

# The Treewidth of MDS and Reed-Muller Codes\*

Navin Kashyap<sup>†</sup>

Andrew Thangaraj<sup>‡</sup>

## Abstract

The constraint complexity of a graphical realization of a linear code is the maximum dimension of the local constraint codes in the realization. The treewidth of a linear code is the least constraint complexity of any of its cycle-free graphical realizations. This notion provides a useful parametrization of the maximum-likelihood decoding complexity for linear codes. In this paper, we prove the surprising fact that for maximum distance separable codes and Reed-Muller codes, treewidth equals trelliswidth, which, for a code, is defined to be the least constraint complexity (or branch complexity) of any of its trellis realizations. From this, we obtain exact expressions for the treewidth of these codes, which constitute the only known explicit expressions for the treewidth of algebraic codes.

## 1 Introduction

A (normal) graphical realization of a linear code  $\mathcal{C}$  consists of an assignment of the coordinates of  $\mathcal{C}$  to the vertices of a graph, along with a specification of linear state spaces and linear “local constraint” codes to be associated with the edges and vertices, respectively, of the graph [4]. Cycle-free graphical realizations, or simply *tree realizations*, are those in which the underlying graph is a tree. Tree realizations of linear codes are interesting because the sum-product algorithm (SPA) on such a realization is an exact implementation of maximum-likelihood (ML) decoding [16]. The notion of constraint complexity of a tree realization was introduced by Forney [5] as a measure of the computational complexity of the corresponding SPA algorithm. It is defined to be the maximum dimension among the local constraint codes constituting the realization. The *treewidth* of a linear code is the least constraint complexity of any of its tree realizations.

The minimal tree complexity measure defined for linear codes by Halford and Chugg [6] is a close relative of treewidth. There are also closely related notions of treewidth defined for graphs [3] and matroids [7]; these relationships are discussed in more detail in [10]. Known facts about the treewidth of graphs and matroids imply that computing the treewidth of a code is NP-hard.

For a length- $n$  linear code over the field  $\mathbb{F}_q$ , the computational complexity of implementing ML decoding, via the SPA on an optimal tree realization, is  $O(nq^t)$ , where  $t$  is the treewidth of the code [10]. In particular, ML decoding is fixed-parameter tractable with respect to treewidth, which means that for codes whose treewidth is bounded by a fixed constant  $t$ , ML decoding can be performed in polynomial time. Thus, treewidth provides a useful parametrization of ML decoding complexity.

Trellis representations (or trellis realizations) of codes are special cases of tree realizations which have received extensive attention in the literature (see e.g., [14]). In the context of trellis representations, constraint complexity is usually called branch complexity. We define here the *trelliswidth* of a code to be the least branch complexity of any of its trellis representations (optimized over all possible orderings

---

\*This work was supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC), Canada.

<sup>†</sup>N. Kashyap is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, and with the Department of Mathematics & Statistics, Queen’s University, Kingston, Ontario, Canada. Email: nkashyap@ece.iisc.ernet.in

<sup>‡</sup>A. Thangaraj is with the Department of Electrical Engineering, Indian Institute of Technology, Madras. Email: andrew@ee.iitm.ac.in

of the coordinates of the code). As trellis representations are instances of tree realizations, trelliswidth is at least as large as treewidth. In fact, it is known that trelliswidth can be much larger than treewidth: it was shown in [11] that the ratio of trelliswidth to treewidth can grow at most logarithmically with blocklength, and that there are codes with arbitrarily large blocklengths that achieve this logarithmic growth rate. The only known code family achieving logarithmic growth rate of this ratio is a family consisting of cut-set codes of a certain class of graphs. The codes in this family all have treewidth equal to 2, and rate approximately 1/4, but minimum distance only 4 [10].

It is not known if there are any other code families for which there is a significant advantage to be gained in going from trellis representations to tree realizations that are topologically more complex. In the only previous investigation reported on this question, Forney [5] considered the family of Reed-Muller codes. He showed that for a certain natural tree realization of Reed-Muller codes, obtained from their well-known recursive  $|u|u + v|$  construction, the constraint complexity is, in general, strictly larger than the trelliswidth of the code. But this still leaves open the possibility that there may be other tree realizations whose constraint complexity beats trelliswidth. In particular, it leaves undecided the question of whether the treewidth of a Reed-Muller code can be strictly less than its trelliswidth.

In this paper, we show that for Reed-Muller codes, treewidth is equal to trelliswidth. The proof of this makes use of structural properties known for optimal trellis realizations of Reed-Muller codes, and also relies strongly on a certain separator theorem for trees. A similar proof strategy also works on the much simpler case of maximum distance separable (MDS) codes, where again we show that treewidth equals trelliswidth. These results yield the first explicit expressions for the treewidth of classical algebraic codes.

The rest of this paper is organized as follows. After providing the necessary definitions and notation in Section 2, we describe, in Section 3, our proof strategy for showing that treewidth equals trelliswidth for certain codes. Sections 4 and 5 deal with MDS and Reed-Muller codes, respectively. The technical details of some of the proofs are given in appendices.

## 2 Preliminaries and Notation

The notation  $[n]$  denotes the set of positive integers from 1 to  $n$ ;  $[a, b]$  denotes the set  $\{i \in \mathbb{Z} : a \leq i \leq b\}$ . An  $(n, k)$  linear code is a code of length  $n$  and dimension  $k$ . The  $n$  coordinates of the code are indexed by the elements of an index set  $I$ ; unless specified otherwise,  $I = [n]$ . Given a linear code  $\mathcal{C}$  with index set  $I$ , for  $J = \{j_1, j_2, \dots, j_s\} \subseteq I$ , the shortening of  $\mathcal{C}$  to the coordinates in  $J$  is denoted  $\mathcal{C}_J$  and defined as follows:

$$\mathcal{C}_J = \{c_{j_1} c_{j_2} \dots c_{j_s} : c_1 c_2 \dots c_n \in \mathcal{C}, c_i = 0 \text{ for } i \notin J\}.$$

The notions of treewidth and trelliswidth are central to this article, and we define these next.

### 2.1 Treewidth and trelliswidth

For brevity, we provide only the necessary definitions and main results; for details, see [5],[10].

A tree is a connected graph with no cycles. The set of nodes and the set of edges of a tree  $T$  are denoted by  $V(T)$  and  $E(T)$ , respectively. Degree-1 nodes in a tree are called *leaves*, and all other nodes are called *internal nodes*. We let  $L(T)$  denote the set of leaves of  $T$ . A tree is a *path* if all its internal nodes have degree 2; and is a *cubic tree* if all its internal nodes have degree 3. A path with at least one edge has exactly two leaves; a cubic tree with  $n$  leaves has  $n - 2$  internal nodes.

Let  $\mathcal{C}$  be an  $(n, k)$  linear code with index set  $I$ . A *tree decomposition* of  $\mathcal{C}$  is a pair  $(T, \omega)$ , where  $T$  is a tree and  $\omega : I \rightarrow V(T)$  is an assignment of coordinates of  $\mathcal{C}$  to the nodes of  $T$ .

Given a tree decomposition  $(T, \omega)$  of  $\mathcal{C}$ , for each node  $v$  of  $T$ , we define a quantity  $\kappa_v$  as follows. Let  $E(v)$  denote the set of edges of  $T$  incident on  $v$ . For  $e \in E(v)$ , let  $T_{e,v}$  denote the component of

$T - e$  ( $T$  with  $e$  removed) not containing  $v$ . Finally, let  $I_{e,v} = \omega^{-1}(V(T_{e,v}))$  be the set of coordinates of  $\mathcal{C}$  that are assigned to nodes in  $T_{e,v}$ . Then,

$$\kappa_v = k - \sum_{e \in E(v)} \dim(\mathcal{C}_{I_{e,v}}). \quad (1)$$

The quantity  $\kappa_v$  above is the dimension of the local constraint code at node  $v$  in the minimal realization of  $\mathcal{C}$  on  $(T, \omega)$ , denoted by  $\mathcal{M}(\mathcal{C}; T, \omega)$ .

Let  $\kappa(\mathcal{C}; T, \omega) = \max_{v \in V(T)} \kappa_v$  denote the constraint complexity of  $\mathcal{M}(\mathcal{C}; T, \omega)$ . The treewidth of a code  $\mathcal{C}$ , denoted by  $\kappa(\mathcal{C})$ , is then defined as

$$\kappa(\mathcal{C}) = \min_{(T, \omega)} \kappa(\mathcal{C}; T, \omega). \quad (2)$$

It is, in fact, enough to perform the minimization in (2) over cubic trees  $T$  with  $n$  leaves, and mappings  $\omega$  that are bijections between  $I$  and  $L(T)$ .

The trelliswidth of  $\mathcal{C}$ , which we will denote by  $\tau(\mathcal{C})$ , can be defined using the above notation as follows:

$$\tau(\mathcal{C}) = \min_{\pi} \kappa(\mathcal{C}; P, \pi), \quad (3)$$

where  $P$  is the path on  $n$  nodes, and the minimization is over mappings  $\pi$  that are bijections between  $I$  and  $V(P)$ . From (2) and (3), it is clear that  $\kappa(\mathcal{C}) \leq \tau(\mathcal{C})$ .

