An Efficient Algorithm for Constructing Minimal Trellises for Codes over Finite Abelian Groups

Vijay V. Vazirani [*]
Huzur Saran [†]
B. Sundar Rajan [‡]

Abstract

We present an efficient algorithm for computing the minimal trellis for a group code over a finite Abelian group, given a generator matrix for the code. We also show how to compute a succinct representation of the minimal trellis for such a code, and present algorithms that use this information to efficiently compute local descriptions of the minimal trellis. This extends the work of Kschischang and Sorokine, who handled the case of linear codes over fields. An important application of our algorithms is to the construction of minimal trellises for lattices.

A key step in our work is handling codes over cyclic groups C_{p^α}, where p is a prime. Such a code can be viewed as a submodule over the ring Z_{p^α}. Because of the presence of zero-divisors in the ring, submodules do not share the useful properties of vector spaces. We get around this difficulty by restricting the notion of linear combination to p-linear combination, and introducing the notion of a p-generator sequence, which enjoys properties similar to that of a generator matrix for a vector space.

Index Terms– Trellis, group codes, codes over rings, lattices, algorithms, Gaussian elimination.

[*]College of Computing, Georgia Institute of Technology, vazirani@cc.gatech.edu
[†]Department of Computer Science and Engg., Indian Institute of Technology, Delhi, hsaran@cse.iitd.ernet.in
[‡]Electrical Engineering Department, Indian Institute of Technology, Delhi, bsrajan@ee.iitd.ernet.in

List of figure captions:

1. The trellis for a single vector (example 1)
2. The trellis from an arbitrary generator matrix (example 2)
3. The trellis for the code generated by a single vector over a ring.
4. The trellis for Example 8
5. The trellis for Example 9
6. The trellis for Example 10
7. The trellis for Example 11
8. The trellis for Example 12

1 Introduction

Ever since the success of trellis coded modulation [22] (which revolutionized transmission rates of modems in bandwidth limited channels), researchers have been studying block coded modulation [7, 12, 14]. Block group codes constitute a basic ingredient for a large class of block coded modulation schemes [7]. The coding gain achieved by these schemes is possible only with soft-decision decoding [22, 21]. Trellises provide a general framework for efficient soft-decision decoding of codes [25], for instance using the Viterbi algorithm [4]. Since the decoding effort is directly related to the size of the trellis, much work has been done on characterizing and constructing minimal trellises for group codes [20, 6, 9, 15, 16].

In this paper, we present an $O(k^2n + s)$ time algorithm for constructing the minimal trellis for a block code over a finite Abelian group, given a generator matrix for the code, where n is the length of the code, k is the number of rows in the generator matrix, and s is the number of states in the minimal trellis; throughout the paper, we will assume that it takes one unit of time to perform an operation over the underlying field, ring or group. For decoding purposes, it is perhaps more important to be able to efficiently compute required local descriptions of the minimal trellis. For this purpose, we show how a succinct description of the minimal trellis for such codes can be computed in $O(k^2n)$ time and occupying $O(kn)$ space; notice that this is polynomial in n amount of space, even though the minimal trellis may be exponentially large. We give algorithms that use this information to compute, for example, all transitions in to or out of a state in $O(k)$ time.

Perhaps the most important application of our work is to the construction of minimal trellises for lattices, since this problem essentially reduces to that of constructing minimal trellises for block codes over Abelian groups. This is elaborated in Section 11. Another application arises as a consequence of the following result: Certain famous non-linear binary codes (including Kerdock, Preparata and Goethals codes) contain more codewords than any known linear code of the same length. In a recent breakthrough result, Hammons, Kumar, Calderbank, Sloane and Sole have shown that under the Gray map from $(Z_2)^2$ to the ring Z_4, these codes turn out to be linear over Z_4 [11]. Note that linear codes over Z_4 are the same as group codes over C_4.

We have built directly on the work of Forney and Trott [9] and Kschischang and Sorokine [15]. Forney and Trott, building on the work of Willems on dynamical systems [23, 24], show that group codes admit unique minimal trellises. Furthermore, they present important structural properties of such trellises, especially in their State Space Theorem (see Section 3). Kschischang and Sorokine have given an $O(k^2n+s)$ time algorithm for constructing the minimal trellis for a linear code over a field, given a generator matrix for the code (see Section 4). They also present an efficient algorithm for computing local descriptions of the minimal trellis.

The essential step in the algorithm of Kschischang and Sorokine is obtaining a special generator matrix for the code: a two-way proper generator matrix. A simpler proof is offered to show that such a generator matrix yields a minimal trellis (Section 4). A key step towards extending this to codes over finite Abelian groups is handling codes over cyclic groups C_{p^α}, where p is a prime. Such codes can be viewed as linear codes over the ring Z_{p^α} and are therefore submodules over Z_{p^α}. The extension is not straightforward; the main difficulty is the presence of zero-divisors in the ring. In Section 5 we state the difficulties encountered because of zero-divisors. Some of these are quite general, e.g., the inability to give satisfactory definitions for basis and dimension of submodules over Z_{p^α}; and others are specific to minimal trellises. We then introduce the notions of p-linear

combinations and p-generator sequences that enable us to get around these difficulties (Section 6). We show how Gaussian elimination can be adapted to this setting, and can be used for obtaining a p-generator sequence for any submodule over Z_{p^α}, given a usual generator matrix for it. These notions should find other applications as well, since they enable one to perform certain operations on submodules over Z_{p^α} similar to the manner in which these operations are performed on subspaces of a vector space.

In Section 7 we give a natural generalization of a two-way proper matrix: a two-way proper p-generator sequence, and we show how Gaussian elimination can be used for obtaining it. Once this is done, a minimal trellis for a linear code over Z_{p^α} can be constructed essentially in the same manner as the field case.

Finally, in Section 8 we consider codes over finite Abelian groups. First, we show that group codes over elementary Abelian groups can be seen as linear codes over an appropriate finite field. We obtain a minimal trellis for this linear code, and from this trellis, using sectionalization, we obtain a minimal trellis for the given group code. For dealing with arbitrary finite Abelian groups, we show that it is sufficient to consider Abelian p-groups. A code over such a group is in turn same as a linear code over a ring Z_{p^α}, and can be handled analogously.

The problem of computing local descriptions of minimal trellises is addressed in Section 9. Two types of problems are solved: Given two states in successive time indices, determine if there is a transition between them, and if so, determine the set of labels on the transition. Also, given a state at time index i, compute all transitions in to it and out of it.

In Section 10, we build on the State Space Theorem to give algebraic structural properties of the set of transitions between two time indices in the minimal trellis for a group code; we call this the Transition Space Theorem. This theorem also defines a succinct representation for the minimal trellis for a group code, from which local descriptions can be computed. This applies to group codes over non-Abelian groups as well; however, in general, the representation may be super-polynomial sized.

2 Preliminaries

In this paper, we will only deal with *block codes*, i.e., codes for which each codeword is of the same length, denoted by n. Let G be a finite group (in this paper, all codes are over finite Abelian groups), and let $W = G^n$ be the n-fold direct product of G. A subgroup $\mathcal{C}$ of W under the componentwise addition operation of G is said to be a *group code over* G. Let I denote the set of positive integers from 1 to n; I will be called the *time axis*. An element $a \in W$ will be called a *sequence*; $a = (a_i, i \in I)$.

Let R be a ring; as a special case, R may also be a field. As before, let $W = R^n$. Let $\mathcal{C}$ be a subgroup of W under the componentwise addition operation of R, and assume furthermore that $\mathcal{C}$ is closed under multiplication with elements of R, again carried out componentwise. Then, $\mathcal{C}$ is said to be a *linear code over* R. Clearly, the class of linear codes over fields is contained in the class of linear codes over rings, which is in turn contained in the class of group codes.

A *trellis, T,* for a group code $\mathcal{C}$ is an edge-labelled directed layered graph. The vertices of T are partitioned into disjoint subsets $V_0, V_1, \cdots V_n$. The set V_i is referred to as the set of *states* at time

index i. V_0 contains a unique start state v_0, and V_n contains a unique terminating state v_n. Edges of T are allowed to run only between states in successive time indices. A *transition* $(u \to v)$, $u \in V_i, v \in V_{i+1}$ is labelled with a non-empty subset of elements from the group G. This transition is said to be *out of* state u and *in to* state v. A state having more than one out-transition (in-transition) will be called *forking state (collapsing state)*. A state u in trellis T will be said to be *forward proper (backward proper)* if the sets of labels on the out-transitions (in-transitions) of u are pairwise disjoint. Finally, trellis T will be said to be *two-way proper* if each of its states is forward proper and backward proper.

A *path* from v_0 to v_n consists of n transitions, $v_0 \to v_1 \to v_2 \cdots \to v_n$, where $v_i \in V_i$. Such a path defines all the n length words $(\alpha_1, \alpha_2, \cdots \alpha_n)$, where α_i is drawn from the set labelling the transition $(v_{i-1} \to v_i)$. We require that each state must be useful, i.e., it must be on some path from v_0 to v_n. Finally, we require that the set of all words defined by all paths in T from v_0 to v_n be exactly the set of codewords in $\mathcal{C}$. We will say that state s is *responsible* for all the codewords whose paths use state s.

Clearly, there exists a trellis for each group code $\mathcal{C}$: create a unique path from v_0 to v_n for each codeword, with unique intermediate states. Such a trellis will have as many states at each time index as the number of codewords in $\mathcal{C}$. For several reasons, including efficient decoding, it is important to obtain a trellis for $\mathcal{C}$ having as few states as possible. Let us say that T is a *minimal trellis* for $\mathcal{C}$ if at each time index, T has the smallest possible number of states.

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two group codes over the same underlying group G, and let T_1 and T_2 be trellises for these codes. Let $\mathcal{C} = \mathcal{C}_1\mathcal{C}_2$ be the product of these two group codes. Notice that in general, $\mathcal{C}$ may not be a group code; however, if G is commutative, $\mathcal{C}$ will be a group code. We can define the operation of taking the *product of trellises* T_1 and T_2 to obtain a trellis T for the code $\mathcal{C}$ as follows: Let $U_i, 0 \le i \le n$ and $V_i, 0 \le i \le n$ be the set of states of T_1 and T_2. Trellis T will have states $W_i, 0 \le i \le n$, where $|W_i| = |U_i||V_i|$, and corresponding to each pair of states $u \in U_i$ and $v \in V_i$, there is a state $(u, v) \in W_i$. There is a transition from $(u, v) \in W_i$ to $(u', v') \in W_{i+1}$ iff $(u \to u')$ and $(v \to v')$ are transitions in T_1 and T_2 respectively. Let α and β be the labels on the transitions $(u \to u')$ and $(v \to v')$. Then, the set of labels on transition $((u, v) \to (u', v'))$ is $\{ab | a \in \alpha, b \in \beta\}$.

3 Structural properties of minimal trellises for group codes

As established by Forney and Trott, structural properties of group codes lead to structural properties of minimal trellises for such codes. In this section, we will review properties essential for our work, especially those following from the State Space Theorem.

Let $J \subseteq I$ be a subset of the time axis. The *projection map* $P_J : W \to W$ sends sequence $a \in W$ to the following sequence b:

$$b_i = \begin{cases} a_i & \text{if } i \in J \\ 0 & \text{if } i \in I - J. \end{cases}$$

Thus, the projection map P_J simply 'zeros out' the $I - J$ components of a sequence. Define *projection* $P_J(\mathcal{C}) = \{P_J(c) | c \in \mathcal{C}\}$, i.e., the image of $\mathcal{C}$ under the projection map P_J. The projection map is a homomorphism, since $P_J(ab) = P_J(a)P_J(b)$. Further, since $\mathcal{C}$ is a group, the image of $\mathcal{C}$ under P_J, $P_J(\mathcal{C})$ is a subgroup of W. If J consists of the first k time indices, we will denote $P_J(\mathcal{C})$

by $P_{k-}(\mathcal{C})$, and $P_{I-J}(\mathcal{C})$ by $P_{k+}(\mathcal{C})$; for $a \in W, P_{k-}(a)$ and $P_{k+}(a)$ are similarly defined. $P_{k-}(\mathcal{C})$ will be called the set of *codeword pasts* and $P_{k+}(\mathcal{C})$ the the set of *codeword futures*.

The *cross section* of $\mathcal{C}$ in J, denoted by $\mathcal{C}_J$, is a subcode of $\mathcal{C}$ consisting of all codewords whose components in $I - J$ are zero, i.e.,

$$\mathcal{C}_J = \{c \in \mathcal{C} \mid c_k = 0, k \in I - J\}.$$

Notice that $\mathcal{C}_J$ is the kernel of the projection map P_{I-J} restricted to $\mathcal{C}$. Again, if J consists of the first k time indices, we will denote $\mathcal{C}_J$ by $\mathcal{C}_{k-}$ and $\mathcal{C}_{I-J}$ by $\mathcal{C}_{k+}$; these are called the *past subcode* and *future subcode*, respectively, in [9]. Since $\mathcal{C}_{k-}$ and $\mathcal{C}_{k+}$ are both normal subgroups of $\mathcal{C}$, $\mathcal{C}_{k-}\mathcal{C}_{k+}$ is also a normal subgroup of $\mathcal{C}$. Furthermore, since $\mathcal{C}_{k-} \cap \mathcal{C}_{k+} = \{0\}$, $\mathcal{C}_{k-}\mathcal{C}_{k+}$ is a direct product.

