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 \( C \) over \( \mathbb{F}_q \) with parameters \((n,k)\) with generator matrix \( G = \{g_1, \ldots, g_k\} \). The linear span of a codeword \( c \in C \) 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 \( c \). 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 \( C \) may be constructed from a generator matrix \( G \) whose rows have been partitioned into linear span rows \( G_i \) and circular span rows \( G_c \).

Define an \( n \times k \) matrix \( E^j = [e_1 \ e_2 \ \ldots \ \ e_k] \) s.t.

\[
e_i = \begin{cases} (0, 0, \ldots, g_{i+1}, g_{i+2}, \ldots, g_n) & \text{if } x \in \{g_i\}, \ g_i \in G_i \ with \ circular \ span \ \{a, b\} \\ 0 & \text{otherwise} \end{cases}
\]

Let \( j \) be the largest integer s.t. the first non-zero position of \( g_j \leq i \). Then the vertex set \( V_j \) at time index \( i \) is defined as follows:

\[
V_j = \left\{ (0, 0, \ldots, c_{i+1}, c_{i+2}, \ldots, c_n) + f : (c_1, \ldots, c_n) = (u_1, \ldots, u_j, 0, \ldots, 0)G, \ f = \sum_{j=0}^{n} v_j e_j \right\}
\]

where \((u_1, \ldots, u_j) \in \mathbb{F}_q^j\). There is an edge \( e \in E_i \) labeled \( g \) from a vertex \( v \in V_{j-1} \) to a vertex \( v' \in V_j \iff \exists \ a pair \ of \ codewords \ c = (c_1, \ldots, c_n), c' = (c'_1, \ldots, c'_n) \ s.t. \ (0, 0, \ldots, 0, c_{i+1}, \ldots, c_n) + f = v, \ (0, 0, \ldots, 0, c'_{i+1}, \ldots, c_n) + f = v' \) (where \( f = \sum_{j=0}^{n} v_j e_j, \ f' = \sum_{j=0}^{n} v'_j e_j \) s.t. \((u_1, \ldots, u_j, 0, \ldots, 0)G = c \ and \ (u'_1, \ldots, u'_j, 0, \ldots, 0)G = c'\), and either \( c = c' \) or \( \beta(c' - c) \) 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 \( T = (V, E, F_q) \) representing an \((n,k)\) linear code \( C \) 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] tail-biting trellis representing \( C \). 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 \( H = [h_1 \ h_2 \ \ldots \ h_n] \) 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 \( C \) given \( G \) and \( H \). Given an \((n,k)\) code \( C \) 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 \( C \) as input and computes a non-mergeable linear tail-biting trellis \( T^+ \) for
Algorithm BCJR-TBT
Input: The matrices $G$ and $H$.
Output: A non-mergeable linear tail-biting trellis $T = (V, E, \mathbb{F}_2)$ representing $C$.
Initialization: $G_{\text{id}} = G$. Let \{d\}_{x \in C}$ as follows:
\[
d_x = \begin{cases} \sum_{j=0}^{n} x_j h_j & \text{if } x \in (g_i), \text{ } g_i \text{ is a row of } G_x \text{ with circular span } (a, b) \\ 0 & \text{otherwise} \end{cases}
\]

**Step 1:** Construct the BCJR-labeled trellis for the subcode generated by the submatrix $G_x$ using the matrix $H$. Let $V_0, V_1, \ldots, V_n$ be the vertices sets created and $E_0, E_1, \ldots, E_n$ be the edge sets created. 

**Step 2:** for each row vector $g$ of $G_x$:
for each $x \in (g_i)$, $y$ in the row space of $G_{\text{id}}$:
\[
\{ \\
\text{let } \mathbf{x} \text{ denote the codeword } x + y. \\
\text{let } \mathbf{d}_x = \mathbf{x} + \mathbf{d}_y. \\
V_i = V_{i-1} \cup \{ \mathbf{d}_x \} \quad (1 \leq i \leq n), \\
V_f = V_n \cup \{ \mathbf{d}_x \}, \\
V_i = V_i \cup \{ \mathbf{d}_x + \sum_{j=i+1}^{n} y_j h_j \} \quad (1 \leq i \leq n). \\
\text{There is an edge } \epsilon = (u, v, w) \in E_i, u \in V_{i-1}, v \in V_i, 1 \leq i \leq n \\
\leftrightarrow \mathbf{d}_x + \sum_{j=0}^{i} y_j h_j = u \text{ and } \mathbf{d}_x + \sum_{j=0}^{i} y_j h_j = v. \\
\} \\
G_{\text{id}} = G_{\text{id}} + g.
\]

Figure 2: The BCJR-TBT algorithm

Algorithm Dual-TBT
Input: The matrices $G$ and $H$.
Output: A non-mergeable tail-biting trellis $T^\perp = (V, E, \mathbb{F}_2)$ representing $C^\perp$.
Initialization:
\[
\{ \\
\text{for each } y = (y_0, y_1, \ldots, y_n) \in C^\perp, \\
\} \\
\text{let } \mathbf{d} = (d_0, d_1, \ldots, d_n)^T \text{ s.t. } d_j = \begin{cases} \sum_{j=0}^{n} y_j h_j & \text{if } 1 \leq j \leq l \\ 0 & \text{otherwise} \end{cases} \\
\text{where } g_j \in G \text{ has circular span } [a, b]. \\
V_0 = V_n = V \cup \{ \mathbf{d} \}, \\
V_i = V_i \cup \{ \mathbf{d} + \sum_{j=0}^{i} y_j g_j \} \quad (1 \leq i \leq n), \\
\text{there is an edge } \epsilon = (u, v, w) \in E_i, u \in V_{i-1}, v \in V_i, 1 \leq i \leq n \quad \leftrightarrow \\
\mathbf{d} + \sum_{j=0}^{i} y_j g_j = u \text{ and } \\
\mathbf{d} + \sum_{j=0}^{i} y_j g_j = v.
\}
\]

Figure 3: The Dual-TBT algorithm

the dual code $C^\perp$. 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 $C$. 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]. Any one 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