Let  $v_1, v_2, \dots, v_n$  be the nodes of the path  $P$ , listed in order from one leaf to the other. For the bijection  $\pi : I \rightarrow V(P)$  that maps  $i$  to  $v_i$  ( $1 \leq i \leq n$ ), we obtain from (1),

$$\kappa_{v_i} = k - \dim(\mathcal{C}_{\pi[1, i-1]}) - \dim(\mathcal{C}_{\pi[i+1, n]}), \quad (4)$$

where  $\pi[a, b] = \{\pi(j) : a \leq j \leq b\}$ .

## 2.2 Generalized Hamming weights

The generalized Hamming weights of a linear code, introduced and studied in [15], limit the possible dimensions of shortened versions of the code. So, they are related to the complexity of tree realizations in a natural way.

Let  $\mathcal{C}$  be an  $(n, k)$  linear code with index set  $I$ . We will use the notation  $\mathcal{D} \sqsubseteq \mathcal{C}$  to say that  $\mathcal{D}$  is a subcode of  $\mathcal{C}$ . For a subcode  $\mathcal{D} \sqsubseteq \mathcal{C}$ , we define its support  $\chi(\mathcal{D}) = \{i : \exists c_1 c_2 \dots c_n \in \mathcal{D} \text{ s.t. } c_i \neq 0\}$ . The  $p$ -th generalized Hamming weight of  $\mathcal{C}$ , denoted  $d_p(\mathcal{C})$ , is the size of the smallest support of a  $p$ -dimensional subcode of  $\mathcal{C}$ , i.e.,  $d_p(\mathcal{C}) = \min\{|\chi(\mathcal{D})| : \mathcal{D} \sqsubseteq \mathcal{C}, \dim(\mathcal{D}) = p\}$  for  $1 \leq p \leq k$ . It is known that  $0 \leq d_1(\mathcal{C}) < d_2(\mathcal{C}) < \dots < d_k(\mathcal{C}) \leq n$ . Also,  $d_1(\mathcal{C})$  is the minimum distance of  $\mathcal{C}$ .

A closely related definition is that of maximal limited-support subcode dimensions. For  $1 \leq s \leq n$ ,  $U_s(\mathcal{C})$  is defined to be the maximum dimension of a subcode of  $\mathcal{C}$  with support at most  $s$ , i.e.,  $U_s(\mathcal{C}) = \max\{\dim(\mathcal{D}) : \mathcal{D} \sqsubseteq \mathcal{C}, |\chi(\mathcal{D})| \leq s\}$ . The maximal limited-support subcode dimensions can be computed using the generalized Hamming weights as follows:

$$U_s(\mathcal{C}) = u \text{ such that } d_u(\mathcal{C}) \leq s < d_{u+1}(\mathcal{C}) \quad (5)$$

with the convention that  $d_0(\mathcal{C}) = 0$  and  $d_{k+1}(\mathcal{C}) = n + 1$ . We also define  $U_0(\mathcal{C}) = 0$ .

## 3 The Proof Strategy

From the relevant definitions, treewidth cannot exceed trelliswidth for any code  $\mathcal{C}$ , i.e.,  $\kappa(\mathcal{C}) \leq \tau(\mathcal{C})$ . We now describe a general strategy that can be used to show the opposite inequality in certain cases.

Consider an  $(n, k)$  linear code  $\mathcal{C}$ , with index set  $I$ . The idea of using maximal limited-support subcode dimensions to study the complexity of trellis realizations of  $\mathcal{C}$  was introduced in [9]. We extend that idea to tree realizations here. For  $J \subseteq I$ ,  $\mathcal{C}_J$  is a subcode of  $\mathcal{C}$  with support at most  $|J|$ . So,  $\dim(\mathcal{C}_J) \leq U_{|J|}(\mathcal{C})$ . Therefore, given any tree decomposition  $(T, \omega)$  of  $\mathcal{C}$ , we obtain from (1) that for any  $v \in V(T)$ ,

$$\kappa_v \geq k - \sum_{e \in E(v)} U_{|I_{e,v}|}(\mathcal{C}). \quad (6)$$

Now, recall from the definition of treewidth that it suffices to carry out the minimization in (2) over tree decompositions  $(T, \omega)$  in which  $T$  is a cubic tree with  $n$  leaves, and  $\omega$  is a bijection between  $I$  and  $L(T)$ . For such a  $(T, \omega)$ , we note that  $|I_{e,v}|$  is simply the number of leaves in  $T_{e,v}$ , and for an internal node  $v \in V(T)$ , the summation in (6) contains exactly three terms.

Let  $n_{e,v}$  denote the number of leaves in  $T_{e,v}$ , and note that these numbers  $n_{e,v}$  are determined purely by the topology of  $T$ . At an internal node  $v$  in a cubic tree  $T$  with  $n$  leaves, we will list the edges in  $E(v)$  in the form of an ordered triple  $[e_1(v) \ e_2(v) \ e_3(v)]$  such that  $1 < n_{e_1(v),v} \leq n_{e_2(v),v} \leq n_{e_3(v),v} < n$ . If the node  $v$  is clear in the context, we will use the simplified notation  $n_i = n_{e_i(v),v}$  for  $i = 1, 2, 3$ .

Suppose that  $T$  is a cubic tree with  $n$  leaves having an internal node  $v$  such that the numbers  $n_1, n_2, n_3$  satisfy  $\sum_{i=1}^3 U_{n_i}(\mathcal{C}) \leq k - \tau(\mathcal{C})$ . Then, by (6), for any bijection  $\omega$  between  $I$  and  $L(T)$ , we have  $\kappa_v \geq \tau(\mathcal{C})$ , and hence  $\kappa(\mathcal{C}; T, \omega) \geq \tau(\mathcal{C})$ . Consequently, if every cubic tree with  $n$  leaves had such a node  $v$ , then we would have  $\kappa(\mathcal{C}) \geq \tau(\mathcal{C})$ . Since the opposite inequality is always true, we have proved the following proposition.

**Proposition 1.** *Let  $\mathcal{C}$  be an  $(n, k)$  linear code with the property that for any cubic tree  $T$  with  $n$  leaves, there always exists an internal node  $v \in V(T)$  such that  $\sum_{i=1}^3 U_{n_i}(\mathcal{C}) \leq k - \tau(\mathcal{C})$ , where  $n_i = n_{e_i(v),v}$ . Then,  $\kappa(\mathcal{C}) = \tau(\mathcal{C})$ .*

A comment on the proof strategy implied by Proposition 1 is in order. To show that  $\kappa(\mathcal{C}) \geq \tau(\mathcal{C})$  (and hence,  $\kappa(\mathcal{C}) = \tau(\mathcal{C})$ ), the obvious strategy would be to show, for each tree decomposition  $(T, \omega)$  of  $\mathcal{C}$ , the existence of a node  $v \in V(T)$  for which  $\kappa_v \geq \tau(\mathcal{C})$ , where  $\kappa_v$  is given by (6). In general, the node  $v$  would depend on the tree  $T$  as well as on the coordinate assignment  $\omega$ . However, in the proof method based upon Proposition 1, the idea is to find, for a given  $(T, \omega)$ , a node  $v \in V(T)$  that depends only on the topology of  $T$ , and thus, is *independent* of  $\omega$ , for which  $\kappa_v \geq \tau(\mathcal{C})$  holds. It is a remarkable fact that this proof strategy can be made to work for MDS and Reed-Muller codes, as we will see in Sections 4 and 5.

The hypothesis of Proposition 1 requires the existence of a node in any cubic tree, whose removal partitions the tree into components with a certain property. The property in this case is that the corresponding partition of the number of leaves,  $n$ , into  $n_1, n_2, n_3$  satisfies  $\sum_{i=1}^3 U_{n_i}(\mathcal{C}) \leq k - \tau(\mathcal{C})$ . Structural results of this form are known as separator theorems (see *e.g.*, [13]).

A classical separator theorem is a theorem of Jordan [8] that states that any tree on  $n$  nodes has an internal node whose removal leaves behind connected components with at most  $n/2$  nodes each. A trivial modification of the simple proof of this theorem shows that the two occurrences of “nodes” in the theorem statement can be replaced by “leaves”. For easy reference, we record this as a proposition for the special case of cubic trees.

**Proposition 2.** *In any cubic tree with  $n \geq 3$  leaves, there exists an internal node  $v$  such that  $n_{e_i(v),v} \leq n/2$  for  $i = 1, 2, 3$ .*

Another classical (edge) separator theorem is the following result (cf. [13]): every cubic tree  $T$  with  $n$  leaves contains an edge  $e$  such that both components of  $T - e$  have at most  $2n/3$  leaves. Now, one of these two components must have at least  $n/2$  leaves; let  $v$  be the node incident with  $e$  for which this component is  $T_{e,v}$ . Then, for this  $v$ , we have  $n_3 \in [n/2, 2n/3]$ . We record this fact below.

**Proposition 3.** *In any cubic tree with  $n \geq 3$  leaves, there exists an internal node  $v$  such that  $n_{e_3(v),v} \in [n/2, 2n/3]$ .*

As we will see in the next two sections, Propositions 2 and 3 allow us to deal with MDS and Reed-Muller codes, respectively. We consider MDS codes first.

