Tail-biting Trellises for Linear Codes and their Duals

Aditya Nori Priti Shankar

IISc-CSA-TR-2003-4

http://archive.csa.iisc.ernet.in/TR/2003/4/

Computer Science and Automation
Indian Institute of Science, India

October 2003

Tail-biting Trellises for Linear Codes and their Duals

Aditya Nori Priti Shankar

Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India 560012
{aditya, priti}@csa.iisc.ernet.in

Abstract

Construction of dual tail-biting trellises from primal ones is an important problem in trellis based decoding algorithms for linear codes. Generalizations of two well known labeling algorithms are presented for the construction of tail-biting trellises. The construction techniques lead directly to an algorithm for construction of a dual trellis from an algebraic description of the primal one.

1 Introduction

Trellis representations of linear block codes are attractive because of their use in soft decision decoding algorithms. An interesting property that is known for conventional trellises is that the minimal conventional trellis (known to be unique) for a linear block code, and its dual have the same state-complexity profile. This interesting property follows from the BCJR construction [1] of the conventional trellis. Tail-biting trellises [3] are known to achieve substantially lower state complexities for the same codes. However, there are different notions of minimality for tail-biting trellises, each of which yield partial orderings which are different and incomparable [6]. Koetter and Vardy have suggested a dual trellis construction which uses a special trellis product called an intersection product [6]. The resultant dual trellis has a state-complexity profile which is less than or equal to (component wise) that of the primal trellis and equal if it is θ-minimal [6]. In this paper we generalize the Massey [9] and BCJR constructions for conventional trellises to obtain analogous labeling schemes for tail-biting trellises. We also give a simple and direct dual construction algorithm yielding dual trellises with exactly the same state-complexity profile as the primal trellises for the class of non-mergeable trellises [8, 10, 11], which properly includes the class of θ-minimal trellises.

2 Preliminaries

In this section, we introduce some concepts related to tail-biting trellises [5].

Definition 2.1 *A* tail-biting trellis $T = (V, E, \Sigma)$ *of depth* n *is an edge-labeled directed graph with the property that the set* V *can be partitioned into* n *vertex classes*

$$V = V_0 \cup V_1 \cup \ldots \cup V_{n-1} \quad (1)$$

such that every edge in T is labeled with a symbol from the alphabet Σ, and begins at a vertex of V_i and ends at a vertex of $V_{i+1 (mod\ n)}$, for some $i \in \{0, 1, \ldots, n-1\}$.

The set of indices $\mathcal{I} = \{0, 1, \ldots, n-1\}$ for the partition in (1) are the *time indices*. We will refer to $\log_{|\Sigma|} |V_i|$ as the *state-complexity* of the trellis at time index i and the sequence $\left\{\log_{|\Sigma|} |V_i|,\ 0 \le i \le n\right\}$ as the *state-complexity profile* of the trellis. We identify $\mathcal{I}$ with $\mathbb{Z}_n$, the residue classes of integers modulo n. An interval of indices $[i, j]$ represents the sequence $\{i, i+1, \ldots j\}$ if $i < j$, and the sequence $\{i, i+1, \ldots n-1, 0, \ldots j\}$ if $i > j$. Every cycle in T starting at a vertex of V_0 defines a vector $(a_1, a_2, \ldots, a_n) \in \Sigma^n$ which is an *edge-label sequence*. We assume that every vertex and every edge in the tail-biting trellis lies on some cycle. The trellis T *represents* a block code $\mathcal{C}$ over Σ if the set of all edge-label sequences in T is equal to $\mathcal{C}$. Let $\mathcal{C}(T)$ denote the code represented by the trellis T. In addition to the labeling of edges, each vertex in the set V_i is labeled by a sequence of length $l_i \ge \lceil \log_{|\Sigma|} |V_i| \rceil$ of elements in Σ , all vertex labels at a given depth being distinct. Thus every cycle in this labeled trellis defines sequences of length $n + l$ (where $l = l_1 + l_2 + \cdots + l_n$) over Σ, consisting of alternating labels of vertices and edges in T. This sequence is called the *label sequence* of T. The set of all label sequences in a labeled tail-biting trellis is called the *label code* represented by T and is denoted by $\mathcal{S}(T)$. A trellis T is said to be *linear* if there exists a vertex labeling of T such that $\mathcal{S}(T)$ is a vector space. The notion of non-mergeability is also useful here. T is *non-mergeable* [8, 10, 11] if there do not exist vertices in the same vertex class of T that can be replaced by a single vertex, while retaining the edges incident on the original vertices, without modifying $\mathcal{C}(T)$. Koetter and Vardy [5] have shown that if a linear trellis is non-mergeable, then it is also biproper. However, though the converse is true for conventional trellises, it is not true in general for tail-biting trellises [5].

