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

1 Introduction

The 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, the Massey [3] and the BCJR [1] 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, satisfying the property that the two trellises have identical state-complexity profiles.

2 Our Results

Consider a linear block code $\mathcal{C}$ over $\mathbb{F}_q$ with parameters (n,k) with generator matrix $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_k\}$. The *linear span* of a codeword $\mathbf{c} \in \mathcal{C}$ is defined to be the semi-open interval $(i,j]$ corresponding to the smallest closed interval $[i,j]$, $j > i$, that contains all the non-zero positions of $\mathbf{c}$. A *circular span* has exactly the same definition with $i > j$. In contrast to the linear span of word (which is unique), circular spans of a word are not unique - they depend on the runs of consecutive zeros chosen for the complement of span with respect to the index set. Koetter and Vardy [2] have shown that any linear trellis for $\mathcal{C}$ may be constructed from a generator matrix G whose rows have been partitioned into linear span rows G_l and circular span rows G_c.

Define an $n \times k$ matrix $E^T = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_k \end{bmatrix}$ 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. the first non-zero position of $\mathbf{g}_j \leq i$. Then the vertex set V_i at time index i is defined 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 $\mathbf{c} = (c_1,\ldots,c_n), \mathbf{c}' = (c_1',\ldots,c_n')$ s.t. $(0,\ldots,0,c_i,\ldots,c_n) + \mathbf{f} = \mathbf{v}$, $(0,\ldots,0,c_{i+1}',\ldots,c_n') + \mathbf{f}' = \mathbf{v}'$ (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 $(u_1',\ldots,u_j',0,\ldots,0)G = \mathbf{c}'$), and either $\mathbf{c} = \mathbf{c}'$ or $\beta(\mathbf{c}' - \mathbf{c})$ equals the j^{th} row of G for some $\beta \in \mathbb{F}_q$.

Figure 1: The Massey Construction for a Tail-Biting Trellis

The *Modified Massey Construction* for a tail-biting trellis $T = (V, E, \mathbb{F}_q)$ representing an (n,k) linear code $\mathcal{C}$ requires a generator matrix G in row-reduced echelon form (and annotated with appropriate spans) as input. The trellis T is a *non-mergeable* [2] linear tail-biting trellis representing $\mathcal{C}$. The general idea of this construction is illustrated in Figure 1. We next describe a BCJR-like labeling scheme for tail-biting trellises [1]. Let $H = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_n \end{bmatrix}$ be the parity check matrix for the code. The algorithm BCJR-TBT shown in Figure 2 constructs a non-mergeable linear tail-biting trellises T for $\mathcal{C}$ given G and H. Given an (n,k) code $\mathcal{C}$ specified by a generator matrix G in row-reduced echelon form (with associated spans) and a parity check matrix H, the Massey and BCJR tail-biting trellises are isomorphic to each other. Moreover, the class of trellises computed by the algorithm BCJR-TBT is exactly the class of non-mergeable trellises.

The Algorithm Dual-TBT shown in Figure 3 takes the generator and parity check matrices G, H respectively, of a linear code $\mathcal{C}$ as input and computes a non-mergeable linear tail-biting trellis $T^{\perp}$ for

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 $\{\mathbf{d_x}\}_{\mathbf{x} \in \mathcal{C}}$ as follows:

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

Step 1: Construct the BCJR labeled trellis for the subcode generated by the submatrix G_l using the matrix H. Let $V_0, V_1 \dots V_n$ be the vertex sets created and $E_1, E_2, \dots E_n$ 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 $\mathbf{d_z} = \mathbf{d_x} + \mathbf{d_y}$.
$V_0 = V_n = V_0 \cup \{\mathbf{d_z}\}$.
$V_i = V_i \cup \left\{\mathbf{d_z} + \sum_{j=1}^{i} z_j \mathbf{h}_j\right\}, 1 \le i < n$.
There is an edge $e = (\mathbf{u}, z_i, \mathbf{v}) \in E_i$, $\mathbf{u} \in V_{i-1}$, $\mathbf{v} \in V_i$, $1 \le i \le n$
$\iff \mathbf{d_z} + \sum_{j=1}^{i-1} z_j \mathbf{h}_j = \mathbf{u}$ and $\mathbf{d_z} + \sum_{j=1}^{i} z_j \mathbf{h}_j = \mathbf{v}$.
}
$G_{int} = G_{int} + \mathbf{g}$.

Figure 2: The BCJR-TBT algorithm

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 $\mathbf{y} = (y_1, y_2, \dots, y_n) \in \mathcal{C}^{\perp}$.
{
let $\mathbf{d} = (d_1 d_2 \dots d_k)^T$ **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 $(a, b]$.
$V_0 = V_n = V_0 \cup \{\mathbf{d}\}$.
$V_i = V_i \cup \left\{\mathbf{d} + \sum_{j=1}^{i} y_j (g_{j,1} g_{j,2} \dots g_{j,k})^T\right\}$.
There is an edge $e = (\mathbf{u}, z_i, \mathbf{v}) \in E_i$, $\mathbf{u} \in V_{i-1}$, $\mathbf{v} \in V_i, 1 \le i \le n$, $\iff$
$\mathbf{d} + \sum_{j=1}^{i} y_j (g_{j,1}, g_{j,2}, \dots, g_{j,k})^T = \mathbf{u}$, and
$\mathbf{d} + \sum_{j=1}^{i} y_j (g_{j,1}, g_{j,2}, \dots, g_{j,k})^T = \mathbf{v}$.
}

Figure 3: The Dual-TBT algorithm

the dual code $\mathcal{C}^{\perp}$. An important property of the dual trellis is that its state-complexity profile is identical to that for the primal trellis. Our main result is stated below.

Theorem 2.1 *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.*

There are several measures of minimality for tail-biting trellises [2]. If any of these definitions requires the trellis to be non-mergeable, it immediately follows from Theorem 2.1 that there exist under that definition of minimality, minimal trellises for a code and its dual with identical state-complexity profiles. Details are available at http://drona.csa.iisc.ernet.in/~priti/tech-reports.html.

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] 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.

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