<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>On viterbi-like algorithms and their application to Reed–Muller codes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Tang, Yuansheng; Ling, San</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2004</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/9839">http://hdl.handle.net/10220/9839</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2004 Elsevier Inc. This is the author created version of a work that has been peer reviewed and accepted for publication by Journal of Complexity, Elsevier Inc. It incorporates referee’s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [<a href="http://dx.doi.org/10.1016/j.jco.2004.01.003">http://dx.doi.org/10.1016/j.jco.2004.01.003</a>].</td>
</tr>
</tbody>
</table>
On Viterbi-like algorithms and their application to Reed–Muller codes

Yuansheng Tang\textsuperscript{a,*,1} and San Ling\textsuperscript{b,2}

\textsuperscript{a}Department of Computer Science, School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore
\textsuperscript{b}Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543, Singapore

Received 3 March 2003; accepted 28 January 2004

Abstract

For a Viterbi-like algorithm over a sectionalized trellis of a linear block code, the decoding procedure consists of three parts: computing the metrics of the edges, selecting the survivor edge between each pair of adjacent vertices and determining the survivor path from the origin to each vertex. In this paper, some new methods for computing the metrics of the edges are proposed. Our method of “partition of index set” for computing the metrics is shown to be near-optimal. The proposed methods are then applied to Reed–Muller (RM) codes. For some RM codes, the computational complexity of decoding is significantly reduced in comparison to the best-known ones. For the RM codes, a direct method for constructing their trellis-oriented-generator-matrices is proposed and some shift invariances are deduced.

Keywords: Sectionalized trellis; Maximum-likelihood decoding; Generator matrix
1. Introduction

Among all trellis-based decoding algorithms for linear block codes, the Viterbi algorithm is the most commonly used one [4,5,7]. If the code length is $N$, the Viterbi algorithm consists of $N$ consecutive stages. At the $i$th stage, for each vertex $v$ at time $i$, among all paths starting at the origin and ending at $v$, the one with the maximum metric, called the survivor path at $v$, is determined. The decoding complexity of the Viterbi algorithm is shown in [5] to be the sum of the number $|E|$ of edges and the expansion index $|E| - |V| + 1$, where $V$ is the set of vertices. In general, the complexity of a trellis-based decoding algorithm can be reduced further if an appropriately sectionalized trellis is employed instead [1,2,4,7]. Clearly, the decoding complexity depends on the sectionalization profile. For some linear block codes, the optimal sectionalizations for some trellis-based decoding algorithms are determined in [4,7,9] further. In general, over a sectionalized trellis, two adjacent vertices are connected by more than one edge. For two adjacent vertices $v$ and $v'$, among the edges which connect $v$ and $v'$, the one with the maximum metric is called the survivor edge between $v$ and $v'$. For most of the decoding algorithms over sectionalized trellises, it is necessary to determine the survivor edge for each pair of adjacent vertices. Some methods for determining the survivor edges are proposed in [4,7]. In this paper, a decoding algorithm over a sectionalized trellis is addressed as a Viterbi-like algorithm if the Viterbi algorithm is applied to the reduced trellis obtained from the sectionalized trellis by deleting all the edges except the survivor edges. The decoding algorithms proposed in [4] are Viterbi-like algorithms, whereas the recursive maximum-likelihood decoding algorithm (RMLD) proposed in [7] is not. Indeed, the RMLD binarily and recursively partitions the time axis and recursively determines the survivor edges at each section $I$ from those at the subsections of $I$ [7,8]. In this paper, we are mainly concerned with the Viterbi-like algorithms. In general, the determination of the survivor edges at a given section can be partitioned into two further parts: computing the metrics of the edges and selecting the survivor edges by comparing the metrics of the edges which connect the same pair of vertices.

In this paper, we propose some new methods for computing the metrics of the edges in Section 2. Our method of “partition of index set” is shown to be a near-optimal method for computing the metrics of the edges. In Section 3, for the Reed–Muller (RM) codes we give a simple method for constructing the trellis-oriented-generator-matrices and show the invariance under a shift of some bits. To decode the RM codes, a Viterbi-like algorithm is proposed by combining the method of “partition of index set” for computing the metrics and a technique proposed in [7] for selecting the survivor edges. For some RM codes, the optimal sectionalizations of the proposed Viterbi-like algorithm are computed. For some cases, the decoding complexity is much smaller than the ones reported in [4].
2. Viterbi-like algorithms of binary linear block codes

2.1. Minimal sectionalized trellises of binary linear block codes

For $0 \leq i < i' \leq N$ and binary $N$-tuple $u = (u_1, u_2, \ldots, u_N)$, let $\rho_{i,i'}(u)$ denote the sub-tuple $(u_{i+1}, u_{i+2}, \ldots, u_{i'})$. Let $C$ be a binary linear $[N, k]$ block code. For $0 \leq i \leq i' \leq N$, let $C_{i,i'}$ denote the set of codewords $c \in C$ for which $\rho_{0,i}(c) = 0_i$ and $\rho_{i',N}(c) = 0_{N-i'}$, where $0_i$ is the all-zero $i$-tuple. $C_{i,i'}$ is a linear subcode of $C$. For simplicity, we use $C_{i,i'}^s$ to denote the shortened code $\rho_{i,i'}(C_{i,i'})$ of $C$.

For any binary $N$-tuple $u \neq 0_N$, let $L(u)$ and $R(u)$ denote the smallest and the largest integers $i$ with $u_i \neq 0$, respectively. A generator matrix $M$ of $C$ is called a trellis-oriented-generator-matrix (TOGM) [3,5] if any two distinct rows $u$ and $u'$ of $M$ satisfy $L(u) \neq L(u')$ and $R(u) \neq R(u')$. Clearly, for $0 \leq i < i' \leq N$, the subcode $C_{i,i'}$ is spanned by the rows $u$ of a TOGM which satisfy

$$L(u) > i \quad \text{and} \quad R(u) \leq i'.$$

Hence, the dimension, denoted $K_{i,i'}$, of $C_{i,i'}$ is equal to the number of rows $u$ of a TOGM which satisfy (1).

For a sectionalization profile $B = \{b_0, b_1, \ldots, b_L\}$ with $b_0 = 0 < b_1 < \cdots < b_L = N$, let $T_B = \left( \bigcup_{i=0}^{L} V_i, \bigcup_{i=1}^{L} E_i \right)$ denote the minimal sectionalized trellis (MST) of $C$, where

1. For $0 \leq i \leq L$, the vertex subset $V_i$ consists of $2^{k - K_{b_0, b_i} - K_{b_i, N}}$ vertices.

   The labels read from the paths passing through a given vertex $v \in V_i$ form a coset $H_{b_i}(v)$ of $C_{0,b} \oplus C_{b_i, N}$ in $C$. For distinct vertices $v$ and $v'$ in $V_i$, the cosets $H_{b_i}(v)$ and $H_{b_i}(v')$ are disjoint.

2. For $1 \leq i \leq L$, the edge subset $E_i$ consists of $2^{k - K_{b_{i-1}, b_i} - K_{b_i, N}}$ edges.

   The labels read from the paths passing through a given edge $e \in E_i$ form a coset of $C_{0,b_{i-1}} \oplus C_{b_i, N}$ in $C$.