## 4 Treewidth of MDS Codes

MDS codes are  $(n, k)$  linear codes for which the minimum distance equals  $n - k + 1$ . Basic facts about MDS codes can be found in [12].

Let  $\mathcal{C}$  be an  $(n, k)$  MDS code, with index set  $I = [n]$ . The generalized Hamming weights of  $\mathcal{C}$  were computed in [15] as follows:

$$d_p(\mathcal{C}) = n - k + p, \quad 1 \leq p \leq k.$$

From this, the maximal limited-support subcode dimensions,  $U_s(\mathcal{C})$  for  $1 \leq s \leq n$ , can be determined using (5). They are given by

$$U_s(\mathcal{C}) = \begin{cases} 0, & 1 \leq s \leq n - k, \\ q, & s = n - k + q, \quad q = 1, 2, \dots, k. \end{cases} \quad (7)$$

Equivalently,  $U_s(\mathcal{C}) = \max\{0, s - (n - k)\}$ . We use this to compute  $\tau(\mathcal{C})$  next.

Let  $H$  be a parity-check matrix for  $\mathcal{C}$ . For a subset  $J \subseteq I$ , the code  $\mathcal{C}_J$  has dimension equal to  $|J| - \text{rank}(H|_J)$ , where  $H|_J$  refers to the restriction of  $H$  to the columns indexed by  $J$ . As  $\mathcal{C}$  is MDS,  $\text{rank}(H|_J) = \min\{|J|, n - k\}$ . Hence,  $\dim(\mathcal{C}_J) = \max\{0, |J| - (n - k)\} = U_{|J|}(\mathcal{C})$ . Therefore, for any permutation  $\pi$  of  $I$ , we have for integers  $1 \leq a \leq b \leq n$ ,  $\dim(\mathcal{C}_{\pi[a,b]}) = U_{b-a+1}(\mathcal{C})$ . Therefore, the right-hand-side of (4) is always equal to  $k - U_{i-1}(\mathcal{C}) - U_{n-i}(\mathcal{C})$ . It follows directly from this that

$$\tau(\mathcal{C}) = \max_{1 \leq i \leq n} (k - U_{i-1}(\mathcal{C}) - U_{n-i}(\mathcal{C})) = k - \min_{1 \leq i \leq n} (U_{i-1}(\mathcal{C}) + U_{n-i}(\mathcal{C})).$$

A straightforward computation using (7) yields

$$\min_{1 \leq i \leq n} (U_{i-1}(\mathcal{C}) + U_{n-i}(\mathcal{C})) = \begin{cases} 0, & \text{if } n - k \geq k, \\ 2k - n - 1, & \text{if } n - k < k. \end{cases}$$

achieved for  $i = n - k + 1$ . We thus have the following result.

**Proposition 4.** *The trelliswidth of an  $(n, k)$  MDS code  $\mathcal{C}$  is given by  $\tau(\mathcal{C}) = \min\{k, n - k + 1\}$ .*

With this, we have

$$k - \tau(\mathcal{C}) = \max\{0, 2k - n - 1\}. \quad (8)$$

We can now prove that the treewidth of an MDS code equals its trelliswidth.

**Theorem 5.** *For an  $(n, k)$  MDS code  $\mathcal{C}$ , we have*

$$\kappa(\mathcal{C}) = \tau(\mathcal{C}) = \min\{k, n - k + 1\}.$$

*Proof.* The statement is trivial for  $n = 1, 2$ , or when  $k = n$ , so we assume  $n \geq 3$  and  $1 \leq n - k$ . Let  $T$  be a cubic tree with  $n$  leaves, and let  $v$  be the node guaranteed by Proposition 2. We will show that  $v$  satisfies the hypothesis of Proposition 1.

Set  $n_i = n_{e_i(v), v}$ ,  $i = 1, 2, 3$ , and recall that, by definition,  $n_1 \leq n_2 \leq n_3$ . By choice of  $v$ , we also have  $n_i \leq n/2$  for  $i = 1, 2, 3$ . For convenience, we write  $U_{n_i}$  for  $U_{n_i}(\mathcal{C})$ .

Case 1:  $n - k \geq k$ .

In this case,  $n_i \leq n/2 \leq n - k$ , so that  $\sum_i U_{n_i} = 0$  by (7). Moreover, by (8),  $k - \tau(\mathcal{C}) = 0$ .

Case 2:  $1 \leq n - k < k$ .

Now, we have  $n_i \leq n/2 < k$ . We must show that  $\sum_i U_{n_i} \leq 2k - n - 1$ . If  $n_3 \leq n - k$ , then  $\sum_i U_{n_i} = 0$ . So, we assume  $n_3 = k - \delta$ , with  $1 \leq \delta < 2k - n$ . Then,  $U_{n_3} = n_3 - (n - k) = 2k - n - \delta$  and  $n_1 + n_2 = n - n_3 = n - k + \delta$ . So, we have

$$\begin{aligned} U_{n_1} + U_{n_2} + U_{n_3} &= \max\{0, k - n + n_1\} + \max\{0, k - n + n_2\} + 2k - n - \delta \\ &\leq \max\{0, k - n + n_1, k - n + n_2, 2k - 2n + n_1 + n_2\} + 2k - n - \delta \\ &= \max\{2k - n - \delta, 3k - 2n + n_2 - \delta, 3k - 2n\} \\ &\leq 2k - n - 1, \end{aligned}$$

where the last inequality holds because  $\delta \geq 1$ ,  $n_2 \leq n - k + \delta - 1$  and  $n - k \geq 1$ .

Thus, in both cases, we see that  $\sum_i U_{n_i} \leq k - \tau(\mathcal{C})$ , and so, by Proposition 1, we have  $\kappa(\mathcal{C}) = \tau(\mathcal{C})$ .  $\square$

## 5 Reed-Muller codes

For a positive integer  $m$  and a non-negative integer  $r$  with  $0 \leq r \leq m$ , the  $r$ -th order binary Reed-Muller code of length  $2^m$ , denoted  $\text{RM}(r, m)$ , is defined as follows. Let  $P_r^m$  denote the set of all Boolean polynomials in  $m$  variables of degree less than or equal to  $r$ . For an integer  $i$ ,  $0 \leq i \leq 2^m - 1$ , with binary expansion  $i = \sum_{j=0}^{m-1} b_j(i)2^j$ ,  $b_j(i) \in \{0, 1\}$ , we let  $\mathbf{b}(i) = (b_0(i), b_1(i), \dots, b_{m-1}(i))$ . For  $f \in P_r^m$ , let  $f(\mathbf{b}(i)) = f(b_0(i), b_1(i), \dots, b_{m-1}(i))$ . The code  $\text{RM}(r, m)$  is defined as

$$\text{RM}(r, m) = \{[f(\mathbf{b}(0)) \ f(\mathbf{b}(1)) \ \dots \ f(\mathbf{b}(2^m - 1))] : f \in P_r^m\}. \quad (9)$$

The code  $\text{RM}(r, m)$  has length  $n = 2^m$ , dimension  $k(r, m) = \sum_{j=0}^r \binom{m}{j}$ , and minimum distance  $2^{m-r}$  [12]. In (9), the order of evaluation of the function  $f$  is according to the index set  $I = [0, 2^m - 1]$ . This is called the standard bit order.

We will denote the treewidth and trelliswidth of  $\text{RM}(r, m)$  by  $\kappa(r, m)$  and  $\tau(r, m)$ , respectively.

### 5.1 Trelliswidth of $\text{RM}(r, m)$

Let  $\mathcal{C}$  be the Reed-Muller code  $\text{RM}(r, m)$  in the standard bit order, so that  $I = [0, 2^m - 1]$ . In this section, we derive an exact expression for the trelliswidth of  $\mathcal{C}$ .

Let  $P$  be the path on  $n = 2^m$  nodes, with  $v_0, v_1, \dots, v_{n-1}$  being the nodes of  $P$ , listed in order from one leaf to the other. For any  $\pi : I \rightarrow V(P)$ , we obtain from (4), in a manner analogous to the derivation of (6),

$$\kappa_{v_i} \geq k(r, m) - U_i(\mathcal{C}) - U_{n-1-i}(\mathcal{C}),$$

for  $i = 0, 1, \dots, n - 1$ . Thus,

$$\kappa(\mathcal{C}; P, \pi) \geq k(r, m) - \min_{0 \leq i \leq n-1} (U_i(\mathcal{C}) + U_{n-1-i}(\mathcal{C})). \quad (10)$$

Note that the right-hand-side is independent of  $\pi$ , so that by (3),

$$\tau(\mathcal{C}) \geq k(r, m) - \min_{0 \leq i \leq n-1} (U_i(\mathcal{C}) + U_{n-1-i}(\mathcal{C})). \quad (11)$$

It is shown in [9] that for  $\text{RM}(r, m)$  in the standard bit order, we have for  $i = 0, 1, \dots, n - 1$ ,

$$\dim(\mathcal{C}_{[0, i]}) = U_{i+1}(\mathcal{C}) \quad \text{and} \quad \dim(\mathcal{C}_{[i, n-1]}) = U_{n-i}(\mathcal{C}). \quad (12)$$

It follows that when  $\pi$  simply maps  $i$  to  $v_i$  for all  $i \in I$ , then we have equality in (10), and hence, in (11). To put this another way, the branch complexity of the minimal trellis representation of  $\text{RM}(r, m)$  in the standard bit order attains the lower bound on, and thus equals, the trelliswidth of the code. Techniques from [2] allow us to compute, with very little effort, the branch complexity of this trellis representation. We give the details of this computation in Appendix A. From this, we obtain the following result.