The *State Space Theorem* of Forney and Trott [9] states that

$$P_{k-}(\mathcal{C})/\mathcal{C}_{k-} \simeq P_{k+}(\mathcal{C})/\mathcal{C}_{k+} \simeq \mathcal{C}/(\mathcal{C}_{k-}\mathcal{C}_{k+}).$$

It will be instructive to consider the following bipartite graph, H_k the *past-future graph at time index k*: Its vertex sets are $P_{k-}(\mathcal{C})$ and $P_{k+}(\mathcal{C})$, and two vertices $u \in P_{k-}(\mathcal{C})$ and $v \in P_{k+}(\mathcal{C})$ are joined by an edge iff $uv \in \mathcal{C}$. We will say that $(A, B), A \subseteq P_{k-}(\mathcal{C}), B \subseteq P_{k+}(\mathcal{C})$ is a *bipartite clique* if for each $u \subseteq A$ and $v \subseteq B$, (u, v) is an edge in H_k. The State Space Theorem shows that H_k consists of disjoint bipartite cliques.

Notice that $\mathcal{C}_{k-} \subseteq P_{k-}(\mathcal{C})$ and $\mathcal{C}_{k+} \subseteq P_{k+}(\mathcal{C})$. Since $\mathcal{C}_{k-}\mathcal{C}_{k+}$ is a direct product, there is a bipartite clique between the corresponding sets of vertices in H_k. This clique will be called the *zero clique* since it corresponds to the subgroup $\mathcal{C}_{k-}\mathcal{C}_{k+}$ of codewords of $\mathcal{C}$; one of its edges corresponds to the all zeros codeword.

For any $c \in \mathcal{C}$, consider the coset $c\mathcal{C}_{k-}\mathcal{C}_{k+}$. The codewords in this set consist of pasts corresponding to the elements of the coset $P_{k-}(\mathcal{C})P_{k-}(c)$ and futures corresponding to the elements of the coset $P_{k+}(\mathcal{C})P_{k+}(c)$. In H_k, there is a bipartite clique between these sets of vertices; the edges of this clique correspond to $c\mathcal{C}_{k-}\mathcal{C}_{k+}$.

The construction of the unique minimal trellis, T, for $\mathcal{C}$ follows from the past-future graphs $H_k, 1 \leq k < n$. T has $|\mathcal{C}/(\mathcal{C}_{k-}\mathcal{C}_{k+})|$ states at time index k; each state is responsible for codewords in one of the cosets. The state that is responsible for codewords in the subgroup $\mathcal{C}_{k-}\mathcal{C}_{k+}$ will be called the *zero state*, and will sometimes be denoted by 0. If u and v are states at time indices k and $k+1$ respectively, then there is a transition from u to v iff the sets of codewords they are responsible for have a non-empty intersection, say A. If so, the set of labels on this transition is the set of symbols in $P_{(k+1)}(A)$, i.e., the projection of A onto the $(k+1)^{th}$ coordinate. Define the *output group* at time index $k+1$ to be $G_{k+1} = P_{(k+1)}(\mathcal{C})$; notice that G_{k+1} is a subgroup of G.

4 Minimal trellises for codes over fields

In this section we will introduce the algorithm of Kschischang and Sorokine [15] for linear codes over fields, since we build directly on it. We will also give a simpler proof for their algorithm. The running time of their algorithm is $O(k^2n + s)$, where the generator matrix has size $k \times n$, and s is the number of states in the minimal trellis.

Our simplified proof of minimality relies on the following characterization established by Willems [23, 24] in the context of dynamical systems. We will use this characterization for proving minimality of trellises constructed for the cyclic group and Abelian group cases as well.

Theorem 4.1 (Willems [23, 24]) *A two-way proper trellis for a block code, $\mathcal{C}$, is the unique minimal trellis for $\mathcal{C}$.*

For a simplified proof of Theorem 4.1, proven in the context of minimal trellises for group codes, see [17, 18]. For further extensions of Willems' results to codes over finite Abelian groups, see [3]. In general, a code may not admit a two-way proper trellis. However, if such a trellis does exist for the code, it is guaranteed to be minimal; see [13] for a proof of this fact.

A linear code, $\mathcal{C}$, over a field, $GF(q)$, is a vector subspace, and can be described by its generator matrix, A. The minimal trellis for the code generated by a single row vector of A can be obtained in a straightforward manner as explained below. Since $\mathcal{C}$ is the sum of the codes generated by the rows of A, the product of the trellises for these codes will be a trellis for $\mathcal{C}$; the operation of computing the product of trellises was introduced by Kschischang and Sorokine [15] for precisely this reason. In general, this trellis may not be minimal. Forney has called a generator matrix that gives rise to a minimal trellis a *trellis-oriented generator matrix* [6]. The key step in the algorithm of Kschischang and Sorokine is efficiently obtaining a trellis-oriented generator matrix, given an arbitrary generator matrix A.

Let $(a_1 a_2 \cdots a_n)$ be a row of A. Let a_i be the first non-zero entry and a_j be the last non-zero entry in this row, i.e., $a_k = 0$ for $k < i$ and for $k > j$. Then, we will say that this row *starts* at i and *ends* at j. Furthermore, a_i will be called the *starting element* of this row and a_j will be called its *ending element*. The minimal trellis for the code generated by this vector has a simple structure: It has a single forking state with q out-transitions at time index $i-1$, and a single collapsing state with q in-transitions at time index j; all other states have one in and one out transition; see Example 1.

Example 1 For the code generated by a single vector over $GF(q)$, the minimal trellis consists of a forking state at the time index at which this vector starts and a collapsing state at the time index at which this vector ends. For example, the minimal trellis for the code over $GF(5)$ generated by (030210) is given in Figure 1.

We will first prove that for establishing minimality of the product of two minimal trellises, it is sufficient to establish two-way properness of the zero-states at each time index. We will prove this in the full generality of group codes. The lemmas below consider only forward properness – analogous statements hold for backward properness. By the set of labels *emanating* from a state we mean the union of the sets of labels on all transitions out of this state.

Lemma 4.2 *Let T be a minimal trellis for a group code, $\mathcal{C}$, over group G and let s_0 and s be the zero state and an arbitrary state at time index i. Let α_0 and α be the sets of labels emanating from s_0 and s respectively, and let G_{i+1} be the output group at time index $i+1$. Then, α_0 is a normal subgroup of G_{i+1}, and α is an element of the quotient group G_{i+1}/α_0.*

Proof: State s_0 is responsible for the set of codewords in $\mathcal{C}_i$–$\mathcal{C}_{i^+}$. Since $\mathcal{C}_i$–$\mathcal{C}_{i^+} \lhd \mathcal{C}$ (note that as usual, "$\lhd$" denotes normal subgroup),

$$\alpha_0 = P_{(i+1)}(\mathcal{C}_i\text{–}\mathcal{C}_{i^+}) \lhd P_{(i+1)}(\mathcal{C}) = G_{i+1}.$$

Since the set of codewords that s is responsible for form a coset of $\mathcal{C}_i$–$\mathcal{C}_{i+}$ in $\mathcal{C}$, using a similar argument, we get that α is an element of the quotient group G_{i+1}/α_0. ■

Lemma 4.3 *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be length n group codes over the same underlying group G and let $\mathcal{C} = \mathcal{C}_1\mathcal{C}_2$ (in general, $\mathcal{C}$ may not be a group code). Let T_1 and T_2 be minimal trellises for $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, and let T be the product of these trellises, which will be a trellis for code $\mathcal{C}$. Let α_0 and β_0 be the set of labels emanating from the zero states, z_1 and z_2, at time index i in T_1 and T_2 respectively. Then, T is forward proper at time index i iff α_0 and β_0 intersect trivially, i.e., $\alpha_0 \cap \beta_0 = \{0\}$, where 0 is the identity element of G.*

Proof: Since α_0 and β_0 are subgroups of G, by a well-known theorem in group theory,

$$|\alpha_0\beta_0| = \frac{|\alpha_0||\beta_0|}{|\alpha_0 \cap \beta_0|}.$$

So, the labels emanating from the zero state at time index i in T, (z_1, z_2), are all distinct iff $|\alpha_0\beta_0| = |\alpha_0||\beta_0|$, which happens iff $\alpha_0 \cap \beta_0 = \{0\}$.

Consider two arbitrary states s_1 and s_2 in T_1 and T_2 respectively at time index i. Let us view the set of labels emanating from s_1 as a left coset of α_0, say $a\alpha_0$, and those emanating from s_2 as a right coset of β_0, say $\beta_0 b$ (notice that in general α_0 and β_0 may be normal subgroups of different groups in G). Then, the of labels emanating from state (s_1, s_2) in T are given by $a\alpha_0\beta_0 b$. As before, these will be all distinct iff $|\alpha_0\beta_0| = |\alpha_0||\beta_0|$. The lemma follows. ■

We will say that a generator matrix is *two-way proper* if every row starts at a distinct point, and every row ends at a distinct point. Following is a restatement of Theorem 2 of Kschischang and Sorokine [15]; we give a simpler proof for it using Theorem 4.1 and Lemma 4.3.

Theorem 4.4 (Kschischang and Sorokine [15]) *A generator matrix for a linear code over a field is trellis-oriented iff it is two-way proper.*

Proof: Since for a field, multiplication by a non-zero element is a one-to-one onto map, in this case, the set of symbols emanating from a zero state is either $\{0\}$ or the entire field. Compute the product of the trellises for the rows of the generator matrix. At any time index i, the zero-state of the product trellis is forward proper iff the sets of labels emanating from zero-states in the component trellises at time index i intersect trivially. This happens iff at most one row of the generator matrix starts at i. Similarly for backward properness. ■

Using two stages of Gaussian elimination, any generator matrix for $\mathcal{C}$ can be converted into a two-way proper generator matrix. The first stage gets the matrix in the usual row echelon form, i.e., row $i+1$ starts at a later point than row i, for $1 \leq i \leq n-1$. Then, by a process of "cancelling upwards", we can ensure that no two rows end at the same point; this process does not affect the starting points.

Example 2 Consider the following generator matrix over $GF(2)$:

$$\begin{pmatrix} 1100 \\ 1010 \end{pmatrix}$$

The trellises for the individual rows as well as the product trellis are shown in Figure 2(a). In this case, the trellis obtained is not two-way proper. However, we may convert the above generator matrix to a two-way proper matrix to obtain

$$\begin{pmatrix} 1100 \\ 0110 \end{pmatrix}$$

The trellis obtained from this matrix is shown in Figure 2(b).

5 Extending to rings Z_{p^α}: the difficulties encountered

A length n linear code, $\mathcal{C}$, over a ring Z_{p^α} is a submodule of the module $Z_{p^\alpha}^n$. Such a submodule can be specified via a generator matrix; linear combinations of the rows of the matrix give vectors of the submodule. So, the question arises whether the notion of a two-way proper generator matrix again helps in obtaining minimal trellises. The answer is, "No". Consider the code generated by the following matrix over Z_4:

$$\begin{pmatrix} 11 \\ 02 \end{pmatrix}$$

. The two rows start at different indices, but end at the same index. However, it is not possible to remedy this by upward Gaussian elimination. The reason is that whereas 1 is a unit, 2 is a zero-divisor in Z_4. In fact, as shown in Example 3, no generator matrix for this code is two-way proper.

Example 3 The following code over Z_4^2 has no two-way proper generator matrix. Moreover, each generator matrix yields a trellis that is not two-way proper, and hence non-minimal.

$$C = \{00, 11, 22, 33, 02, 13, 20, 31\}$$

Since this code has eight codewords, we need at least two rows in the generator matrix. Since the multiple of a zero-divisor cannot give a unit, one of the rows in the generating matrix must contain a unit and so must be drawn from $\{11, 33, 13, 31\}$. But then the other row will either have the same starting point or the same ending point as this row.

Unlike the field case, the trellis for the code generated by a single vector can have several forking states and several collapsing states. For example, the minimal trellis for the length 6 code generated by (241014) over Z_8 is shown in Figure 3.

We will need the following definitions: For $a \in Z_{p^\alpha}$ if the additive subgroup generated by a has p^k elements, then say that the *order* of a is k. For example, the order of 0 is zero, and the order of 1 is α. For $a, b \in Z_{p^\alpha}$, a and b will be said to be *associates* if there is a unit $u \in Z_{p^\alpha}$ such that $a = ub$. Notice that a and b are associates iff they have the same order. In the example, notice that as the orders of elements in the given vector first increase, we get forking states. Finally, as the orders decrease, we get collapsing states.

Let us point out some more general difficulties in working over modules, arising because of zero-divisors (and more generally, the fact that elements of the ring have different orders). There are two natural ways of defining *linear dependence* of a set of vectors V:

- a non-trivial linear combination of the vectors in V gives the zero vector
- one of the vectors in V can be expressed as a linear combination of the rest.

In the case of a vector space, these two definitions are equivalent. However, in the case of a module dependence in the first sense need not imply dependence in the second sense. For example, over Z_4, the vectors (12) and (10) are dependent by the first definition, but not the second.

Another difficulty is that we cannot give a suitable definition of dimension of a submodule. For example, over Z_4, (20) and (02) form a basis for the submodule they generate. On the other hand (10) and (01) form a basis for a submodule that strictly contains the first submodule. Consequently, defining the dimension of a submodule as the cardinality of its basis is not very meaningful. Also notice that the vectors of the first submodule are not uniquely generated by linear combinations of the basis vectors.

