comm.utoronto.ca).

Communicated by G. D. Forney, Jr., Guest Editor.
 Publisher Item Identifier S 0018-9448(01)00731-3.

the decoding of TGs [10], [27], [28]. For some codes, such as turbo codes and LDPC codes, iterative decoding, though suboptimal, is the only practical means of decoding the code. Even for the optimal (maximum-likelihood) decoding, some codes have graphs which lead to decoding algorithms with smaller complexity than the application of the Viterbi algorithm to the least complex trellis [22]. Despite the excellent performance of these techniques and their reasonable complexity in many applications, little is known so far in general about the analysis and synthesis of codes based on graphs and the corresponding iterative decoding algorithms.

Historically, and partly because of their more complicated structure, lattices became an active subject of research in the coding community only after the corresponding problems were solved for linear block codes. For lattices, therefore, even less is known about the nontrellis graph representations. Nevertheless, the increasing interest in lattice codes for signaling over band-limited channels and for vector quantization, and also the capability of lattice codes to achieve channel capacity [7], make the study of such problems of importance.

For nontrellis TGs of lattices, the only work of which we are aware is [26]. In [26], a TG construction for lattices, using a method different from the one discussed here, was briefly sketched. In this work, we elaborate the method of [26], and establish relationships between the two constructions. In general, the method of [26] appears to be computationally more complex in searching for simple TGs of a lattice.

The construction of a TG for a linear block code is well known, and is based on using a parity-check matrix of the code [24], [27]. In [8], the authors derive upper bounds on the minimum distance of linear block codes that can be represented by cycle-free TGs, and show that cycle-free TGs cannot support good codes. In this work, we develop a TG construction for the more general category of Abelian group block codes (here simply called group codes) with arbitrary alphabets at different coordinates. An important application of this construction is to represent the label code of a lattice.

In Section II, we consider group codes that are subgroups of $\mathbf{G} = Z_{g_1} \times \cdots \times Z_{g_n}$, where Z_{g_i} is the additive cyclic group of integers modulo g_i . We then define an inner product (or pairing) between elements of \mathbf{G} . Given any subgroup \mathbf{L} of \mathbf{G} , the set \mathbf{L}^* of elements of \mathbf{G} whose inner product with each element of \mathbf{L} is zero is a subgroup of \mathbf{G} that can be viewed as the dual of \mathbf{L} . Our approach here is closely related to the conventional way of defining the dual group as a subgroup of the character group of \mathbf{G} . However, it is somewhat easier to apply, since the dual code is drawn from the same underlying group. Using this approach, we then construct a TG for \mathbf{L} based on the modular linear constraints that a set of generators for \mathbf{L}^* imposes on the symbols of \mathbf{L} . Structural properties of group codes such as trimness and fully dynamical property are also discussed in this section, and it is shown that \mathbf{L} is trim if and only if \mathbf{L}^* is fully dynamical. This is then used for discussions on the TG structure of isomorphic codes.

In Section III, using the label code of a lattice, we apply the TG construction of Section II to lattices. The comparison with the construction of [26] is also given in this section.

In Section IV, we describe an efficient algorithm that computes a generating set for \mathbf{L}^* , given the codewords for \mathbf{L} . This is done through the introduction of generalized Construction A (GCA) for lattices.

In Section V, we study the complexity of TGs for group codes and lattices. Tight lower bounds on lattice label-code complexity are derived, and it is shown that minimal TGs for many important lattices do have cycles.

II. ABELIAN GROUP BLOCK CODES

A. Definitions

Let \mathbf{L} be an Abelian group block code defined over the alphabet sequence space $\mathbf{G} = G_1 \times \cdots \times G_n$, where G_i is the finite Abelian group representing the i th alphabet of the code. Since any finite Abelian group can be expressed as a direct product of finite cyclic groups, without loss of generality, we assume that G_i is cyclic. Suppose that $|G_i| = g_i$, $i = 1, \dots, n$. The group G_i is then isomorphic to the additive cyclic group of integers modulo g_i ($Z_{g_i} = \{0, 1, \dots, g_i - 1\}$), so we assume that $\mathbf{G} = Z_{g_1} \times Z_{g_2} \times \cdots \times Z_{g_n}$ under componentwise addition, and that \mathbf{L} is a subgroup of \mathbf{G} . We also use the notation A to denote a general Abelian group. Particular cases of interest are $A = Z_m$, $m \in Z^+$, and $A = \mathbf{G}$.

Let A be a finite Abelian group with a binary addition operation $+$ and identity element 0 . For $a \in A$, and for positive integers m , we use the notation ma to denote the m -fold addition of a with itself, $ma = \sum_{i=1}^m a$. By convention, we extend this notation to arbitrary integer values of m by defining $0a = 0$, and when $m < 0$, $ma = (-m)(-a)$, where $-a$ is the inverse of a . A set $\{a_1, \dots, a_k\}$ of nonzero elements of A is said to be *independent* if the only solutions to the equation $\sum_{i=1}^k m_i a_i = 0$ with integer unknowns m_1, \dots, m_k are the trivial ones in which $m_i a_i = 0, \forall i$. We say that the set is a *generating set* of A if every element $a \in A$ can be written as $a = \sum_{i=1}^k m_i a_i$, for some integers m_i . The set is called a *basis* of A if it is a generating set and the elements are independent.

Given an Abelian group A , a *character* of A is a homomorphism Ψ from A into the additive circle group of real numbers modulo 1 ($R_{[0,1]}$). The circle group is sometimes denoted as R/Z . It is isomorphic to the group of complex numbers with unit magnitude under multiplication. The *character group*¹ \hat{A} of A is the Abelian group of all homomorphisms $\Psi: A \rightarrow R_{[0,1]}$ under the operation defined by

$$(\Psi_1 \circ \Psi_2)(a) = \Psi_1(a) + \Psi_2(a) \quad \forall a \in A.$$

The identity element of \hat{A} , also referred to as the neutral element or the zero element, is denoted by $\Psi_{\mathcal{N}}$, and maps all the elements of A into 0. It is well known that when A is finite, then \hat{A} is isomorphic to A [14].

B. TG Construction

A TG for a linear $[n, k]$ block code C can be obtained from any parity-check matrix $H = [h_{ij}]$ for C . The n symbol

¹Some authors refer to the character group \hat{A} as the *dual group* of A . However, we choose to use the nomenclature “character group” to prevent any confusion with the “dual group” defined later.

nodes of the graph correspond to the codeword coordinates (v_1, \dots, v_n) , and the $n - k$ (or possibly more) check nodes correspond to the $n - k$ check equations, represented in matrix form as $(v_1, \dots, v_n)H^T = \mathbf{0}$, that each codeword must satisfy. An edge joins symbol node v_j to check node i if and only if v_j is involved in the i th check equation, i.e., if and only if $h_{ij} \neq 0$. The i th check equation is regarded as a ‘‘local constraint,’’ enforcing the condition that $\sum_{j=1}^n h_{ij}v_j = 0$. A given configuration (v_1, \dots, v_n) is a valid codeword if and only if all local constraints are satisfied.

In the following, we generalize this construction to Abelian group block codes with arbitrary alphabets at different coordinates. This covers the important case of the label code of a lattice.

1) *Pairing, Dual Group, and Check Constraints:* For an Abelian group A , the pairing $\langle \cdot, \cdot \rangle: \hat{A} \times A \rightarrow R_{[0,1]}$ is defined for every pair $(\Psi, a) \in \hat{A} \times A$ into the circle group $R_{[0,1]}$ by $\langle \Psi, a \rangle = \Psi(a)$. Clearly, the pairing is bihomomorphic. The elements $\Psi \in \hat{A}$ and $a \in A$ are called *orthogonal* if $\langle \Psi, a \rangle = 0$. Let H be a subgroup of A . We define H^* as the set of elements $h^* \in \hat{A}$ such that $\langle h^*, h \rangle = 0, \forall h \in H$. The set H^* , which is a subgroup of \hat{A} , is called the *dual group* of H . It is known that H^* is in fact the character group for the quotient group A/H , [14]. We then have $|H^*||H| = |A|$, and thus H^* is nontrivial (contains more than just the zero element) if and only if $H \neq A$.

For a direct product of cyclic groups

$$\mathbf{G} = Z_{g_1} \times Z_{g_2} \times \dots \times Z_{g_n}$$

we have

$$\hat{\mathbf{G}} = \hat{Z}_{g_1} \times \hat{Z}_{g_2} \times \dots \times \hat{Z}_{g_n}$$

which is isomorphic to \mathbf{G} . A character Ψ of \mathbf{G} can then be represented as (Ψ_1, \dots, Ψ_n) for some $\Psi_i \in \hat{Z}_{g_i}, i = 1, \dots, n$. It maps an element $\mathbf{c} = (c_1, \dots, c_n) \in \mathbf{G}$ to $\Psi_1(c_1) + \dots + \Psi_n(c_n)$, where the additions are performed in $R_{[0,1]}$. For $\mathbf{c} = (c_1, \dots, c_n) \in \mathbf{G}$ and $\Psi = (\Psi_1, \dots, \Psi_n) \in \hat{\mathbf{G}}$, we thus have

$$\langle \Psi, \mathbf{c} \rangle = \langle \Psi_1, c_1 \rangle + \langle \Psi_2, c_2 \rangle + \dots + \langle \Psi_n, c_n \rangle.$$

For a subgroup \mathbf{L} of \mathbf{G} , let the dual \mathbf{L}^* have a generating set $\mathcal{C}^* = \{\Psi_1, \dots, \Psi_r\}$ with r generators. Due to the duality property $(\mathbf{L}^*)^* = \mathbf{L}$, the set of orthogonality constraints

$$\langle \Psi_i, \mathbf{c} \rangle = \langle \Psi_{i1}, c_1 \rangle + \dots + \langle \Psi_{in}, c_n \rangle = 0, \quad i = 1, \dots, r \quad (1)$$

characterizes the codewords $\mathbf{c} \in \mathbf{L}$. To construct a TG for \mathbf{L} , we associate a check node to each such constraint; i.e., there are r check nodes, and each symbol c_j of \mathbf{L} is represented by a symbol node. There is an edge between symbol node j and check node i if and only if $\Psi_{ij} \neq \Psi_N$.

For computational purposes, it is convenient to parameterize the character group $\hat{\mathbf{G}}$ by \mathbf{G} , using the fact that there is a (non-canonical) isomorphism Φ between them. This in turn reduces the pairing $\langle \cdot, \cdot \rangle$ to an inner product $(\cdot, \cdot)_\Phi$ between the elements of \mathbf{G} , and results in a dual code \mathbf{L}_Φ^* for \mathbf{L} which is a subgroup of \mathbf{G} (rather than a subgroup of $\hat{\mathbf{G}}$). The orthogonality constraints

(1) associated with the check nodes will then translate to modular linear equations imposed on the code symbols by a set of generators for \mathbf{L}_Φ^* . The complete derivation follows.