**Proposition 6.** *The trelliswidth of the Reed-Muller code  $RM(r, m)$  is given by*

$$\tau(r, m) = \begin{cases} \sum_{j=0}^r \binom{m-2j-1}{r-j} & \text{if } m \geq 2r+1, \\ 1 + \sum_{j=0}^{m-r-1} \binom{m-2j-1}{r-j} & \text{if } m < 2r+1. \end{cases}$$

Recall that the dimension of the code  $RM(r, m)$  is given by  $k(r, m) = \sum_{j=0}^r \binom{m}{j}$ . We will find it convenient to define  $k(r', m')$  to be  $\sum_{j=0}^{r'} \binom{m'}{j}$  for all non-negative integers  $r', m'$ , including when  $r' > m'$ , with the usual conventions that  $\binom{0}{0} = 1$  and  $\binom{m'}{j} = 0$  for  $j > m'$ . Thus, for  $r' \geq m' \geq 0$ ,  $k(r', m') = 2^{m'}$ . Following these conventions, we give an expression for the difference  $k(r, m) - \tau(r, m)$ .

**Proposition 7.** *For the Reed-Muller code  $RM(r, m)$ , we have*

$$k(r, m) - \tau(r, m) = \sum_{i=0}^{\min\{2(r-1), m-1\}} k(r-1 - \lceil i/2 \rceil, m-1-i).$$

We present the algebraic manipulations required to prove this proposition in Appendix A.

It is instructive to explicitly write out some of the terms of the summation in the last proposition. When  $m \geq 2r$ , we have

$$\begin{aligned} k(r, m) - \tau(r, m) &= k(r-1, m-1) + k(r-2, m-2) + k(r-2, m-3) \\ &\quad + k(r-3, m-4) + k(r-3, m-5) \\ &\quad + \cdots + k(0, m-2r+2) + k(0, m-2r+1), \end{aligned} \tag{13}$$

and when  $m \leq 2r-1$ , we have

$$\begin{aligned} k(r, m) - \tau(r, m) &= k(r-1, m-1) + k(r-2, m-2) + k(r-2, m-3) \\ &\quad + k(r-3, m-4) + k(r-3, m-5) \\ &\quad + \cdots + k(r-1 - \lceil \frac{m-2}{2} \rceil, 1) + k(r-1 - \lceil \frac{m-1}{2} \rceil, 0). \end{aligned} \tag{14}$$

## 5.2 Treewidth of $RM(r, m)$

We state below our main result showing that the treewidth of a Reed-Muller code equals its trelliswidth.

**Theorem 8.** *The treewidth of the Reed-Muller code  $RM(r, m)$  is given by*

$$\kappa(r, m) = \tau(r, m) = \begin{cases} \sum_{j=0}^r \binom{m-2j-1}{r-j} & \text{if } m \geq 2r+1, \\ 1 + \sum_{j=0}^{m-r-1} \binom{m-2j-1}{r-j} & \text{if } m < 2r+1. \end{cases}$$

The rest of this section is devoted to a proof of the above result, which follows the strategy outlined in Section 3. Some of the technical details of the proof are presented in Appendices B and C.

Let  $RM(r, m)$  be given. If  $m \leq 2$ , or  $r = m$ , then  $RM(r, m)$  is an MDS code, which has been dealt with in Section 4. Henceforth, we will assume  $m \geq 3$  and  $r \leq m-1$ .

Let  $T$  be a cubic tree with  $n = 2^m$  leaves,  $m \geq 3$ , and let  $W = \{v \in V(T) : n_{e_3(v), v} \in [n/2, 2n/3]\}$ . By Proposition 3,  $W$  is non-empty. Let  $v^* \in W$  be a node that achieves  $\max\{n_{e_3(v), v} : v \in W\}$ . Write  $n_i^* = n_{e_i(v^*), v^*}$ ,  $i = 1, 2, 3$ .

**Lemma 9.** *We have  $n/6 < n_2^* < n/3$ .*

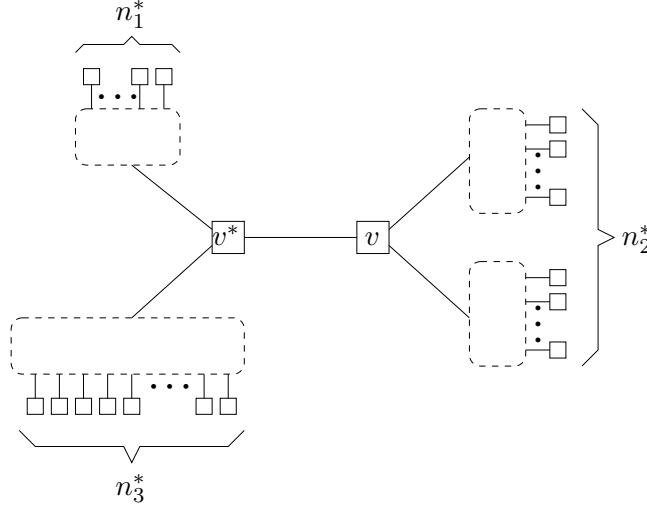


Figure 1: For  $v$ , we have  $n_3 = n_1^* + n_2^*$ .

*Proof.* If  $n_2^* < n/6$ , then from the fact that  $n_1^* \leq n_2^*$ , we obtain  $n_1^* + n_2^* < n/3$ , so that  $n_3^* > 2n/3$ , a contradiction. So,  $n_2 \geq n/6$ . However,  $n/6$  is not an integer for  $n = 2^m$ , and so,  $n_2^* > n/6$ .

If  $n_2^* \geq n/3$ , then  $n_1^* + n_3^* \leq 2n/3$ . Let  $v$  be the neighbour of  $v^*$  incident with edge  $e_2(v^*)$ . Then, setting  $n_3 = n_{e_3(v),v}$ , we see that  $n_3 = n_1^* + n_3^*$ ; see Figure 1. But this means that  $n_3^* < n_3 \leq 2n/3$ , which contradicts our choice of  $v^*$ .  $\square$

We will show that  $\sum_{i=1}^3 U_{n_i^*} \leq k(r, m) - \tau(r, m)$ , which will prove Theorem 8 by virtue of Proposition 1. Here, and in all that follows, we use  $U_h$  as shorthand for  $U_h(\text{RM}(r, m))$ ,

Denote by  $\alpha^{(m)}$  and  $\beta^{(m)}$  the largest integers in  $[0, 2n/3]$  and  $[0, n/3]$ , respectively. Explicitly,

$$\alpha^{(m)} = \begin{cases} \frac{2}{3} \cdot 2^m - \frac{1}{3} & \text{if } m \text{ is odd,} \\ \frac{2}{3} \cdot 2^m - \frac{2}{3} & \text{if } m \text{ is even,} \end{cases} \quad (15)$$

and

$$\beta^{(m)} = \begin{cases} \frac{1}{3} \cdot 2^m - \frac{2}{3} & \text{if } m \text{ is odd,} \\ \frac{1}{3} \cdot 2^m - \frac{1}{3} & \text{if } m \text{ is even.} \end{cases} \quad (16)$$

Equivalently, in binary form,

$$\mathbf{b}(\alpha^{(m)}) = \begin{cases} (1, 0, 1, 0, 1, \dots, 0, 1) & \text{if } m \text{ is odd,} \\ (0, 1, 0, 1, \dots, 0, 1) & \text{if } m \text{ is even,} \end{cases} \quad (17)$$

and

$$\mathbf{b}(\beta^{(m)}) = \begin{cases} (0, 1, 0, 1, 0, \dots, 1, 0) & \text{if } m \text{ is odd,} \\ (1, 0, 1, 0, \dots, 1, 0) & \text{if } m \text{ is even.} \end{cases} \quad (18)$$

When there is no ambiguity, we will drop the superscripts from  $\alpha^{(m)}$  and  $\beta^{(m)}$  for notational ease.

Now, what we know is that  $n_3^* \in [2^{m-1}, \alpha]$  and  $n_2^* \in [\lceil \frac{1}{6} 2^m \rceil, \beta]$ . In fact, it can be directly verified from the expression for  $\alpha$  that  $\lceil \frac{1}{6} 2^m \rceil = \alpha - 2^{m-1} + 1$ . We wish to show that  $\sum U_{n_i^*} \leq k(r, m) - \tau(r, m)$ . We will do this in two steps: first, we show in Lemma 10 below that  $\sum U_{n_i^*} \leq U_\alpha + U_\beta + U_1$ , and then, we prove in Lemma 11 that  $U_\alpha + U_\beta + U_1 = k(r, m) - \tau(r, m)$ .

Write  $n_3^* = \alpha - i$  and  $n_2^* = \beta - j$ , so that  $n_1^* = 2^m - (n_3^* + n_2^*) = i + j + 1$ , where  $i \in [0, \alpha - 2^{m-1}]$  and  $j \in [0, \beta - (\alpha - 2^{m-1} + 1)]$ . The following lemma shows that  $\sum U_{n_i^*} \leq U_\alpha + U_\beta + U_1$ .