3. For $1 \leq i \leq L$, $v_{i-1} \in V_{i-1}$ and $v_i \in V_i$,

   (a) The vertices $v_{i-1}$ and $v_i$ are connected by some edges in $E_i$ if and only if $H_{b_{i-1}}(v_{i-1}) \cap H_{b_i}(v_i) \neq \emptyset$.

   (b) If $H_{b_{i-1}}(v_{i-1}) \cap H_{b_i}(v_i) \neq \emptyset$, the vertices $v_{i-1}$ and $v_i$ are connected by just $2^{K_{b_{i-1}, b_i}}$ edges whose labels are the tuples in the set $\rho_{b_{i-1}, b_i}(H_{b_{i-1}}(v_{i-1}) \cap H_{b_i}(v_i))$, which is a coset of $C_{b_{i-1}, b_i}^s$ in $\rho_{b_{i-1}, b_i}(C)$.

2.2. Viterbi-like algorithms

Suppose that the code $C$ is used for error control over the additive white Gaussian noise (AWGN) channel with BPSK signaling. If codeword $c = (c_1, c_2, \ldots, c_N) \in C$ is transmitted, then the received sequence $r = (r_1, r_2, \ldots, r_N)$ is an $N$-tuple in $\mathbb{R}^N$ which can be written as $s(c) + w$, where $s(c) \triangleq (-1)^{c_1}, (-1)^{c_2}, \ldots, (-1)^{c_N}$ is the bipolar sequence corresponding to $c$ and the components of $w$ are independent Gaussian
random variables each with mean 0 and variance $N_0/2$. The distribution function of a component of $w$ is

$$Q(w) \triangleq \frac{1}{(\pi N_0)^{1/2}} \int_{-\infty}^{w} e^{-y^2/N_0} \, dy.$$  

(2)

The density function of the received sequence $r$ is

$$p(r|c) = \frac{1}{(\pi N_0)^{N/2}} e^{-d(r,s(c))^2/N_0},$$  

(3)

where $d(r,s(c))$ is the Euclidean distance between $r$ and $s(c)$. For each received sequence $r$, maximum-likelihood (ML) decoding outputs the most likely codeword $c_{\text{opt}}$ which satisfies

$$p(r|c_{\text{opt}}) = \max_{c \in C} p(r|c).$$  

(4)

For any codeword $c$, the sum

$$m_r(c) \triangleq \sum_{i=1}^{N} (-1)^{c_i} r_i$$  

(5)

is called the metric of $c$. Clearly, the codeword $c_{\text{opt}}$ maximizes the metric.

Suppose that the Viterbi algorithm is implemented over an MST $T_B$ of $C$, where $B = \{b_0, b_1, \ldots, b_L\}$ with $b_0 = 0 < b_1 < \cdots < b_L = N$. For every time the decoding is used, we relabel the metric attached to each path of $T_B$. For $0 \leq i < j \leq L$ and a path which starts from $V_i$ and ends at $V_j$, its metric is defined as the sum of the real numbers $r_{b_i+1}(-1)^{a_1}, r_{b_i+2}(-1)^{a_2}, \ldots, r_{b_j}(-1)^{a_{b_j-b_i}}$, where $a_1 a_2 \cdots a_{b_j-b_i}$ is the label read from the path. For any vertex $v$, among all of the paths which connect $v$ and the origin, one with the maximum metric is called a survivor path at $v$. For $1 \leq i \leq L$, let $\mathcal{P}_i$ denote the set of survivor paths at the vertices in $V_i$. Thus, $\mathcal{P}_L$ consists of a unique path which corresponds the codeword $c_{\text{opt}}$.

The Viterbi algorithm consists of $L$ stages. At the first stage, the survivor paths in $\mathcal{P}_1$ are determined by the edges in $E_1$. At the $i$th stage with $1 < i \leq L$, the survivor paths in $\mathcal{P}_i$ are determined by the edges in $E_i$ and the survivor paths in $\mathcal{P}_{i-1}$.

The decoding procedure of the Viterbi algorithm over $T_B$ can also be divided into two parts: computing the metrics of edges and decoding the relabeled trellis. Since the Viterbi algorithm itself does not imply any concrete ideas for performing these two parts of decoding procedure, it is better to address such a decoding algorithm over sectionalized trellises as a Viterbi-like algorithm. We will investigate some details of the foresaid two parts of decoding procedure for Viterbi-like algorithms in this section. For the evaluation of complexity, we count only the number of operations of real numbers and assume that negation is costless.

### 2.2.1. Computing the metrics of edges

Assume that the truncated code, denoted $E$, of $C$ in a given section is a binary linear $[n,k]$ block code. To compute the metrics of the codewords of $E$, Lafourcade and Vardy listed in [4] three methods, namely “exhaustive computation”, “Gray
codes” and “projection on the code”. The numbers of operations for these methods are, respectively, [4]

\[ M_c \triangleq (n - 1)2^{n-1}, \]

\[ M_G \triangleq \begin{cases} 2^{n-2} + 2^{n/2} + n - 4 & \text{if } 1_n \in E^\perp \text{ and } n \equiv 0 \mod 2, \\ 2^{n-1} + n - 2 & \text{otherwise,} \end{cases} \]

\[ M_c \triangleq (n - 1)M_{\text{neg}}(E), \]

where \(1_n\) is the all-one \(n\)-tuple and \(M_{\text{neg}}(E)\) is defined as

\[ M_{\text{neg}}(E) \triangleq \begin{cases} \frac{|E|}{2} & \text{if } 1_n \in E, \\ |E| & \text{otherwise}. \end{cases} \]

We note that, for the “Gray codes” method, to compute the metric of a codeword from the metric of another codeword, one needs to add or to subtract twice the metric of a bit, but the operations for doubling the metrics of the bits are neglected in the analysis of complexity given in [4]. However, since the number of such operations is bounded above by \(n\), for the sake of clarity, we assume that twice the metric of each bit is already known before decoding. Below we propose some new methods to compute the metrics.

**Minimal distance:** Clearly, if the all-one tuple \(1_n\) belongs to the code \(E\), then we can consider only a subcode \(E'\) of \(E\), which contains just one of each complementary pair of codewords of \(E\). Hence, without loss of generality, we assume that \(1_n \notin E\). Let \(d(E)\) denote the minimal Hamming distance of \(E\), and \(E_{\text{min}}\) the subcode of \(E\) spanned by the codewords of Hamming weight \(d(E)\). Clearly, there is at least one generator matrix of \(E_{\text{min}}\) whose rows are codewords of Hamming weight \(d(E)\). Thus the codewords of a coset of \(E_{\text{min}}\) can be arranged in an order similar to that of the Gray code such that the Hamming distance between two consecutive codewords is \(d(E)\). From the metric of one codeword, one can use \(d(E)\) operations of real numbers to get the metric of the next codeword. Hence, for any coset of \(E_{\text{min}}\) in \(E\), from the metric of a coset leader, the metrics of the remaining codewords in the coset can be computed with \((|E_{\text{min}}| - 1)d(E)\) operations. Since the number of cosets of \(E_{\text{min}}\) in \(E\) is \(|E|/|E_{\text{min}}|\), the metrics of the codewords in \(E\) can be computed with at most

\[ M_d \triangleq ((n - 1) + (|E_{\text{min}}| - 1)d(E)) \frac{|E|}{|E_{\text{min}}|} \]

\[ = d(E)|E| + (n - 1 - d(E)) \frac{|E|}{|E_{\text{min}}|} \]

operations of real numbers. Since \(M_{\text{neg}}(E) = |E|\) and \(|E_{\text{min}}| \geq 2\), if both the minimal Hamming distance \(d(E)\) and the dimension of \(E\) are much smaller than the length \(n\) of the code \(E\), the above method is much better than the methods listed in [4].