2) *Inner Product, Associated Dual, and Construction of Modular Linear Check Constraints:* For the special case $A = Z_m$, an isomorphism Φ between Z_m and \hat{Z}_m is obtained by

$$\begin{aligned} \Phi: Z_m &\longrightarrow \hat{Z}_m \\ a &\longrightarrow a\Psi \end{aligned} \quad (2)$$

where Ψ is a generating character for \hat{Z}_m , and $a\Psi$ represents the a -fold ‘‘o’’ operation on Ψ . A generating character Ψ generates the dual group \hat{Z}_m and is defined by $\Psi(c) = ci/m \in R_{[0,1]}$, where i is an integer relatively prime to m ($\gcd(i, m) = 1$). Associated with the isomorphism Φ and the generating character Ψ , we now define an *inner product* $(a, c)_\Phi$ into $R_{[0,1]}$ for every pair $(a, c) \in Z_m \times Z_m$ by $(a, c)_\Phi = \langle \Phi(a), c \rangle$. It is easy to see that $(a, c)_\Phi = cai/m \pmod{1}$.

To parameterize $\hat{\mathbf{G}} = \hat{Z}_{g_1} \times \hat{Z}_{g_2} \times \dots \times \hat{Z}_{g_n}$ by \mathbf{G} , one chooses a generating character $\Psi^{(j)}$ for each component cyclic group \hat{Z}_{g_j} , and defines isomorphisms $\Phi_j: Z_{g_j} \rightarrow \hat{Z}_{g_j}$ as in (2). This results in the following isomorphism between \mathbf{G} and $\hat{\mathbf{G}}$:

$$\begin{aligned} \Phi: \mathbf{G} &\longrightarrow \hat{\mathbf{G}} \\ (a_1, \dots, a_n) &\longrightarrow (\Phi_1(a_1), \dots, \Phi_n(a_n)). \end{aligned} \quad (3)$$

The isomorphism Φ induces an inner product on $\mathbf{G} \times \mathbf{G}$ given by $(\cdot, \cdot)_\Phi = \langle \Phi(\cdot), \cdot \rangle$. Given two elements $\mathbf{a}, \mathbf{c} \in \mathbf{G}$, this inner product can be written as

$$\begin{aligned} (\mathbf{a}, \mathbf{c})_\Phi &= \langle \Phi(\mathbf{a}), \mathbf{c} \rangle \\ &= \langle \Phi_1(a_1), c_1 \rangle + \langle \Phi_2(a_2), c_2 \rangle + \dots + \langle \Phi_n(a_n), c_n \rangle \\ &= \frac{c_1 a_1 i_1}{g_1} + \frac{c_2 a_2 i_2}{g_2} + \dots + \frac{c_n a_n i_n}{g_n} \pmod{1} \end{aligned} \quad (4)$$

for some integers $i_j \in \{1, \dots, g_j - 1\}, j = 1, \dots, n$, relatively prime to $g_j, j = 1, \dots, n$, respectively.

It is clear that based on the isomorphism Φ , the dual \mathbf{L}^* can be parameterized by $\mathbf{L}_\Phi^* \triangleq \Phi^{-1}(\mathbf{L}^*)$, and the generating set \mathcal{C}^* for \mathbf{L}^* can be characterized by elements of \mathbf{G} by setting $\mathcal{C}_\Phi^* \triangleq \Phi^{-1}(\mathcal{C}^*)$. Clearly, \mathcal{C}_Φ^* is a generating set for \mathbf{L}_Φ^* , and conversely, any generating set of \mathbf{L}_Φ^* is mapped via Φ to a generating set of \mathbf{L}^* of the same cardinality.

From the above derivation, it follows that instead of working with the dual \mathbf{L}^* and the pairing $\langle \cdot, \cdot \rangle$, one can work with the transformed dual \mathbf{L}_Φ^* and the inner product $(\cdot, \cdot)_\Phi$; i.e., the following proposition holds.

Proposition 1: Let $\mathbf{G} = Z_{g_1} \times Z_{g_2} \times \dots \times Z_{g_n}$. Then any choice of integers i_j such that $\gcd(i_j, g_j) = 1, j = 1, \dots, n$, defines an isomorphism (3) and an inner product (4) with the following properties.

- i) There are one-to-one correspondences among a subgroup $\mathbf{L} \subset \mathbf{G}$, its dual $\mathbf{L}^* \subset \hat{\mathbf{G}}$, and its transformed dual $\mathbf{L}_\Phi^* \subset \mathbf{G}$. In particular, \mathbf{L}_Φ^* is orthogonal to \mathbf{L} with respect to the inner product $(\cdot, \cdot)_\Phi$ if and only if \mathbf{L}^* is orthogonal to \mathbf{L} with respect to the pairing $\langle \cdot, \cdot \rangle$.

- ii) The TG for \mathbf{L} based on the generating set \mathcal{C}^* and the pairing $\langle \cdot, \cdot \rangle$ is the same as the TG for \mathbf{L} based on the generating set \mathcal{C}_{Φ}^* and the inner product $(\cdot, \cdot)_{\Phi}$.

Proof: Property i) follows directly from the definition of the dual and the properties of isomorphism. For property ii), since $|\mathcal{C}_{\Phi}^*| = |\mathcal{C}^*|$, both TGs have the same number of check nodes. Moreover, let the elements of \mathcal{C}_{Φ}^* be denoted by $\mathbf{c}_i^* = \Phi^{-1}(\Psi_i)$, $i = 1, \dots, r$. Then the orthogonality constraints (1) are equivalent to

$$(\mathbf{c}_i^*, \mathbf{c})_{\Phi} = 0, \quad \text{for } i = 1, \dots, r. \quad (5)$$

In particular, the condition $\Psi_{ij} = \Psi_{\mathcal{N}}$, which determines the edges of the TG, is equivalent to the condition $c_{ij}^* = 0$ in Z_{g_j} . Note that this condition is independent of the particular choice of the generating character $\Psi^{(j)}$ and its corresponding isomorphism Φ_j . \square

As explained above, the choice of the inner product in (4), or equivalently, the choice of the dual code \mathbf{L}_{Φ}^* , does not influence the TG of \mathbf{L} . In the remainder of the paper and for the sake of simplicity, we choose the inner product (4) with $i_j = 1$, $\forall j$, and use the notation $\langle \cdot, \cdot \rangle$ for it. We also use the notation \mathbf{L}^* to denote the corresponding dual. We refer to these as ‘‘inner product’’ and ‘‘dual group of \mathbf{L} ,’’ respectively.

To summarize, for the inner product, we have

$$\langle \mathbf{c}, \mathbf{c}' \rangle = \sum_{i=1}^n \frac{c_i c'_i}{g_i} \in R_{[0,1)}, \quad \forall \mathbf{c}, \mathbf{c}' \in Z_{g_1} \times Z_{g_2} \times \dots \times Z_{g_n}$$

where the multiplications and divisions are performed in the field of real numbers. To construct a TG, we use the following set of check equations, which fully describes the codewords $\mathbf{c} = (c_1, \dots, c_n) \in \mathbf{L}$:

$$\sum_{i=1}^n c_{ki}^* c_i / g_i = 0 \pmod{1}, \quad k = 1, \dots, r \quad (6)$$

or equivalently

$$\sum_{i=1}^n c_{ki}^* c_i / g_i \in Z, \quad k = 1, \dots, r \quad (7)$$

where $\mathcal{C}^* = \{\mathbf{c}_1^*, \dots, \mathbf{c}_r^*\}$ is a generating set for \mathbf{L}^* with r generators. For linear block codes, these equations reduce to the well-known parity-check equations.

Example 1: Let $\mathbf{G} = Z_2 \times Z_6 \times Z_6 \times Z_2$, and

$$\mathbf{L} = \left\{ \begin{array}{cccccc} 0000, & 0031, & 0220, & 0251, & 0440, & 0411, \\ 1300, & 1331, & 1520, & 1551, & 1140, & 1111 \end{array} \right\}.$$

We then have

$$\mathbf{L}^* = \left\{ \begin{array}{cccccc} 0000, & 0240, & 0420, & 1511, & 1151, & 0031, \\ 1300, & 1331, & 0451, & 1540, & 1120, & 0211 \end{array} \right\}.$$

It can be seen that $\{1151, 0240, 0031\}$ is a generating set for \mathbf{L}^* . The corresponding TG is given in Fig. 1. It is easy to see that the modular constraints of Fig. 1 are the same as the check equations in (6).

To construct a TG for a code \mathbf{L} , one needs to find a generating set for \mathbf{L}^* . Later, in Section IV, we develop an algorithm to perform this task efficiently.

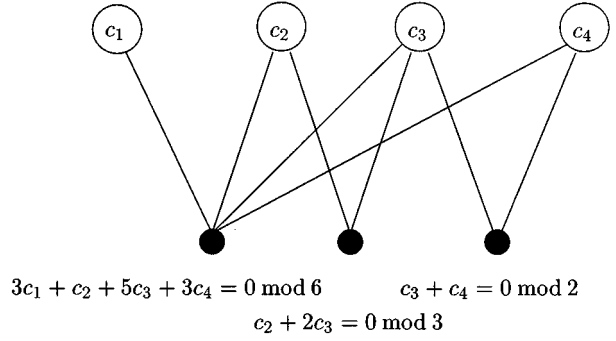


Fig. 1. A TG for the code \mathbf{L} of Example 1.

C. Fully Dynamical Property, Trimness, and TG Structure of Isomorphic Codes

In this subsection, we discuss two properties of group codes which appear to be important in the following discussions.

We call \mathbf{L} *fully dynamical* if it contains no codeword of Hamming weight 1. The minimal trellis of a fully dynamical code does not contain any parallel edges. For codes which are not fully dynamical, we adopt the notation $\mathbf{q}(\mathbf{L})$ of [11] to denote the fully dynamical component of \mathbf{L} . We thus have the coset decomposition $\mathbf{L} = \mathbf{q}(\mathbf{L}) + \mathbf{L}'$, where \mathbf{L}' is a subgroup of \mathbf{L} generated by all codewords of Hamming weight 1, and $\mathbf{q}(\mathbf{L}) = \mathbf{L}/\mathbf{L}'$. The minimal trellis for \mathbf{L}' has only one state at each level. If \mathbf{L}' is isomorphic to $Z_{k_1} \times \dots \times Z_{k_n}$ for some $\mathbf{k} \in (Z^+)^n$, then $\mathbf{q}(\mathbf{L})$ is a group code defined over $Z_{\frac{g_1}{k_1}} \times \dots \times Z_{\frac{g_n}{k_n}}$.²

We call \mathbf{L} *trim* [11], if for every coordinate i , the projection of \mathbf{L} onto that coordinate is equal to the corresponding symbol alphabet G_i .