**Lemma 10.** For  $i \in [0, \alpha - 2^{m-1}]$ , and  $j \in [0, \beta - (\alpha - 2^{m-1} + 1)]$ , we have

$$(U_\alpha - U_{\alpha-i}) + (U_\beta - U_{\beta-j}) \geq U_{i+j+1} - U_1.$$

*Proof.* See Appendix B. □

**Lemma 11.**  $U_\alpha + U_\beta + U_1 = k(r, m) - \tau(r, m)$ .

*Proof.* The minimum distance of  $\text{RM}(r, m)$  is  $2^{m-r}$ . Since we have assumed  $r \leq m-1$ , the minimum distance is at least 2, and hence,  $U_1 = 0$ . In Appendix C, we show the following: when  $m \geq 2r$ ,

$$U_s = \begin{cases} \sum_{i=0}^{r-1} k(r-1-i, m-1-2i) & \text{if } s = \alpha, \\ \sum_{i=1}^{r-1} k(r-1-i, m-2i) & \text{if } s = \beta. \end{cases} \quad (19)$$

Examining the above summations term-by-term, it may be verified that the alternate terms on the right-hand side of (13), beginning with  $k(r-1, m-1)$ , sum to  $U_\alpha$ , while the remaining terms sum to  $U_\beta$ . Hence, when  $m \geq 2r$ , the statement of the lemma holds.

When  $m < 2r$ , we show in Appendix C that

$$U_\alpha = \begin{cases} \sum_{i=0}^{\frac{m-1}{2}} k(r-1-i, m-1-2i) & \text{if } m \text{ is odd,} \\ \sum_{i=0}^{\frac{m-2}{2}} k(r-1-i, m-1-2i) & \text{if } m \text{ is even.} \end{cases} \quad (20)$$

and

$$U_\beta = \begin{cases} \sum_{i=1}^{\frac{m-1}{2}} k(r-1-i, m-2i) & \text{if } m \text{ is odd,} \\ \sum_{i=1}^{\frac{m}{2}} k(r-1-i, m-2i) & \text{if } m \text{ is even.} \end{cases} \quad (21)$$

This time, it can be seen that the alternate terms on the right-hand side of (14), beginning with  $k(r-1, m-1)$ , sum to  $U_\alpha$ , while the remaining terms sum to  $U_\beta$ . This completes the proof of the lemma. □

With this, the proof of Theorem 8 is complete.

## 6 Concluding Remarks

In this paper, we proved the surprising fact that for the families of MDS and Reed-Muller codes, if we use the maximum dimension of local constraint codes to measure the complexity of a graphical realization, then there is no advantage to be gained in going from trellis realizations to cycle-free realizations on more complex tree topologies. This is particularly surprising for Reed-Muller codes, given that they have a natural binary-tree structure arising from the recursive  $|u|u+v|$  construction (see e.g. [5]). Of course, the situation could be different if we used some other measure for the complexity of a graphical realization, for example, the sum of the local constraint dimensions.

It is also quite remarkable that the proof strategy outlined in Section 3 – namely, identifying in any cubic tree  $T$  a node  $v \in V(T)$  such that  $\kappa_v \geq \tau(\mathcal{C})$  for every tree decomposition of the code  $\mathcal{C}$  on  $T$  — succeeds for MDS and Reed-Muller codes. As noted in that section, this strategy ignores the role played by the coordinate assignment  $\omega$  in determining the local constraint code dimension,  $\kappa_v$ . It seems unlikely that this method of proof would succeed for other code families. It would of course be interesting to devise a set of tools that could be used to compute treewidth, or simply to determine whether or not treewidth can be strictly less than trelliswidth, for other families of algebraic codes.

## Appendix A: Proofs of Propositions 6 and 7

In this appendix, we compute the branch complexity of the minimal trellis representation of  $\text{RM}(r, m)$  in the standard bit order, from which the expressions in Proposition 6 and 7 are obtained. We refer the reader to the survey by Vardy [14] for the necessary background on the theory of trellis representations.

Let  $\tau(r, m)$  and  $\sigma(r, m)$  denote, respectively, the branch complexity and state complexity of the minimal trellis representation of  $\text{RM}(r, m)$  in the standard bit order. Berger and Be'ery [1] gave an explicit expression for  $\sigma(r, m)$ :

$$\sigma(r, m) = \sum_{j=0}^{\min\{r, m-r-1\}} \binom{m-2j-1}{r-j}.$$

A different derivation of the above was given by Blackmore and Norton [2]. We rely heavily on tools from [2] to prove the following result, which is equivalent to Proposition 6.

**Proposition 12.**

$$\tau(r, m) = \begin{cases} \sigma(r, m) & \text{if } m \geq 2r+1, \\ \sigma(r, m) + 1 & \text{if } m < 2r+1. \end{cases}$$

We introduce some terminology and notation that will be needed in the proof of the proposition. Let  $\mathcal{C}$  be the code  $\text{RM}(r, m)$  in the standard bit order, and let  $n = 2^m$ . Let  $\mathcal{T}$  be the minimal trellis of  $\mathcal{C}$ . For  $i = 0, 1, \dots, n$ , the dimension of the state space at depth  $i$  in  $\mathcal{T}$  is denoted  $\sigma_i$ . Thus,  $\sigma(r, m) = \max_i \sigma_i$ . For  $i = 0, 1, \dots, n-1$ , we denote by  $\tau_i$  the dimension of the branch space between the state spaces at depths  $i$  and  $i+1$ ; then,  $\tau(r, m) = \max_i \tau_i$ .

The following definitions were made in [2] for  $0 \leq i \leq n-1$ :

- (a) if  $\dim(\mathcal{C}_{[i+1, n-1]}) = \dim(\mathcal{C}_{[i-1, n-1]}) - 1$ , then  $i$  is called a *point of gain* of  $\mathcal{C}$ ; and
- (b) if  $\dim(\mathcal{C}_{[0, i]}) = \dim(\mathcal{C}_{[0, i-1]}) + 1$ , then  $i$  is called a *point of fall* of  $\mathcal{C}$ .

As per our notation from Section 5,  $\mathbf{b}(i)$  denotes the  $m$ -bit binary representation of  $i$ ,  $0 \leq i \leq n-1$ . Let  $|\mathbf{b}(i)|_0$  and  $|\mathbf{b}(i)|_1$  denote the number of 0s and 1s, respectively, in  $\mathbf{b}(i)$ .

**Lemma 13** ([2], Proposition 2.2). *For  $0 \leq i \leq n-1$ ,*

- (a)  *$i$  is a point of gain of  $\mathcal{C}$  iff  $|\mathbf{b}(i)|_1 \leq r$ ;*
- (b)  *$i$  is a point of fall of  $\mathcal{C}$  iff  $|\mathbf{b}(i)|_0 \leq r$ .*

*Proof of Proposition 12.* It is a fact that for any minimal trellis representation, branch complexity either is equal to the state complexity or is exactly one more than the state complexity. In particular,  $\sigma(r, m) \leq \tau(r, m) \leq \sigma(r, m) + 1$ . So, to prove Proposition 12, it suffices to show that

$$\tau(r, m) = \sigma(r, m) + 1 \text{ iff } m \leq 2r. \quad (22)$$

Suppose that  $\tau(r, m) = \tau_i$  for some  $i \in [0, n-1]$ . From the local behaviour of  $\mathcal{T}$  described in [2, p. 44], it follows that we can have  $\tau_i = \sigma(r, m) + 1$  iff  $\sigma_i = \sigma(r, m)$  and  $i+1$  is a point of gain as well as a point of fall of  $\mathcal{C}$ .

Thus, if  $\tau_i = \sigma(r, m) + 1$ , then by Lemma 13,  $m = |\mathbf{b}(i+1)|_1 + |\mathbf{b}(i+1)|_0 \leq 2r$ . This proves the “only if” direction of (22).

Conversely, suppose  $m \leq 2r$ . The proposition is clearly true if  $m = r$ , since  $\text{RM}(m, m) = \{0, 1\}^{2^m}$ , and we have  $\sigma(m, m) = 0$  and  $\tau(m, m) = 1$ . So, we may assume  $m \geq r+1$ . Take  $i$  to be

such that  $\mathbf{b}(i) = (0, 0 \dots, 0, 1, 0, 1, 0, \dots, 1, 0)$ , with  $|\mathbf{b}(i)|_1 = m - r - 1$ . Then, by Theorem 2.11 in [2],  $\sigma_i = \sigma(r, m)$ . Also,  $\mathbf{b}(i+1) = (1, 0, \dots, 0, 1, 0, 1, 0, \dots, 1, 0)$ , with  $|\mathbf{b}(i+1)|_1 = m - r \leq r$  and  $|\mathbf{b}(i+1)|_0 = m - (m - r) = r$ . Hence, by Lemma 13,  $i+1$  is a point of gain as well as a point of fall of  $\mathcal{C}$ . Hence,  $\tau_i = \sigma(r, m) + 1$ , which completes the proof of (22), and hence, of Proposition 12.  $\square$

We next present the algebraic manipulations needed to prove Proposition 7.

*Proof of Proposition 7.* We divide the proof into three cases.