6 p-linear combinations and p-generator sequences

In this section, we will introduce the notions of p-linear combinations and p-generator sequences which enable us to get around the difficulties mentioned in the previous section in working over submodules of Z_{p^α}. We will show that p-linear combinations of p-generator sequences enjoy properties similar to those of a basis for a vector subspace: they uniquely generate the elements of the submodule, a suitable definition of dimension of a submodule can be given, and the two notions of linear dependence turn out to be equivalent.

Let $V = \{\vec{v_1}, \cdots, \vec{v_k}\}$ be a set of vectors over Z_{p^α}. We will say that $\sum_{i=1}^{k} a_i \vec{v_i}$ is a *p-linear combination* of these vectors if all coefficients $a_i \in \{0, 1, \cdots (p-1)\}$. Notice that the elements $1, \cdots, (p-1)$ are all units in Z_{p^α}. We will denote by *p-span*(V) the set of all vectors generated as p-linear combinations of vectors in V, and by *span*(V) the set of vectors generated as (ordinary) linear combinations of vectors in V. We will say that a given linear combination (p-linear combination) *uses* vector $\vec{v_i}$ if its coefficient is non-zero in the linear combination (p-linear combination).

An ordered sequence of vectors $V = (\vec{v_1}, \cdots, \vec{v_k})$ over Z_{p^α} is said to be a *p-generator sequence* if for $1 \leq i \leq k, p\vec{v_i}$ is a p-linear combination of the vectors $\vec{v}_{i+1}, \cdots, \vec{v_k}$ (in particular, $p\vec{v_k}$ is the zero vector). For each vector $\vec{v_i}$ one such p-linear combination is designated the *canonical p-linear combination for* $\vec{v_i}$. If $i < j$, for convenience, we will say that $\vec{v_i}$ *is earlier than* $\vec{v_j}$ and that $\vec{v_j}$ *is later than* $\vec{v_i}$.

For an arbitrary set of vectors V, *p-span*(V) may not be a submodule, for example, if $p\vec{v_i}$ is not a p-linear combination of the vectors in V, for some $\vec{v_i} \in V$. However, ensuring this condition is not sufficient as shown in Example 4. On the other hand, this condition together with the order among the vectors, as stated in the definition of a p-generator sequence, turns out to be sufficient; this is established in Theorem 6.2.

Example 4 Consider the following set of vectors over Z_9.

$$\vec{v_1} = (3106), \quad \vec{v_2} = (2270), \quad \vec{v_3} = (8510), \quad \vec{v_4} = (3533)$$

Here,

$$3\vec{v_1} = \vec{v_2} + 2\vec{v_3}, \quad 3\vec{v_2} = \vec{v_1} + \vec{v_4}$$

$$3\vec{v}_3 = \vec{v}_1 + \vec{v}_4, 3\vec{v}_4 = \vec{v}_1 + 2\vec{v}_2 + \vec{v}_3 + \vec{v}_4$$

The *p-span* of these vectors is not a submodule, since it does not contain the vector $5\vec{v}_4 = (6766)$. The reason is that we cannot order the vectors so they satisfy the definition of a p-generator sequence.

Example 5 The following set of vectors over Z_9 form a p-generator sequence:

$$\vec{v}_1 = (0101), \vec{v}_2 = (2500), \vec{v}_3 = (5203), \vec{v}_4 = (3300)$$

$$3\vec{v}_1 = 2\vec{v}_2 + \vec{v}_3, \quad 3\vec{v}_2 = 2\vec{v}_4, \quad 3\vec{v}_3 = 2\vec{v}_4, \quad 3\vec{v}_4 = 0$$

Let us see how to obtain a p-linear combination equivalent to $\vec{u} = 7\vec{v}_1 + 4\vec{v}_2 + \vec{v}_3 + 2\vec{v}_4 = (1801)$:

$$\begin{aligned}
\vec{u} &= 7\vec{v}_1 + 4\vec{v}_2 + \vec{v}_3 + 2\vec{v}_4 \\
&= \vec{v}_1 + 2(3\vec{v}_1) + 4\vec{v}_2 + \vec{v}_3 + 2\vec{v}_4 \\
&= \vec{v}_1 + 2(2\vec{v}_2 + \vec{v}_3) + 4\vec{v}_2 + \vec{v}_3 + 2\vec{v}_4 \\
&= \vec{v}_1 + 8\vec{v}_2 + 3\vec{v}_3 + 2\vec{v}_4 \\
&= \vec{v}_1 + 2\vec{v}_2 + 2(2\vec{v}_4) + 3\vec{v}_3 + 2\vec{v}_4 \\
&= \vec{v}_1 + 2\vec{v}_2 + 2\vec{v}_4 + 6\vec{v}_4 \\
&= \vec{v}_1 + 2\vec{v}_2 + 2\vec{v}_4
\end{aligned}$$

Remark: Let us give an intuitive justification for the definition of p-generator sequences. The definition is motivated by computational considerations. If the vectors can be ordered as required in the definition, computations with them proceed in an orderly fashion along the ordering; this is proven rigorously in Theorem 6.2. Otherwise, computations get "entangled" in loops. In fact, we conjecture that if the vectors in V cannot be ordered, then either p-linear combinations of V do not generate a submodule, or the two notions of dependence do not turn out to be equivalent (see Theorem 6.3); we expect the proof of this to be quite involved. Examples 4 and 5 illustrate this.

Lemma 6.1 *Let V be a p-generator sequence, with $|V| = k$. Let $\vec{v} = \sum_{i=1}^{k} a_i\vec{v}_i$ be any linear combination of vectors in V, and let $\vec{v}_l$ be the earliest vector used in this linear combination. Then, $\vec{v}$ can be expressed as a p-linear combination of $\vec{v}_l$ and later vectors of V.*

Proof: The coefficients occurring in any linear or p-linear combination can be written as a k dimensional vector. Let $(b_1, \cdots, b_k)$ and $(c_1, \cdots, c_k)$ be two such vectors, and let b_i and c_j be their first non-zero coefficient. We will say that $(b_1, \cdots, b_k)$ is lexicographically larger than $(c_1, \cdots, c_k)$ if either $i < j$, or $i = j$ and $b_i > c_i$.

Now consider the coefficient vector $(a_1, \cdots, a_k)$. If all coefficients are in the range $\{0, 1, \cdots, p-1\}$, then we are done. Otherwise, let a_j be the first coefficient that is $\geq p$. Let $a_j = ap + b$. Write $ap\vec{v}_j$ using the canonical p-linear combination for $\vec{v}_j$. This uses vectors occurring later than $\vec{v}_j$. Substituting, we will get a vector equivalent to $(a_1, \cdots, a_k)$, which is the same in the first $j-1$ places, and has b in the j^{th} place. So, this vector is lexicographically smaller than $(a_1, \cdots, a_k)$. Now, this process can be continued until we get an equivalent p-linear combination. Clearly, the process terminates, and the final vector will have zero coefficients in the first $l-1$ places. ■

Theorem 6.2 *If V is a p-generator sequence then* p-span(V) = span(V).

Proof: Clearly, *p-span*$(V) \subseteq$ *span*(V). Since by Lemma 6.1 every vector in *span*(V) can also be expressed as a p-linear combination of vectors in V, the other direction also follows. ■

Theorem 6.3 *Let V be a p-generator sequence. W.r.t. p-linear combinations over V, the two notions of linear dependence are equivalent, i.e., there is a non-trivial p-linear combination of vectors from V that is 0 iff there is a vector in V that can be expressed as a p-linear combination of the remaining vectors in V.*

Proof: First suppose that there is a non-trivial p-linear combination $\sum_{i=1}^{k} a_i \vec{v_i} = 0$. Let $\vec{v_l}$ be the earliest vector used by this p-linear combination. Now we have

$$a_l \vec{v_l} = -(\sum_{i=l+1}^{k} a_i \vec{v_i}) = \sum_{i=l+1}^{k} (p^\alpha - 1) a_i \vec{v_i}.$$

Since a_l is a unit in Z_{p^α}, we get

$$\vec{v_l} = (a_l)^{-1} \sum_{i=l+1}^{k} (p^\alpha - 1) a_i \vec{v_i}.$$

Now, by Lemma 6.1, the linear combination on the r.h.s. can be expressed as a p-linear combination which does not use $\vec{v_l}$. So, $\vec{v_l}$ has been expressed as a p-linear combination of the remaining vectors.

To prove the other direction, suppose $\vec{v_l} = \sum_{i \neq l} a_i \vec{v_i}$. Let $\vec{v_j}$ be the earliest vector used in the r.h.s. There are two cases:

Case 1: $l < j$. By Lemma 6.1, $-(\sum_{i \neq l} a_i \vec{v_i})$ can be expressed as a p-linear combination using vectors later than $\vec{v_l}$ only. Hence,

$$\vec{v_l} - (\sum_{i \neq l} a_i \vec{v_i})$$

can be written as a non-trivial p-linear combination that is 0.

Case 2: $l > j$. By Lemma 6.1, $-\vec{v_l}$ can be expressed as a p-linear combination that does not use $\vec{v_j}$. Hence,

$$\sum_{i \neq l} a_i \vec{v_i} - \vec{v_l}$$

can be written as a non-trivial p-linear combination that is 0.

■

We will say that a p-generator sequence V is *p-linearly independent* if there is no non-trivial p-linear combination of its vectors that is 0. A p-linearly independent p-generator sequence will be called a *p-basis*. Clearly, the p-linear combinations of the elements of a p-basis V uniquely generate the elements of the submodule *p-span*(V). So, if $|V| = k$, the submodule has p^k elements. We will define the *p-dimension* of this submodule to be k.

Remark: Note that the notions of p-dimension and p-generator sequence of a submodule over Z_{p^α} correspond exactly to the notions of *composition length* and *generating system along a composition chain* of a module in commutative ring theory (see [19]). The reason for our choice of terminology is that it is more suggestive of properties of vector subspaces that we are attempting to endow submodules over Z_{p^α} with.

Lemma 6.4 *Every submodule over Z_{p^α} has a p-generator sequence.*

Proof: Let $U = \{\vec{v_1}, \cdots, \vec{v_k}\}$ be a usual generating set for the submodule. Let V be the ordered sequence consisting of multiples of these vectors by $p^i, 0 \leq i \leq \alpha - 1$, i.e.,

$$V = (\vec{v_1}, p\vec{v_1}, \cdots, p^{\alpha-1}\vec{v_1}, \ \cdots \ , \vec{v_k}, p\vec{v_k}, \cdots, p^{\alpha-1}\vec{v_k}).$$

Then, clearly V is a p-generator sequence with p-*span*(V) = *span*(U). Notice that if any of the vectors in V is 0, it can be dropped. ■

Our next goal is to show that every submodule over Z_{p^α} has a p-basis; we will accomplish this by adapting Gaussian elimination to this setting. Let us first recall the process of Gaussian elimination when performed on vectors from a vector space. Let $V = \{\vec{v_1}, \cdots, \vec{v_k}\}$ be the generator set for a subspace of F^n, the n-dimensional vector space over field F. The process of Gaussian elimination is based on the following fact: Let $\vec{v} = \sum_{i=1}^{k} a_i\vec{v_i}$ be a linear combination of the vectors in V. Then, for any vector $\vec{v_i}$ that is used by this linear combination, $V + \vec{v} - \vec{v_i}$ generates the same subspace as V. Using this principle, Gaussian elimination starts with an arbitrary generator set for a subspace, and brings it into "row echelon" form, i.e., all non-zero vectors have distinct starting points, and are sorted by starting point, with the 0 vectors being listed last. Now, the non-zero vectors are linearly independent, and form a basis for the subspace. Carrying out this process is somewhat more involved for p-generator sequences.

Lemma 6.5 *Let $V = (\vec{v_1}, \cdots, \vec{v_k})$ be a p-generator sequence, and let $\vec{v} = \sum_{i=1}^{k} a_i\vec{v_i}$ be a p-linear combination of its vectors. Let $\vec{v_l}$ be the earliest vector in this ordering that is used by the p-linear combination, and U be obtained by replacing $\vec{v_l}$ by $\vec{v}$ in the ordered set V. Then, U is also a p-generator sequence with the same span as V.*

Proof: Since a_l is a unit, we can write

$$\vec{v_l} = (a_l)^{-1}(\vec{v} - \sum_{i \neq l} a_i\vec{v_i})$$

Therefore, corresponding to any linear combination of the vectors of V there is an equivalent linear combination of the vectors of U and vice versa. Hence, U has the same span as V.