Example 2: The group code $\mathbf{L} = \{00, 31\}$ defined over $Z_6 \times Z_2$ is fully dynamical. However, the code is not trim since its projection onto the first coordinate is $\{0, 3\} \neq Z_6$. \mathbf{L} is in fact isomorphic to $\{00, 11\}$ defined over $Z_2 \times Z_2$.

We have

$$\mathbf{L}^* = \{00, 11, 20, 31, 40, 51\}$$

which is trim but not fully dynamical. It is easy to see that $\mathbf{q}(\mathbf{L}^*) = \{00, 11\}$, which is a group code over $Z_2 \times Z_2$, and $\mathbf{L}^* = \{00, 11\} + \{00, 20, 40\}$.

Lemma 1: A group code \mathbf{L} is trim if and only if \mathbf{L}^* is fully dynamical.

Proof: Suppose that \mathbf{L}^* is not fully dynamical. Then it has a nonzero element $\mathbf{c}' = (0, \dots, 0, c'_i, 0, \dots, 0)$ of weight one. For all $\mathbf{c} \in \mathbf{L}$, the inner product $\langle \mathbf{c}', \mathbf{c} \rangle = \langle c'_i, c_i \rangle = 0$, and thus $c'_i \in P_i(\mathbf{L}^*)$, the dual group of the projection of \mathbf{L} onto coordinate i . Since c'_i is nonzero, $P_i(\mathbf{L}^*)$ is nontrivial, which implies that $P_i(\mathbf{L}) \neq G_i$, the i th symbol alphabet. Therefore, \mathbf{L} is not trim.

Now suppose \mathbf{L} is not trim. Then for some i , $P_i(\mathbf{L}) \neq G_i$, and hence $P_i(\mathbf{L})^*$ is nontrivial. From any nonzero $a \in P_i(\mathbf{L})^*$, form the word $\mathbf{c}' = (0, \dots, 0, a, 0, \dots, 0)$, where a occurs in

²In [11], $\mathbf{q}(\mathbf{L})$ and \mathbf{L}' are referred to as the label code and the parallel transition code of \mathbf{L} , respectively, and the decomposition $\mathbf{L} = \mathbf{q}(\mathbf{L}) + \mathbf{L}'$ is called the label code decomposition.

coordinate i . Then, for all $\mathbf{c} \in \mathbf{L}$, we have $\langle \mathbf{c}', \mathbf{c} \rangle = \langle a, c_i \rangle = 0$; hence \mathbf{c}' is an element of \mathbf{L}^* of Hamming weight one, which implies that \mathbf{L}^* is not fully dynamical. \square

The following example shows that there is not necessarily a one-to-one correspondence between the TGs of two isomorphic codes defined over different alphabet sequence spaces.

Example 3: Consider the group code $\mathbf{L}_1 = \{00, 11\}$ defined over $\mathbf{G}_1 = Z_2 \times Z_2$. The code \mathbf{L}_1 is self-dual, and is described by the check equation $c_1 + c_2 = 0 \pmod{2}$, resulting from the only possible generating set $\mathcal{C}_1^* = \{11\}$ for \mathbf{L}_1^* .

Now, let

$$\mathbf{L}_2 = \{00, 22\} \subset \mathbf{G}_2 = Z_4 \times Z_4.$$

Clearly, \mathbf{L}_2 is isomorphic to \mathbf{L}_1 , but is not trim. The dual code is

$$\mathbf{L}_2^* = \{00, 11, 22, 33, 13, 31, 02, 20\}.$$

It can be seen that \mathbf{L}_2^* is not cyclic, and thus any TG for \mathbf{L}_2 must have at least two check nodes. As an example of check equations that describe \mathbf{L}_2 , we have $c_1 + c_2 = 0 \pmod{4}$ and $c_1 + 3c_2 = 0 \pmod{4}$, resulting from the generating set $\mathcal{C}_2^* = \{11, 13\}$ for \mathbf{L}_2^* . Note that \mathbf{L}_1 does not have any corresponding TG description.

Despite the lack of a one-to-one correspondence between the TGs of two isomorphic codes, in the following, we show that if the group code \mathbf{L}_1 is trim, then any TG of \mathbf{L}_1 has a correspondence, essentially with the same structure, in the set of TGs for of any group code \mathbf{L}_2 isomorphic to \mathbf{L}_1 and defined over a larger sequence space.

Theorem 1: Let a trim group code \mathbf{L}_1 be defined over the alphabet sequence space $\mathbf{G}_1 = Z_{g_{11}} \times \cdots \times Z_{g_{1n}}$. Suppose that a larger sequence space $\mathbf{G}_2 = Z_{g_{21}} \times \cdots \times Z_{g_{2n}}$ is defined such that $g_{2i}/g_{1i} = k_i, i = 1, \dots, n$, for some $\mathbf{k} \in (Z^+)^n$. Also, let \mathbf{L}_2 be a group code isomorphic to \mathbf{L}_1 and defined over \mathbf{G}_2 . Then for any TG of \mathbf{L}_1 , there is a TG for \mathbf{L}_2 with the same structure.

Proof: Since \mathbf{L}_1 is trim, \mathbf{L}_1^* is fully dynamical. A TG T_1 for \mathbf{L}_1 can be constructed based on a generating set \mathcal{C}_1^* of \mathbf{L}_1^* . The code \mathbf{L}_2 , however, is not trim. This means that \mathbf{L}_2^* is not fully dynamical, and can be decomposed as $\mathbf{L}_2^* = \mathbf{q}(\mathbf{L}_2^*) + \mathbf{L}'$, where \mathbf{L}' is isomorphic to $Z_{k_1} \times \cdots \times Z_{k_n}$, and $\mathbf{q}(\mathbf{L}_2^*) = \mathbf{L}_2^*/\mathbf{L}'$ is a group code over $Z_{\frac{g_{21}}{k_1}} \times \cdots \times Z_{\frac{g_{2n}}{k_n}} = \mathbf{G}_1$. In fact, $\mathbf{q}(\mathbf{L}_2^*)$ is the same as \mathbf{L}_1^* ; i.e., both \mathbf{L}_1^* and \mathbf{L}_2^* have the same dynamical structure. It can be seen that \mathcal{C}_1^* along with the following codewords of \mathbf{G}_2 form a generating set for \mathbf{L}_2^* :

$$\begin{pmatrix} g_{11} & 0 & 0 & \cdots & 0, \\ 0 & g_{12} & 0 & \cdots & 0, \\ & & & \cdots & \\ 0 & 0 & 0 & \cdots & g_{1n}. \end{pmatrix} \quad (8)$$

In the following, we show that the TG constructed for \mathbf{L}_2 based on this generating set has the same structure as T_1 .

The check equation for \mathbf{L}_2 resulting from the i th codeword of (8) is $c_{2i}g_{1i}/g_{2i} = c_{2i}/k_i = 0 \pmod{1}$, just constraining the i th symbol node. (Note that if $k_i = 1$ for some i , then $g_{1i} = g_{2i}$, and the corresponding codeword is equal to the all-zero codeword, creating a redundant equation.) This constraint can also

be applied by defining a new variable $c'_i = c_{2i}/k_i$, and assuming that $c'_i \in Z_{g_{1i}}$.

Let $\mathbf{c}^* \in \mathcal{C}_1^*$. Then the resulting check equation for $\mathbf{c}_2 \in \mathbf{L}_2$ is

$$\sum_{i=1}^n c_{2i}c_i^*/g_{2i} = 0 \pmod{1}.$$

Combining this with $g_{2i} = k_i g_{1i}$, and replacing symbols c_{2i} with $k_i c'_i$, we obtain

$$\sum_{i=1}^n c'_i c_i^*/g_{1i} = 0 \pmod{1}$$

which is the same as the check equation for \mathbf{L}_1 resulting from \mathbf{c}^* . \square

Theorem 1 implies that to obtain a TG for a group code, one can first find a trim isomorph of the code defined on a smaller sequence space. The TG of this code can be then used to represent the original code through simple symbol-by-symbol transformations.

Example 3 (Continued): Codewords $\{11, 02, 20\}$ form a generating set for \mathbf{L}_2^* . The corresponding check equations for \mathbf{L}_2 are $\frac{c_1}{4} + \frac{c_2}{4} = 0 \pmod{1}$, $c_1 = 0 \pmod{2}$, and $c_2 = 0 \pmod{2}$. Using the change of variables $c_1 = 2c'_1$, $c'_1 \in Z_2$, and $c_2 = 2c'_2$, $c'_2 \in Z_2$, the check equations will be reduced to

$$\frac{c'_1}{2} + \frac{c'_2}{2} = 0 \pmod{1}$$

which is the same as the check equation describing \mathbf{L}_1 .

In the following section, by defining the label code of a lattice, we describe how the construction presented in Section II-B can be used to obtain TGs for lattices.

III. LATTICES

A. Definitions

Let R^m be the m -dimensional (m -D) real vector space with the standard inner product $\langle \cdot, \cdot \rangle$, and Euclidean norm $\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}}$. The subspace generated by a subset S of R^m is denoted by $\text{span}(S)$, and its orthogonal complement by $\text{span}(S)^\perp$. A discrete, additive subgroup Λ of R^m is called a *lattice*. Every lattice Λ can be generated as an Abelian group by the integer linear combinations of some set of vectors $\mathbf{b}_1, \dots, \mathbf{b}_n \in \Lambda$. These vectors form a *generating set* for the lattice, and the $n \times m$ matrix $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ which has the generator vectors as its rows is called a *generator matrix* of Λ . We use the brief notation $\text{span}(\Lambda)$ to denote the real span of the set of generator vectors; i.e., $\text{span}(\Lambda) = \text{span}(\mathbf{b}_1, \dots, \mathbf{b}_n)$. If the generators are linearly independent, they form a *basis* for Λ , and the matrix B is called a *basis matrix* of Λ . In this case, the integer n ($\leq m$) is referred to as the *dimension* of Λ . We call a lattice *full-dimensional* if $n = m$. A lattice Λ is called *orthogonal* (rectangular) if it has a basis B with mutually orthogonal vectors.

The determinant (or volume) of an n -D lattice Λ , $\det(\Lambda)$, is defined as the common volume of the (n -D) fundamental regions of Λ , where a fundamental region of Λ is defined as any building-block region which, when translated by the vectors of

Λ , partitions $\text{span}(\Lambda)$ with just one lattice point in each copy. If B is a basis matrix for Λ , then $\det(\Lambda) = [\det(BB^T)]^{1/2}$, where B^T denotes the transpose of B .

A *sublattice* Λ_1 of a lattice Λ is a subgroup of Λ . The quotient group Λ/Λ_1 is finite if and only if the dimension of Λ_1 is equal to the dimension of Λ . In this case, $|\Lambda/\Lambda_1| = \det(\Lambda_1)/\det(\Lambda)$.