Furthermore, the above method can be used recursively in the following manner. Let \(E_{\text{min}}\) be a linear subcode of \(E\) of minimum dimension such that \(E = E_{\text{min}} \oplus E_{\text{min}}\).
Since the leaders of the cosets of $E_{\text{min}}$ in $E$ can be chosen as the codewords in $E_{\text{min}}$, the metrics of the codewords in $E_{\text{min}}$ can also be computed in the same way. Define 
\[ E^0 \triangleq E, \quad E_{\text{min}}^0 \triangleq (E^0)_{\text{min}} = E_{\text{min}} \quad \text{and for } i \geq 0, \]
\[ E^{i+1} \triangleq (E^i)_{\text{min}}, \quad E_{\text{min}}^{i+1} \triangleq (E_{\text{min}}^i)_{\text{min}}. \] (11)

Clearly,
\[ |E^i| = |(E^i)_{\text{min}}| \cdot |E^{i+1}|, \] (12)

and there are some integers $j$ such that $E^j = E_{\text{min}}^j$. Let $\mu$ denote the smallest of such numbers $j$. Then, $E^{\mu+1}$ consists of the all-zero $n$-tuple $0_n$ only and the number of operations for computing the metrics of the codewords in $E^\mu$ is
\[ M_{d,\mu} \triangleq n - 1 + d(E^\mu) \cdot (|E^\mu| - 1). \] (13)

For $i = \mu - 1, \mu - 2, \ldots, 0$, the number of operations for computing the metrics of the codewords in a coset of $E_{\text{min}}^i$ in $E^i$ from a coset leader is
\[ M_{d,i} \triangleq d(E^i) \cdot (|E_{\text{min}}^i| - 1). \] (14)

From (12) to (14), the number of operations needed for computing the metrics of the codewords of $E$ is at most
\[ M_d' \triangleq M_{d,\mu} + \sum_{i=0}^{\mu-1} M_{d,i} \cdot |E^{i+1}| = n - 1 + \sum_{i=0}^{\mu} d(E^i)(|E^i| - |E^{i+1}|). \] (15)

In general, the number $M_d'$ depends on the choice of the codes $E^i$ for $i \geq 1$. For many cases, because $|E^0| = |E| \gg |E^i|$ for $i \geq 1$, both $M_d$ and $M_d'$ are dominated by $d(E)|E|$.

**Example 3.1.** Assume that $E$ is the binary linear $[7, 4, 2]$ block code generated by the following matrix:
\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix},
\] (16)

Then, the generator matrices of $E_{\text{min}}$ and $E_{\text{min}}^0$ can be chosen as
\[
\begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix},
\] (17)

respectively. $E_{\text{min}}^0$ is a binary linear $[7, 2, 3]$ block code. Thus, $M_d = 2 \times 2^4 + (7 - 1 - 2)2^2 = 48$ and $M_d' = (7 - 1) + 3 \times (2^2 - 1) + 2 \times (2^4 - 2^2) = 39$.

**Partition of index set:** Assume that $\{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ is a partition of the index set $\{1, 2, \ldots, n\}$. Let $\mathcal{P}(\gamma_i, E)$ the truncated code of $E$ in $\gamma_i$. Then, the metrics of the
codewords of $E$ can be computed in the following two steps:

1. Compute the metrics of the codewords in the truncated codes $\mathcal{P}(g_i, E)$ for $i = 1, 2, \ldots, m$.
2. For every codeword $c$ in $E$, if the metric of the complementary codeword of $c$ is already known, set the metric of $c$ to be the negation of the metric of the complementary codeword. Otherwise, set the metric of $c$ to be the sum of the metrics of its truncated codewords in the truncated codes.

The number of operations needed in the second step of the above method is $(m - 1)M_{\text{neg}}(E)$. Let $M_{\text{min}}(E)$ denote the minimum number of operations for computing the metrics of the codewords of $E$. Then, we have the following lemma which gives an upper bound for $M_{\text{min}}(E)$.

**Lemma 1.** For any partition $\Gamma = \{g_1, g_2, \ldots, g_m\}$ of $\{1, 2, \ldots, n\}$,

$$M_{\text{min}}(E) \leq \sum_{i=1}^{m} M_{\text{min}}(\mathcal{P}(g_i, E)) + (m - 1)M_{\text{neg}}(E).$$  \hspace{1cm} (18)

We note that the method of index partition can be used recursively. Namely, the metrics of the codewords in the truncated codes $\mathcal{P}(g_i, E)$ can also be computed by the method of index partition. Since the dimensions of the truncated codes $\mathcal{P}(g_i, E)$ are much smaller than that of the original code $E$ for many cases, the recursive method may be much more efficient. Obviously, the complexity is dependent on the partition. In general, it is a good method to partition the index set into two subsets such that the larger of the dimensions of the two truncated codes is minimized. We will see in the next section that, for any RM code, the number of operations for the recursive method with a natural partition of the index set is very close to the lower bound given in the following theorem.

**Theorem 1.** For any binary linear $[n, k]$ block code $E$,

$$n - 2 + M_{\text{neg}}(E) \leq M_{\text{min}}(E) \leq \left\lfloor \frac{n}{k} \right\rfloor \left( 2^k + 2^{k-1} + k - 2 \right) - 2^k.$$  \hspace{1cm} (19)

**Proof.** Consider an arbitrary method for computing the metrics of the codewords in $E$. Before outputting the metric of the first codeword, it needs at least $n - 1$ operations. For any other codeword which is not the complementary codeword of any codeword with known metric, it needs at least one operation to output its metric. Hence, we have the following lower bound:

$$M_{\text{min}}(E) \geq n - 1 + (M_{\text{neg}}(E) - 1) = n - 2 + M_{\text{neg}}(E).$$  \hspace{1cm} (20)

To show the upper bound in (19), we partition the index set into $\{g_1, g_2, \ldots, g_m\}$ with $|g_i| = k$ for $1 \leq i \leq m - 1$, where $k$ is the dimension of the code $E$. Then, $m = \left\lfloor \frac{n}{k} \right\rfloor$ and $0 \leq |g_m| < k$. If we use the method of “Gray codes” to compute the metrics of the codewords of the truncated codes $\mathcal{P}(g_i, E)$, then from (18) we get the
following upper bound

\[ M_{\text{min}}(E) \leq \left\lceil \frac{n}{k} \right\rceil (2^{k-1} + k - 2) + \left( \left\lceil \frac{n}{k} \right\rceil - 1 \right) 2^k \]

\[ = \left\lceil \frac{n}{k} \right\rceil (2^k + 2^{k-1} + k - 2) - 2^k. \quad (21) \]

Then, (19) follows from (20) and (21). □

**Example 3.2.** We compute the metrics of codewords in the code \( E \) given in Example 3.1 by the method of “partition of index set”. Assume the index set \([1, 2, \ldots, 7]\) is partitioned into \( \gamma_1 = \{1, 2, 3, 4\} \) and \( \gamma_2 = \{5, 6, 7\} \). The metrics of the codewords in the truncated codes \( P(E, \gamma_1) \) and \( P(E, \gamma_2) \) are computed by the method of “Gray code”. Then, the metrics of codewords in the code \( E \) can be computed within \((2^{4-1} + 4 - 2) + (2^{3-1} + 3 - 2) + 2^4 = 31\) operations.