Next we show that U is a p-generator sequence, i.e., for each vector $\vec{v_j} \in U$, $p\vec{v_j}$ can be expressed as a p-linear combination of vectors later than $\vec{v_j}$ in U. This is clearly true for $j > l$. For vector $\vec{v}$,

$$p\vec{v} = p\vec{v_l} + \sum_{i \neq l} pa_i\vec{v_i}.$$

Using Lemma 6.1, and the fact that there is a p-linear combination for $p\vec{v_l}$ using vectors later than l, the r.h.s. can be expressed as a p-linear combination of vectors later than l. Finally, consider

$j < l$. If the canonical p-linear combination for $p\vec{v_j}$ in V does not use $\vec{v_l}$, we will simply use this same p-linear combination. Otherwise, we will substitute for $\vec{v_l}$ using the first equation given above, and use Lemma 6.1 to obtain a p-linear combination that uses vectors of U later than $\vec{v_j}$. ■

Corollary 6.6 *Let $V = (\vec{v_1}, \cdots, \vec{v_k})$ be a p-generator sequence. Let $\vec{v} = \vec{v_i} + a\vec{v_j}$, where $i < j$ and $a \in Z_{p^\alpha}$. Then, replacing $\vec{v_i}$ by $\vec{v}$ in V gives an equivalent p-generator sequence.*

Proof: The proof follows by observing that $a\vec{v_j}$ can be written as a p-linear combination of $\vec{v_j}$ and later vectors of V. ■

Say that a p-generator sequence, V, is *proper* if for each pair of non-zero vectors, $\vec{u}, \vec{v} \in V$, if $\vec{u}$ and $\vec{v}$ have the same starting point, then their starting elements are not associates.

Lemma 6.7 *Every submodule of $Z_{p^\alpha}^n$ has a proper p-generator sequence.*

Proof: Let V be a p-generator sequence that is not proper. Say that vectors $\vec{u}$ and $\vec{v}$ are *in conflict* if they have the same starting point, and their starting elements are associates. Among all conflicting pairs having the earliest starting point, pick a pair whose starting elements have the highest order. Let $\vec{v_i}$ and $\vec{v_j}$ be this pair, with $i > j$. Now, we can find $a \in Z_{p^\alpha}$ such that adding $a\vec{v_j}$ to $\vec{v_i}$ zeros out the starting element of $\vec{v_i}$, i.e., $\vec{v} = \vec{v_i} + a\vec{v_j}$ has a later starting point than $\vec{v_i}$. By Corollary 6.6, replacing $\vec{v_i}$ by $\vec{v}$ in V gives an equivalent p-generator sequence. Clearly, this process must terminate, and yield a proper p-generator sequence. ■

We will say that a proper p-generator sequence $V = (\vec{v_1}, \cdots, \vec{v_k})$ is in *row echelon form* if for $1 \leq i < j \leq k$ either:

1. $\vec{v_i}$ has an earlier starting point than $\vec{v_j}$, or
2. $\vec{v_i}$ and $\vec{v_j}$ have the same starting point, and the starting element of $\vec{v_i}$ has higher order than the starting element of $\vec{v_j}$.

Lemma 6.8 *Let $V = \{\vec{v_1}, \cdots, \vec{v_k}\}$ be a p-generator sequence in row echelon form. If $\vec{v_i}$ is non-zero, then a p-linear combination for $p\vec{v_i}$ cannot use any vector $\vec{v_j}$ with $j < i$.*

Proof: Suppose not, and let $\vec{v_l}$ be the earliest vector used. V has at most one vector with a given starting point and order of starting element. Therefore, the remaining vectors used in the p-linear combination cannot zero out the starting element of $\vec{v_l}$. So, this p-linear combination will either start before $\vec{v_i}$, or will start at the same point as $\vec{v_i}$ but with an element of higher order than the starting element of $\vec{v_i}$. In either case we get a contradiction. ■

Corollary 6.9 *Let V be a proper p-generator sequence. Then, permuting its vectors so they are in row echelon form gives an equivalent p-generator sequence.*

Lemma 6.10 *The non-zero vectors of a p-generator sequence in row echelon form are p-linearly independent.*

Proof: The proof is along the same lines as Lemma 6.8. Consider any non-trivial p-linear combination of the vectors. Then, the starting element of the earliest vector used cannot by cancelled out by the remaining vectors. Hence, no non-trivial p-linear combination of the vectors can be 0. ■

Theorem 6.11 *Every submodule of $Z_{p^\alpha}^n$ has a p-basis.*

Finally, we give below the Gaussian elimination procedure that starts with an arbitrary p-generator sequence for a submodule of $Z_{p^\alpha}^n$, and finds a p-generator sequence in row echelon form. This procedure is designed along the lines of the usual Gaussian elimination procedure for obtaining a basis in row echelon form for the field case; it simultaneously carries out the process in Lemma 6.7, together with the permutation of vectors given in Corollary 6.9.

ALGORITHM GAUSSIAN ELIMINATION

1). $S \longleftarrow V$.
2). While there is a non-zero vector in S do:
3). Find $S' \subseteq S$, vectors of S having the earliest starting point.
4). Find $S'' \subseteq S'$, vectors of S' having the highest order starting element.
5). Pick the last vector $\vec{v} \in S''$, list it, and set $S \longleftarrow S - \{\vec{v}\}$.
6). For each remaining $\vec{u} \in S''$, replace $\vec{u}$ in S by $(\vec{u} + a\vec{v})$,
where $a \in Z_{p^\alpha}$ such that $(\vec{u} + a\vec{v})$ starts later than $\vec{u}$.
7). end.

Theorem 6.12 *The process of Gaussian elimination starts with an arbitrary p-generator sequence for a submodule, and finds a proper p-generator sequence in row echelon form. Its running time is bounded by $O(k^2 n)$ operations over Z_{p^α}, where k is the number of vectors in the p-generator sequence.*

Example 6 Consider the code over Z_8 generated by:

$$\begin{pmatrix} 1212 \\ 2042 \\ 0044 \end{pmatrix}$$

A p-generator sequence for it is is given below. Since there are two rows starting with a 2 in column one, we add 3 times row 3 to row 2. Next, we add row 6 to row 5. Finally, discarding duplicate rows we get a p-basis.

$$\begin{pmatrix} 1212 \\ 2042 \\ 2424 \\ 0044 \\ 4004 \\ 4040 \end{pmatrix} \longrightarrow \begin{pmatrix} 1212 \\ 0426 \\ 2424 \\ 0044 \\ 4004 \\ 4040 \end{pmatrix} \longrightarrow \begin{pmatrix} 1212 \\ 0426 \\ 2424 \\ 0044 \\ 0044 \\ 4040 \end{pmatrix} \longrightarrow \begin{pmatrix} 1212 \\ 0426 \\ 2424 \\ 0044 \\ 4040 \end{pmatrix}$$

7 Minimal trellises for codes over rings Z_{p^α}

In this section we will present a polynomial time algorithm for constructing a minimal trellis for a linear code over a ring Z_{p^α}, given a generator matrix for it. Let us first give a natural generalization of the notion of a two-way proper matrix as defined for the field case.

A p-generator sequence, V will be said to be *two-way proper* if:

1. for each pair of vectors $\vec{u}, \vec{v} \in V$, if $\vec{u}$ and $\vec{v}$ start at the same point, then their starting elements are not associates, and
2. for each pair of vectors $\vec{u}, \vec{v} \in V$, if $\vec{u}$ and $\vec{v}$ end at the same point, then their ending elements are not associates.

Below we give an algorithm that starts with a proper p-generator sequence in row echelon form, V, and finds a two-way proper p-generator sequence having the same span.

ALGORITHM TWO-WAY PROPER P-GENERATING SET

1). $S \longleftarrow V$.
2). While S is not two-way proper do:
3). Find $S' \subseteq S$, with $|S'| > 1$, vectors having the latest ending point,
and moreover their ending elements being associates.
4). Let $\vec{v}$ be the last vector in S'.
5). For each remaining $\vec{u} \in S'$, replace $\vec{u}$ in S by $(\vec{u} + a\vec{v})$,
where $a \in Z_{p^\alpha}$ such that $(\vec{u} + a\vec{v})$ ends earlier than $\vec{u}$.
6). end.

Lemma 7.1 ALGORITHM TWO-WAY PROPER P-GENERATING SET *starts with a proper p-generator sequence in row echelon form, and finds a two-way proper p-generator sequence with the same span. Its running time is bounded by $O(k^2 n)$ operations over Z_{p^α}, where k is the number of vectors in the p-generator sequence.*

Example 7 The p-basis obtained in Example 6 is not two-way proper. We add row 2 to row 1 and row 4 to row 3 to obtain the final two-way proper p-basis:

$$\begin{pmatrix} 1630 \\ 0426 \\ 2460 \\ 0044 \\ 4040 \end{pmatrix}$$

Using two-way proper p-generator sequences, our trellis construction algorithm has the same overall structure as the field case. The trellis for a single vector $\vec{v}$ of the two-way proper p-generator

sequence, V, is required to generate all codewords that can be generated as p-linear combinations of this vector, i.e., $\{0, \vec{v}, 2\vec{v}, \cdots, (p-1)\vec{v}\}$. This trellis is similar to the trellis for a single vector in the field case: it has a p-way fork at the time index at which $\vec{v}$ starts, and a p-way collapse at the time index at which $\vec{v}$ ends. The p out-transitions will be labelled with 0 and associates of the starting element of $\vec{v}$, and the p in-transitions will be labelled with 0 and associates of the ending element of $\vec{v}$. We will next show that the product of the trellises for the vectors of V is two-way proper and hence minimal.

Let $a_1, \cdots a_k \in Z_{p^\alpha}$. For notational convenience, it will be useful to regard these elements as one dimensional vectors belonging to the module $Z^1_{p^\alpha}$. We can then talk about all p-linear combinations of these elements.

Lemma 7.2 *Let $a_1, \cdots a_k \in Z_{p^\alpha}$, so that they are pairwise non-associates. Then, their p-linear combinations give distinct elements of Z_{p^α}.*

Proof: Notice that if we start with any α elements of Z_{p^α} that are pairwise non-associates, they will form a p-basis for the one dimensional module $Z^1_{p^\alpha}$. Hence, in particular the p-linear combinations of $a_1, \cdots a_k$ will generate distinct elements in $Z^1_{p^\alpha}$. ■

It is easy to see that the converse of the statement in Lemma 7.2 is not true, i.e., one can get an example in which even though $a_1, \cdots a_k$ are not pairwise non-associates, their p-linear combinations may still generate distinct elements of Z_{p^α}. Yet the following holds:

Theorem 7.3 *Let V be a p-basis for a submodule of $Z^n_{p^\alpha}$. Then, the product of trellises for the vectors of V is a minimal trellis for the submodule generated by V iff V is two-way proper.*

Proof: Suppose V is two-way proper. Let us show that the product trellis will be forward proper; the proof that it is backward proper is similar. If V has k_i vectors, say V_i, that start at time index i, then any state, s, in the product trellis at time index $i-1$ will have p^{k_i} out-transitions. Since the starting elements of the vectors in V_i are non-associates, by Lemma 7.2, the p-linear combinations of these vectors will all start with distinct elements. The set of symbols on the out-transitions of s consist of some element $a \in Z_{p^\alpha}$ added to these distinct elements, and so s is forward proper.

Next, suppose V is not two-way proper. Suppose there are two vectors $\vec{u}$ and $\vec{v}$ starting at index i, so that their starting elements are associates; the proof in case V has two vectors whose ending points are the same, but ending elements are associates is similar.

Now, there are units $b, c \in Z_{p^\alpha}$ such that $b\vec{u} + c\vec{v}$ starts at a later index than i. Let V_i be the set of vectors of V that start at index i. Using the fact that V is a p-generator sequence it can be argued that $b\vec{u} + c\vec{v}$ can be written as a p-linear combination of vectors in V_i; clearly, this p-linear combination is non-trivial. In addition, the trivial p-linear combination of the vectors in V_i also gives a vector that is 0 at time index i.

Finally, let s and a be as defined above. Now, there are two p-linear combinations of vectors in V_i that give out-transitions on symbol a from state s. Therefore, the product trellis is not two-way proper, and hence it is not minimal.

■

Example 8 Consider the code over Z_4 generated by the following two-way proper p-generator sequence:

$$\begin{pmatrix} 1230 \\ 2020 \\ 0222 \end{pmatrix}$$

The trellises for the individual rows as well as the product trellis are shown in Figure 4.

8 Minimal trellises for codes over Abelian groups

We will first extend our construction algorithm to codes over elementary Abelian groups; this will illustrate in a simpler setting the main ideas in the extension to codes over arbitrary finite Abelian groups. An elementary Abelian group G is isomorphic to a direct product of cyclic p-groups, i.e., $G \simeq C_p^m$, where p is a prime.

Lemma 8.1 *A length n group code $\mathcal{C}$ over $G \simeq C_p^m$ can be viewed as a linear code, S, of length mn over $GF(p)$.*

Proof: Using the natural isomorphism between C_p and the additive group of $GF(p)$, we can view $\mathcal{C}$ as a length mn code over $GF(p)$, say S. Since $\mathcal{C}$ is a group code, so is S. Further, since multiplication in $GF(p)$ is simply repeated addition, S is a linear code over $GF(p)$. ■

Biglieri and Elia [1] have shown that $\mathcal{C}$ can be specified by a $k \times n$ generator matrix Ψ whose entries are endomorphisms, $\psi_{i,j} : C_p^m \to C_p^m$. Any information vector $\vec{v} \in (C_p^m)^k$ yields the code word $\vec{v}\Psi$, and the set of all code words constructed in this manner constitute $\mathcal{C}$.

Lemma 8.2 *Given a generator matrix Ψ for $\mathcal{C}$, we can obtain a $km \times mn$ generator matrix over $GF(p)$, A, for S.*

Proof: As shown in [1], an endomorphism $\psi : C_p^m \to C_p^m$ can be viewed as an $m \times m$ matrix, M_ψ, over C_p. View an element $a \in C_p^m$ as an m-dimensional row vector $\vec{a}$ with entries from C_p. Then, $\vec{a}M_\psi = \psi(a)$.