The notation $\lambda(\Lambda)$ is used to denote the length of a shortest nonzero vector in Λ . This is also equal to the minimum distance between lattice points. The *coding gain* of Λ is defined as [6]

$$\gamma(\Lambda) \triangleq \lambda^2(\Lambda)[\det(\Lambda)]^{-2/n}. \quad (9)$$

To any ordered basis of Λ , say $\mathbf{b}_1, \dots, \mathbf{b}_n \in R^m$, one can associate a set of *Gram–Schmidt (G-S)* vectors $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_n \in R^m$, such that $\hat{\mathbf{b}}_1 = \mathbf{b}_1$ and for $i > 1$, $\hat{\mathbf{b}}_i$ is the projection of \mathbf{b}_i on $\text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})^\perp$. Clearly, the vectors $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_n$ are mutually orthogonal.

Two lattices Λ_1 and Λ_2 are called *equivalent*, denoted by $\Lambda_1 \cong \Lambda_2$, if they are the same up to rotation, reflection, and scaling.

If Λ is an n -D lattice, then the set of all vectors in $\text{span}(\Lambda)$ whose inner product with all elements of Λ is integer is an n -D lattice Λ^* , called the *dual lattice* of Λ . If $\Lambda^* = \Lambda$, then Λ is called *self-dual*. If $\Lambda^* \cong \Lambda$, then Λ is called *iso-dual*. Many important lattices are self- or iso-dual [6].

If B is a basis matrix for a lattice Λ , then there exists a matrix B' such that $B(B')^T = I$. It can be seen that B' is a basis matrix for Λ^* . Thus, for a full-dimensional lattice Λ , $(B^{-1})^T$ forms a basis matrix for Λ^* . For every lattice, we also have

$$\det(\Lambda^*) = \frac{1}{\det(\Lambda)}. \quad (10)$$

Let lattices Λ_1 and Λ_2 have basis matrices B_1 and B_2 , respectively. The *direct sum lattice* $\Lambda_1 \oplus \Lambda_2$ is defined by the following basis matrix:

$$B = B_1 \oplus B_2 \triangleq \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}. \quad (11)$$

It is easy to see that $\det(\Lambda_1 \oplus \Lambda_2) = \det(\Lambda_1) \det(\Lambda_2)$. We also use the notation Λ^m for the m -fold direct sum of Λ with itself.

B. Label Code of a Lattice

A set of check equations which characterizes a lattice Λ , and is similar to $\mathbf{c}H^T = \mathbf{0}$ for linear block codes, is $\mathbf{v}(V^*)^T \in Z^r$, where $\mathbf{v} \in R^m$ belongs to Λ , and $V^* = \{\mathbf{v}_1^*, \dots, \mathbf{v}_r^*\}$ is a generating set with r generators for the dual lattice Λ^* . Although this set of equations fully describes the lattice, it does not provide an appropriate form for constructing a TG for Λ , simply because \mathbf{v} is a real vector. (Note that for the two-way algorithm to be applicable to a TG, the sizes of the symbol alphabets must be finite.) To resolve this, one needs to transform \mathbf{v} to a vector from a finite alphabet space. This task is performed in the following by using the label code of Λ . The same notations as used in Section II are adopted here.

Let Λ be an n -D lattice defined in an m -D real vector space R^m . Suppose that Λ has an n -D orthogonal sublattice Λ' , and let Λ' have a set of basis vectors along the orthogonal one-dimensional (1-D) subspaces $\mathcal{S} = \{W_i\}_{i=1}^n$. A useful representation of Λ is a description in terms of the quotient group Λ/Λ' . In fact,

the trellis of Λ [9] is a combinatorially efficient graph representation of Λ in terms of the cosets of Λ' in Λ . In this light, it is natural to also think of the construction of a TG for Λ based on Λ/Λ' .

For the rest of the paper, we assume that Λ' has the smallest determinant among all the orthogonal sublattices of Λ with their basis vectors along \mathcal{S} . Such an orthogonal sublattice, which results in a minimal trellis for Λ with no parallel edges, is called *primitive* (with respect to \mathcal{S}). In the trellis of Λ , to label the cosets of Λ' , one uses the elements of the Abelian group $G_i = P_{W_i}(\Lambda)/\Lambda_{W_i}$ for labeling the edges of the trellis section i , for $i = 1, \dots, n$, where $P_{W_i}(\Lambda)$ and Λ_{W_i} denote the projection and cross section of Λ on W_i , respectively [3]. The groups G_i , $i = 1, \dots, n$, are thus called the *label groups* of Λ in the coordinate system \mathcal{S} , referred to as the *graph coordinate system*, or briefly the “coordinate system” hereafter. (In the following, we often assume that the vectors are presented in the graph coordinate system. In this system, all the vectors belong to R^n .) The set of all label sequences, denoted by $\bar{\mathbf{L}}(\Lambda)$, is then called the *label code*. The label code, which is isomorphic to Λ/Λ' , is apparently an Abelian group block code defined over the alphabet sequence space $\mathbf{G} = G_1 \times \dots \times G_n$. Note that in general the symbol alphabets at different coordinates differ in size. By definition, the label code of a lattice is fully dynamical and trim.

For simplicity, and based on the isomorphism $G_i \cong Z_{g_i}$, we assume $\mathbf{G} = Z_{g_1} \times Z_{g_2} \times \dots \times Z_{g_n}$. Moreover, let $\Lambda_{W_i} = Z\mathbf{v}$, where \mathbf{v} is a generator for the 1-D lattice Λ_{W_i} . To uniquely specify the isomorphism, we map

$$\Lambda_{W_i} + \det(P_{W_i}(\Lambda)) \frac{\mathbf{v}}{\|\mathbf{v}\|} \in G_i$$

to $1 \in Z_{g_i}$.

Example 4: The following basis matrix generates the checkerboard lattice $\Lambda = D_4$:

$$B = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 \end{pmatrix}.$$

The associated Gram–Schmidt vectors are

$$\begin{aligned} \hat{\mathbf{b}}_1 &= (1, 1, 0, 0) \\ \hat{\mathbf{b}}_2 &= (1/2, -1/2, 1, 0) \\ \hat{\mathbf{b}}_3 &= (-1/3, 1/3, 1/3, 1) \\ \hat{\mathbf{b}}_4 &= (1/2, -1/2, -1/2, 1/2). \end{aligned}$$

In the graph coordinate system $\{W_i\}_{i=1}^4 = \{\text{span}(\hat{\mathbf{b}}_i)\}_{i=1}^4$, we obtain the following projection and cross-section lattices:

$$\begin{aligned} P_{W_1}(\Lambda) &= \frac{Z}{\sqrt{2}} \frac{\hat{\mathbf{b}}_1}{\|\hat{\mathbf{b}}_1\|} & \Lambda_{W_1} &= \sqrt{2}Z \frac{\hat{\mathbf{b}}_1}{\|\hat{\mathbf{b}}_1\|} \\ P_{W_2}(\Lambda) &= \frac{Z}{\sqrt{6}} \frac{\hat{\mathbf{b}}_2}{\|\hat{\mathbf{b}}_2\|} & \Lambda_{W_2} &= \sqrt{6}Z \frac{\hat{\mathbf{b}}_2}{\|\hat{\mathbf{b}}_2\|} \\ P_{W_3}(\Lambda) &= \frac{Z}{\sqrt{3}} \frac{\hat{\mathbf{b}}_3}{\|\hat{\mathbf{b}}_3\|} & \Lambda_{W_3} &= 2\sqrt{3}Z \frac{\hat{\mathbf{b}}_3}{\|\hat{\mathbf{b}}_3\|} \\ P_{W_4}(\Lambda) &= Z \frac{\hat{\mathbf{b}}_4}{\|\hat{\mathbf{b}}_4\|} & \Lambda_{W_4} &= 2Z \frac{\hat{\mathbf{b}}_4}{\|\hat{\mathbf{b}}_4\|}. \end{aligned}$$

This results in the following label groups for D_4 :

$$G_1 = Z_2 \quad G_2 = Z_6 \quad G_3 = Z_6 \quad G_4 = Z_2$$

which corresponds to the \mathbf{G} given in Example 1. In fact, the label code is also the group code \mathbf{L} of Example 1.

When a lattice Λ has label code \mathbf{L} in some (graph) coordinate system \mathcal{S} , it can be decomposed as

$$\Lambda = Z^n C(\Lambda) + \mathbf{L}P(\Lambda) \quad (12)$$

where

$$\begin{aligned} C(\Lambda) &= \text{diag}(\det(\Lambda_{W_1}), \dots, \det(\Lambda_{W_n})) \\ P(\Lambda) &= \text{diag}(\det(P_{W_1}(\Lambda)), \dots, \det(P_{W_n}(\Lambda))) \end{aligned}$$

and $\text{diag}(\dots)$ is a diagonal matrix. The decomposition (12) means that in \mathcal{S} , a vector $\mathbf{v} \in R^n$ belongs to Λ if and only if it can be expressed as $\mathbf{v} = \mathbf{k}C(\Lambda) + \mathbf{c}P(\Lambda)$ for some $\mathbf{k} \in Z^n$ and $\mathbf{c} \in \mathbf{L}$. [Note that in (12), the orthogonal sublattice Λ' is equal to

$$Z^n C(\Lambda) = \det(\Lambda_{W_1})Z \oplus \dots \oplus \det(\Lambda_{W_n})Z$$

and $\mathbf{L}P(\Lambda) = \Lambda/\Lambda'$.]

The following theorem is of key importance for the rest of the paper.

Theorem 2: Let a lattice Λ have a label code \mathbf{L} in a graph coordinate system \mathcal{S} . Then \mathbf{L}^* is the label code of the dual lattice Λ^* in \mathcal{S} , i.e., $\mathbf{L}(\Lambda^*)^* = \mathbf{L}(\Lambda^*)$.³

Proof: In \mathcal{S} , let Λ^* have a label code \mathbf{M} and the corresponding coset decomposition $\Lambda^* = Z^n C(\Lambda^*) + \mathbf{M}P(\Lambda^*)$, similar to (12). Since the label complexity profiles of dual lattices in the same coordinate system are the same [3], \mathbf{M} is defined over the same alphabet sequence space \mathbf{G} as \mathbf{L} . From the duality results $(\Lambda_{W_i})^* = P_{W_i}(\Lambda^*)$ and $P_{W_i}(\Lambda)^* = (\Lambda^*)_{W_i}$ [9], combined with (10), we find that $C(\Lambda^*) = P^{-1}(\Lambda)$ and $P(\Lambda^*) = C^{-1}(\Lambda)$. Thus, $\Lambda^* = Z^n P^{-1}(\Lambda) + \mathbf{M}C^{-1}(\Lambda)$, where \mathbf{M} is a group code over \mathbf{G} .