Case 1:  $m \geq 2r + 1$ . We have

$$\begin{aligned}
k(r, m) - \tau(r, m) &= \sum_{j=0}^r \binom{m}{j} - \sum_{j=0}^r \binom{m-2j-1}{r-j} \\
&= \sum_{j=0}^r \binom{m}{j} - \sum_{j=0}^r \binom{m-2(r-j)-1}{j} \\
&= \sum_{j=1}^r \left[ \binom{m}{j} - \binom{m-2(r-j)-1}{j} \right] \\
&\stackrel{(a)}{=} \sum_{j=1}^r \sum_{i=0}^{2(r-j)} \binom{m-1-i}{j-1} \\
&\stackrel{(b)}{=} \sum_{i=0}^{2(r-1)} \sum_{j=1}^{r-\lceil i/2 \rceil} \binom{m-1-i}{j-1} \\
&= \sum_{i=0}^{2(r-1)} k(r-1-\lceil i/2 \rceil, m-1-i).
\end{aligned}$$

In the above chain of equalities, equality (a) uses the fact that for integers  $a < b$  and  $j \geq 1$ , we have  $\binom{b}{j} - \binom{a}{j} = \sum_{q=a}^{b-1} \binom{q}{j-1}$ ; this is just repeated application of the identity  $\binom{b}{j} = \binom{b-1}{j-1} + \binom{b-1}{j}$ . Equality (b) is obtained by exchanging the order of the summations in  $i$  and  $j$ .

Case 2:  $m = 2r$ . Here,

$$\begin{aligned}
k(r, m) - \tau(r, m) &= \sum_{j=0}^r \binom{m}{j} - 1 - \sum_{j=0}^{r-1} \binom{m-2j-1}{r-j} \\
&= \sum_{j=1}^r \binom{m}{j} - \sum_{j=1}^r \binom{m-2(r-j)-1}{j} \\
&= \sum_{j=1}^r \left[ \binom{m}{j} - \binom{m-2(r-j)-1}{j} \right],
\end{aligned}$$

and now we carry on from equality (a) of Case 1.

Case 3:  $m \leq 2r - 1$ . This is the most tedious case. We start with

$$\begin{aligned}
k(r, m) - \tau(r, m) &= \sum_{j=0}^r \binom{m}{j} - 1 - \sum_{j=0}^{m-r-1} \binom{m-2j-1}{r-j} \\
&= \sum_{j=1}^r \binom{m}{j} - \sum_{j=2r-m+1}^r \binom{m-2(r-j)-1}{j} \\
&= \sum_{j=1}^{2r-m} \binom{m}{j} + \sum_{j=2r-m+1}^r \left[ \binom{m}{j} - \binom{m-2(r-j)-1}{j} \right] \\
&= \sum_{j=1}^{2r-m} \binom{m}{j} + \sum_{j=2r-m+1}^r \sum_{i=0}^{2(r-j)} \binom{m-1-i}{j-1} \\
&= \sum_{j=1}^{2r-m} \binom{m}{j} + \sum_{i=0}^{2(m-r-1)} \sum_{j=2r-m+1}^{r-\lceil i/2 \rceil} \binom{m-1-i}{j-1}. \tag{23}
\end{aligned}$$

Now, for  $j \geq 1$ , write  $\binom{m}{j} = \binom{m}{j} - \binom{0}{j} = \sum_{i=0}^{m-1} \binom{m-1-i}{j-1}$ . Hence,

$$\sum_{j=1}^{2r-m} \binom{m}{j} = \sum_{i=0}^{m-1} \sum_{j=1}^{2r-m} \binom{m-1-i}{j-1}. \tag{24}$$

Also,

$$\sum_{i=0}^{2(m-r-1)} \sum_{j=2r-m+1}^{r-\lceil i/2 \rceil} \binom{m-1-i}{j-1} = \sum_{i=0}^{m-1} \sum_{j=2r-m+1}^{r-\lceil i/2 \rceil} \binom{m-1-i}{j-1}, \tag{25}$$

as when  $i \geq 2(m-r-1) + 1$ , we have  $r - \lceil i/2 \rceil \leq 2r - m$ , so that the inner summation  $\sum_{j=2r-m+1}^{r-\lceil i/2 \rceil}$  is empty. Plugging (24) and (25) into (23), we find that

$$k(r, m) - \tau(r, m) = \sum_{i=0}^{m-1} \sum_{j=1}^{r-\lceil i/2 \rceil} \binom{m-1-i}{j-1} = \sum_{i=0}^{m-1} k(r-1-\lceil i/2 \rceil, m-1-i).$$

This completes the proof of Proposition 7.  $\square$

## Appendix B: Proof of Lemma 10

We recast the statement of Lemma 10 into an equivalent statement about binary representations of integers. From (12) and the notion of points of fall from [2] (see Appendix A), we see that for  $1 \leq s \leq 2^m$ ,  $U_s$  is equal to the number of points of fall of  $\text{RM}(r, m)$  within the interval  $[0, s-1]$ . Thus, by Lemma 13,  $U_s$  is equal to the number of integers in  $[0, s-1]$  whose  $m$ -bit binary representations have at least  $m-r$  1s.

For an integer  $j \in [0, 2^m - 1]$ , let  $\text{wt}(j)$  denote the Hamming weight of (i.e., the number of 1s in) the binary representation  $\mathbf{b}(j)$ . For a subset  $S \subseteq [0, 2^m - 1]$ , let  $w_i(S)$  denote the number of integers  $j \in S$  with  $\text{wt}(j) \geq i$ . We set  $w_i(\emptyset) = 0$ . Then, Lemma 10 is equivalent to the following assertion: for  $i \in [0, \alpha - 2^{m-1}]$  and  $j \in [0, \beta - (\alpha - 2^{m-1} + 1)]$ , we have

$$w_{m-r}([\alpha - i, \alpha - 1]) + w_{m-r}([\beta - j, \beta - 1]) \geq w_{m-r}([1, i + j]). \tag{26}$$

Since Lemma 10 needs to be shown for any  $\text{RM}(r, m)$  with  $0 \leq r \leq m-1$ , we see that (26) must be shown for any  $m-r \in \{1, 2, \dots, m\}$ . With this in mind, we define for  $S \subseteq [0, 2^m - 1]$ ,

$$\mathbf{w}^{(m)}(S) = [w_1(S) \ w_2(S) \ \dots \ w_m(S)].$$

As usual, we will drop the superscript  $(m)$  when it can be gleaned unambiguously from the context.

**Proposition 14.** *For  $m \geq 2$  and  $0 \leq i, j \leq \alpha^{(m)} - 2^{m-1}$ , we have*

$$\mathbf{w}^{(m)}([\alpha^{(m)} - i, \alpha^{(m)} - 1]) + \mathbf{w}^{(m)}([\beta^{(m)} - j, \beta^{(m)} - 1]) \geq \mathbf{w}^{(m)}([1, i + j]), \quad (27)$$

with the inequality above holding componentwise.

Observe that this proposition is slightly stronger than Lemma 10, since the latter only requires  $0 \leq j \leq \beta^{(m)} - (\alpha^{(m)} - 2^{m-1} + 1)$ . It is easy to verify that  $\beta^{(m)} - (\alpha^{(m)} - 2^{m-1} + 1) \leq \alpha^{(m)} - 2^{m-1}$ . The remainder of this appendix is devoted to a proof of Proposition 14. The proof is by induction on  $m$ , which is why we have taken care to include the superscripts on  $\alpha$  and  $\beta$  in the statement of the proposition. The main ingredients in the inductive proof are the simple facts that for a non-negative integer  $j$ ,  $\text{wt}(2j) = \text{wt}(j)$  and  $\text{wt}(2j + 1) = \text{wt}(j) + 1$ . The rest is merely careful bookkeeping.

Let  $P^{(m)}(i, j)$  denote the inequality in (27). The induction argument is built upon certain implications among the  $P^{(m)}(i, j)$ , as stated in the series of lemmas below. We introduce here some notation that we will use in the proofs of these lemmas. For a set of integers  $S$ , we write  $2S$  and  $2S + 1$  to mean the sets  $\{2j : j \in S\}$  and  $\{2j + 1 : j \in S\}$ , respectively. By  $\mathbf{1}_{[a,b]}^{(m)}$ , with  $1 \leq a \leq b \leq m$ , we mean the vector  $[z_1 z_2 \dots z_m]$ , with  $z_i = 1$  for  $a \leq i \leq b$ , and  $z_i = 0$  otherwise. Again, we will drop the superscript  $(m)$  when there is no ambiguity.