In the discussion that follows, we restrict ourselves to trellises representing linear block codes over the alphabet $\Sigma = \mathbb{F}_q$. Any linear trellis, conventional or tail-biting, for an (n, k) linear code $\mathcal{C}$ can be constructed as a *trellis product* [7] of the representation of the individual trellises corresponding to the k rows of the generator matrix G for $\mathcal{C}$ [5]. The trellis product T of a pair of trellises T_1 and T_2 will have at index i a set of vertices which is the Cartesian product of vertices of T_1 and T_2 at that time index, with every edge between time indices i and $i+1$ from (v_1, v_2) to (v_1', v_2'), where (v_1, a, v_2) is an edge between vertices at time indices i and $i+1$ in T_1, and (v_1', a', v_2') is an edge between vertices at time indices i and $i+1$ in T_2, having label $a + a'$ where $+$ denotes addition in the field $\mathbb{F}_q$. Let $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k\}$ be the rows of a generator matrix G for the linear code $\mathcal{C}$. Each vector $\mathbf{g}_i$ generates a one-dimensional subcode of $\mathcal{C}$, which we denote by $\langle \mathbf{g}_i \rangle$. Therefore $\mathcal{C} = \langle \mathbf{g}_1 \rangle + \langle \mathbf{g}_2 \rangle + \cdots + \langle \mathbf{g}_k \rangle$, and the trellis representing $\mathcal{C}$ is given by $T = T_1 \times T_2 \times \cdots \times T_k$, where T_i is the trellis for $\langle \mathbf{g}_i \rangle$, $1 \le i \le k$. To specify the component trellises in the trellis product above, we will need to introduce the notions of linear and circular spans and elementary trellises [5]. Given a codeword $\mathbf{c} \in \mathcal{C}$, the *linear span* of $\mathbf{c}$, is the semi-open interval $(i, j] \in \mathcal{I}$ corresponding to the smallest closed interval $[i, j]$, $j > i$, which contains all the non-zero positions of $\mathbf{c}$. A *circular span* has exactly the same definition with $i > j$. Note that for a given vector, the linear span is unique, but circular spans are not– they depend on the runs of consecutive zeros chosen for the complement of the span with respect to the index set $\mathcal{I}$. For a vector $\mathbf{x} = (x_1, \ldots, x_n)$ over the field $\mathbb{F}_q$, there is a unique *elementary trellis* representing $\langle \mathbf{x} \rangle$ [5]. This trellis has q vertices at those positions that belong to the chosen span (linear or circular), and a single vertex at other positions. Consequently, T_i in the trellis product mentioned earlier, is the elementary trellis representing $\langle \mathbf{g}_i \rangle$ for some choice of span (either linear or circular).

Koetter and Vardy [5] have shown that any linear trellis, conventional or tail-biting (we will refer to this trellis as the KV trellis) can be constructed from a generator matrix whose rows can be partitioned into two sets, those which have linear span, and those taken to have circular span. Then the trellis is formed as a product of the elementary trellises corresponding to these rows. We will represent such a generator matrix as $G = \left[\begin{array}{c} G_l \\ \hline G_c \end{array}\right]$, where G_l is the submatrix consisting of rows with linear span, and G_c the submatrix of rows with circular span.

3 The Massey Tail-biting trellis

We will first describe the Massey construction [9] of the minimal conventional trellis for a linear block code before describing our "Massey" scheme for constructing tail-biting trellises. The Massey construction uses parity check symbols that have yet to be observed after the current time index to label states. To achieve this it has to put the generator matrix for the code into a "systematic" form, that is, a form in which certain symbols positions are reserved information symbols and the others for check symbols. It is well known that every generator matrix has a unique *row-reduced-echelon* (RRE) form [4] and this is the starting point for the Massey algorithm. The Massey trellis for an (n, k) linear block code $\mathcal{C}$ with generator matrix G (in RRE form) is computed by associating the vertices V_i at time index i with parity symbols that are determined by information symbols that have already been observed at time i, with the remaining information symbols being treated as zeros. Let j be the largest integer such that the leading non-zero component of $\mathbf{g}_j$ (denoted by $\triangleright(\mathbf{g}_j)) \leq i$. Then

$$V_i = \{(c_{i+1}, \ldots, c_n) : (c_1, \ldots, c_n) = (u_1, \ldots, u_j, 0, \ldots, 0)G\}$$

where $(u_1, \ldots, u_j) \in \mathbb{F}_q^j$. By convention, we have $V_0 = \{0\}$ and $V_n = \{\epsilon\}$. There is an edge $e \in E_i$ labeled c'_i from a vertex $\mathbf{v} \in V_{i-1}$ to a vertex $\mathbf{v}' \in V_i \iff \exists$ a pair of codewords $\mathbf{c} = (c_1, \ldots, c_n), \mathbf{c}' = (c'_1, \ldots, c'_n)$ s.t. $(c_i, \ldots, c_n) = \mathbf{v}$, $(c'_{i+1}, \ldots, c'_n) = \mathbf{v}'$ and either $\mathbf{c} = \mathbf{c}'$ or $\beta(\mathbf{c}' - \mathbf{c})$ equals the j^{th} row of G for some $\beta \in \mathbb{F}_q$. The resulting trellis is isomorphic to the BCJR conventional trellis [11]. We will now describe the modified Massey construction for a tail-biting trellis $T = (V, E, \mathbb{F}_q)$ for an (n, k) block code $\mathcal{C}$ with generator matrix G (in RRE form and annotated with appropriate spans).