Now let

$$W = Z^n P^{-1}(\Lambda) + \mathbf{L}^* C^{-1}(\Lambda)$$

and $\mathbf{u} = \mathbf{j}P^{-1}(\Lambda) + \mathbf{c}^* C^{-1}(\Lambda)$ be an arbitrary element of W . Similarly, let $\mathbf{v} = \mathbf{k}C(\Lambda) + \mathbf{c}P(\Lambda)$ be an element of Λ . Then

$$\begin{aligned} \mathbf{u} \cdot \mathbf{v} &= \mathbf{j} \cdot (\mathbf{k}C(\Lambda)P^{-1}(\Lambda)) + \mathbf{j} \cdot (\mathbf{c}P(\Lambda)P^{-1}(\Lambda)) \\ &\quad + \mathbf{c}^* \cdot (\mathbf{k}C(\Lambda)C^{-1}(\Lambda)) + \mathbf{c}^* \cdot (\mathbf{c}P(\Lambda)C^{-1}(\Lambda)). \end{aligned} \quad (13)$$

Each term on the right-hand side of (13) is an integer. This is the case because

$$\begin{aligned} C(\Lambda)P^{-1}(\Lambda) &= \text{diag}(g_1, \dots, g_n) \\ P(\Lambda)P^{-1}(\Lambda) &= C(\Lambda)C^{-1}(\Lambda) = I \end{aligned}$$

where I is the identity matrix, and that for $\mathbf{c} \in \mathbf{L}$ and $\mathbf{c}^* \in \mathbf{L}^*$

$$\mathbf{c}^* \cdot (\mathbf{c}P(\Lambda)C^{-1}(\Lambda)) = \sum_{i=1}^n \frac{c_i^* c_i}{g_i} \in Z$$

³It can be proved that the theorem still holds if \mathbf{L} is not fully dynamical, i.e., if the orthogonal sublattice Λ' is not primitive.

by the definition of the dual code. The inner product $\mathbf{u} \cdot \mathbf{v}$ is thus an integer, and since \mathbf{u} is chosen arbitrarily, $W \subset \Lambda^*$, and thus $\mathbf{L}^* \subset \mathbf{M}$.

Now let \mathbf{u} be an arbitrary element of Λ^* , and write \mathbf{u} as $\mathbf{u} = \mathbf{j}P^{-1}(\Lambda) + \mathbf{m}C^{-1}(\Lambda)$, where $\mathbf{m} \in \mathbf{M}$. With \mathbf{v} as above, $\mathbf{u} \cdot \mathbf{v}$ can be written as in (13) with \mathbf{c}^* replaced by \mathbf{m} . Since $\mathbf{u} \cdot \mathbf{v}$ must be an integer and the terms corresponding to the first three terms of (13) are integers, it follows that the last term, i.e.,

$$\mathbf{m} \cdot (\mathbf{c}P(\Lambda)C^{-1}(\Lambda)) = \sum_{i=1}^n \frac{m_i c_i}{g_i}$$

is an integer and hence \mathbf{m} is an element of \mathbf{L}^* . Thus, $\mathbf{M} \subset \mathbf{L}^*$. Since $\mathbf{L}^* \subset \mathbf{M}$ and $\mathbf{M} \subset \mathbf{L}^*$, we have $\mathbf{M} = \mathbf{L}^*$. \square

Corollary 1: The label code of a self-dual lattice in any coordinate system is a self-dual group block code.

C. TGs for Lattices

The TG construction for group codes, developed in Section II-B, can be applied to the label code of a lattice as a special case. In [26], Tarokh sketched another TG construction for a lattice Λ based on Λ/Λ' , using a basis of Λ^* . In the following, we elaborate this construction and explicitly derive the check equations.

Proposition 2: Let $\mathbf{c} \in \mathbf{G} = Z_{g_1} \times Z_{g_2} \times \dots \times Z_{g_n}$. Then $\mathbf{c} \in \mathbf{L}(\Lambda)$ if and only if

$$\mathbf{c}P(\Lambda)V^{*T} \in Z^r \quad (14)$$

where V^* is a generator matrix for Λ^* in the graph coordinate system, r is the number of generators in V^* (number of rows), and $P(\Lambda)$ is defined in (12).

Proof: If $\mathbf{c} \in \mathbf{L}(\Lambda)$, then in the graph coordinate system, the vector $\mathbf{c}P(\Lambda)$ belongs to Λ and is in the coset of Λ' which corresponds to \mathbf{c} . It thus satisfies (14).

Conversely, if (14) is satisfied then $\mathbf{c}P(\Lambda) \in \Lambda$. This along with $\mathbf{c} \in \mathbf{G}$ and the definition of label code proves that $\mathbf{c} \in \mathbf{L}(\Lambda)$. \square

Proposition 2 provides us with r check equations to characterize $\mathbf{L}(\Lambda)$. Now, a natural question is: "How can the two constructions be compared, and are they related?" As the first step in comparing the two constructions, we notice that there are infinitely many generator matrices for Λ^* . However, in (7), the number of possible generating sets for \mathbf{L}^* is finite. In fact, it appears that searching for a simple TG based on the construction of (14) is more difficult than using (7). In the following, we establish relationships between the two constructions, and compare them.

D. Comparison of the Two Constructions

Let $\{\mathbf{v}_1, \dots, \mathbf{v}_r\}$ be a generating set for the lattice Λ . Also, in a given coordinate system and for $\mathbf{v} \in \Lambda$, let $\mathbf{v} = \mathbf{k}C(\Lambda) + \mathbf{c}P(\Lambda)$ for $\mathbf{c} \in \mathbf{L}$ and $\mathbf{k} \in Z^n$. We define a homomorphism $\Phi: \Lambda \rightarrow \mathbf{L}$ by mapping the point $\mathbf{v} \in \Lambda$ to the codeword $\mathbf{c} \in \mathbf{L}$ (similarly, we define $\Phi^*: \Lambda^* \rightarrow \mathbf{L}^*$). It can then be seen that $\{\Phi(\mathbf{v}_1), \dots, \Phi(\mathbf{v}_r)\}$ forms a generating set for \mathbf{L} . Note that the derivation of $\Phi(\mathbf{v}_i)$ from \mathbf{v}_i is quite easy. To obtain the j th

coordinate, one needs to divide the j th graph coordinate of \mathbf{v}_i by $\det(P_{W_j}(\Lambda))$, and then evaluate the result modulo g_j .

On the other hand, if $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_r\}$ is a generating set for \mathbf{L} , then $\mathcal{C}P(\Lambda)$ along with the following vectors in the graph coordinate system form a generating set for Λ :

$$\begin{pmatrix} \det(\Lambda_{W_1}), & 0, & 0, & \dots, & 0, & 0 \\ 0, & \det(\Lambda_{W_2}), & 0, & \dots, & 0, & 0 \\ & & \dots & & & \\ 0, & 0, & 0, & \dots, & 0, & \det(\Lambda_{W_n}) \end{pmatrix} \quad (15)$$

Combining these with Theorem 2 implies that under the proper transformation of the generating sets for the dual lattice and its label code, the two constructions (7) and (14) result in essentially the same set of check equations (up to straightforward modular simplifications). This is proved in the following theorem.

Theorem 3: For creating check equations to describe the label code of a lattice, the two methods of (7) and (14) are equivalent in the sense that any set of check equations created by one can also be created by the other.

Proof: Let \mathcal{C}^* be a generating set for \mathbf{L}^* . We first prove that by using $\mathcal{C}^*P(\Lambda^*)$ as V^* in (14), we obtain the same set of check equations as in (7). Substituting $\mathcal{C}^*P(\Lambda^*)$ as V^* in (14), and using $P(\Lambda^*) = C^{-1}(\Lambda)$, we obtain $\mathcal{C}P(\Lambda)C^{-1}(\Lambda)\mathcal{C}^{*T} \in Z^r$. Combining this with

$$P(\Lambda)C^{-1}(\Lambda) = \text{diag}(1/g_1, \dots, 1/g_n)$$

results in (7). Note that for Λ^* , substituting the generator vectors of the form (15) in (14) results in redundant check equations since $\det(P_{W_i}(\Lambda))\det((\Lambda^*)_{W_i}) = 1$.

Now let $V^* = \{v_1^*, \dots, v_r^*\}$ be a generating set for Λ^* . We then prove that the set of equations (7) is the same as (14) (up to modular simplifications) if one uses

$$\mathcal{C}^* = \Phi^*(V^*) = \{\Phi^*(v_1^*), \dots, \Phi^*(v_r^*)\}$$

in (7). We start from (7), where \mathcal{C}^* is substituted by $\Phi^*(V^*)$. This results in the set of equations

$$\sum_{i=1}^n c_i (\Phi^*(v_j^*))_i / g_i \in Z, \quad j = 1, \dots, r$$

and thus

$$\begin{aligned} \sum_{i=1}^n \left[\frac{(v_j^*)_i}{\det(P_{W_i}(\Lambda^*))} + k_{ji}g_i \right] \frac{c_i}{g_i} \\ = \sum_{i=1}^n c_i \det(P_{W_i}(\Lambda))(v_j^*)_i + \sum_{i=1}^n c_i k_{ji} \in Z \end{aligned} \quad (16)$$

for some $k_{ji} \in Z, i = 1, \dots, n, j = 1, \dots, r$. For the last step, we have used $g_i \det(P_{W_i}(\Lambda^*)) = 1/\det(P_{W_i}(\Lambda))$. Since the second term in (16) is integer, the first term must also be integer, i.e.

$$\sum_{i=1}^n c_i \det(P_{W_i}(\Lambda))(v_j^*)_i \in Z, \quad j = 1, \dots, r.$$

This is the same as (14). \square

However, it is worth noting that although the two constructions are equivalent in the sense described in Theorem 3, for $n \geq 2$ there exist infinitely many generating sets V^* for Λ^* that

are mapped to the same generating set \mathcal{C}^* for \mathbf{L}^* . It is thus easier to search for an ‘‘appropriate’’ \mathcal{C}^* than for an ‘‘appropriate’’ V^* .

In the following section, by introducing a lattice construction based on an arbitrary group code \mathbf{L} , we devise an efficient algorithm to find a generating set for \mathbf{L}^* .

IV. COMPUTING THE DUAL OF A GROUP BLOCK CODE

A. Generalized Construction A (GCA)

Let \mathbf{L} be a group block code defined over $\mathbf{G} = Z_{g_1} \times \dots \times Z_{g_n}$. We then construct

$$\Lambda = Z^n A + \mathbf{L}B \quad (17)$$

where A and B are positive diagonal matrices, and $B^{-1}A = \text{diag}(g_1, \dots, g_n)$. It can be seen that Λ is discrete and has a group structure and is therefore a lattice. If $A = \text{diag}(a_1, \dots, a_n)$, then $B = \text{diag}(\frac{a_1}{g_1}, \dots, \frac{a_n}{g_n})$ and

$$\Lambda = \Lambda' + \mathbf{L} \text{diag} \left(\frac{a_1}{g_1}, \dots, \frac{a_n}{g_n} \right)$$

where $\Lambda' = a_1 Z \oplus \dots \oplus a_n Z$ is an orthogonal sublattice of Λ .