**Lemma 15.** *For even  $m$ ,  $P^{(m)}(i, j)$  implies  $P^{(m+1)}(2i + 1, 2j)$ . For odd  $m$ ,  $P^{(m)}(i, j)$  implies  $P^{(m+1)}(2i, 2j + 1)$ .*

*Proof.* For even  $m$ , we have  $\alpha^{(m+1)} = 2\alpha^{(m)} + 1$ , and  $\beta^{(m+1)} = 2\beta^{(m)}$ . Set  $S = [\alpha^{(m)} - i, \alpha^{(m)} - 1]$  and  $T = [\beta^{(m)} - j, \beta^{(m)} - 1]$ . Now,  $P^{(m)}(i, j)$  implies

$$\mathbf{w}^{(m+1)}(2S) + \mathbf{w}^{(m+1)}(2T) \geq \mathbf{w}^{(m+1)}(2[1, i + j]) \quad (28)$$

$$\mathbf{w}^{(m+1)}(2S + 1) + \mathbf{w}^{(m+1)}(2T + 1) \geq \mathbf{w}^{(m+1)}(2[1, i + j] + 1) \quad (29)$$

since  $\text{wt}(2j) = \text{wt}(j)$  and  $\text{wt}(2j + 1) = \text{wt}(j) + 1$  for any non-negative integer  $j$ . Henceforth, all the  $\mathbf{w}$ 's in this proof are  $\mathbf{w}^{(m+1)}$ 's. Combining (28) and (29), we have

$$\mathbf{w}([2\alpha^{(m)} - 2i, 2\alpha^{(m)} - 1]) + \mathbf{w}([2\beta^{(m)} - 2j, 2\beta^{(m)} - 1]) \geq \mathbf{w}([2, 2i + 2j + 1]),$$

which is the same as

$$\mathbf{w}([\alpha^{(m+1)} - 2i - 1, \alpha^{(m+1)} - 2]) + \mathbf{w}([\beta^{(m+1)} - 2j, \beta^{(m+1)} - 1]) \geq \mathbf{w}([2, 2i + 2j + 1]). \quad (30)$$

Now,  $\mathbf{w}([1, 2i + 2j + 1]) = \mathbf{w}([2, 2i + 2j + 1]) + \mathbf{1}_{[1,1]}^{(m+1)}$ . Also,  $\mathbf{w}([\alpha^{(m+1)} - 2i - 1, \alpha^{(m+1)} - 1]) = \mathbf{w}([\alpha^{(m+1)} - 2i - 1, \alpha^{(m+1)} - 2]) + \mathbf{1}_{[1,m/2]}^{(m+1)}$ , since  $\text{wt}(\alpha^{(m+1)} - 1) = \text{wt}(2\alpha^{(m)}) = \text{wt}(\alpha^{(m)}) = m/2$ , by (17). Therefore,

$$\mathbf{w}([a^{(m+1)} - 2i - 1, a^{(m+1)} - 1]) + \mathbf{w}([b^{(m+1)} - 2j, b^{(m+1)} - 1]) \geq \mathbf{w}([1, 2i + 2j + 1]), \quad (31)$$

which is  $P^{(m+1)}(2i + 1, 2j)$ .

The proof for odd  $m$  is along similar lines.  $\square$

**Lemma 16.** *(a) When  $\alpha^{(m)} - i$  is even, the two inequalities  $P^{(m)}(i, j)$  and  $P^{(m)}(i + 2, j)$  together imply  $P^{(m)}(i + 1, j)$ .*

*(b) When  $\beta^{(m)} - j$  is even, the two inequalities  $P^{(m)}(i, j)$  and  $P^{(m)}(i, j + 2)$  together imply  $P^{(m)}(i, j + 1)$ .*

*Proof.* We only prove (a), as the proof of (b) is completely analogous. In this proof, all omitted superscripts are to be taken to be  $(m)$ .

Let  $x = \text{wt}(\alpha - i - 1)$  and  $y = \text{wt}(i + j + 1)$ . We have  $\mathbf{w}([\alpha - i - 1, \alpha - 1]) = \mathbf{w}([\alpha - i, \alpha - 1]) + \mathbf{1}_{[1, x]}$ , and  $\mathbf{w}([1, i + j + 1]) = \mathbf{w}([1, i + j]) + \mathbf{1}_{[1, y]}$ . We want to show  $P^{(m)}(i + 1, j)$ :

$$\mathbf{w}([\alpha - i, \alpha - 1]) + \mathbf{1}_{[1, x]} + \mathbf{w}([\beta - j, \beta - 1]) \geq \mathbf{w}([1, i + j]) + \mathbf{1}_{[1, y]}. \quad (32)$$

If  $x \geq y$ , then  $P^{(m)}(i, j)$  clearly implies (32). So, suppose  $x < y$ . Then, (32) becomes

$$\mathbf{w}([\alpha - i, \alpha - 1]) + \mathbf{w}([\beta - j, \beta - 1]) \geq \mathbf{w}([1, i + j]) + \mathbf{1}_{[x+1, y]},$$

or equivalently,

$$w_l([\alpha - i, \alpha - 1]) + w_l([\beta - j, \beta - 1]) \geq \begin{cases} w_l([1, i + j]) + 1 & \text{if } x + 1 \leq l \leq y \\ w_l([1, i + j]) & \text{otherwise.} \end{cases} \quad (33)$$

Let  $x' = \text{wt}(\alpha - i - 2)$  and  $y' = \text{wt}(i + j + 2)$ . Since  $\alpha - i$  is even, we see that  $x' + 1 = x$  or  $x' < x$ . Now, we have

$$\mathbf{w}([\alpha - i - 2, \alpha - 1]) = \mathbf{w}([\alpha - i, \alpha - 1]) + \mathbf{1}_{[1, x]} + \mathbf{1}_{[1, x']} \quad (34)$$

$$\mathbf{w}([1, i + j + 2]) = \mathbf{w}([1, i + j]) + \mathbf{1}_{[1, y]} + \mathbf{1}_{[1, y']} \quad (35)$$

Thus,  $P^{(m)}(i + 2, j)$  is equivalent to

$$\mathbf{w}([\alpha - i, \alpha - 1]) + \mathbf{1}_{[1, x']} + \mathbf{w}([\beta - j, \beta - 1]) \geq \mathbf{w}([1, i + j]) + \mathbf{1}_{[x+1, y]} + \mathbf{1}_{[1, y']}. \quad (36)$$

Using the fact that  $x' < x$ , (36) implies that for  $x + 1 \leq l \leq y$ ,

$$w_l([\alpha - i, \alpha - 1]) + w_l([\beta - j, \beta - 1]) \geq w_l([1, i + j]) + 1.$$

Since  $P^{(m)}(i, j)$  clearly implies the “otherwise” part of (33), we have shown that  $P^{(m)}(i, j)$  and  $P^{(m)}(i + 2, j)$  together imply (33), i.e.,  $P^{(m)}(i + 1, j)$ .  $\square$

**Lemma 17.** For even  $m$ , the following implications hold:

- (a)  $P^{(m)}(i, j) \implies P^{(m+1)}(2i + 1, 2j)$ ;
- (b)  $P^{(m)}(i - 1, j) \wedge P^{(m)}(i, j) \implies P^{(m+1)}(2i, 2j)$ ;
- (c)  $P^{(m)}(i, j) \wedge P^{(m)}(i, j + 1) \implies P^{(m+1)}(2i + 1, 2j + 1)$ ;
- (d)  $P^{(m)}(i - 1, j) \wedge P^{(m)}(i, j) \wedge P^{(m)}(i - 1, j + 1) \wedge P^{(m)}(i, j + 1) \implies P^{(m+1)}(2i, 2j + 1)$ .

*Proof.* (a) follows directly from Lemma 15.

(b): If  $P^{(m)}(i - 1, j)$  and  $P^{(m)}(i, j)$  are true, then by Lemma 15, we have  $P^{(m+1)}(2i - 1, 2j)$  and  $P^{(m+1)}(2i + 1, 2j)$  being true. Since  $m + 1$  is odd,  $\alpha^{(m+1)}$  is odd (see (17)). It now follows from Lemma 16(a) that  $P^{(m+1)}(2i, 2j)$  holds.

(c): This follows by an argument similar to part (b), except that Lemma 16(b) is applied.

(d): By part (b),  $P^{(m+1)}(2i, 2j)$  and  $P^{(m+1)}(2i, 2j + 2)$  hold. Therefore, by Lemma 16(b),  $P^{(m+1)}(2i, 2j + 1)$  holds.  $\square$

Arguments similar to those used in the above proof show the next result.

**Lemma 18.** For odd  $m$ , the following implications hold:

- (a)  $P^{(m)}(i, j) \implies P^{(m+1)}(2i, 2j + 1);$
- (b)  $P^{(m)}(i, j - 1) \wedge P^{(m)}(i, j) \implies P^{(m+1)}(2i, 2j);$
- (c)  $P^{(m)}(i, j) \wedge P^{(m)}(i + 1, j) \implies P^{(m+1)}(2i + 1, 2j + 1);$
- (d)  $P^{(m)}(i, j - 1) \wedge P^{(m)}(i, j) \wedge P^{(m)}(i + 1, j - 1) \wedge P^{(m)}(i + 1, j) \implies P^{(m+1)}(2i + 1, 2j).$

We are now in a position to prove Proposition 14.

*Proof of Proposition 14.* Set  $\ell^{(m)} = \alpha^{(m)} - 2^{m-1}$ . We wish to show that for  $m \geq 2$ ,  $P^{(m)}(i, j)$  holds for  $0 \leq i, j \leq \ell^{(m)}$ . It is easy to verify this directly for  $m = 2$  and  $m = 3$ , so we start the induction by assuming that for some odd  $m \geq 3$ ,  $P^{(m)}(i, j)$  holds for  $0 \leq i, j \leq \ell^{(m)}$ .