Let $E = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_k \end{bmatrix}$ be a $k \times n$ matrix s.t.

$$\mathbf{e}_i = \begin{cases} (0, 0, \ldots, g_{i,a}, g_{i,a+1}, \ldots, g_{i,n}) & \text{if } \mathbf{x} \in \langle \mathbf{g}_i \rangle,\ \mathbf{g}_i \in G_c \text{ with circular span } (a, b] \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Let j be the largest integer s.t. $\triangleright(g_j) \leq i$. Then the vertex set V_i at time index i as follows:

$$V_i = \left\{ (0, 0, \ldots, c_{i+1}, c_{i+2}, \ldots, c_n) + \mathbf{f} : (c_1, \ldots, c_n) = (u_1, \ldots, u_j, 0, \ldots, 0)G,\ \mathbf{f} = \sum_{i=0}^{j} u_i \mathbf{e}_i \right\}$$

where $(u_1, \ldots, u_j) \in \mathbb{F}_q^j$. There is an edge $e \in E_i$ labeled c'_i from a vertex $\mathbf{v} \in V_{i-1}$ to a vertex $\mathbf{v}' \in V_i \iff \exists$ a pair of codewords $\mathbf{c} = (c_1, \ldots, c_n), \mathbf{c}' = (c'_1, \ldots, c'_n)$

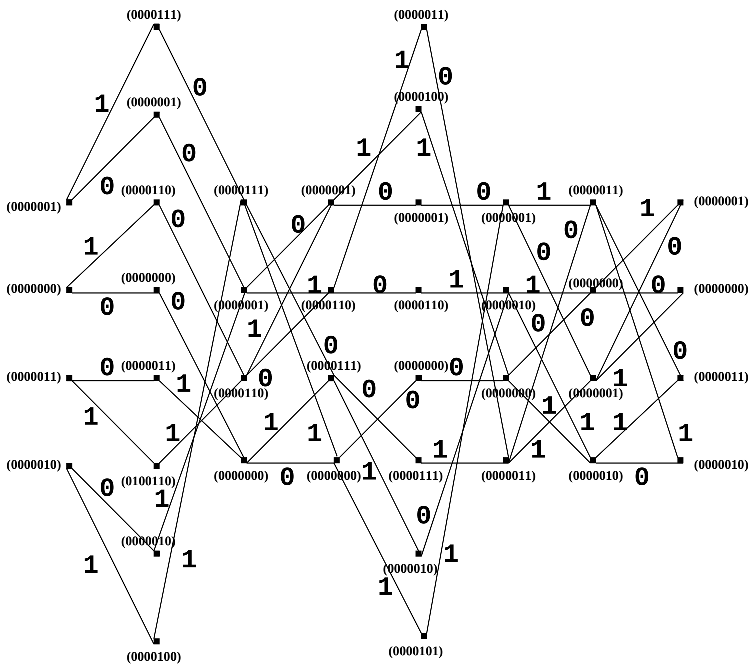

Figure 1: A Massey tail-biting trellis for the $(7,4)$ Hamming code

s.t. $(0,\ldots,0,c_i,\ldots,c_n)+\mathbf{f}=\mathbf{v}$, $\big(0,\ldots,0,c'_{i+1},\ldots,c'_n\big)+\mathbf{f}'=\mathbf{v}'$ (where $\mathbf{f}=\sum_{i=0}^{j}u_i\mathbf{e}_i$, $\mathbf{f}'=\sum_{i=0}^{j}u'_i\mathbf{e}_i$ s.t. $(u_1,\ldots,u_j,0,\ldots,0)G=\mathbf{c}$ and $\big(u'_1,\ldots,u'_j,0,\ldots,0\big)G=\mathbf{c}'$), and either $\mathbf{c}=\mathbf{c}'$ or $\beta(\mathbf{c}'-\mathbf{c})$ equals the j^{th} row of G for some $\beta\in\mathbb{F}_q$ We will show in Section 4 that T is a non-mergeable linear trellis that represents $\mathcal{C}$.

Example 1 *Consider the* $(7,4)$ *Hamming code defined by the parity check matrix* H *and the generator matrix* G *(annotated with spans):*

$$G=\begin{array}{c}\left[\begin{array}{ccccccc}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&0&1\end{array}\right]\end{array}\begin{array}{c}(1,6]\\(6,2]\\(3,7]\\(7,5]\end{array}\qquad H=\left[\begin{array}{ccccccc}1&1&0&0&1&0&1\\1&1&1&0&0&1&0\\0&1&1&1&0&0&1\end{array}\right]$$

The Massey tail-biting trellis for this code is shown in Figure 1.