The construction reduces to the so-called ‘‘Construction A’’ [6] for $a_i = g_i = 2, \forall i$. For the rest of the paper, we assume that none of the coordinates of \mathbf{L} is always zero. This eliminates the trivial cases where the GCA lattice Λ can be decomposed as the direct sum of an orthogonal lattice and a nonorthogonal lattice.

Proposition 3: In the GCA given in (17), \mathbf{L} is the label code of Λ if \mathbf{L} is fully dynamical and trim. Otherwise, the label code is a trim group code which is isomorphic to $\mathbf{q}(\mathbf{L})$ in the coset decomposition $\mathbf{L} = \mathbf{L}' + \mathbf{q}(\mathbf{L})$, where \mathbf{L}' is the subgroup of \mathbf{L} generated by all codewords of Hamming weight one.

Proof: Suppose that \mathbf{L}' is isomorphic to $Z_{k_1} \times \dots \times Z_{k_n}$. It can then be seen that $\Lambda = \Lambda'' + \mathbf{q}(\mathbf{L})B$, where Λ'' is the primitive sublattice of Λ in the coordinate system along the orthogonal basis of Λ' , and $\mathbf{q}(\mathbf{L})$ is a fully dynamical code defined over $Z_{\frac{g_1}{k_1}} \times \dots \times Z_{\frac{g_n}{k_n}}$. The result then follows by the definition of the label code. \square

Example 5: Let $\mathbf{L} = \{00, 22, 40, 62\}$ be defined over $\mathbf{G} = Z_8 \times Z_4$. Clearly \mathbf{L} is not fully dynamical, and we have $\mathbf{q}(\mathbf{L}) = \{00, 22\}$ over $Z_4 \times Z_4$. The label code of GCA(\mathbf{L}) is therefore $\{00, 11\}$ over $Z_2 \times Z_2$.

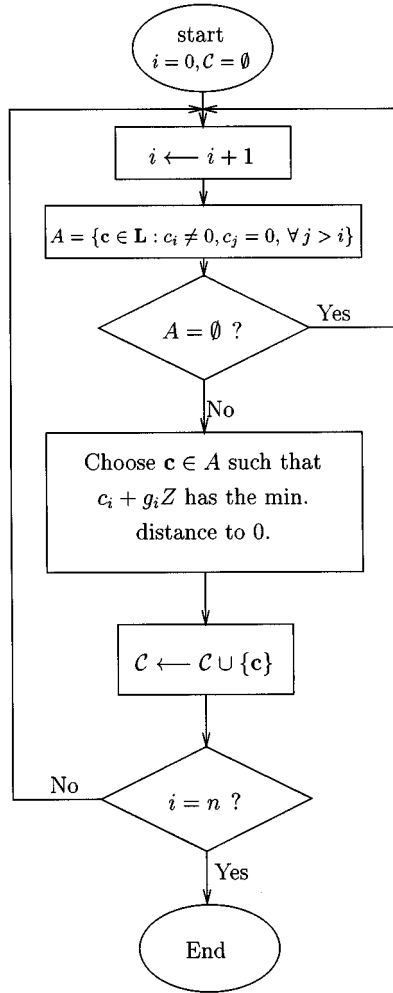
The following proposition and example show the generality and the potential strength of GCA.

Proposition 4: Every lattice with a finite label code in some coordinate system can be constructed by a GCA in that coordinate system.

Example 6: For the repetition code $\mathbf{L} = \{00, 11\} \subset Z_2 \times Z_2$, maximizing the coding gain of GCA(\mathbf{L}) with respect to a_1 and a_2 results in the hexagonal lattice with $\gamma = 2/\sqrt{3}$. This is achieved for $\{a_1, a_2\} = \{1, \sqrt{3}\}$ or $\{1, 1/\sqrt{3}\}$.

Note that the Construction A lattice of \mathbf{L} has a coding gain of just one.

It is important to note that although the coding gain of Λ depends on the values of a_1, \dots, a_n , its TG structure in the corresponding coordinate system is independent of these values.

Fig. 2. Flowchart for obtaining a generating set for the group code \mathbf{L} .

For the rest of the paper, to simplify the discussions, we assume that \mathbf{L} is fully dynamical and trim. This covers the important case of the label code of a lattice.

B. Algorithm for Generating the Dual Code

In the following, we first develop an algorithm which obtains a generating set \mathcal{C} with at most n codewords for a group code \mathbf{L} of length n , given the codewords of \mathbf{L} . The flowchart for the algorithm is given in Fig. 2. The main idea is to construct a lattice Λ from \mathbf{L} by GCA, then find a basis B for Λ , and finally apply the homomorphism $\Phi: \Lambda \rightarrow \mathbf{L}$ to B to obtain \mathcal{C} . The following proposition explains this in more detail.

Proposition 5: The algorithm of the flowchart of Fig. 2 results in a generating set \mathcal{C} for a group code $\mathbf{L} \subset Z_{g_1} \times \cdots \times Z_{g_n}$.

Proof: We first construct a lattice Λ from \mathbf{L} by GCA and by selecting $a_i = g_i, \forall i$. Let $V_0 = \{0\}$ and $V_i = V_{i-1} \oplus W_i, 1 \leq i \leq n$, where W_i is the 1-D subspace corresponding to the i th coordinate. We then construct a basis $B = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ for Λ . We begin by selecting \mathbf{b}_1 as a shortest vector of the lattice $\Lambda \cap V_1$. Now suppose that vectors $\mathbf{b}_1, \dots, \mathbf{b}_i$ have been chosen ($1 \leq i \leq n-1$). We then choose \mathbf{b}_{i+1} to be a lattice vector in $\Lambda \cap V_{i+1}$ with a minimum nonzero distance to V_i . This results in a basis for Λ [3]. To select \mathbf{b}_i in the above algorithm, one only

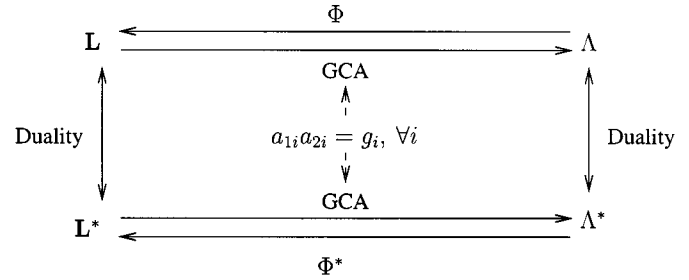


Fig. 3. Relationships among code, its GCA lattice, and their duals.

needs to check the codewords of A in the flowchart of Fig. 2. If $A = \emptyset$, then $\mathbf{b}_i = g_i \mathbf{u}_i$, where \mathbf{u}_i is the i th unit coordinate vector. If $A \neq \emptyset$, then \mathbf{b}_i will be a modification of a codeword $\mathbf{c} \in A$ for which $c_i + g_i Z$ has the minimum distance to zero. The modification is to replace c_i with the closest point of $c_i + g_i Z$ to zero. Let J and J^\perp denote the set of indices for which $A = \emptyset$ and its complement, respectively.

By construction, the basis matrix B is lower triangular. To obtain a generating set for \mathbf{L} , we apply the homomorphism $\Phi: \Lambda \rightarrow \mathbf{L}$ to B . For every $i \in J$, \mathbf{b}_i is mapped to the zero codeword, and for $i \in J^\perp$, it is mapped to the codeword $\mathbf{c} \in A$ such that $c_i + g_i Z$ has the minimum distance to zero. \square

We can now proceed to obtain a basis for Λ^* using the relationship $B^* = (B^{-1})^T$, where B is the basis of Λ described in the above proof. Since by Theorem 2 \mathbf{L}^* is in fact the label code of Λ^* , its generating set \mathcal{C}^* can be obtained by applying the homomorphism $\Phi^*: \Lambda^* \rightarrow \mathbf{L}^*$ to B^* . This procedure is explained by the following example.

Example 1 (Continued): By applying the above algorithm to the group code \mathbf{L} of Example 1, we obtain the following matrices as a basis for Λ (GCA lattice) and a generator for \mathbf{L} , respectively:

$$B = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 0 & 4 & -2 & 0 \\ 0 & 0 & 3 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 0 & 3 & 1 \end{pmatrix}.$$

It can be seen that C is in fact a basis of \mathbf{L} . We then have

$$B^* = \begin{pmatrix} \frac{1}{2} & -\frac{1}{6} & -\frac{1}{3} & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & -2 \\ 0 & 0 & -\frac{1}{2} & \frac{3}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad C^* = \begin{pmatrix} 1 & 5 & 4 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 0 & 3 & 1 \end{pmatrix}.$$

Note that the proposed algorithm is much more efficient than an exhaustive search, particularly for long codes with large alphabets.

C. Relationships Between Dual Codes and Dual Lattices

We end this section by further investigating the relationships among a group code, its GCA lattice, and their duals. These are summarized in Fig. 3.

Theorem 4: Let $\Lambda^{(1)} = Z^n A_1 + \mathbf{L} B_1$ and $\Lambda^{(2)} = Z^n A_2 + \mathbf{L}^* B_2$ be the lattices constructed by GCA from codes \mathbf{L} and

\mathbf{L}^* , respectively. Then $\Lambda^{(2)} = (\Lambda^{(1)})^*$ if and only if $A_1 A_2 = \text{diag}(g_1, \dots, g_n)$, or equivalently, $a_{1i} a_{2i} = g_i, \forall i$.

Proof: If $\Lambda^{(2)} = (\Lambda^{(1)})^*$, then

$$a_{2i} = \det(\Lambda_{W_i}^{(2)}) = 1/\det(P_{W_i}(\Lambda^{(1)})).$$

This combined with

$$\det(P_{W_i}(\Lambda^{(1)})) = \det(\Lambda_{W_i}^{(1)})/g_i = a_{1i}/g_i \quad \forall i$$

results in $a_{1i} a_{2i} = g_i \forall i$.