### 2.2.2. Decoding the relabeled trellis

Assume that the survivor paths in \( P_{i-1} \) and the metrics of the edges \( E_i \) are known. Consider determining the survivor paths in \( P_i \). Clearly, the concatenation of a path \( \delta \in P_{i-1} \) and an edge \( e \in E_i \) which connects \( \delta \) is a candidate of the survivor path at the vertex at which \( e \) ends. The number of such candidate paths is equal to the number of edges in \( E_i \). One needs \(|E_i|\) additions to obtain the metrics for these candidate paths. For a vertex \( v \in V_i \), if there are \( \tau \) candidate paths ending at \( v \), the survivor path at \( v \) is then determined after \( \tau - 1 \) comparisons. The total number of such comparisons is equal to \(|E_i| - |V_i|\). Thus, the number of operations for determining the survivor paths in \( P_i \) is [4]

\[ D_{\text{min},i} \leq D_{\text{v},i} \triangleq \begin{cases} 2|E_i| - |V_i| & \text{if } i > 1, \\ |E_i| - |V_i| & \text{if } i = 1, \end{cases} \quad (22) \]

where, for the case \( i = 1 \), no addition is needed in practical computation.

For any pair of adjacent vertices \( v_{i-1} \in V_{i-1} \) and \( v_i \in V_i \), they are connected by \(|C_{b_{i-1},b_i}|\) parallel edges whose labels form a coset of \( C_{b_{i-1},b_i}^s \) in \( \rho_{b_{i-1},b_i}(C) \). An edge with the maximum metric among such parallel edges is called a survivor edge between \( v_{i-1} \) and \( v_i \). Clearly, the candidates of the survivor paths in \( P_i \) can be limited to the concatenations of the survivor paths in \( P_{i-1} \) and the survivor edges. Since the number of survivor edges is \(|E_i|/|C_{b_{i-1},b_i}|\), the number of additions can be reduced to \(|E_i|/|C_{b_{i-1},b_i}|\), the number of comparisons for determining the survivor edges is \((M_{\text{neg}}(C_{b_{i-1},b_i}) - 1)|E_i|/|C_{b_{i-1},b_i}|\) and the number of comparisons for determining the survivor paths in \( P_i \) from the concatenations of the survivor paths in \( P_{i-1} \) and the survivor edges is \(|E_i|/|C_{b_{i-1},b_i}| - |V_i|\). Thus [4]

\[ D_{\text{min},i} \leq D_{i,i} \triangleq \begin{cases} \frac{|E_i|(M_{\text{neg}}(C_{b_{i-1},b_i}) + 1)}{|C_{b_{i-1},b_i}|} - |V_i| & \text{if } i > 1, \\ \frac{|E_i|M_{\text{neg}}(C_{b_{i-1},b_i})}{|C_{b_{i},b_i}|} - |V_i| & \text{if } i = 1. \end{cases} \quad (23) \]
It is noticed in [7] that some pairs of adjacent vertices may be connected by some edges with the same set of labels and thus the survivor edges between each of such pairs of adjacent vertices should also have the same label. Hence, it is not necessary to use $M_{\text{neg}}(C_{\text{bi},b_1}) - 1$ comparisons to determine the survivor edge for each pair of adjacent vertices $v_{i-1} \in V_{i-1}$ and $v_i \in V_i$. Indeed, it is enough to determine only the label with the maximum metric in each coset of $C_{\text{bi},b_1}$ in $\rho_{\text{bi},b_1}(C)$. Since the number of cosets of $C_{\text{bi},b_1}$ in $\rho_{\text{bi},b_1}(C)$ is $|\rho_{\text{bi},b_1}(C)|/|C_{\text{bi},b_1}|$, which is usually much smaller than the number of the survivor edges, we have (cf. [7])

$$D_{\text{min},i} \leq D_{p,i} \triangleq \begin{cases} \frac{|\rho_{\text{bi},b_1}(C)|(M_{\text{neg}}(C_{\text{bi},b_1}) - 1)}{|C_{\text{bi},b_1}|} + \frac{2|E_i| - |V_i|}{|C_{\text{bi},b_1}|} & \text{if } i > 1, \\ \frac{|\rho_{\text{bi},b_1}(C)|(M_{\text{neg}}(C_{\text{bi},b_1}) - 1)}{|C_{\text{bi},b_1}|} & \text{if } i = 1. \end{cases} \quad (24)$$

To compute $D_{p,i}$, we need to compute the dimension of $\rho_{\text{bi},b_1}(C)$, which cannot be read directly from a TOGM of $C$ in general. However, since $\rho_{\text{bi},b_1}(C)$ is the dual code of $(C^\perp)_{\text{bi},b_1}$ (cf. [1]), i.e.,

$$\left(\rho_{\text{bi},b_1}(C)\right)^\perp = (C^\perp)_{\text{bi},b_1}, \quad (25)$$

the dimension of $\rho_{\text{bi},b_1}(C)$ can be read simply from a TOGM of the dual code $C^\perp$ of $C$.

### 3. Application of the Viterbi algorithm to RM codes

#### 3.1. TOGM of RM codes

For positive integers $m$ and $j$ with $1 \leq j \leq m$, let $v_{j,m}$ denote the binary $2^m$-tuple $(v_1, v_2, \ldots, v_{2^m})$ which satisfies

$$v_i = \begin{cases} 0, & \text{if } \left\lfloor (i-1)/2^{m-j} \right\rfloor \text{ is even,} \\ 1, & \text{if } \left\lfloor (i-1)/2^{m-j} \right\rfloor \text{ is odd,} \end{cases} \quad i = 1, 2, \ldots, 2^m. \quad (26)$$

$v_{j,m}$ is the concatenation of $2^{j-1}$ identical $2^{m-j+1}$-tuples. In each such $2^{m-j+1}$-tuple, the first $2^{m-j}$ bits are zeros and the rest are ones, i.e.,

$$v_{j,m} = (0_{2^{m-j}}, 1_{2^{m-j}}, 0_{2^{m-j}}, 1_{2^{m-j}}, \ldots, 0_{2^{m-j}}, 1_{2^{m-j}}). \quad (27)$$

For any two binary $n$-tuples $u = (u_1, u_2, \ldots, u_n)$ and $u' = (u'_1, u'_2, \ldots, u'_n)$, their Boolean product, denoted $uu'$, is defined as the $n$-tuple $(u_1u'_1, u_2u'_2, \ldots, u_nu'_n)$.