4 The BCJR Tail-biting trellis

The original BCJR algorithm [1] constructs the minimal conventional trellis for a linear block code in the following way. Let H be the parity check matrix for a (n,k) linear block code over $\mathbb{F}_q$, and let $\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_n$ be the columns of H. Each codeword $\mathbf{c}=(c_1,\ldots,c_n)$ of the code gives rise to a sequence of states $\{\mathbf{s}_i\}_{i=0}^{n}$, each state being labeled by an $(n-k)\times 1$ vector as follows:

$$\mathbf{s}_i=\begin{cases}\mathbf{0} & \text{if } i=0\\ \mathbf{s}_{i-1}+c_i\mathbf{h}_i & \text{otherwise}\end{cases}$$

Clearly, there will be a single state at time index n as $H\mathbf{c}^T=0$ for all codewords $\mathbf{c}$. We refer to such a labeling as a BCJR labeling of the trellis in the following section. It is well known that the set of vectors that are labels at each time index form a vector space whose dimension is the state-complexity at that time index. The algorithm is now generalized to construct labeled tail-biting trellises for any linear code with the trellis satisfying the following two properties.

1. The trellis formed is isomorphic to the KV trellis.
2. The state labels at each time index form a vector space.

Let $\mathcal{C}$ be an (n, k) linear code over $\mathbb{F}_q$ with generator matrix $G = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k\}$ and parity check matrix $H = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_n \end{bmatrix}$. The algorithm BCJR-TBT constructs a non-mergeable linear tail-biting trellis T, given G (with appropriate spans) and H. Informally speaking, the algorithm first constructs the conventional BCJR labeled trellis for the subcode consisting of the rows of linear span, and then adds states and edges for linear combinations of each row $\mathbf{g} \in G_c$ with circular span, in turn, by offsetting BCJR-like label computations with a state vector that is formed by beginning the label computations for $\mathbf{g}$ at the start of the circular span and proceeding in a circular order, rather than beginning at time index 0 and proceeding in linear order, as is performed for rows of linear span. The intermediate generator matrix which is formed by including one row of circular span at a time is called G_{int}.

Algorithm BCJR-TBT
`Input:` The matrices G and H.
`Output:` A non-mergeable linear tail-biting trellis $T = (V, E, \mathbb{F}_q)$ representing $\mathcal{C}$.

Initialization: $G_{int} = G_l$. Let $\{\mathbf{d}_\mathbf{x}\}_{\mathbf{x}\in\mathcal{C}}$ as follows:

$$\mathbf{d}_\mathbf{x} = \begin{cases} \sum_{j=a}^{n} x_j \mathbf{h}_j & \text{if } \mathbf{x} \in \langle \mathbf{g}_i \rangle,\ \mathbf{g}_i \text{ is a row of } G_c \text{ with circular span } (a, b] \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Step 1: Construct the BCJR labeled trellis for the subcode generated by the submatrix G_l, but using the matrix H instead of the parity check matrix for the code G_l. Let $V_0, V_1 \ldots V_n$ be the vertex sets created and $E_1, E_2, \ldots E_n$ be the edge sets created.

Step 2: **for** each row vector $\mathbf{g}$ of G_c
  for each $\mathbf{x} \in \langle \mathbf{g} \rangle$, $\mathbf{y}$ in the rowspace of G_{int}.
  {
   let $\mathbf{z}$ denote the codeword $\mathbf{x} + \mathbf{y}$.
   let $\mathbf{d}_\mathbf{z} = \mathbf{d}_\mathbf{x} + \mathbf{d}_\mathbf{y}$.
   $V_0 = V_n = V_0 \cup \{\mathbf{d}_\mathbf{z}\}$.
   $V_i = V_i \cup \left\{\mathbf{d}_\mathbf{z} + \sum_{j=1}^{i} z_j \mathbf{h}_j\right\}, 1 \le i < n$.
   There is an edge $e = (\mathbf{u}, z_i, \mathbf{v}) \in E_i$, $\mathbf{u} \in V_{i-1}$, $\mathbf{v} \in V_i$, $1 \le i \le n$
    $\iff \mathbf{d}_\mathbf{z} + \sum_{j=1}^{i-1} z_j \mathbf{h}_j = \mathbf{u}$ and $\mathbf{d}_\mathbf{z} + \sum_{j=1}^{i} z_j \mathbf{h}_j = \mathbf{v}$.
  }
  $G_{int} = G_{int} + \mathbf{g}$.

The properties of the resulting trellis T are given by the following lemmas.