To prove the other direction of the claim, let $\mathbf{v} = \mathbf{k}A_1 + \mathbf{c}B_1$ and $\mathbf{v}' = \mathbf{j}A_2 + \mathbf{c}'B_2$ be arbitrary vectors in $\Lambda^{(1)}$ and $\Lambda^{(2)}$, respectively, where $\mathbf{c} \in \mathbf{L}$, $\mathbf{c}' \in \mathbf{L}^*$ and $\mathbf{k}, \mathbf{j} \in \mathbb{Z}^n$. We then have

$$\mathbf{v} \cdot \mathbf{v}' = \mathbf{k}A_1 A_2^T \mathbf{j}^T + \mathbf{k}A_1 B_2^T \mathbf{c}'^T + \mathbf{c}B_1 A_2^T \mathbf{j}^T + \mathbf{c}B_1 B_2^T \mathbf{c}'^T. \quad (18)$$

Using the assumption $A_1 A_2 = \text{diag}(g_1, \dots, g_n)$, we have $A_1 B_2^T = B_1 A_2^T = I$, and $B_1 B_2^T = \text{diag}(1/g_1, \dots, 1/g_n)$. It is then easy to see that each term in (18) is integer, and thus the inner product $\mathbf{v} \cdot \mathbf{v}'$ is integer. Since \mathbf{v} and \mathbf{v}' were chosen arbitrarily, this means $\Lambda^{(1)} \subset (\Lambda^{(2)})^*$ and $\Lambda^{(2)} \subset (\Lambda^{(1)})^*$.

Using Theorem 2, we have

$$(\Lambda^{(1)})^* = \mathbb{Z}^n B_1^{-1} + \mathbf{L}^* A_1^{-1}$$

and

$$(\Lambda^{(2)})^* = \mathbb{Z}^n B_2^{-1} + \mathbf{L} A_2^{-1}.$$

By choosing arbitrary elements $\mathbf{v} \in (\Lambda^{(1)})^*$, $\mathbf{v}' \in (\Lambda^{(2)})^*$, and by taking similar steps as in the previous part of the proof, it can be shown that $\mathbf{v} \cdot \mathbf{v}'$ is also integer, and therefore $(\Lambda^{(2)})^* \subset \Lambda^{(1)}$ and $(\Lambda^{(1)})^* \subset \Lambda^{(2)}$.

Putting the two parts together, we have $\Lambda^{(2)} = (\Lambda^{(1)})^*$. \square

V. TG COMPLEXITY

A. Group Codes

Based on the construction of Section II-B, the problem of finding an optimal TG (with respect to a certain measure) for a group code \mathbf{L} can be reduced to the problem of finding an appropriate set of generators for \mathbf{L}^* . We call a TG of \mathbf{L} *minimal* if it minimizes both the number of check nodes and the number of edges. This corresponds to a minimal set of generators with minimum number of nonzero elements for \mathbf{L}^* , and for a cycle-free graph, it translates to the minimum decoding complexity for \mathbf{L} . Note that in a conventional TG for a linear block code, the number of check nodes is fixed and is equal to the rank of the parity-check matrix. For a group code, however, this can vary considerably depending on the selected set of generators for the dual code.

Example 1 (Continued): It is easy to see that \mathbf{L}^* is not cyclic, and thus cannot be generated by a single generator. It can also be verified that an optimum set of generators which results in a minimal TG is the basis $\{1120, 0031\}$. The corresponding TG is given in Fig. 4. Note that this graph is much simpler than the TG of Fig. 1.

B. Lattices

For lattices, the problem of finding a low-complexity TG is more complicated, since one has also the freedom of selecting

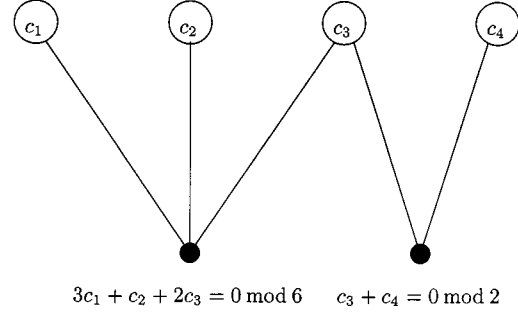


Fig. 4. A minimal TG for the group code of Example 1.

the graph coordinate system. For a given coordinate system, however, the problem is reduced to that of a group code. In the following, to tackle the problem for a lattice, we divide it into two subproblems: 1) finding a graph coordinate system which minimizes the label-code complexity; 2) obtaining a minimal TG for the label code obtained in the first part. In the following section, we discuss the first subproblem.

1) Lower Bounds on Label-Code Complexity: There are many possible complexity measures for the label code \mathbf{L} , among which are the sizes of the label groups ($g_i, i = 1, \dots, n$), and the size of the label code ($|\mathbf{L}|$). The sizes of the label groups play an important role in the graph-based decoding complexity of the label code [27], as g_i determines the size of the symbol alphabet at the i th position of the code. On the other hand, the size of the label code is equal to the number of paths in the trellis of the lattice (constructed in the same coordinate system), which is a fundamental measure of trellis complexity [3]. In this work, we take $|\mathbf{G}| = \prod_{i=1}^n g_i$ as a measure of label-code complexity. Since $|\mathbf{G}| = |\mathbf{L}| |\mathbf{L}^*|$, this measure is also related to $|\mathbf{L}|$. In fact, by Theorem 2, for the important class of self-dual lattices, minimizing $|\mathbf{G}|$ is equivalent to minimizing $|\mathbf{L}|$. We call a graph coordinate system *optimal* if it minimizes $|\mathbf{G}|$, and *strictly optimal* if it also minimizes $|\mathbf{L}|$.

In the following, we derive tight lower bounds on label-code complexity in terms of the coding gain of the lattice and its dual, and show that for many important lattices, there are coordinates in which the bounds are achieved.

Theorem 5: For any n -D lattice Λ , and in any graph coordinate system

$$g_i \geq \lambda(\Lambda)\lambda(\Lambda^*) = [\gamma(\Lambda)\gamma(\Lambda^*)]^{1/2}, \quad i = 1, \dots, n. \quad (19)$$

For each i , the bound is achieved if and only if both Λ and Λ^* have a vector of minimum length along the graph coordinate i .

Proof: We have

$$\begin{aligned} g_i &= \frac{\det(\Lambda_{W_i})}{\det(P_{W_i}(\Lambda))} = \det(\Lambda_{W_i}) \det((\Lambda^*)_{W_i}) \geq \lambda(\Lambda)\lambda(\Lambda^*) \\ &= [\gamma(\Lambda)\gamma(\Lambda^*)]^{1/2} \end{aligned}$$

where for the last step we have used (10). \square

The bound of (19) imposes constraints on the coding gain of lattices constructed from codes. For instance, the bound would explain why ‘‘Construction A’’ cannot result in lattices with high coding gains.

Corollary 2: For any lattice Λ , and in any graph coordinate system

$$|\mathbf{G}| \geq [\gamma(\Lambda)\gamma(\Lambda^*)]^{n/2} \quad (20)$$

where the bound is achieved if and only if both Λ and Λ^* have n mutually orthogonal vectors of minimum length along the graph coordinates.

For a self-dual lattice Λ , minimizing $|\mathbf{G}|$ is equivalent to minimizing $|\mathbf{L}|$, which in turn is the same as minimizing the number of paths in the trellis of Λ constructed in the same coordinate system. The latter problem has been extensively studied in [2]–[4]. Based on the results of [2], for many important self-dual lattices such as the Leech lattice and the Barnes–Wall lattices BW_n , $n = 2^m$, m odd, the lower bounds of (19) and (20) are achieved.

It is also interesting to note that (19) implies that the coding gain of a self-dual lattice must be an integer.

The following corollary follows from Theorem 5 and the fact that g_i must be an integer for every i .

Corollary 3: For any lattice Λ , and in any graph coordinate system

$$g_i \geq \lceil \gamma(\Lambda)^{1/2} \gamma(\Lambda^*)^{1/2} \rceil, \quad i = 1, \dots, n$$

$$|\mathbf{G}| \geq \lceil \gamma(\Lambda)^{1/2} \gamma(\Lambda^*)^{1/2} \rceil^n.$$

The lower bounds of Corollary 3 are achieved for many well-known lattices such as D_n , D_n^* , $n \geq 3$, and E_7 , E_7^* [2]. For many other important lattices, however, we are able to further improve the lower bounds of Corollary 3. This is explained in the following.

Proposition 6: For Barnes–Wall lattices BW_n , $n = 2^m$, m even, in any coordinate system, we have

$$g_i \geq \sqrt{2}\gamma = \sqrt{n}, \quad \forall i \quad \text{and} \quad |\mathbf{G}| \geq (\sqrt{2}\gamma)^n = n^{n/2}.$$

Proof: These lattices are iso-dual. Thus, $g_i \geq \gamma \forall i$. However, $\gamma = \sqrt{n/2}$ is not an integer, and therefore there does not exist any coordinate W_i such that both Λ and Λ^* have a vector of minimum length along W_i . The best scenario to minimize g_i is thus for one of the lattices Λ or Λ^* to have a vector of minimum length and for the other one to have a vector of its second shell along W_i . This results in $g_i \geq \sqrt{2}\lambda(\Lambda)\lambda(\Lambda^*) = \sqrt{2}\gamma$. \square

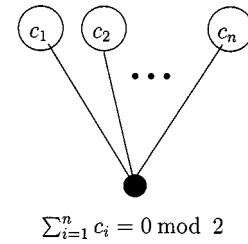
Strictly optimal coordinate systems which achieve these lower bounds can be found in [2].

Lattices E_6 and E_6^* are, respectively, the densest packing and the best quantizer in six dimensions. For these lattices, the lower bounds of Corollary 3 are $g_i \geq 2, \forall i$ and $|\mathbf{G}| \geq 64$. These are improved in the following proposition.

Proposition 7: For E_6 and E_6^* , in any coordinate system, we have $\{g_i, i = 1, \dots, 6\} \geq \{2^4, 4^2\}$; i.e., at least two of the label groups have a size of at least 4, and $|\mathbf{G}| \geq 256$.

Proof: We consider a version of E_6 such that E_6 and E_6^* have the following lengths of nonzero vectors in increasing order:

$$E_6: \sqrt{2}, 2, \sqrt{6}, \sqrt{8}, \sqrt{10}, \dots$$



$$|\mathbf{G}| = 2^n, \quad |\mathbf{L}| = 2^{n-1}$$

Fig. 5. A minimal TG for D_n .

$$E_6^*: 2/\sqrt{3}, \sqrt{2}, \sqrt{10/3}, 2, 4/\sqrt{3}, \sqrt{6}, \dots$$

For $g_i = \det(\Lambda_{W_i}) \det((\Lambda^*)_{W_i})$ to be equal to two, $\det(\Lambda_{W_i})$ has to be $\sqrt{2}$ for E_6 . Examination of the minimal vectors of E_6 (given in [6, p. 126]) shows that there exist at most four of them which are mutually orthogonal. Moreover, by examining the above lengths of the vectors for E_6 and E_6^* , it can be seen that g_i cannot be equal to three for any value of i . This completes the proof. \square

Strictly optimal coordinate systems for E_6 and E_6^* can be found in [3].