For odd  $m$ , the implications in Lemma 18 are enough to show that  $P^{(m+1)}(i, j)$  holds for  $1 \leq i \leq 2\ell^{(m)}$  and  $1 \leq j \leq 2\ell^{(m)} + 1$ . Note also that for odd  $m$ , we have  $\ell^{(m+1)} = 2\ell^{(m)}$ , as can be verified from (15). Since  $P^{(m+1)}(0, 0)$ ,  $P^{(m+1)}(0, 1)$  and  $P^{(m+1)}(1, 0)$  trivially hold, we have that  $P^{(m+1)}(i, j)$  holds for  $0 \leq i \leq \ell^{(m+1)}$  and  $0 \leq j \leq \ell^{(m+1)} + 1$ .

Now,  $m + 1$  is even, and we have shown above that  $P^{(m+1)}(i, j)$  is true for  $0 \leq i \leq \ell^{(m+1)}$  and  $0 \leq j \leq \ell^{(m+1)} + 1$ . The implications in Lemma 17 are then sufficient to show that  $P^{(m+2)}(i, j)$  holds for  $1 \leq i, j \leq 2\ell^{(m+1)} + 1$ . Again,  $P^{(m+2)}(0, 0)$ ,  $P^{(m+2)}(0, 1)$  and  $P^{(m+2)}(1, 0)$  can be seen to hold trivially, so  $P^{(m+2)}(i, j)$  in fact holds for  $0 \leq i, j \leq 2\ell^{(m+1)} + 1$ . This completes the induction step, since for even  $m + 1$ , it follows from (15) that  $\ell^{(m+2)} = 2\ell^{(m+1)} + 1$ .  $\square$

As observed earlier, Proposition 14 proves Lemma 10.

## Appendix C: Computing $U_\alpha$ and $U_\beta$

To derive the expressions in (19)–(21), we make use of (5) and a result of Wei [15] that explicitly determines the generalized Hamming weight hierarchy of  $\text{RM}(r, m)$ . Any non-negative integer  $u < k(r, m)$  can be uniquely expressed as a sum

$$u = \sum_{i=1}^{\ell} k(r_i, m_i), \quad (37)$$

where  $r > r_1 \geq r_2 \geq \dots \geq r_\ell \geq 0$ ,  $m > m_1 \geq m_2 \geq \dots \geq m_\ell \geq 0$ , and for all  $i$ ,  $m_i - r_i = m - r + 1 - i$  [15, Lemma 2]. The above representation is called the  $(r, m)$ -canonical representation of  $u$ .

**Theorem 19** ([15], Corollary 6). For  $0 \leq u < k(r, m)$ , given the unique  $(r, m)$ -canonical representation of  $u$  as in (37), we have  $d_u(\text{RM}(r, m)) = \sum_{i=1}^{\ell} 2^{m_i}$ .

For convenience, we will henceforth write  $d_u(\text{RM}(r, m))$  simply as  $d_u$ .

Assume that  $m \geq 2r$ . We want to show that (19) holds. We will only prove here the result for  $s = \alpha$ , as the result for  $s = \beta$  can be proved analogously. Let  $\hat{u}$  be the integer given by

$$\hat{u} = \sum_{i=1}^r k(r - i, m + 1 - 2i). \quad (38)$$

Note that the above is the  $(r, m)$ -canonical representation of  $\hat{u}$ . By Theorem 19, we have  $d_{\hat{u}} = \sum_{i=1}^r 2^{m+1-2i}$ . In binary form,  $\mathbf{b}(d_{\hat{u}}) = (0, 0, \dots, 0, 0, 1, 0, 1, \dots, 0, 1)$ , the number of 1s in  $\mathbf{b}(d_{\hat{u}})$  being  $r$ . Comparing this with the binary form of  $\alpha$  given in (17), it is clear that  $d_{\hat{u}} \leq \alpha$ .

Next, write  $\hat{u} + 1$  as

$$\hat{u} + 1 = \sum_{i=1}^r k(r - i, m + 1 - 2i) + k(0, m - 2r),$$

using the fact that  $k(0, m - 2r) = \binom{m-2r}{0} = 1$ . This is again in  $(r, m)$ -canonical form, and hence by Theorem 19, we have  $d_{\hat{u}+1} = \sum_{i=1}^r 2^{m+1-2i} + 2^{m-2r}$ . In binary form, this is  $\mathbf{b}(d_{\hat{u}+1}) = (0, 0, \dots, 0, 1, 1, 0, 1, \dots, 0, 1)$ , the number of 1s here being  $r + 1$ . Comparing with (17), we see that  $\alpha < d_{\hat{u}+1}$ .

Since  $d_{\hat{u}} \leq \alpha < d_{\hat{u}+1}$ , we have by (5),  $U_\alpha = \hat{u}$ . Observe that  $\hat{u}$  as given by (38) is precisely equal to the claimed value of  $U_\alpha$  in (19).

Now, assume  $m < 2r$ . We wish to show (20) and (21). We sketch the proof for (21) here; the proof for (20) is similar. Set

$$\check{u} = \begin{cases} \sum_{i=1}^{\frac{m-1}{2}} k(r - 1 - i, m - 2i) & \text{if } m \text{ is odd,} \\ \sum_{i=1}^{\frac{m}{2}} k(r - 1 - i, m - 2i) & \text{if } m \text{ is even.} \end{cases}$$

The above is the  $(r, m)$ -canonical representation of  $\check{u}$ , and hence,

$$d_{\check{u}} = \begin{cases} \sum_{i=1}^{\frac{m-1}{2}} 2^{m-2i} & \text{if } m \text{ is odd,} \\ \sum_{i=1}^{\frac{m}{2}} 2^{m-2i} & \text{if } m \text{ is even.} \end{cases}$$

Comparing  $\mathbf{b}(d_{\check{u}})$  with  $\mathbf{b}(\beta)$  given in (18), it can be seen that  $d_{\check{u}} \leq \beta$ .

The  $(r, m)$ -canonical representation of  $\check{u} + 1$  is given by

$$\begin{cases} \sum_{i=1}^{\frac{m-1}{2}} k(r - 1 - i, m - 2i) + k(r - 1 - \frac{m-1}{2}, 0) & \text{if } m \text{ is odd,} \\ \sum_{i=1}^{\frac{m}{2}-1} k(r - 1 - i, m - 2i) + k(r - \frac{m}{2}, 1) & \text{if } m \text{ is even.} \end{cases}$$

Again,  $d_{\check{u}+1}$  can be obtained from Theorem 19, and the subsequent comparison of binary forms shows that  $\beta < d_{\check{u}+1}$ . Hence, by (5), we have  $U_\beta = \check{u}$ , which proves (21).

## Acknowledgement

A. Thangaraj thanks Rakesh Pokala for several helpful discussions.

## References

- [1] Y. Berger and Y. Be'ery, "Bounds on the trellis size of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 203–209, 1993.
- [2] T. Blackmore and G.H. Norton, "On trellis structures for Reed-Muller codes," *Finite Fields and Their Applications*, vol. 6, pp. 39–70, 2000.
- [3] H.L. Bodlaender, "A tourist guide through treewidth," *Acta Cybernetica*, vol. 11, pp. 1–23, 1993.
- [4] G.D. Forney Jr., "Codes on graphs: normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.
- [5] G.D. Forney Jr., "Codes on graphs: constraint complexity of cycle-free realizations of linear codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1597–1610, July 2003.

- [6] T.R. Halford and K.M. Chugg, “The extraction and complexity limits of graphical models for linear codes,” *IEEE Trans. Inform. Theory*, vol. 54, no. 9, pp. 3884–3906, Sept. 2008.
- [7] P. Hliněný and G. Whittle, “Matroid tree-width,” *Europ. J. Combin.*, vol. 27, pp. 1117–1128, 2006.
- [8] C. Jordan, “Sur les assemblages des lignes,” *Journal für reine und angewandte Mathematik*, vol. 70, pp. 185–190, 1869.
- [9] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, “On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes,” *IEEE Trans. Inform. Theory*, vol. 39, no. 1, pp. 242–245, Jan. 1993.
- [10] N. Kashyap, “On minimal tree realizations of linear codes,” *IEEE Trans. Inform. Theory*, vol. 55, no. 8, pp. 3501–3519, Aug. 2009.
- [11] N. Kashyap, “Constraint complexity of realizations of linear codes on arbitrary graphs,” *IEEE Trans. Inform. Theory*, vol. 55, no. 11, pp. 4864–4877, Nov. 2009.
- [12] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.
- [13] W.D. Smith and N. Wormald, “Geometric separator theorems and applications,” manuscript, 1998. Available at <http://www.math.uwaterloo.ca/~nwormald/papers/geomsep.ps.gz>.
- [14] A. Vardy, “Trellis Structure of Codes,” in *Handbook of Coding Theory*, R. Brualdi, C. Huffman and V. Pless, Eds., Amsterdam, The Netherlands: Elsevier, 1998.
- [15] V.K. Wei, “Generalized Hamming weights for linear codes,” *IEEE Trans. Inform. Theory*, vol. 37, no. 5, pp. 1412–1418, Sept. 1991.
- [16] N. Wiberg, *Codes and Decoding on General Graphs*, Ph.D. thesis, Linköping University, Linköping, Sweden, 1996.