Lemma 4.1 *The trellis T is linear and represents $\mathcal{C}$.*

Proof We first prove that $\mathcal{C}(T) = \mathcal{C}$. Assume to the contrary that $\exists \mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{C}(T)$ s.t. $\mathbf{x} \notin \mathcal{C}$. Then by construction we have $\mathbf{d_x} + \sum_{i=1}^{n} x_i \mathbf{h}_i = \mathbf{d_x} \Rightarrow H\mathbf{x}^T = \mathbf{0} \Rightarrow \mathbf{x} \in \mathcal{C}$, thus contradicting our assumption. Let $\mathbf{x}$, $\mathbf{y} \in \mathcal{C}(T)$ and let $\mathbf{x}'$, $\mathbf{y}' \in \mathcal{S}(T)$ respectively, be their labeled codewords. Since $\mathcal{C}(T) = \mathcal{C}$, we have $\mathbf{z} = \mathbf{x} + \mathbf{y} \in \mathcal{C}(T)$. In order to prove linearity of T, we need to show that $\mathbf{z}' = \mathbf{x}' + \mathbf{y}'$ also belongs to $\mathcal{S}(T)$. Then from the construction, we have $\mathbf{x}' = \langle \mathbf{d_x}, x_1, \mathbf{u}_1, \ldots, x_n, \mathbf{d_x} \rangle$, s.t. $\mathbf{u}_i \mid_{1 \leq i \leq n} = \mathbf{d_x} + \sum_{j=1}^{i} x_j \mathbf{h}_j$ and $\mathbf{y}' = \langle \mathbf{d_y}, y_1, \mathbf{v}_1, \ldots, y_n, \mathbf{d_y} \rangle$, s.t. $\mathbf{v}_i \mid_{1 \leq i \leq n} = \mathbf{d_y} + \sum_{j=1}^{i} y_j \mathbf{h}_j$. Therefore, $\mathbf{z}' = \langle \mathbf{d_x} + \mathbf{d_y}, x_1 + y_1, \mathbf{u}_1 + \mathbf{v}_1, \ldots, \mathbf{x}_n + \mathbf{y}_n, \mathbf{d_x} + \mathbf{d_y} \rangle = \langle \mathbf{d_z}, z_1, \mathbf{d_z} + z_1 \mathbf{h}_1, \ldots, \mathbf{z}_n, \mathbf{d_z} \rangle$, which shows that $\mathbf{z}'$ is the labeled codeword in $\mathcal{S}(T)$ representing the codeword $\mathbf{z}$, thus proving that T is indeed a linear trellis representing $\mathcal{C}$. ■

Lemma 4.2 *The trellis T is non-mergeable.*

Proof Assume to the contrary that T is mergeable. Then there exist *distinct* vertices $\mathbf{v}_1$, $\mathbf{v}_2 \in V_i$, for some time index i, which can be replaced by a single vertex, without modifying $\mathcal{C}(T)$. Let $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$, $\mathbf{y} = \langle y_1, y_2, \ldots, y_n \rangle \in \mathcal{C}(T)$ s.t. their corresponding cycles in T contain the vertices $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively. By construction, $\mathbf{v}_1 = \mathbf{d_x} + \sum_{j=1}^{i} x_j \mathbf{h}_j$ and $\mathbf{v}_2 = \mathbf{d_y} + \sum_{j=1}^{i} y_j \mathbf{h}_j$. As T is mergeable, the following equation must hold: $\mathbf{d_y} = \mathbf{d_x} + \sum_{j=1}^{i} x_j \mathbf{h}_j + \sum_{j=i+1}^{n} y_j \mathbf{h}_j$ (this represents the equation that must hold for the path from the vertex $\mathbf{d_x}$ to the vertex $\mathbf{d_y}$). Therefore $\mathbf{d_y} = \mathbf{d_x} + \sum_{j=1}^{i} x_j \mathbf{h}_j - \sum_{j=1}^{i} y_j \mathbf{h}_j \Rightarrow \mathbf{d_x} + \sum_{j=1}^{i} x_j \mathbf{h}_j = \mathbf{d_y} + \sum_{j=1}^{i} y_j \mathbf{h}_j \Rightarrow \mathbf{v}_1 = \mathbf{v}_2$, contrary to our original assumption. Therefore T is non-mergeable. ■

Lemma 4.3 *The algorithm* BCJR-TBT *constructs a tail-biting trellis isomorphic to the KV trellis using the generator matrix G iff the KV trellis is non-mergeable.*

Proof Let $I = G_l \cup J$, where $J \subseteq G_c$. Note that $I = G \Rightarrow J = G_c$. We will prove the lemma by mathematical induction on the size of the set J. The lemma is true for $|J| = 0$ as the product construction for the vectors in the matrix $I = G_l$ and the BCJR construction for the subcode generated by G_l give isomorphic conventional trellises. Assume that the lemma is true for some $|J| = m$. Let us add $\mathbf{g} \in G_c$, with circular span $(a, b]$ to J and verify that the lemma holds for $|J| = m + 1$. let T_m be the trellis representing the code generated by I. From the inductive hypothesis, this is the non-mergeable trellis computed by the algorithm BCJR-TBT and by the KV product construction. Denote by $T_{\mathbf{g}}$ the elementary trellis representing $\langle \mathbf{g} \rangle$. By design, the algorithm BCJR-TBT computes a trellis isomorphic to $T_{\mathbf{g}}$ when given $\{\mathbf{g}\}$ as input. The KV product operation $T_m \times T_{\mathbf{g}}$, results in a trellis T_{m+1} whose structure is as follows. T_{m+1} is isomorphic to T_m in the time interval $\mathcal{I} \setminus (a, b]$, otherwise, each state of T_m is replicated q (size of the code alphabet) times. T_{m+1} must also be non-mergeable (as this is assumed in the statement of the lemma). It can be easily seen that under these conditions, the algorithm BCJR-TBT essentially computes $T_m \times T_{\mathbf{g}}$ which proves that the hypothesis is true for $|J| = m + 1$, thus proving the lemma. ■