Another important lattice is the Coxeter–Todd lattice K_{12} , which is the best known packing and quantizer in $n = 12$. For K_{12} , the lower bounds of Corollary 3 are $g_i \geq 3 \forall i$, and $|\mathbf{G}| \geq 3^{12}$. The following proposition improves these bounds. We omit the proof, since it is similar to those of previous propositions.

Proposition 8: For K_{12} , in any coordinate system, we have $g_i \geq 4 \forall i$, and $|\mathbf{G}| \geq 4^{12}$.

A strictly optimal coordinate system for K_{12} can be found in [2].

2) *Tanner Graph Structure of Specific Lattices:* In this subsection, we study the TG structure of some important lattices in (strictly) optimal coordinate systems.

Example 7: A well-known construction for lattices D_n , $n \geq 3$, is based on applying Construction A to the binary linear single-parity-check code $(n, n-1, 2)$. In the corresponding coordinate system, $|\mathbf{L}| = 2^{n-1}$ and $g_i = 2, \forall i$. To construct the TG in this coordinate system, the only possible selection for the generating set of \mathbf{L}^* is the all-one vector, which results in the TG of Fig. 5.

In [3], another coordinate system which also minimizes $|\mathbf{L}|$ was introduced. In this strictly optimal system, the label code is the binary linear code $(n, \frac{n-2}{2}, 4)$ or $(n, \frac{n-1}{2}, 3)$, for n even or odd, respectively. The corresponding minimal TGs are given in Fig. 6. They are constructed based on the following generating sets for \mathbf{L}^* , for n even and odd, respectively:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ & & & & & \dots & & \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & \dots & 1 & 0 \end{pmatrix}$$

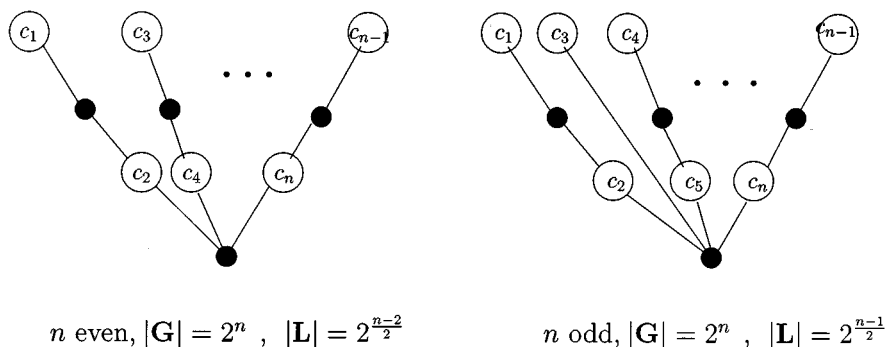


Fig. 6. Minimal TGs for D_n in a strictly optimal coordinate system.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & \cdots & 0 & 0 \\ & & & & & & \cdots & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

It is worth noting that compared to the TG of Fig. 5, the TGs of Fig. 6 require a smaller number of computations to decode D_n using a two-way algorithm.

In [8], the authors derived upper bounds on the minimum distance of a linear block code that can be represented by a TG without cycles. It is easy to see that for both Figs. 5 and 6, the label codes are optimal cycle-free linear block codes in the sense that they achieve the upper bounds of [8]. Using these bounds, we also show that the optimal label codes of some important lattices do not have conventional cycle-free TGs. For some other lattices, however, the label codes are not defined over a finite field, and thus the results of [8] are not applicable. In the following, we study a few such cases, and prove that cycle-free TGs, in fact, do not support the corresponding label codes. Although in this work, we consider only a few categories of lattices, the same ideas can also be applied to other lattices.

Example 8: In the strictly optimal coordinate system of [3], the label code of E_8 is the first-order binary (8, 4, 4) Reed–Muller code. This code does not satisfy the upper bound of $d \leq 2$, for cycle-free linear codes of length 8 and dimension 4 [8]. It thus does not have a cycle-free TG. In fact, it can be seen that any minimal TG for this code has 16 edges, four check nodes, and cycles of minimum length 4.

For E_7 and E_7^* also, the label codes in the strictly optimal coordinate systems of [3] are linear block codes. They are in fact the little Hamming code (7, 3, 4) and the Hamming code (7, 4, 3), respectively. For both codes, the upper bound of [8] proves that there does not exist any cycle-free TG. For E_6 , E_6^* , and BW_n , $n = 2^m$, $m \geq 4$, however, the optimal label codes

are not linear block codes, and therefore the results of [8] cannot be applied. For all these cases, we have proved that the label codes cannot, in fact, be supported by cycle-free TGs. In the following, we provide the proof only for E_6 .

Example 9: In an strictly optimal coordinate system for E_6 , we have the following label code and its dual over

$$\mathbf{G} = Z_2 \times Z_2 \times Z_4 \times Z_4 \times Z_2 \times Z_2$$

as shown at the bottom of the page. Let $|E|$ and $|V|$ denote the number of edges and the number of nodes in any TG for \mathbf{L} , respectively. The maximum order of any codeword of \mathbf{L}^* is 4, and none of the codewords of order 4 are independent. This implies that any generating set for \mathbf{L}^* will have $n \geq 3$ codewords. If $n = 3$, at least one of the codewords must have a weight of 4, because any codeword with a weight different than 4 has an order of only 2, and thus three of them cannot generate \mathbf{L}^* . This combined with the fact that there exists only one nonzero codeword of weight 2 results in

$$|E| \geq 2 + 3 + 4 > 8 = |V| - 1$$

which in turn implies that the TG contains a cycle. If $n > 3$, we have

$$|E| \geq 2 + 3(n - 1) > 6 + n - 1 = |V| - 1$$

which again implies that the TG has a cycle.

VI. CONCLUDING REMARKS

One of the main results of this paper is Theorem 2, which shows that by properly defining the dual \mathbf{L}^* of an Abelian group block code \mathbf{L} , the dual of the label code of a lattice Λ is the label code of the dual lattice Λ^* ; i.e., $\mathbf{L}(\Lambda)^* = \mathbf{L}(\Lambda^*)$. Using this theorem, given a group code \mathbf{L} , we can find a generating set for \mathbf{L}^* . This can be then used to form a set of modular linear

$$\mathbf{L} = \left\{ \begin{matrix} 000000, & 103301, & 002200, & 101101, & 013310, & 112211, & 011110, & 110011, \\ 110200, & 013101, & 112000, & 011301, & 103110, & 002011, & 101310, & 000211 \end{matrix} \right\}$$

$$\mathbf{L}^* = \left\{ \begin{matrix} 000000, & 103310, & 002200, & 101110, & 013301, & 112211, & 011101, & 110011, \\ 110200, & 013110, & 112000, & 011310, & 103101, & 002011, & 101301, & 000211 \end{matrix} \right\}.$$

constraints which defines a Tanner graph for L , and as a special case, for every generalized Construction A (GCA) lattice.

There are still many unanswered questions and open problems, and we hope that this paper will stimulate more work in this area. Probably the most important question is: "Which TGs are good for decoding?" It is also interesting to rigorously state and prove the following conjecture: "Good lattices cannot be supported by cycle-free TGs." GCA seems to be a promising approach to the construction of dense lattices with low iterative decoding complexity, probably based on a group replica of LDPC codes. How GCA can be used in such a construction remains to be studied.

ACKNOWLEDGMENT

The authors wish to acknowledge the anonymous referees for their helpful comments and suggestions, which greatly improved the presentation of the paper. They also wish to thank Dr. G. D. Forney, Jr. for handling this paper throughout the review process.

REFERENCES

- [1] S. M. Aji, G. B. Horn, and R. J. McEliece, "Iterative decoding on graphs with a single cycle," in *Proc. 1998 IEEE Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 276.
- [2] A. H. Banihashemi, "Trellis structure and decoding complexity of lattices," Ph.D. dissertation, E&CE Dept., Univ. Waterloo, Waterloo, ON, Canada, 1997.
- [3] A. H. Banihashemi and I. F. Blake, "Trellis complexity and minimal trellis diagrams of lattices," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1829–1847, Sept. 1998.
- [4] —, "On the trellis complexity of root lattices and their duals," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2168–2172, Sept. 1999.
- [5] A. R. Calderbank, G. D. Forney Jr., and A. Vardy, "Minimal tail-biting trellises: The Golay code and more," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1435–1455, July 1999.
- [6] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 2nd ed. New York: Springer-Verlag, 1993.
- [7] R. de Buda, "Some optimal codes have structure," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 893–899, Aug. 1989.
- [8] T. Etzion, A. Trachtenberg, and A. Vardy, "Which codes have cycle-free Tanner graphs?," in *Proc. 1998 IEEE Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 207.
- [9] G. D. Forney Jr., "Density/length profiles and trellis complexity of lattices," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1753–1772, Nov. 1994.
- [10] —, "On iterative decoding and the two-way algorithm," in *Int. Symp. Turbo Codes*, Brest, France, Sept. 1997.
- [11] G. D. Forney Jr. and M. D. Trott, "The dynamics of group codes: state spaces, trellis diagrams, and canonical encoders," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1491–1513, Sept. 1993.
- [12] B. J. Frey, R. Kötter, and A. Vardy, "Skewness and pseudo-codewords in iterative decoding," in *Proc. 1998 IEEE Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 148.
- [13] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.
- [14] M. Hall, *The Theory of Groups*. New York: Macmillan, 1959.
- [15] R. Kötter and A. Vardy, "Factor graphs: constructions, classification, and bounds," in *Proc. 1998 IEEE Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 14.
- [16] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," in *IEEE J. Select. Areas Commun.*, vol. 16, Feb. 1998, pp. 219–230.
- [17] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.
- [18] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs and belief propagation," in *Proc. 1998 IEEE Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 117.
- [19] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.
- [20] R. J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1072–1092, July 1996.
- [21] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, Feb. 1998.
- [22] R. J. McEliece and M. Xu, "Junction tree representations for linear block codes," in *Proc. 1998 IEEE Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 253.
- [23] A. Reznik, "Iterative decoding of codes defined on graphs," M.Sc. thesis, Dept. Elec. Eng. Comp. Sci., MIT, Cambridge, MA, June 1998.
- [24] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.
- [25] —, "Codes with sparse graphs: Transform analysis and constructions," in *Proc. 1998 IEEE Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 116.
- [26] V. Tarokh, "Minimal tanner graphs for block codes and lattices," in *Proc. 5th Can. Workshop on Information Theory*, Toronto, ON, Canada, June 1997, pp. 9–10.
- [27] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Dept. Elec. Eng., Univ. Linköping, Linköping, Sweden, Apr. 1996.
- [28] N. Wiberg, H.-A. Loeliger, and R. Kötter, "Codes and iterative decoding on general graphs," *Euro. Trans. Telecommun.*, vol. 6, pp. 513–526, Sept. 1995.