A BCJR-like Labeling Algorithm for Tail-Biting Trellises

Aditya Nori and Priti Shankar
Department of Computer Science and Automation,
Indian Institute of Science,
Bangalore, India
{aditya, priti}@csm.iisc.ernet.in

I. INTRODUCTION

We describe two constructions for tail-biting trellises that are very similar to the well-known BCJR construction for conventional trellises. The constructions lead to a simple proof of the fact that there exist linear tail-biting trellises for a linear code and its dual, which have the same state-complexity profiles.

II. MAIN RESULTS

Let $C$ be an $(n, k)$ linear code over $\mathbb{F}_q$ with generator matrix $G = \{g_1, g_2, \ldots, g_k\}$. Given a codeword $c \in C$, the linear span of $c$, is the semi-open interval $[i, j]$ corresponding to the smallest closed interval $[i, j]$ with $i < j$, which contains all the non-zero positions of $c$. 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 $T$.

Koetter and Vardy [1] have shown that any linear trellises may be constructed from a generator matrix whose rows have been partitioned into linear span rows and circular span rows. Let $G_l$ and $G_c$ denote the sub-matrices of $G$ containing vectors of linear span and circular span respectively. Let $H = [\ h_1 \ h_2 \ \ldots \ h_n \ ]$ be the parity check matrix for the code. The algorithms BCJR-TBT and BCJR-TBT$^{-1}$ respectively, construct non-mergeable [3] linear tail-bitting trellises $T$ and $T^{-1}$ for $C$ and its dual $C^\perp$, given $G$ and $H$.

Algorithm BCJR-TBT

Input: The matrices $G$, $H$ and a span (linear or circular) associated with every $g \in G$.
Output: A non-mergeable linear tail-bitting trellis $T = (V, E, E_q)$ representing $C$.

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

\[
\text{if } x \in (g_i), \ g_i \in G_c, \ \text{with circular span } (a, b) \ \text{then } 0
\]

otherwise

Step 1: Construct the BCJR labeled trellis for the subcode generated by $G_l$, 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 $g$ of $G_c$

For each $x \in \langle g \rangle$, $y$ in the rowspace of $G_{int}$.

\[
d_x = \sum_{j=1}^{n} x_j h_j = u \quad \text{and} \quad d_x + \sum_{j=1}^{n} y_j h_j = v.
\]

$G_{int} = G_{int} + g$.

Algorithm BCJR-TBT$^{-1}$

Input: The matrices $G$ and $H$.
Output: A non-mergeable tail-bitting trellis $T^{-1} = (V, E, E_q)$ representing $C^\perp$.

Initialization: $V_i \cap [0 \leq i \leq n] = E_i \cap [1 \leq i \leq n] = \phi$.

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

\[
\text{let } d = \langle d_1, d_2, \ldots, d_n \rangle \text{ s.t.}
\]

\[
d_i = \begin{cases} 0 & \text{if } 1 \leq i \leq l \\ \sum_{j=n}^{i} y_j g_i, j & \text{otherwise} \end{cases}
\]

where $g_i \in G$ has circular span $(a, b)$.

$V_0 = V_n = V_0 \cup \{d\}$.

$V_i = V_i \cup \{d + \sum_{j=1}^{n} y_j (g_{j+1} g_{j+2} \ldots g_i)^T \}$.

There is an edge $e = (u, z, v) \in E_i$, $u \in V_{i-1}$, $v \in V_i$, $1 \leq i \leq n$, if

\[
d + \sum_{j=1}^{n} y_j (g_{j+1} g_{j+2} \ldots g_i)^T = u, \quad \text{and}
\]

\[
d + \sum_{j=1}^{n} y_j (g_{j+1} g_{j+2} \ldots g_i)^T = v.
\]

The preceding algorithms lead us to our main result.

Theorem 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 $T$.

Finally, as we know that for tail-biting trellises there are several measures of minimality [2], 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