For $1 \leq j \leq m$, let $\mathcal{V}_{j,m}$ denote the set of Boolean products of precisely $j$ distinct $2^m$-tuples in the set $\{v_{1,m}, v_{2,m}, \ldots, v_{m,m}\}$. For an integer $r$ with $0 \leq r \leq m$, the set $\mathcal{V}_{r,m} \triangleq \bigcup_{0 \leq j \leq r} \mathcal{V}_{j,m}$ of binary $2^m$-tuples is a basis of the RM code RM$(r, m)$ with the natural lexicographic coordinate ordering, where $\mathcal{V}_{0,0} \triangleq \{1_{2^m}\}$. The dimension of
Theorem 2. The matrix whose rows are the tuples in the following set:

\[ G_{r,m} \triangleq \{ g(\sigma, l) : \sigma \in \mathcal{O}_j^m, \quad 0 < l < 2^{r-j+1}, 0 \leq j \leq r \} \]  

is a TOGM of the RM code RM(r, m).
3.2. Shift invariance of RM codes

For convenience, we define \( \text{RM}(r, m) \) as \( \text{RM}(m, m) \), the set of all binary \( 2^m \)-tuples, for \( r \geq m \), and as the set consisting of only the all-zero tuple \( 0_{2^m} \), for \( r < 0 \), respectively. From the definition of RM codes and

\[
\rho_{0,2^m-1}(v_{j,m}) = \begin{cases} v_{j-1,m-1} & \text{if } 1 < j \leq m, \\ 0_{2^m-1} & \text{if } j = 1, \end{cases}
\]

we see easily that

\[
\rho_{0,2^m-1}(\text{RM}(r, m)) = \text{RM}(r, m - 1).
\]

One can also easily deduce

\[
S_{2^m-2}(v_{j,m}) = \begin{cases} v_{j,m} & \text{if } 2 < j \leq m, \\ v_{2,m} + 1_{2^m} & \text{if } j = 2, \\ v_{1,m} + v_{2,m} + 1_{2^m} & \text{if } j = 1. \end{cases}
\]

Thus, from the definition of RM codes and (36), we see that (cf. [6])

\[
S_{2^m-2}(\text{RM}(r, m)) = \text{RM}(r, m),
\]

that is, the RM code \( \text{RM}(r, m) \) is invariant under a shift of \( 2^{m-2} \) bits.

**Lemma 2.** For \( 0 < i < m \) and \( 0 < l < 2^{i+1} \),

\[
\rho_{(l-1)2^{m-i-1},(l+1)2^{m-i-1}}(\text{RM}(r, m)) = \text{RM}(r, m - i). \tag{38}
\]

**Proof.** If \( i = 1 \), (38) becomes

\[
\rho_{(l-1)2^{m-2},(l+1)2^{m-2}}(\text{RM}(r, m)) = \text{RM}(r, m - 1), \text{ for } 0 < l < 4,
\]

which is a direct application of (35) and (37).

Assume that \( j \) is an integer with \( 1 < j < m \) and (38) is valid for \( i = j - 1 \), i.e.,

\[
\rho_{(l-1)2^{m-j-1},(l+1)2^{m-j-1}}(\text{RM}(r, m)) = \text{RM}(r, m - j + 1), \text{ for } 0 < l < 2^j. \tag{40}
\]

Consider the case \( i = j \). From (35) and (40),

\[
\begin{align*}
\rho_{(l-1)2^{m-j-1},(l+1)2^{m-j-1}}(\text{RM}(r, m)) \\
&= \rho_{0,2^m-1}(\rho_{(l-1)2^{m-j-1},(l+1)2^{m-j}}(\text{RM}(r, m))) \\
&= \rho_{0,2^m-1}(\text{RM}(r, m - j + 1)) \\
&= \text{RM}(r, m - j - 1),
\end{align*}
\]

Therefore, (38) holds for \( i = j \) as well.
which means that (38) is valid for the case \( i = j \) and \( l = 1 \). Suppose \( 1 < l < 2^{i+1} \) below. Let \( l_1 = \lfloor l/2 \rfloor \) and \( l_2 = l - 2l_1 + 2 \). Then, \( 0 < l_1 < 2^i \) and \( 2 \leq l_2 \leq 3 \) and

\[
(l - 1)2^{m-j-1} = (l_1 - 1)2^{m-j} + (l_2 - 1)2^{m-j-1} > (l_1 - 1)2^{m-j},
\]

(42)

\[
(l + 1)2^{m-j-1} = (l_1 - 1)2^{m-j} + (l_2 + 1)2^{m-j-1} \leq (l_1 + 1)2^{m-j}.
\]

(43)

Hence, from (39), (40), (42) and (43),

\[
\rho_{(l-1)2^{m-j-1},(l+1)2^{m-j-1}}(\text{RM}(r, m))
\]

\[
= \rho_{(l_1-1)2^{m-j},(l_1+1)2^{m-j-1}}(\rho_{(l_1-1)2^{m-j},(l_1+1)2^{m-j}}(\text{RM}(r, m)))
\]

\[
= \rho_{(l_1-1)2^{m-j},(l_1+1)2^{m-j-1}}(\text{RM}(r, m - j + 1))
\]

\[
= \text{RM}(r, m - j), \text{ for } 1 < l < 2^{i+1}.
\]

(44)

Then, from (41) and (44), equality (38) is valid for \( i = j \) and \( 0 < l < 2^{i+1} \). The proof of (38) is complete. \( \square \)

From Lemma 2, we have the following corollary.

**Corollary 1.** For \( 0 < i < m \) and \( 0 < l < 2^{i+1} \),

\[
\text{RM}(r, m)_{(l-1)2^{m-i-1},(l+1)2^{m-i-1}} = \text{RM}(r - i, m - i).
\]

(45)

**Proof.** It is well known that \( \text{RM}(r, m) \) is the dual code of \( \text{RM}(m - r - 1, m) \) (cf. [1]), i.e.,

\[
\text{RM}(r, m) = (\text{RM}(m - r - 1, m))^\perp.
\]

(46)

For \( 0 < i < m \) and \( 0 < l < 2^{i+1} \), from (25), (38) and (46),

\[
\text{RM}(r, m)_{(l-1)2^{m-i-1},(l+1)2^{m-i-1}}
\]

\[
= (\rho_{(l-1)2^{m-i-1},(l+1)2^{m-i-1}}(\text{RM}(m - r - 1, m)))^\perp
\]

\[
= (\text{RM}(m - r - 1, m - i))^\perp
\]

\[
= \text{RM}(r - i, m - i). \quad \square
\]

(47)

Clearly, for odd integer \( l = 2l' + 1 \), equalities (38) and (45) can be rewritten as

\[
\rho_{p2^{m-i},(p+1)2^{m-i}}(\text{RM}(r, m)) = \text{RM}(r, m - i),
\]

(48)

\[
\text{RM}(r, m)_{p2^{m-i},(p+1)2^{m-i}} = \text{RM}(r - i, m - i),
\]

(49)
both (48) and (49) can be found in [8,10]. However, for even integer \( l \), (38) and (45) are new, as far as we know.