Proposition 4.4 *The class of trellises computed by the algorithm* BCJR-TBT *is exactly the class of non-mergeable linear trellises.*

Proof This follows from Lemmas 4.2, 4.3 and from the result of Koetter and Vardy [5] that all linear tail-biting trellises arise from product constructions. ■

Let G_i and H_i respectively, denote the submatrices consisting of the first i columns of G and H. Recall that the first l rows of G have linear span and the rest of the $k - l$

rows have circular span. Define a matrix $D = \left[\begin{array}{cccc} \mathbf{d}_1 & \mathbf{d}_2 & \ldots & \mathbf{d}_k \end{array}\right]$ as follows:

$$\mathbf{d}_i = \begin{cases} \mathbf{0} & \text{if } 1 \le i \le l \\ \sum_{j=a}^{n} g_{ij}\mathbf{h}_j & \text{otherwise} \end{cases},$$

where $\mathbf{g}_i = (g_{i1}, \ldots, g_{in})$ has circular span $(a, b]$. Let M_i be defined such that

$$M_i = \left[\begin{array}{c} H_i {G_i}^T + D \end{array}\right]$$

Then we have the following lemma.

Lemma 4.5 *For all time indices $i \in \{0, 1, \ldots, n\}$, V_i the state cardinality of T at time index i equals the column space of M_i.*

Proof Direct consequence of algorithm BCJR-TBT. ■

The BCJR-TBT trellis for the $(7, 4)$ Hamming code from Example 1 is illustrated in Figure 2.

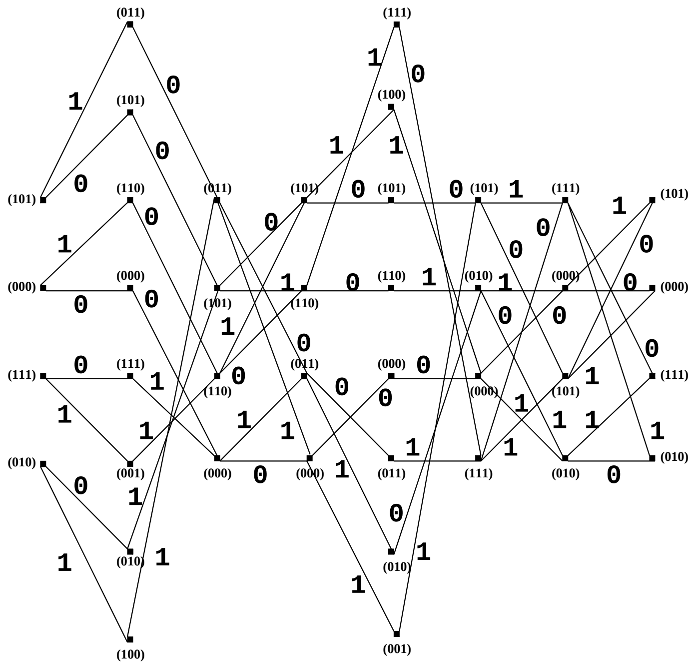

Figure 2: A BCJR-TBT trellis for the $(7, 4)$ Hamming code

Lemma 4.6 *Given an (n, k) code $\mathcal{C}$ with a fixed generator matrix G (in RRE form with associated spans) and parity check matrix H, we have for all time indices $i \in \{0, 1, \ldots, n\}$, the Massey and BCJR tail-biting trellises are isomorphic to each other.*

Proof Let m denote the largest integer such that $\triangleright(\mathbf{g}_m) \le i$. Define

$$K = \left[\begin{array}{c|c} \mathbf{0} & G_{m,n-i} \end{array}\right] + \left[\begin{array}{c} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_m \end{array}\right]$$

