

# On the Closed-form Weight Enumeration of Polar Codes: $1.5d$ -weight codewords

Mohammad Rowshan<sup>\*†</sup>, Vlad-Florin Drăgoi<sup>\*‡</sup>, and Jinhong Yuan<sup>†</sup>, *Fellow, IEEE*

<sup>†</sup>School of Electrical Eng. and Telecom., University of New South Wales (UNSW), Sydney, Australia

<sup>‡</sup>Faculty of Exact Sciences, Aurel Vlaicu University, Arad, Romania  
m.rowshan@unsw.edu.au, vlad.dragoi@uav.ro, j.yuan@unsw.edu.au

**Abstract**—The weight distribution of error correction codes is a critical determinant of their error-correcting performance, making enumeration of utmost importance. In the case of polar codes, the minimum weight  $w_{\min}$  (which is equal to minimum distance  $d$ ) is the only weight for which an explicit enumerator formula is currently available. Having closed-form weight enumerators for polar codewords with weights greater than the minimum weight not only simplifies the enumeration process but also provides valuable insights towards constructing better polar-like codes. In this paper, we contribute towards understanding the algebraic structure underlying higher weights by analyzing Minkowski sums of orbits. Our approach builds upon the lower triangular affine (LTA) group of decreasing monomial codes. Specifically, we propose a closed-form expression for the enumeration of codewords with weight  $1.5 w_{\min}$ . Our simulations demonstrate the potential for extending this method to higher weights.

**Index Terms**—Polar codes, Reed-Muller codes, monomial codes, lower triangular affine group, algebraic properties, minimum Hamming distance, minimum weight codeword, weight distribution, enumeration, error coefficient.

## I. INTRODUCTION

Polar codes [1] are the first class of channel codes with explicit construction that achieve the symmetric (Shannon) capacity of a binary-input discrete memoryless channel (BI-DMC). As short and medium-length polar codes have shown a remarkable error correction performance, they had been chosen as a coding scheme for logical control channels in the 5th generation of mobile broadband standard [2]. In the past decade, polar codes were the focus of attention in the field of coding theory. The efforts were mostly focused on low-complexity methods for code construction, and performance improvement through code concatenation and decoding schemes. The more fundamental problems such as the algebraic characteristics of polar codes were less investigated. One of the major problems is the algebraic investigation of weight distribution and finding closed-form expressions for the enumeration of the codewords with certain weights.

The weight distribution of a code determines the error correction performance of a code. The upper bound for the block error rate (BLER) of linear codes under maximum likelihood (ML) decoding can be obtained by Union bound where the number of codewords with small weights, such as the

minimum weight, the second minimum weight, and larger ones take the role of coefficient in the largest terms [3, Sect. 10.1]. It was shown in [4] that polar codes along with Reed-Muller codes belong to a larger family of codes named decreasing monomial codes. On top of the polynomial structure that both Reed-Muller and polar codes possess, the decreasing property induces new algebraic properties. Exploited for the first time in [4], [5] they allowed the discovery of new structural properties such as duals of decreasing monomial codes are still decreasing monomial codes, the permutation group of decreasing monomial codes admits as a subgroup, the lower triangular affine group ( $LTA(m, 2)$ ). Having a slightly better understanding of the algebraic structure of polar codes allowed the scientific community to propose practical applications. In [5] the permutation group revealed the complete structure of the minimum weight codewords of decreasing monomial codes, thus implicitly of polar codes. In [6] a sub-linearity code construction was proposed. The permutation group has a significant contribution in parallelized decoding of polar codes [7]–[13]. Also, the permutation group was used as an optimisation tool in the weight enumeration algorithm of Yao, Fazeli and Vardy [14].

Finding closed-form expressions for the weight distribution of Reed-Muller codes is still an open problem after more than five decades since notable progress was made towards this goal. Kasami and Tokura [15] characterized the codewords of Reed-Muller codes of weight up to twice the minimum weight of the code. The results were extended to the enumeration of codewords with weights less the 2.5 times the minimum weight in [16]. In the case of polar codes, the first progress was made by the closed-form enumeration of minimum weight codewords in [5] and it was stopped there.