From (38), the metrics of the codewords in \( \text{RM}(r,m) \) can be computed by the combination of the methods of “partition of index set” and “Gray codes” within

\[
M_{r,m} \triangleq \sum_{i=1}^{m-r} 2^{i-1} M_{\text{neg}}(\text{RM}(r, m - i + 1)) + 2^{m-r}(2^{2^{i-1} + 2^{r} - 2})
\]

\[
= \sum_{i=1}^{m-r} 2^{i-2 + \Sigma_{j=0}^{i} (m-j+1)} + 2^{m-r}(2^{2^{i-1} + 2^{r} - 2}) \tag{50}
\]

operations. We note that \( M_{1,m} = m^{2m} \) and \( M_{r,m} \) is dominated by \( M_{\text{neg}}(\text{RM}(r,m)) \) if \( r > 1 \). Hence, from Theorem 1, the combination of the methods of “partition of index set” and “Gray codes” is a near-optimum method for computing the metrics of the codewords of \( \text{RM} \) codes.

3.3. Complexity of the Viterbi algorithm for \( \text{RM} \) codes

Suppose that the Viterbi algorithm is implemented over a minimal s-trellis \( T_B = (V, E, B) \) of \( \text{RM}(r, m) \), where \( 1 \leq r \leq m - 2 \) and \( B = \{b_0, b_1, \ldots, b_L\} \).

Since the shortened code \( \text{RM}(m - r - 1, m)_{b,b'} \) is the dual code of the truncated code \( \rho_{b,b'}(\text{RM}(r, m)) \) for any \( 0 \leq b < b' \leq 2^m \), the dimension of \( \rho_{b,b'}(\text{RM}(r, m)) \) can be easily read from the TOGM of \( \text{RM}(m - r - 1, m) \). We compute the metrics of tuples in \( \rho_{b,b'}(\text{RM}(r, m)) \) by the method of “partition of index set” recursively in the following manner:

1. If the dimension of \( \rho_{b,b'}(\text{RM}(r, m)) \) is \( b' - b \), i.e., \( \rho_{b,b'}(\text{RM}(r, m)) \) is the set of all binary \((b' - b)\)-tuples, compute the metrics by the method of “Gray codes”.
2. If the dimension of \( \rho_{b,b'}(\text{RM}(r, m)) \) is less than \( b' - b \), partition the index set \( \{b + 1, b + 2, \ldots, b'\} \) into two subsets \( \{b + 1, b + 2, \ldots, b''\} \) and \( \{b'' + 1, b'' + 2, \ldots, b'\} \) such that the sum of the dimensions of \( \rho_{b,b'}(\text{RM}(r, m)) \) and \( \rho_{b'',b'}(\text{RM}(r, m)) \) achieves the minimum.

Let \( M_{p,i} \) denote the number of operations for computing the metrics of the edges in \( E_i \) by this method. If \( b_{l} - b_{l-1} = 2^{i} \) and \( 2^{i-1} | b_{l-1} \), then \( M_{p,i} = M_{r,s} \).

The following lemma is used in the computation of the numbers \( D_{p,i} \) for \( C = \text{RM}(r, m) \).

**Lemma 3.** For \( 0 \leq b < b' \leq 2^m \), the tuple \( 1_{b'-b} \) belongs to \( \text{RM}(r, m)_{b,b'}^{s} \) if and only if \( 2^{m-r-1}|b \) and \( 2^{m-r}|(b' - b) \).

**Proof.** If-part: For \( 0 \leq i < (b' - b)/2^{m-r} \), from (45) the tuple \((0_{b+i2^{m-r}}, 1_{2^{m-r}}, 0_{2^{m-b-(i+1)2^{m-r}}}) \) belongs to \( \text{RM}(r, m)_{b+i2^{m-r},b+(i+1)2^{m-r}} \subset \text{RM}(r, m)_{b,b'} \) and thus \( \text{RM}(r, m)_{b,b'} \) contains the tuple \((0_{b}, 1_{b'-b}, 0_{2^{m-b'}}) \), i.e., \( 1_{b'-b} \in \text{RM}(r, m)_{b,b'}^{s} \).
Only-if-part: Since $\rho_{b,b'}(\text{RM}(m - r - 1, m))$ is the dual code of $\text{RM}(r, m)^s_{b,b'}$, from $1_{b'} \in \text{RM}(r, m)^s_{b,b'}$ we see that $\rho_{b,b'}(\text{RM}(m - r - 1, m))$ contains only the tuples of even weights. Assume $2^s(b - b')$ and $2^t(b - m)$. If $s = 0$, then $\rho_{b,b'}(1_{b'}^m) \in \rho_{b,b'}(\text{RM}(m - r - 1, m))$ is a tuple of odd weight. Hence, $s \geq m - r$. If $0 \leq t < m - r - 1$, then $\rho_{b,b'}(v_{m-t, m}) \in \rho_{b,b'}(\text{RM}(m - r - 1, m))$ is a tuple of odd weight. Hence, $t \geq m - r - 1$. □

In general, the computational complexity is dependent on the sectionalization profile $B$. For some RM codes $\text{RM}(r, m)$, the optimal sectionalization $B_{r,m}$ under the optimality criterion $\sum_{i=1}^L (M_{P,i} + D_{P,i})$ are computed. Some computational results are shown in Table 1. The computational complexities of decoding for some RM codes, such as $\text{RM}(1,4)$, $\text{RM}(1,5)$, $\text{RM}(1,6)$, $\text{RM}(2,6)$ and $\text{RM}(3,6)$, are much smaller than the ones shown in [4].

4. Conclusion

In this paper, we investigate the Viterbi-like algorithms over sectionalized trellises of binary linear block codes. Some new methods for computing the metrics are proposed. The method of “partition of index set” for the computation of metrics is shown to be near-optimal. For the RM codes, a direct method for constructing their trellis-oriented-generator-matrices is proposed and some shift invariances are deduced. By combining our methods for computing the metrics and a technique of [7], a Viterbi-like algorithm is proposed to decode RM codes. For some RM codes, the proposed Viterbi-like algorithm performs much better than the ones proposed in [4].

Acknowledgments

The authors thank the anonymous reviewers for their helpful and valuable comments.

Appendix A. Proof of Theorem 2

To prove Theorem 2, we show some lemmas first.

Lemma A.1. (1) For binary $n$-tuples $u$, $u'$ and monomials $\sigma = q_{a_1} \cdots q_{a_i}$, $\sigma' = q_{a_1} \cdots q_{a_i} \in 2$ of the same degree,

\[
(\sigma(u))(\sigma'(u')) = q_{a_1d_1}q_{a_2d_2} \cdots q_{a_id_i}(uu').
\]  

(A.1)
Thus, one can get (A.1) by repeatedly applying (A.3).