Thus, Ψ can now be viewed as a $k \times n$ matrix whose elements are $m \times m$ matrices over C_p. Intuitively, the $km \times mn$ matrix A is obtained by simply "removing the demarkations" of the element matrices of Ψ. Formally, for $1 \leq i \leq km$ and $1 \leq j \leq mn$, divide i and j by m to obtain quotients and remainders q_i, q_j and r_i, r_j respectively. Now, let the $(i,j)^{th}$ entry of A be the $(r_i, r_j)^{th}$ entry of the matrix corresponding to ψ_{q_i,q_j}, i.e.,

$$A[i,j] = M_{\psi_{q_i,q_j}}[r_i, r_j].$$

Clearly, A is the generator matrix for code S. ■

The algorithm for obtaining a minimal trellis for $\mathcal{C}$ is as follows: First, obtain a minimal trellis T for the linear code S. This trellis will have length mn. Next, *sectionalize* this trellis by collapsing m successive layers into one layer to obtain trellis T': T' has length n, and the set of states in layer i

in T' is the same as the set of states in layer ki in T. States u and v in successive layers of T' have a transition iff there is a path from u to v in T. If so, each such path gives a symbol from group G (of course, if there are multiple labels on transitions of T, we will get multiple symbols from the same path); these symbols constitute the set of labels on this transition. It is easy to see that if T is two-way proper, then so is T' (notice that T' may be two-way proper even though T is not).

Lemma 8.3 *T' is a minimal trellis for $\mathcal{C}$.*

Example 9 For the elementary Abelian group with four elements, $C_2 \times C_2 = \{1, x\} \times \{1, y\}$, consider the following length 4 group code. This is an MDS code, and it cannot be seen as a linear code over $GF(4)$.

$$\begin{array}{cccc|cccc|cccc|cccc}
(1 & 1 & 1 & 1) & (1 & x & xy & y) & (1 & y & y & x) & (1 & xy & x & xy) \\
(x & 1 & xy & xy) & (x & x & 1 & x) & (x & y & x & y) & (x & xy & y & 1) \\
(y & 1 & y & y) & (y & x & x & 1) & (y & y & 1 & xy) & (y & xy & xy & x) \\
(xy & 1 & x & x) & (xy & x & y & xy) & (xy & y & xy & 1) & (xy & xy & 1 & y)
\end{array}$$

Under the map

$$1 \to (00), x \to (01), y \to (10), xy \to (11)$$

this code is same as a length 8 linear code over $GF(2)$. A generator matrix for the original group code is

$$\begin{pmatrix}
\begin{pmatrix} 00 \\ 01 \end{pmatrix} & \begin{pmatrix} 01 \\ 00 \end{pmatrix} & \begin{pmatrix} 11 \\ 11 \end{pmatrix} & \begin{pmatrix} 10 \\ 11 \end{pmatrix} \\
\begin{pmatrix} 01 \\ 11 \end{pmatrix} & \begin{pmatrix} 10 \\ 01 \end{pmatrix} & \begin{pmatrix} 01 \\ 10 \end{pmatrix} & \begin{pmatrix} 10 \\ 11 \end{pmatrix}
\end{pmatrix}$$

and a generator matrix for the corresponding code over $GF(2)$ is given below from which we derive the following two-way proper generating matrix:

$$\begin{pmatrix} 00011110 \\ 01001111 \\ 01100110 \\ 11011011 \end{pmatrix} \longrightarrow \begin{pmatrix} 11101100 \\ 01111000 \\ 00101001 \\ 00011110 \end{pmatrix}$$

Applying the trellis contruction procedures for linear codes we obtain the trellis given in Figure 5(a). Sectionalizing this trellis, and applying the reverse map from $GF(2)^2$ to $C_2 \times C_2$ we obtain, in Figure 5(b), the minimal trellis for the original code.

Finally, let us consider the case when G is an arbitrary finite Abelian group. Then, G is isomorphic to the direct product of cyclic groups, i.e.,

$$G \simeq (C_{p_1^{\alpha_{11}}} \bigotimes \cdots \bigotimes C_{p_1^{\alpha_{1m_1}}}) \bigotimes \cdots \bigotimes (C_{p_l^{\alpha_{l1}}} \bigotimes \cdots \bigotimes C_{p_l^{\alpha_{lm_l}}})$$

where $p_1, \cdots, p_l$ are distinct primes.

Lemma 8.4 *Let $\mathcal{C}$ be a length n group code over Abelian group G. Let $G \simeq G_1 \bigotimes G_2$, where the orders of G_1 and G_2 are relatively prime. Then, there are length n group codes $\mathcal{C}_1$ and $\mathcal{C}_2$ over G_1 and G_2 respectively such that $\mathcal{C} \simeq \mathcal{C}_1 \bigotimes \mathcal{C}_2$.*

As a consequence of Lemma 8.4, we can decompose $\mathcal{C}$ into l codes $\mathcal{C}_1, \cdots, \mathcal{C}_l$. Obtain minimal trellises for these l codes; however, view labels in these trellises as if they were elements of G, by using the natural injection maps. Now, using Theorem 4.1, it is easy to see that the product of these trellises will be a minimal trellis for $\mathcal{C}$. Hence, it is sufficient to consider the case

$$G \simeq (C_{p^{\alpha_1}} \bigotimes \cdots \bigotimes C_{p^{\alpha_m}})$$

where $\alpha_1 \leq \cdots \leq \alpha_m$. The proofs of the next two lemmas are identical to those of Lemma 8.1 and 8.2.

Lemma 8.5 *A length n group code $\mathcal{C}$ over $(C_{p^{\alpha_1}} \bigotimes \cdots \bigotimes C_{p^{\alpha_m}})$, where $\alpha_1 \leq \cdots \leq \alpha_m$, is same as a linear code, A, of length mn over Z_{p^α}, where $\alpha = \alpha_m$.*

Once again, by the result of Biglieri and Elia [1], $\mathcal{C}$ can be specified by a $k \times n$ generator matrix Ψ whose entries are endomorphisms, $\psi_{i,j} : G \to G$.

Lemma 8.6 *Given a generator matrix Ψ for $\mathcal{C}$, we can obtain a $km \times mn$ generator matrix over Z_{p^α}, A, for S.*

Now, the structure of the algorithm is similar to the elementary Abelian case. We can obtain a generator matrix for the code over Z_{p^α} from that for $\mathcal{C}$, and construct a minimal trellis for it. Finally, sectionalizing this trellis will give a minimal trellis for $\mathcal{C}$. Hence, we get:

Theorem 8.7 *There is an $O(k^2 n + s)$ time algorithm that given a generator matrix for a group code over an Abelian group, constructs a minimal trellis for the code. In addition, this algorithm computes s in $O(k^2 n)$ time.*

We illustrate the core ideas in the extension to the case of general finite Abelian groups, below:

Example 10 Consider the length 3 group code over $C_2 \times C_4$ given by the following generator matrix:

$$\begin{pmatrix} \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 2 \\ 1 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix}$$

Observe that here the entries of the first column of every endomorphism matrix are over Z_2 whereas the second column is over Z_4. Since Z_2 can be embedded into Z_4 by the map $i \longrightarrow 2i, i \in Z_2$, we can obtain an equivalent generator matrix over Z_4 by using this map on the first column of each

endomorphism matrix. Now, as in the previous example, we can view this code as as a length 6 code over Z_4. Applying our algorithm we obtain the following two-way proper generating matrix:

$$\begin{pmatrix} 000101 \\ 012100 \\ 222000 \\ 000022 \\ 000220 \\ 020200 \end{pmatrix}$$

Applying the trellis contruction procedures for linear codes we obtain the trellis given in Figure 6(a). Sectionalizing this trellis, and applying the reverse map we obtain, in Figure 6(b), the minimal trellis for the original code.

9 Computing local descriptions of minimal trellises

We will present efficient algorithms for the following two problems:

Problem I: Given states s and t at time indices i and $i+1$, determine if there is a transition from s to t, and if so, the set of labels on this transition.

Problem II: Given state s at time index i, compute all states at time index $i+1$ that s has transitions to, and the sets of labels on these transitions. The problem of computing all states at time index $i-1$ that have transitions in to s is analogous.

Kschischang and Sorokine [15] have given algorithms for these problems for linear codes over fields. Using the notion of p-linearity developed in Sections 6 and 7, these algorithms extend to linear codes over rings Z_{p^α}. In turn, using the concepts developed in Section 8, these algorithms extend to codes over Abelian groups. Rather than directly presenting the algorithms for codes over Abelian groups, we show below the natural progression of ideas; this will help state the algorithms more clearly. For the field case, we have modified the algorithms of Kschischang and Sorokine, so they start with a two-way proper generator matrix, rather than an arbitrary generator matrix.

9.1 The field case

Let A be a $k \times n$ two-way proper generator matrix for a linear code, $\mathcal{C}$, over $GF(q)$. For each $i, 1 \leq i < n$, compute a_i, b_i and c_i as follows:

- a_i is the set of rows of A that are zero in columns $i+1$ to n. Linear combinations of these rows of A generate codewords in $\mathcal{C}_{i-}$.
- b_i is the set of rows of A that are zero in columns 1 to i. Linear combinations of these rows of A generate codewords in $\mathcal{C}_{i+}$.
- c_i is the remaining set of rows of A. Linear combinations of these rows of A generate coset representatives for $\mathcal{C}/(\mathcal{C}_{i-}\mathcal{C}_{i+})$, and are therefore in one-to-one correspondence with the set of states at time index i in the minimal trellis for $\mathcal{C}$. We will denote state s at time index

i by a k-dimensional information vector v_s that is zero in the components specified by a_i and b_i. Thus v_sA is the coset representative for state s, and $v_sA + (\mathcal{C}/(\mathcal{C}_{i-}\mathcal{C}_{i+}))$ is the set of codewords that it is responsible for.

A succinct representation of the minimal trellis for $\mathcal{C}$ consists of A together with a_i, b_i and c_i for each $i, 1 \leq i < n$. This information can be computed in $O(k^2n)$ time, and requires $O(kn)$ space. In the following, we will denote the i^{th} column of A by A_i.

The algorithm for Problem I is now straightforward: there is a transition from s to t iff the sets of codewords they are responsible for have non-empty intersection. This happens iff v_s and v_t are identical on the components specified by $c_i \cap c_{i+1}$. If so, the set of codewords that use this transition are:

$$\bigcup_u \{uA\},$$

where u ranges over all k-dimensional vectors that agree with v_s on positions specified by c_i and with v_t on positions specified by c_{i+1}. So, the set of labels on this transition are given by:

$$\bigcup_u \{uA_{i+1}\},$$

where u is as specified above. This expression can be simplified considerably. There are two cases: if $b_i \cap a_{i+1} = \emptyset$, then on the positions specified by $\overline{(c_i \cup c_{i+1})}$, A_{i+1} is zero. So, the label on the transition is given by uA_{i+1}, where from the vectors u given above, we have picked one that is zero on positions $\overline{(c_i \cup c_{i+1})}$. Otherwise, $b_i \cap a_{i+1}$ is a single row of A which is zero everywhere except in column $i+1$. In this case, the set of labels on the transition is all of $GF(q)$.

The algorithm is summarized below. It is easy to check that it runs in $O(k)$ time.

ALGORITHM LOCAL DESCRIPTION, PROBLEM I

1). If v_s and v_t are not identical on components specified by $c_i \cap c_{i+1}$,
then there is no transition from s to t.
2). If $b_i \cap a_{i+1} \neq \emptyset$,
then the transition from s to t is labelled with $GF(q)$;
3). else, the transition from s to t is labelled with uA_{i+1},
where u agrees with v_s on positions specified by c_i, with v_t on positions specified by c_{i+1}, and is zero on the remaining positions.
4). end.

Next, we give an algorithm for Problem II. If $c_{i+1} - c_i = \emptyset$, then there is only one transition out of state s. It goes to state t which is given by v_t, where v_t agrees with v_s on positions specified by c_{i+1} and is zero everywhere else. The set of labels on this transition can be computed as above. Otherwise, $c_{i+1} - c_i$ is a single row of A. In this case, there are q transitions out of s, in to states defined by the following vectors: for each $e \in GF(q)$, consider the vector that is e in the position specified by $c_{i+1} - c_i$, agrees with v_s on positions specified by $c_i \cap c_{i+1}$, and is zero elsewhere. The

label is computed as in Step 3 of the algorithm given above. This can be made more efficient by pre-computing $v_s A_{i+1}$, and adding to this ef, where f is the symbol in A at row $c_{i+1} - c_i$ and column $i+1$, for each $e \in GF(q)$. Clearly, this takes $O(k)$ time.

Example 11 Consider the code over Z_3 generated by the following matrix:

$$\begin{pmatrix} 100202 \\ 011000 \\ 001100 \\ 000110 \end{pmatrix}$$

Then,

$$a_3 = \{2\}, b_3 = \{4\}, c_3 = \{1, 3\},$$

and

$$a_4 = \{2, 3\}, b_4 = \emptyset, c_4 = \{1, 4\}.$$

Since $b_3 \cap a_4 = \emptyset$, there is only one label on transitions from time index 3 to time index 4. The label on the transition from state (1020) at time index 3 to state (1001) at time index 4 is given by multiplying (1021) with the fourth column of the matrix, giving 2. Since $c_4 - c_3 \neq \emptyset$, each state at time index 3 has transitions to 3 states at time index 4.