It can be verified that $|V|$=rank K. Therefore we have

$$\text{rank } K = \text{rank } HK^T = \text{rank } \left(H \left[\begin{array}{c|c} \mathbf{0} & G_{m,n-i} \end{array} \right]^T + H \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_m \end{bmatrix}^T \right) = \text{rank } \left(H_i G_i^T + D\right),$$

where D is the matrix defined in Section 4. But rank $\left(H_i G_i^T + D\right)$ is the state cardinality of the BCJR tail-biting trellis at time index i - which proves that the Massey and BCJR tail-biting trellises have the same state-complexity profiles. Consider an edge $e = (\mathbf{u}, c_i, \mathbf{v}) \in E_i$ in the Massey trellis and let $\mathbf{c} = (c_1, \ldots, c_n)$ be a codeword whose cycle in the trellis contains e. Then we have $\mathbf{u} = (0, \ldots, 0, c_i, \ldots, c_n) + \mathbf{f}$ and $\mathbf{v} = (0, \ldots, 0, c_{i+1}, \ldots, c_n) + \mathbf{f}$ (we assume for the sake for simplicity that both $\mathbf{u}$ and $\mathbf{v}$ have labels induced by the same codeword), where $\mathbf{f}$ is as defined in Section 3. This is equivalent to saying that there is an edge $e = \left(H\mathbf{u}^T, c_i, H\mathbf{v}^T\right) \in E_i$
$\iff \left(H_i(c_i, \ldots, c_n)^T + H\mathbf{f}^T, c_i, H_{i+1}(c_{i+1}, \ldots, c_n)^T + H\mathbf{f}^T\right) \in E_i$
$\iff \left(\left(\mathbf{d_c} + \sum_{j=1}^{i} c_j \mathbf{h}_j\right), c_i, \left(\mathbf{d_c} + \sum_{j=1}^{i+1} c_j \mathbf{h}_j\right)\right) \in E_i$ - this is precisely the condition for edge placement in the BCJR tail-biting trellis, thus proving that both the Massey and BCJR tail-biting trellises are isomorphic to each other. ■

Since the BCJR tail-biting trellis is a non-mergeable linear trellis, it follows from the above lemma that the *Massey tail-biting trellis is also a non-mergeable linear trellis.*

5 The Dual Tail-biting trellis

Apart from their theoretical properties, dual trellises are of interest because it is sometimes advantageous to decode over them [2]. Koetter and Vardy [6] have defined a special product operation called the *intersection product* to construct a dual linear tail-biting trellis directly from a generator matrix for the primal code. This results in a linear tail-biting trellis $T^\perp$ for the dual code that has the same state-complexity profile if the primal trellis T is θ-minimal (that is, minimal under component-wise ordering), otherwise $T^\perp$ has a smaller state-complexity profile than T. We will now describe the algorithm Dual-TBT that takes G (annotated with spans) and H as inputs and computes a non-mergeable linear tail-biting trellis $T^\perp$ for the dual code $\mathcal{C}^\perp$. $T^\perp$ will have the property that its state-complexity profile is equal to the state-complexity profile of T.

Algorithm Dual-TBT
`Input:` The matrices G and H.
`Output:` A non-mergeable tail-biting trellis $T^\perp = (V, E, \mathbb{F}_q)$ representing $\mathcal{C}^\perp$.

Initialization: $V_i \mid_{0 \le i \le n} = E_i \mid_{1 \le i \le n} = \phi$.

for each $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \mathcal{C}^\perp$.

{

let $\mathbf{d} = (d_1 d_2 \ldots d_k)^T$ **s.t.** $d_i = \begin{cases} 0 & \text{if } 1 \le i \le l \\ \sum_{j=a}^{n} y_j g_{i,j} & \text{otherwise} \end{cases}$

where $\mathbf{g}_i \in G$ has circular span $(a, b]$.

$V_0 = V_n = V_0 \cup \{\mathbf{d}\}$.
$V_i = V_i \cup \left\{\mathbf{d} + \sum_{j=1}^{i} y_j (g_{j,1} g_{j,2} \ldots g_{j,k})^T\right\}$.
There is an edge $e = (\mathbf{u}, z_i, \mathbf{v}) \in E_i$, $\mathbf{u} \in V_{i-1}$, $\mathbf{v} \in V_i, 1 \leq i \leq n$, $\iff$
$\mathbf{d} + \sum_{j=1}^{i} y_j (g_{j,1}, g_{j,2}, \ldots, g_{j,k})^T = \mathbf{u}$, and
$\mathbf{d} + \sum_{j=1}^{i} y_j (g_{j,1}, g_{j,2}, \ldots, g_{j,k})^T = \mathbf{v}$.

}

The following lemma states the properties of $T^{\perp}$.

Lemma 5.1 *The trellis $T^{\perp}$ is a non-mergeable linear trellis that represents $\mathcal{C}^{\perp}$.*

Proof Proof similar to those of Lemmas 4.1 and 4.2. ■

Let M_i be the matrix from Lemma 4.5.

Lemma 5.2 *For all time indices $i \in \{0, 1, \ldots, n\}$, $V_i^{\perp}$ the state cardinality of $T^{\perp}$ at time index i equals the column space of M_i^T.*

Proof Follows directly from the algorithm Dual-TBT. ■

The Dual-TBT$^{\perp}$ trellis for the Hamming code from Example 1 is shown in Figure 3 and has the same state-complexity profile as its primal counterpart (Figure 2).