Proof. (1) For $a, a' \in \{0, 1\}$ and binary $n$-tuples $u$ and $u'$, from the definitions of $q_0$ and $q_1$, we see easily that

$$q_a(\mathbf{u})q_{a'}(\mathbf{u'}) = q_{aa'}(\mathbf{uu'}) = q_{aa'}(\mathbf{uu'}).$$

Thus, one can get (A.1) by repeatedly applying (A.3).
(2) It is obvious that, for $1 \leq i \leq m$,
\begin{equation}
\mathbf{v}_{i,m} = q_1^{j-1} q_0 q_1^{m-i}(\mathbf{1}_1).
\end{equation}
Hence, (A.2) follows from (A.1), (A.4) and $\mathbf{1}_1 \mathbf{1}_1 = \mathbf{1}_1$.  \qed 

Let $\mathcal{L}_{r,m}$ denote the set $\{L(\mathbf{u}) : \mathbf{u} \in \mathcal{Y}^{-\infty}_{r,m}\}$. Since, for $1 \leq i_1 < i_2 < \cdots < i_j \leq m$,
\begin{equation}
L(\mathbf{v}_{i_1,m} \mathbf{v}_{i_2,m} \cdots \mathbf{v}_{i_j,m}) = 2^{m-i_1} + 2^{m-i_2} + \cdots + 2^{m-i_j} + 1,
\end{equation}
it follows that
\begin{equation}
|\mathcal{L}_{r,m}| = |\mathcal{Y}^{-\infty}_{r,m}|.
\end{equation}
For $\sigma \in \mathbb{Z}_{2}^{r,m}$, let
\begin{equation}
v(\sigma) \triangleq q_1^{r-j+1} \sigma(\mathbf{1}_1).
\end{equation}
For $0 \leq j \leq r$, let $\mathcal{Y}_{r,m}^{j} \triangleq \{ (\sigma, l) : \sigma \in \mathbb{Z}_{2}^{r,m}, 0 < l < 2^{r-j+1} \}$. Clearly,
\begin{equation}
L(g(\sigma, l)) = (l - 1) 2^{m-r+j-1} + L(v(\sigma)), \text{ for } (\sigma, l) \in \mathcal{Y}_{r,m}^{j}.
\end{equation}
For $(\sigma, l) \in \mathcal{Y}_{r,m}^{j}$, since the integer $l - 1$ can be expressed as the sum of at most $r - j$ integers of the form $2^{r-j-i-1}$ with $1 \leq i \leq r + 1$ and, from (A.2), the tuple $v(\sigma)$ is the Boolean product of $j$ distinct tuples in $v_{r-j+3,m}, v_{r-j+4,m}, \ldots, v_{m,m}$, it follows from (A.5) and (A.8) that
\begin{equation}
L(g(\sigma, l)) \in \mathcal{L}_{r,m}, \text{ for } (\sigma, l) \in \mathcal{Y}_{r,m}^{j}.
\end{equation}

**Lemma A.2.** Assume that $w = 2^{m-i_1} + 2^{m-i_2} + \cdots + 2^{m-i_{j_0}} + 1$ with $1 \leq i_1 < i_2 < \cdots < i_{j_0} \leq m$ and $0 \leq j_0 \leq r$. Then, $L(g(\sigma, l)) = w$ for some pair $(\sigma, l) \in \mathcal{Y}_{r,m}^{j}$ if and only if
\begin{align}
0 &\leq j \leq j_0, \quad \text{(A.10)} \\
i_{j_0-j} &\leq r - j + 1, \quad \text{if } j < j_0, \quad \text{(A.11)} \\
i_{j_0-j+1} &\geq r - j + 3, \quad \text{if } j > 0. \quad \text{(A.12)}
\end{align}

**Proof.** If-part: Assume that (A.10) to (A.12) are valid. Define
\begin{align}
\sigma &\triangleq \left\{ \begin{array}{ll}
q_1^{i_{j_0-j+1}-1-(r+j+1)} q_0 q_1^{i_{j_0-j+2}+1} \cdots q_0 q_1^{i_{j_0-1}-1} q_0 q_1^{m-i_{j_0}} & \text{if } j > 0, \\
q_1^{r-j-1} & \text{otherwise},
\end{array} \right. \quad \text{(A.13)} \\
l &\triangleq \left\{ \begin{array}{ll}
2^{r-j+1-j_1} + 2^{r-j+1-j_2} + \cdots + 2^{r-j+1-j_{0-j}} + 1 & \text{if } j < j_0, \\
1 & \text{otherwise}. \end{array} \right. \quad \text{(A.14)}
\end{align}

From (A.12), the monomial $\sigma$ defined by (A.13) is led by $q_1$. Clearly, the 0-degree and 1-degree of $\sigma$ are $j$ and $m - r - 1$, respectively. From (A.11) and $j_0 - j \leq r - j$, the
integer $l$ defined by (A.14) satisfies $0 < l < 2^{r-j+1}$. Hence, $(\sigma, l) \in \mathcal{W}^{r,m}$. From (A.2) and (A.13), we see that
\begin{equation}
  v(\sigma) = v_{i_0-j+1,m} v_{i_0-j+2,m} \cdots v_{i_0,m},
\end{equation}
(A.15)
\begin{equation}
  L(v(\sigma)) = 2^{m-i_0} + 2^{m-i_0-1} + \cdots + 2^{m-i_0} + 1.
\end{equation}
(A.16)

It follows from (A.8), (A.14) and (A.16) that $L(g(\sigma, l)) = w$.

**Only-if part:** Assume that $L(g(\sigma, l)) = w$ for $(\sigma, l) \in \mathcal{W}^{r,m}$. Since $v(\sigma)$ is the Boolean product of $j$ different tuples in $v_{r-j+3,m}, v_{r-j+4,m}, \ldots, v_{m,m}$, there must exist $j$ different integers between $r-j+3$ and $m$, say $r-j+3 \leq i'_1 < i'_2 < \cdots < i'_j \leq m$, such that
\begin{equation}
  L(v(\sigma)) = 2^{m-i'_1} + 2^{m-i'_2} + \cdots + 2^{m-i'_j} + 1.
\end{equation}
(A.17)

Since $1 \leq l < 2^{r-j+1}$, there are at most $r-j$ different integers between 1 and $r-j+1$, say $1 \leq i'_1 < i'_2 < \cdots < i'_m \leq r-j+1$ with $0 \leq j_1 \leq r-j$, such that
\begin{equation}
  l - 1 = 2^{r-j-1-i'_1} + 2^{r-j-1-i'_2} + \cdots + 2^{r-j-1-i'_m}.
\end{equation}
(A.18)

Thus, from $L(g(\sigma, l)) = (l - 1)2^{m-r+j-1} + L(v(\sigma)) = w$,
\begin{equation}
  2^{m-i'_1} + 2^{m-i'_2} + \cdots + 2^{m-i'_m} + 2^{m-i'_1} + 2^{m-i'_2} + \cdots + 2^{m-i'_j}
  = 2^{m-i_0} + 2^{m-i_2} + \cdots + 2^{m-i_0}.
\end{equation}
(A.19)

From (A.19), $1 \leq i'_1 < i'_2 < \cdots < i'_m \leq r-j+1 < r-j+3 \leq i'_1 < i'_2 < \cdots < i'_j \leq m$ and $1 \leq i_1 < i_2 < \cdots < i_{j_0} \leq m$, we get
\begin{equation}
  j_0 = j_1 + j,
\end{equation}
(A.20)
\begin{equation}
  i'_k = \begin{cases} 
  i''_k & \text{if } 1 \leq k \leq j_1, \\
  i''_{k-j_1} & \text{if } j_1 + 1 \leq k \leq j_0.
\end{cases}
\end{equation}
(A.21)

Then, (A.10) follows from (A.20) and $j_1 \geq 0$. (A.11) and (A.12) follow from (A.21), $i''_{j_1} \leq r-j+1$ and $i'_1 \geq r-j+3$. \qed

**Lemma A.3.** For each integer $w$ in $\mathcal{L}_{r,m}$, there is a unique pair $(\sigma, l) \in \bigcup_{0 \leq j \leq r} \mathcal{W}^{r,m}$ such that $L(g(\sigma, l)) = w$.