9.2 Extending to rings Z_{p^α}

The algorithm and proof are similar to the field case; the main difference being that "linear combination" is replaced by "p-linear combination". Let A be a two-way proper p-generator sequencefor code $\mathcal{C}$. As in the field case, for $1 \leq i < n$, we will compute a_i, b_i and c_i, the rows whose p-linear combinations generate $\mathcal{C}_{i-}$, $\mathcal{C}_{i+}$ and coset representatives for $\mathcal{C}/(\mathcal{C}_{i-}\mathcal{C}_{i+})$. Suppose A has k rows. Then, a state s at time index i will be represented by a k-dimensional information vector v_s that is 0 in positions specified by a_i and b_i. The components of v_s specified by c_i give the p-linear combinationof these rows of A such that $v_s A$ is a coset representative for the codewords that s is responsible for.

For Problem I, there is a transition from s to t iff v_s and v_t agree on the positions specified by $c_i \cap c_{i+1}$. Again, there are two cases: If $b_i \cap a_{i+1} = \emptyset$, then on the positions specified by $\overline{(c_i \cup c_{i+1})}$, A_{i+1} is zero. In this case, there will be one symbol on the transition, given by uA_{i+1}, where u agrees with v_s on positions specified by c_i and with v_t on positions specified by c_{i+1}, and is zero in the remaining positions. Otherwise, each row specified by $b_i \cap a_{i+1}$ has a non-zero entry only in the $(i+1)^{th}$ column. Let β be the order of the highest order element among these non-zero entries. Then, using ideas from Section 6, it is easy to show that $|b_i \cap a_{i+1}| = \beta$, and these β non-zero entries have distinct orders. In this case, the transition from s to t will have p^β symbols: add each p-linear combinationof the β non-zero entries to uA_{i+1}, where u is as defined above.

For Problem II, let $|c_{i+1} - c_i| = \beta$. Again, it is easy to show that $\beta \leq \alpha$. State s will have transitions to p^β states: construct v_t by using an arbitrary p-linear combination on the positions specified by $c_{i+1} - c_i$, letting v_t agree with v_s on positions specified by $c_i \cap c_{i+1}$, and be zero elsewhere. The set of labels is computed as for Problem I, and this can be made more efficient by doing certain precomputations as in the field case. Once the succinct representation is computed, the time for solving both problems is $O(k)$ (we are assuming that the ring, and hence α, is fixed).

9.3 Extending to Abelian groups

Let $\mathcal{C}$ be a length n code over a finite Abelian group G. As in Section 8, using the Chinese Remainder Theorem, one can show that it is sufficient to consider the case

$$G \simeq (C_{p^{\alpha_1}} \bigotimes \cdots \bigotimes C_{p^{\alpha_m}})$$

where $\alpha_1 \leq \cdots \leq \alpha_m$. Also, using Lemmas 8.5 and 8.6, the problem reduces to the case of rings Z_{p^α}. Let A be the $km \times mn$ p-generator sequenceover Z_{p^α}. The main differences are: w.r.t. A, s and t correspond to states at time indices im and $(i+1)m$. So, for example, s has a transition to t iff v_s and v_t agree on components specified by $c_{im} \cap c_{(i+1)m}$. Let B be the submatrix of A given by columns $im+1$ to $(i+1)m$. Then, the set of labels on this transition is given by:

$$\bigcup_u \{uD\},$$

where u ranges over all k-dimensional p-linear combinations that agree with v_s on positions specified by c_{im} and with v_t on positions specified by $c_{(i+1)m}$. This expression can be simplified as shown in the ring case. The number of symbols on this transition will be p^r, where $r = |b_{im} \cap a_{(i+1)m}|$. It is easy to show that $r \leq m\alpha$. Also, if $|c_{(i+1)m} - c_{im}| = r'$, then there are $p^{r'}$ transitions out of s, and again $r' \leq m\alpha$.

Theorem 9.1 *For any finite Abelian group G, there is an algorithm that pre-computes $O(kn)$ information, in $O(k^2 n)$ time, and solves Problem I and Problem II in $O(k)$ time.*

10 Transition Space Theorem

In this section, we will present the Transition Space Theorem. This theorem helps define a succinct representation for minimal trellises for group codes, using which local descriptions of the trellis can be efficiently computed. However, unlike the succinct representation given for group codes over Abelian groups in Section 9, in general this representation will not be of polynomial size, and so is less useful. Perhaps more importantly, the Transition Space Theorem gives algebraic structural properties of transitions in a minimal trellis for a group code. This theorem is derived as a corollary of the State Space Theorem of Forney and Trott, and can be viewed as a complementary theorem: the State Space Theorem characterizes states in a minimal trellis, and the Transition Space Theorem characterizes transitions.

Let Σ_k and σ_k denote the set of states and the state map at time index k, $\sigma_k : \mathcal{C} \to \Sigma_k$. The set of transitions from time index k to $k+1$, is called the *branch space* and is denoted by $\Sigma_{k,k+1}$ in [9]. Let $\sigma_k \times \sigma_{k+1}$ be the Cartesian product of homomorphisms σ_k and σ_{k+1},

$$\sigma_k \times \sigma_{k+1} : \mathcal{C} \to \Sigma_k \times \Sigma_{k+1}.$$

Then, $\Sigma_{k,k+1} = [\sigma_k \times \sigma_{k+1}](\mathcal{C})$. For conciseness, let us denote $\Sigma_{k,k+1}$ by $\mathcal{S}^{(k)}$.

We will view $\mathcal{S}^{(k)}$ as a length two group code. Then, at time index 1, the codeword pasts,

$$P_-(\mathcal{S}^{(k)}) = \Sigma_k \simeq \mathcal{C}/(\mathcal{C}_{k^-}\mathcal{C}_{k^+}),$$

and the codeword futures,

$$P_+(\mathcal{S}^{(k)}) = \Sigma_{k+1} \simeq \mathcal{C}/(\mathcal{C}_{(k+1)^-}\mathcal{C}_{(k+1)^+}).$$

Denote the past subcode at time index 1 by $\mathcal{S}_-^{(k)}$, and the future subcode by $\mathcal{S}_+^{(k)}$, i.e.,

$$\mathcal{S}_-^{(k)} = \{(a,0) \mid (a,0) \in \mathcal{S}^{(k)}\}$$

$$\mathcal{S}_+^{(k)} = \{(0,b) \mid (0,b) \in \mathcal{S}^{(k)}\}.$$

Then, by applying the State Space Theorem to this length two code, we get:

Theorem 10.1 (Transition Space Theorem)

$$\frac{\mathcal{S}^{(k)}}{\mathcal{S}_-^{(k)}\mathcal{S}_+^{(k)}} \simeq \frac{\Sigma_k}{\mathcal{S}_-^{(k)}} \simeq \frac{\Sigma_{k+1}}{\mathcal{S}_+^{(k)}}.$$

Let A and B be sets of states at time indices k and $k+1$ in the minimal trellis,T, for code $\mathcal{C}$. We will say that there is a *clique of transitions* from A to B if every state in A has a transition to every state in B in T. The Transition Space Theorem shows that the branch space $\mathcal{S}^{(k)}$ is partitioned into disjoint cliques of transitions, each corresponding to a coset of $\mathcal{S}_-^{(k)}\mathcal{S}_+^{(k)}$ in $\mathcal{S}^{(k)}$. In particular, the set of transitions in $\mathcal{S}_-^{(k)}\mathcal{S}_+^{(k)}$ will be called the *zero clique*. See Figure 7 from Example 11 for a good illustration of this fact.

The Transition Space Theorem enables us to give a succinct representation of the minimal trellis for a group code: for each time index, k, store the zero clique of transitions and quotient group

$$\frac{\mathcal{S}^{(k)}}{\mathcal{S}_-^{(k)}\mathcal{S}_+^{(k)}}.$$

Clearly, this information is sufficient to compute the local description of the trellis.

11 Lattice Codes

Our algorithm can be used for constructing minimal trellises for lattices. Lattice codes constitute an important class of coset codes. An excellent treatment of lattice codes and trellises for lattice codes can be found in [6].

A real N-dimensional lattice is said to be *rectangular* if it has a generator matrix which is diagonal [2]. A lattice has a finite state trellis diagram with respect to a given set of coordinates if and only if it contains a sublattice that is rectangular with respect to the same set of coordinates. Let $M = diag(a_1, a_2, \ldots, a_N)$ be a generator matrix of a N-dimensional rectangular lattice, say Λ, and Λ_R the maximal rectangular sublattice in Λ. Then Λ_R has "trivial dynamics" and is the "non-dynamical component" of Λ [9]. The quotient Λ/Λ_R is a finite Abelian group and the techniques presented in our work can be used to study the dynamical structure of this group. We illustrate this with the following example:

Example 12 Let the lattice Λ be generated by

$$M = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & -2 & 0 & 0 \\ 0 & 2 & -3 & 0 \\ 0 & 0 & 3 & -4 \end{bmatrix}$$

The maximal rectangular sublattice Λ_R is generated by $diag(2, 4, 6, 8)$ and the quotient Λ/Λ_R is isomorphic to a subgroup of $Z_2 \times Z_4 \times Z_6 \times Z_8$. The generator matrix for this subgroup can be obtained by taking the i^{th} column modulo a_i and discarding redundant rows and is shown below.

$$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 2 & 3 & 0 \\ 0 & 0 & 3 & 4 \end{bmatrix}$$

The corresponding trellis diagram is shown in figure 8(a). The trellis diagram for Λ can be obtained by taking the product of this trellis with the trellis for Λ_R shown in figure 8(b).

12 Discussion

A natural first step in extending our work to codes over non-Abelian groups is to consider group codes over semi-direct product groups, for example Dihedral groups. Group codes over such groups, obtainable using multilevel constructions, have been characterized in [10] and a theorem (Theorem 3) on trellis construction has been proposed. We believe that if we obtain minimal trellises for the component codes of the group code first using our algorithm and then take the product of the resulting trellises, we shall get the minimal trellis for the block group codes discussed in [10].

A further extension to group codes over arbitrary finite non-Abelian groups seems difficult at present since we do not know of a generator matrix description for such codes. In particular, the set of endomorphisms of a non-Abelian group in general do not form a ring.

Another research direction worth investigating is using the succinct representation of minimal trellises to obtaining faster decoding algorithms.

13 Acknowledgements

We wish to thank the anonymous reviewer for suggesting several important improvements, including pointing out the application to lattices, and giving Example 12.

References

[1] E. Biglieri and M. Elia, "Construction of linear block codes over groups," in proceedings of 1993 IEEE International Symposium on Information Theory, San Antonio, Texas.

[2] J.H. Conway and N.J.A. Sloane, "Sphere Packing, Lattices and Groups," 2nd edition, *Springer Verlag*, 1988.

[3] F. Fagnani and S. Zampieri, "Dynamical systems and convolutional codes over finite Abelian groups," *IEEE Trans. on Inform. Theory*, this issue.

[4] G. D. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE* , vol. 61, 1973.

[5] G. D. Forney, "Coset codes - part I: introduction and geometrical classification," *IEEE Trans. on Inform. Theory*, vol. IT-34, pp. 1123-1151, 1988.

[6] G. D. Forney, "Coset codes - part II: binary lattices and related codes," *IEEE Trans. on Inform. Theory*, vol. IT-34, pp. 1152-1187, 1988.

[7] G. D. Forney, "Geometrically uniform codes," *IEEE Trans. on Inform. Theory*, vol. IT-37, pp. 1241-1260, 1991.

[8] G. D. Forney, "Dimension/length profiles and trellis complexity of linear block codes," *IEEE Trans. on Inform. Theory*, vol. IT-40, pp. 1741-1752, 1994.

[9] G. D. Forney and M. Trott, "The dynamics of group codes: state spaces, trellis diagrams, and canonical encoders," *IEEE Trans. on Inform. Theory*, vol. IT-39, pp. 1491-1513, 1993.

[10] R. Garello and S. Benedetto, "Multi-level construction of block and trellis group codes," *IEEE Trans. on Inform. Theory*, vol. IT-41, pp. 1257-1264, 1995.

[11] A. R. Hammons, P. Vijay Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Sole, "The Z_4-linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. on Inform. Theory*, vol. IT-40, pp. 301-319, 1994.

[12] M. Isaksson and L. H. Zetterberg, "Block-coded M-PSK modulation over $GF(M)$," *IEEE Trans. on Inform. Theory*, vol. IT-39, pp. 337-346, 1993.

[13] F. R. Kschischang, "The Trellis structure of maximal fixed-cost codes," *IEEE Trans. on Inform. Theory*, this issue.

[14] F. R. Kschischang, P.G. de Buda and S. Pasupathy, "Block coset codes for M-ary phase shift keying," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 900-913, 1989.

[15] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Trans. on Inform. Theory*, vol. IT-41, no. 6, Nov 1995.

[16] A. Lafourcade and A. Vardy, "Lower bounds on trellis complexity of block codes," manuscript, 1995.

[17] H. Loeliger and T. Mittelholzer, "Convolutional codes over groups," *IEEE Trans. on Inform. Theory*, this issue.

[18] H. Loeliger, G. D. Forney, T. Mittelholzer and M. Trott, "Minimality and observability of group systems," *Linear Algebra and its Applications* Vol. 205-206, pp. 937-963, 1994.

[19] H. Matsumura, *Commutative Ring Theory*, Cambridge University Press, 1986.