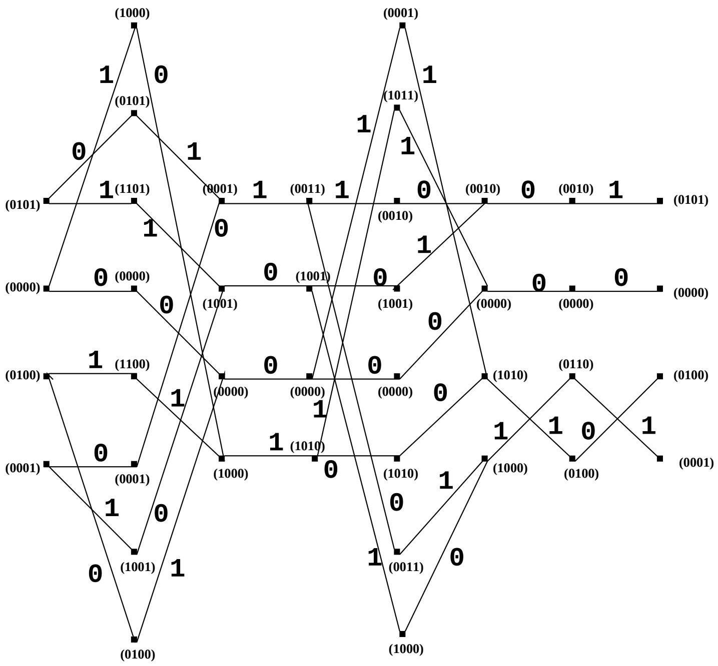

Figure 3: Dual-TBT trellis for the $(7, 4)$ Hamming code

Lemma 5.3 *Let T and $T^{\perp}$ be the trellises computed by the algorithms* BCJR-TBT *and* Dual-TBT *respectively. Then for all all time indices $i \in \{0, 1, \ldots, n\}$, the state cardinality V_i of T at level i equals the state cardinality $V_i^{\perp}$ of $T^{\perp}$ at level i. In other words, $|V_i| = |V_i^{\perp}|$.*

Proof From Lemmas 4.5 and 5.2 we have $|V_i|$ equal to the column space of M_i and $\left|V_i^{\perp}\right|$ equal to the column space of M_i^T. Therefore, by the "row rank=column rank" theorem of linear algebra [4], $|V_i| = \left|V_i^{\perp}\right|$. ■

We conclude with a theorem stating our main result.

Theorem 5.4 *Let T be a non-mergeable linear trellis, either conventional or tail-biting, for a linear code $\mathcal{C}$. Then there exists a non-mergeable linear dual trellis $T^{\perp}$ for $C^{\perp}$ such that the state-complexity profile of $T^{\perp}$ is identical to the state-complexity profile of T.*

Proof Follows from Lemmas 4.1, 4.2, 4.3, 5.1 and 5.3. ■

Finally, as we know that for tail-biting trellises there are several measures of minimality, if any of these definitions requires the trellis to be non-mergeable, it immediately follows that there exist under that definition of minimality, minimal trellises for a code and its dual with identical state-complexity profiles.

References

[1] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, Optimal decoding of linear codes for minimizing symbol error rate, *IEEE Trans. Inform. Theory*, **20**(2), March 1974, pp. 284-287.

[2] J. Berkmann and C. Weiß, On Dualizing Trellis-Based APP Decoding Algorithms, *IEEE Trans. on Communications*, **50**(11), November 2002, pp. 1743-1757.

[3] A.R. Calderbank, G.D. Forney,Jr., and A. Vardy, Minimal Tail-Biting Trellises: The Golay Code and More, *IEEE Trans. Inform. Theory*, **45**(5), July 1999, pp. 1435-1455.

[4] K. Hoffman and R. Kunze, Linear Algebra, Englewood Cliffs, N.J., Prentice, 1961.

[5] R. Koetter and A. Vardy, On the theory of linear trellises, *Information, Coding and Mathematics* (M. Blaum, Editor), Boston:Kluwer, May 2002.

[6] R. Koetter and A. Vardy, The Structure of Tail-Biting Trellises: Minimality and Basic Principles, http://tesla.csl.uiuc.edu/~koetter/publications.html, May 2002.

[7] F.R. Kschischang and V. Sorokine, On the trellis structure of block codes, *IEEE Trans. Inform. Theory*, **41**(6), Nov 1995, pp. 1924-1937.

[8] F.R. Kschischang, The trellis structure of maximal fixed-cost codes, *IEEE Trans. Inform. Theory*, **42**, 1996, pp. 1828-1838.

[9] J.L. Massey, Foundations and methods of channel encoding, *Proc. Int. Conf. Information Theory and Systems*, **65**, NTG-Fachberichte, Berlin, 1978, pp. 148-157.

[10] A. Vardy and F.R. Kschischang, Proof of a conjecture of McEliece regarding the expansion index of the minimal trellis, *IEEE Trans. Inform. Theory*, **42**, November 1996, pp. 2027-2034.

[11] A. Vardy, Trellis structure of codes, *Handbook of Coding Theory*, V.S. Pless and W.C. Huffman, Elsevier, 1998.