**Proof.** Assume that $w = 2^{m-i_1} + 2^{m-i_2} + \cdots + 2^{m-i_{j_0}} + 1$ for $1 \leq i_1 < i_2 < \cdots < i_{j_0} \leq m$, $0 \leq j_0 \leq r$. If $j$ satisfies (A.10) to (A.12), then
\begin{equation}
  i_{j_0-j} \geq i_{j_0-j+1} + (j-j' - 1) \geq (r-j+3) + (j-j' - 1) > r-j' + 1 \quad \text{if } 0 \leq j' < j,
\end{equation}
(A.22)
\[ i_{j_0-j''+1} \leq i_{j_0-j} - (j'' - j - 1) \leq (r-j+1) - (j'' - j - 1) < r - j'' + 3 \quad \text{if } j < j'' \leq j_0. \]

(A.23)

and thus \( j \) is the unique integer satisfying (A.10) to (A.12).

Below we prove that such an integer \( j \) does exist and thus, by Lemma A.2, there is a pair \((\sigma, l) \in \mathcal{W}_{j}^{r,m}\) such that \( L(g(\sigma, l)) = w \).

Let \( s(j') \) denote the number of integers \( i_k \) between \( j' + 3 \) and \( m \), i.e.,
\[ s(j') = |\{k : j' + 3 \leq i_k \leq m, 1 \leq k \leq j_0\}|. \]

(A.24)

Let \( s^*(j') = s(j') \). Then, \( j' \leq s^*(j') \leq j' + j_0 \) and
\[ s^*(j' - 1) \leq s^*(j') \leq s^*(j' - 1) + 1. \]

(A.25)

Thus, there is some integer \( j' \) such that \( s^*(j') = r \). Let \( j_2 \) be the smallest integer such that \( s^*(j_2) = r \). Then,
\[ 0 \leq s(j_2) \leq j_0, \]

(A.26)
\[ j_2 + 2 \notin \{i_1, i_2, \ldots, i_{j_0}\}, \]

(A.27)
\[ i_k \leq j_2 + 2, \quad \text{if } 1 \leq k \leq j_0 - s(j_2), \]

(A.28)
\[ i_k \geq j_2 + 3, \quad \text{if } j_0 - s(j_2) + 1 \leq k \leq j_0. \]

(A.29)

It follows from (A.27) and (A.28) that
\[ i_{j_0 - s(j_2)} \leq j_2 + 1 = r - s(j_2) + 1, \quad \text{if } s(j_2) < j_0. \]

(A.30)

Thus, from (A.26), (A.29) and (A.30), we see that (A.10) to (A.12) are valid for \( j = s(j_2) \).

Finally, we note that \( L(g(\sigma, l)) \neq L(g(\sigma', l')) \) is obviously valid for distinct pairs \((\sigma, l)\) and \((\sigma', l')\) in the same set \( \mathcal{W}_{j}^{r,m} \).

**Proof of Theorem 2.** From (A.6), (A.9) and Lemma A.3, we see that
\[ L(g(\sigma, l)) \neq L(g(\sigma', l')), \quad \text{for } (\sigma, l) \neq (\sigma', l'), \]

(A.31)
\[ |\mathcal{G}_{r,m}| = |\mathcal{W}_{r,m}^{r}|. \]

(A.32)

From (A.31), the tuples in \( \mathcal{G}_{r,m} \) are linearly independent.

Since, for any \( \sigma \in \mathcal{W}_j^{r,m} \) and \( 1 \leq l < 2^r - j + 1 \), we have
\[ R(g(\sigma, l)) = 2^m + 1 - L(g(\sigma, 2^r - j + 1 - l)), \]

(A.33)

we see also that \( R(g(\sigma, l)) \neq R(g(\sigma', l')) \) for different pairs \((\sigma, l)\) and \((\sigma', l')\).

Hence, to prove that \( \mathcal{G}_{r,m} \) is a TOGM of the RM code \( \text{RM}(r, m) \), it remains to prove that \( g(\sigma, l) \) is a codeword in \( \text{RM}(r, m) \) for any pair \((\sigma, l) \in \mathcal{W}_{j}^{r,m} \).
For \(1 \leq i \leq m\) and \(0 \leq l < 2^i\), let
\[
v(l, i) \triangleq (0_{2^{m-i}}, 1_{2^m}, 0_{(2^i-1)2^{m-i}}).
\] (A.34)

The tuple \(v(l, i)\) can also be expressed as the following Boolean product:
\[
v(l, i) = (v_{1,m} + h_{i-1}(l)1_{2^m})(v_{2,m} + h_{i-2}(l)1_{2^m}) \cdots (v_{i,m} + h_0(l)1_{2^m}),
\] (A.35)

where \(h_k(l)\) are the integers in \(\{0, 1\}\) satisfying
\[
\sum_{k=0}^{i-1} (1 - h_k(l))2^k = l.
\] (A.36)

It follows from (29), (A.7) and (A.34) that
\[
g(\sigma, l) = v(l - 1, r - j + 1)q^{r-j+1}_1(1) + v(l, r - j + 1)q^{r-j+1}_1(1)
\]
\[
= v(l - 1, r - j + 1)v(\sigma) + v(l, r - j + 1)v(\sigma).
\] (A.37)

We note that \(v(\sigma)\) belongs to \(\mathcal{V}_{r,m}\). From (A.35), the tuple \(v(l - 1, r - j + 1)v(\sigma)\) is the sum of \(v_{1,m} \cdots v_{r-j+1,m}v(\sigma)\) and some tuples in \(\mathcal{V}^*_{r,m}\). If \(v(\sigma) = v_{i_1,m}v_{i_2,m} \cdots v_{i_j,m}\), it is obvious that
\[
v(\sigma) = (v_{i_1,m} + 1_{2^m})(v_{i_2,m} + 1_{2^m}) \cdots (v_{i_j,m} + 1_{2^m}),
\] (A.38)

and thus, from (A.35), the tuple \(v(l, r - j + 1)v(\sigma)\) is also the sum of \(v_{1,m} \cdots v_{r-j+1,m}v(\sigma)\) and some tuples in \(\mathcal{V}^*_{r,m}\). Hence, from (A.37), the tuple \(g(\sigma, l)\) can be expressed as the sum of some tuples in \(\mathcal{V}^*_{r,m}\), i.e., \(g(\sigma, l)\) is a codeword in \(\text{RM}(r, m)\). \(\square\)

References