[20] D. J. Muder, "Minimal trellises for block codes," *IEEE Trans. on Inform. Theory*, vol. IT-34, pp. 1049-1053, 1988.

[21] J. G. Proakis, *Digital Communications*, McGraw Hill, 1989.

[22] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. on Inform. Theory*, vol. IT-28, pp. 55-67, 1982.

[23] J. C. Willems, "From time series to linear systems, Part I," *Automatica* , vol. 32, pp. 561-580, 1986.

[24] J. C. Willems, "From time series to linear systems, Part II," *Automatica* , vol. 32, pp. 675-694, 1986.

[25] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. on Inform. Theory*, vol. IT-24, pp. 76-80, 1978.

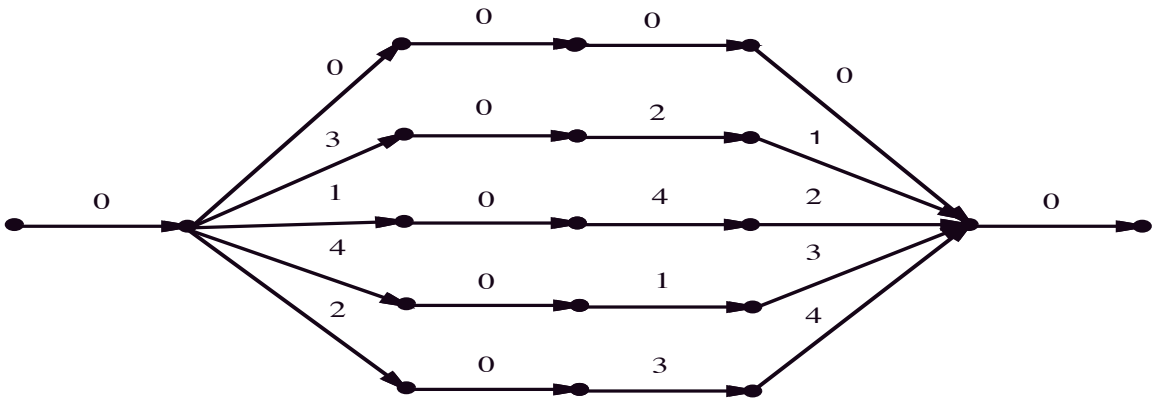

Figure 1: The trellis for a single vector (example 1)

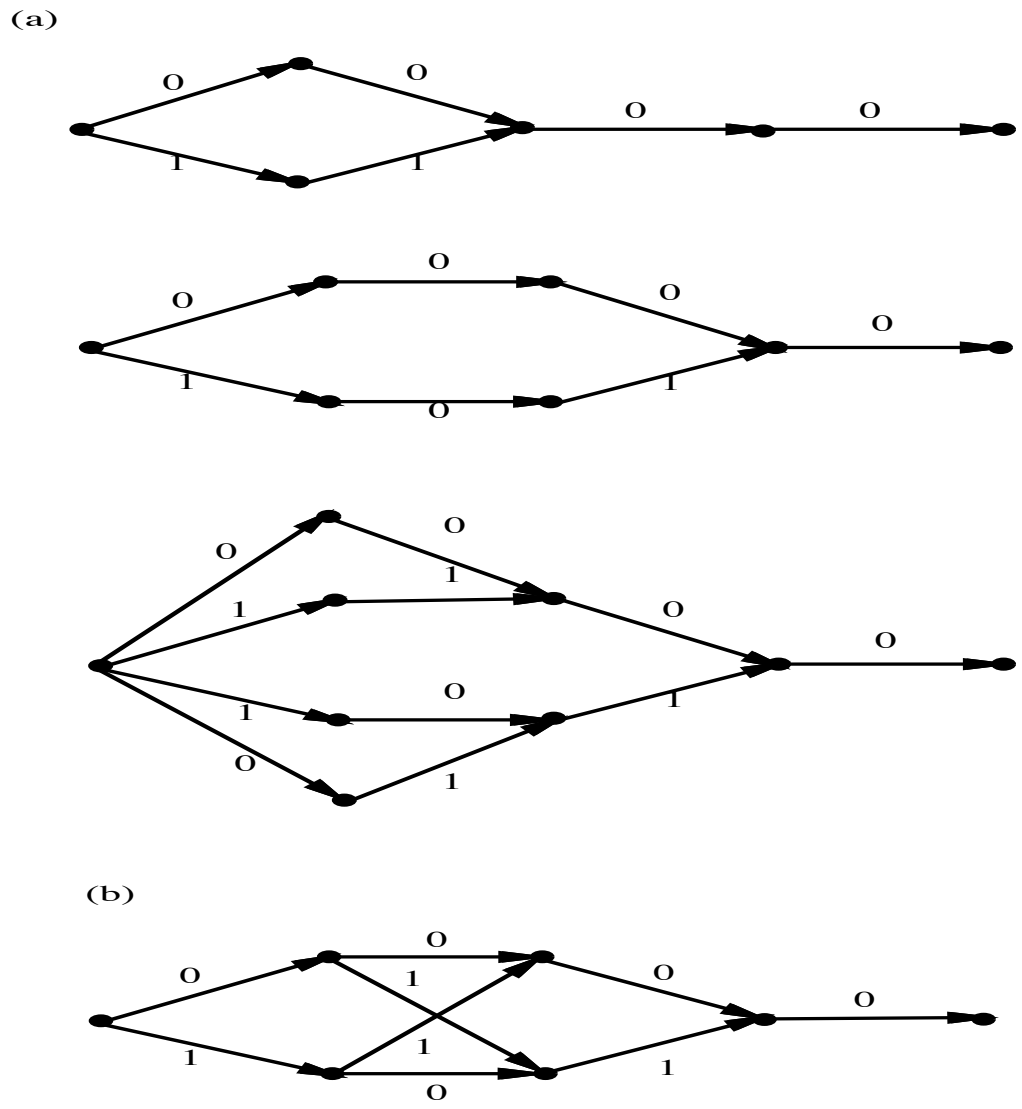

Figure 2: The trellis from an arbitrary generator matrix (example 2)

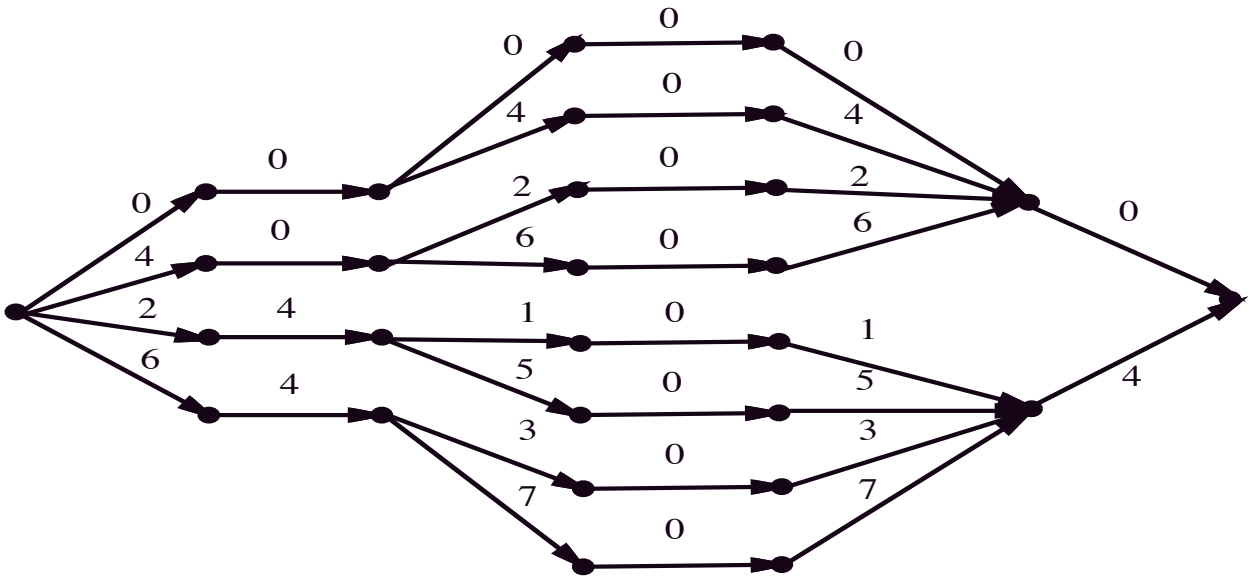

Figure 3: The trellis for the code generated by a single vector over a ring.

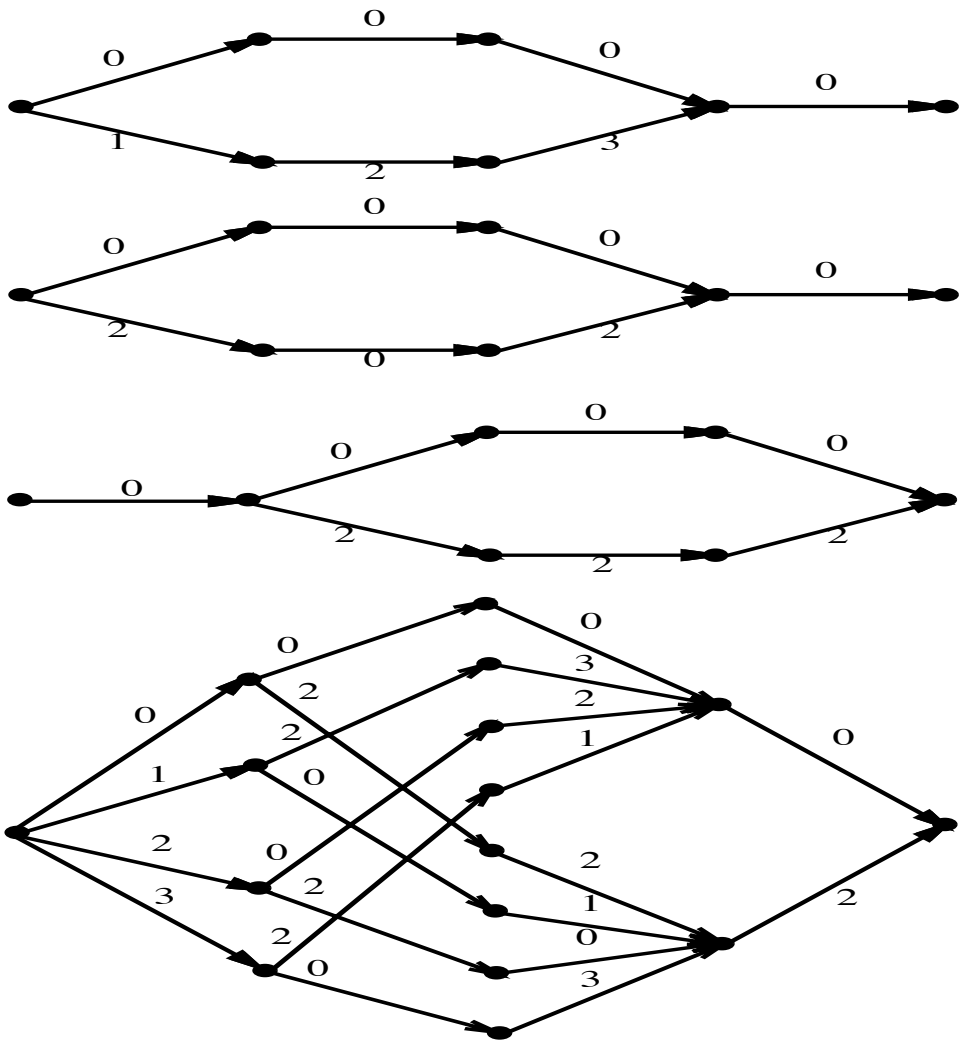

Figure 4: The trellis for Example 8

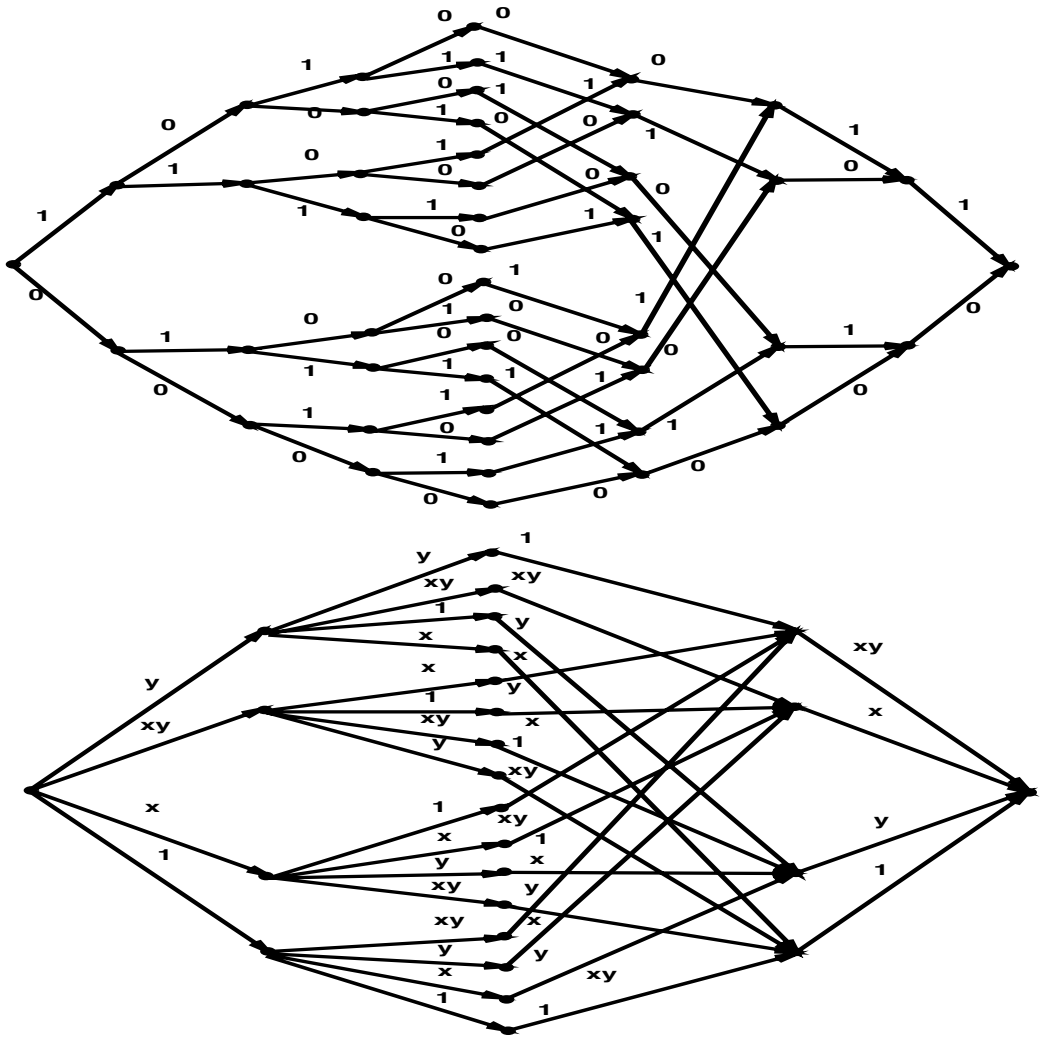

Figure 5: The trellis for Example 9

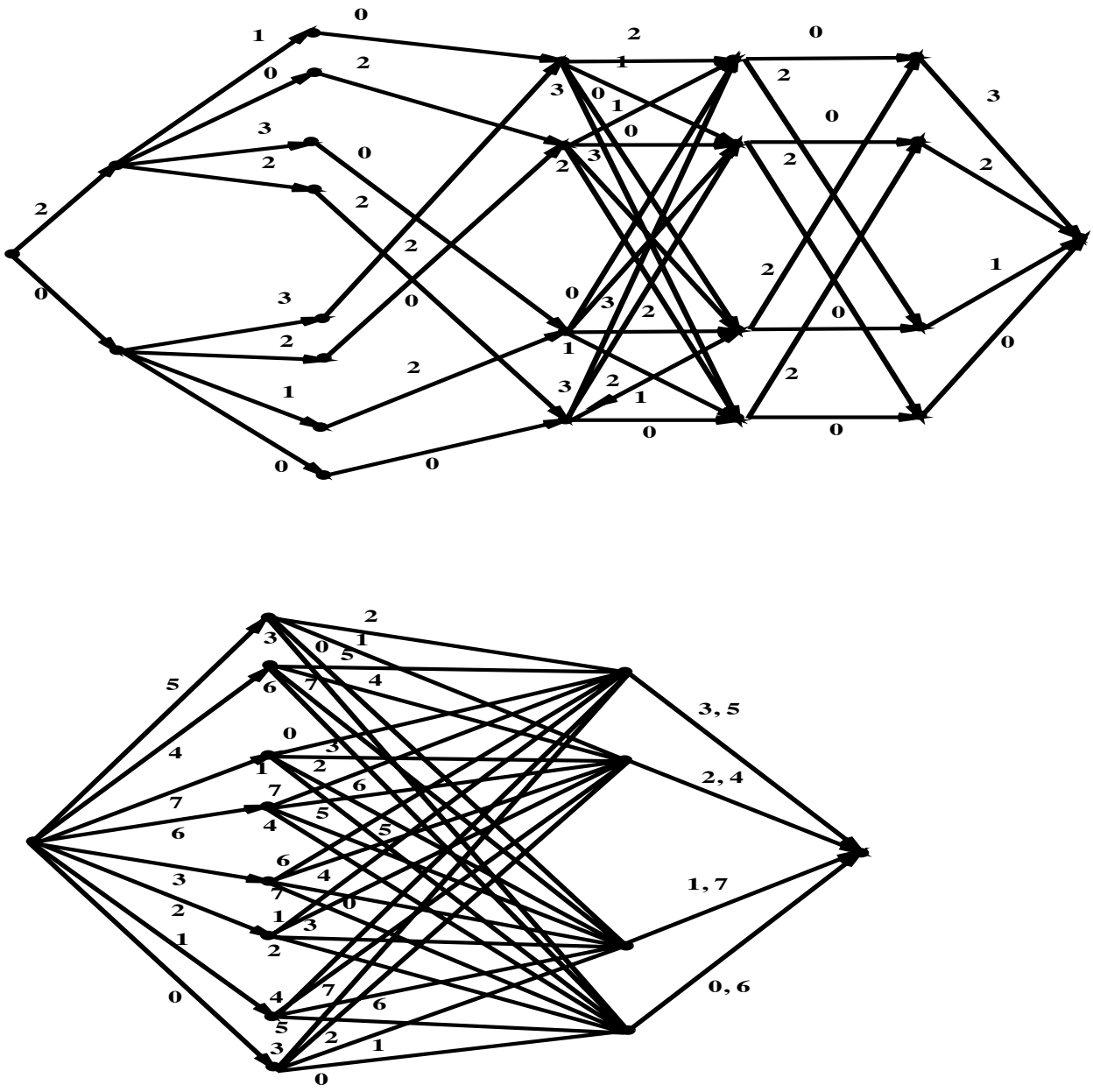

Figure 6: The trellis for Example 10

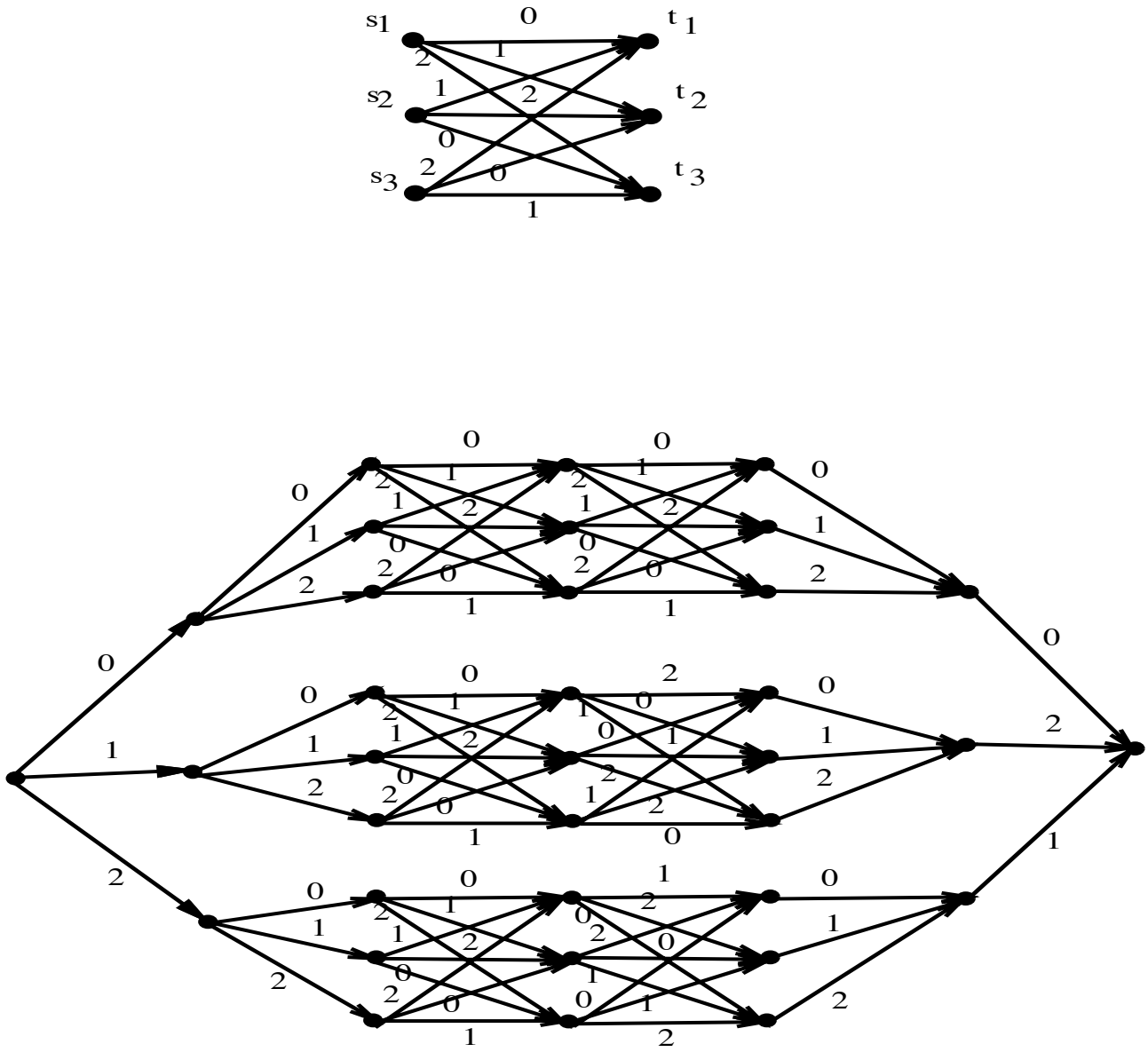

Figure 7: The trellis for Example 11

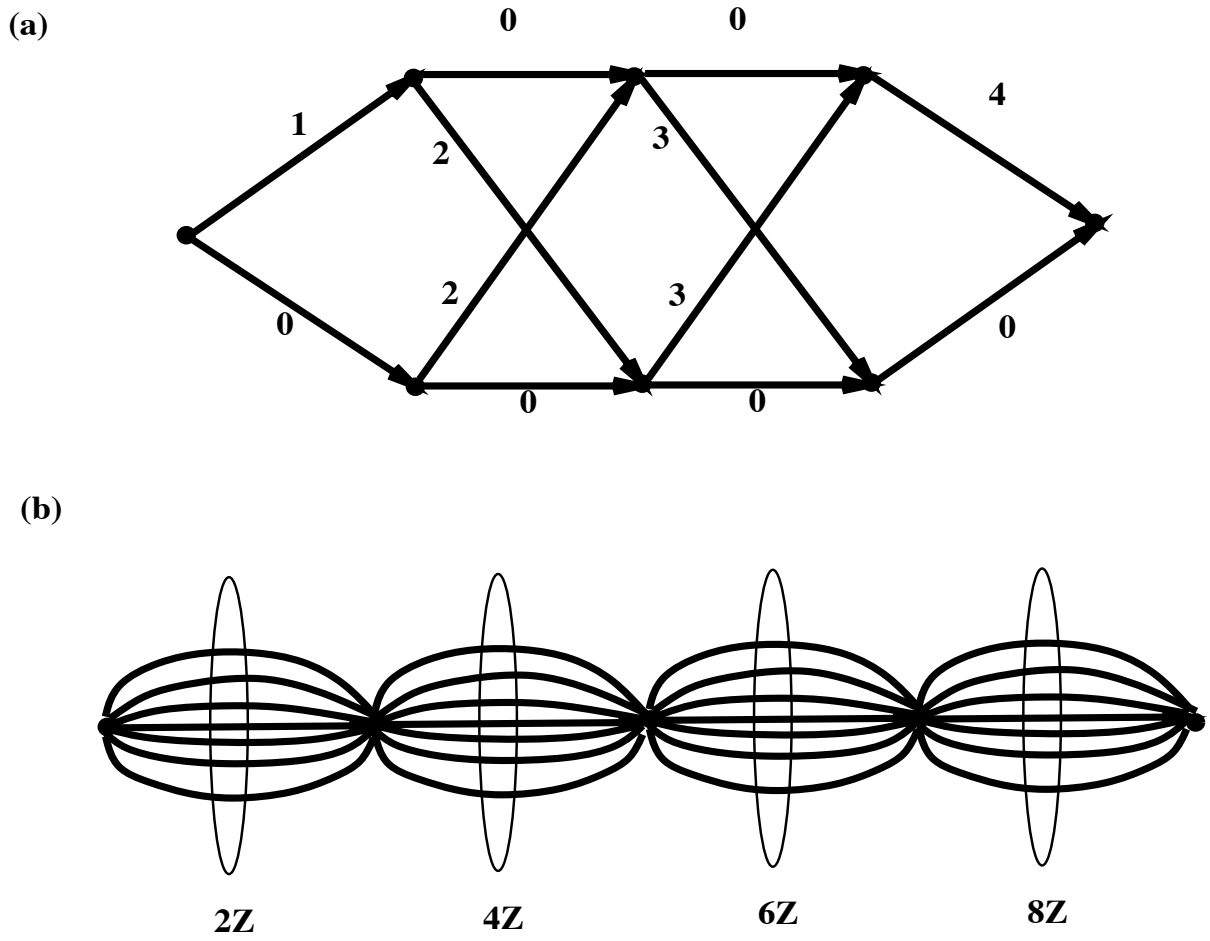

Figure 8: The trellis for Example 12