Due to the importance of the weight distribution of polar codes, the rest of the attempts were focused on the algorithmic solutions for either exact or approximate enumeration of polar codewords with various weights. In [17], the authors proposed to send the all-zero codeword over a channel with low noise or to receive at very high SNRs, and count the re-encoded messages with certain weights at the output of a successive cancellation list decoder with very large list size. The method presented in [18] suggests computing a probabilistic weight distribution expression efficiently. The authors in [19] proposed a way to get an approximate distance spectrum of polar

<sup>\*</sup>These authors contributed equally (Corresponding author: Vlad-Florin Drăgoi).

codes with large lengths using the spectrum of short codes and a probabilistic assumption on appearing ones in codewords. Based on the weight distribution of  $|u|u + v|$  constructed codes in [20], the weight distribution of the words generated by polar transform was found recursively in [21]. This work and the work in [5] inspired the authors of [14] to propose a deterministic recursive algorithm to count all the codewords of polar codes with any weights although due to high complexity, it cannot be used for medium and long block-lengths.

In this work, we take one step forward and partially address the long-standing problem of weight distribution for polar codes by enumerating codewords of weight  $1.5 w_{\min}$ . Our result applies to any decreasing monomial code. In order to address this problem we need to understand that closed formulae for enumeration of codewords of a given weight are indeed complex and challenging problems for polar codes. Dealing with the general case of decreasing monomial codes has several advantages and could unify old results (the case of Reed-Muller codes) with the case of polar codes. In order to achieve our goal we built our results on two key ingredients which are the permutation group  $(LTA(m, 2))$  and a classification theorem by Kasami and Tokura for weight smaller than  $2 w_{\min}$  [15]. The theorem in [15] applies to any polynomial code and gives the polynomial shape (up to linear affine transformation) of any codeword of weight smaller than  $2 w_{\min}$ . Such codewords can be defined as sums of minimum weight codewords, which are defined as the evaluation of polynomials in orbits  $LTA(m, 2) \cdot f$  for a monomial  $f$  of maximum degree, let's say  $\deg(f) = r$ . Hence, the weight of such a sum is given by the number of positions on which two minimum weight codewords overlap. This leads us to one of the key ingredients in our quest, i.e., the understanding of the Minkowski sum of two orbits  $LTA(m, 2) \cdot f + LTA(m, 2) \cdot g$  where  $f, g$  are monomials of maximum degree ( $f, g$  define minimum weight codewords).

The initial step involves transposing the findings from [15] to the scenario of decreasing monomial codes. To accomplish this, we will establish the following:

- find the subgroups that generate the complete orbits for sums and product of sums;
- given a codeword  $c$  of weight  $1.5 w_{\min}$ , compute the maximum degree monomials, say  $f$  and  $g$  s.t.  $c$  is the evaluation of a polynomial that belongs to the set  $LTA(m, 2) \cdot f + LTA(m, 2) \cdot g$ .

The next step will be to determine the cardinal of a Minkowski sum of orbits. We demonstrate that if there is a particular order relation on the variables of  $f$  and  $g$  then, the cardinal of  $LTA(m, 2) \cdot f + LTA(m, 2) \cdot g$  is equal to the product of cardinals of the two sets  $LTA(m, 2) \cdot f$  and  $LTA(m, 2) \cdot g$ . To prove our result we will introduce the concept of polynomial collision. Informally, two pair of distinct polynomials  $(P, Q), (P^*, Q^*) \in LTA(m, 2) \cdot f \times LTA(m, 2) \cdot g$  form a collision if  $P + q = P^* + Q^*$ . We demonstrate that the existence of a collision at the level of monomials ( $f, g$ ), by a factorization procedure (which is possible due

to the decreasing order relation), can be transposed down to  $(f/h, g/h)$  where  $h = \gcd(f, g)$ . Next, we characterize all possible cases of collisions and thus retrieve the set of all invariants. A direct consequence of our result is that we give a simple formula to compute the cardinal of a Minkowski sum  $LTA(m, 2) \cdot f + LTA(m, 2) \cdot g$  for  $f$  and  $g$  of degree 2.

While the aforementioned results are rather theoretical, we do apply them for a practical challenge. We determine closed formulae for weight  $1.5 w_{\min}$ . Let's start by recalling that words of weight  $1.5 w_{\min}$  belong to sum of orbits of some monomials  $f$  and  $g$  of maximum degree  $\deg(f) = \deg(g) = r$  and such that  $\deg(\gcd(f, g)) = r - 2$ . We demonstrate that distinct pairs of such monomials define disjoint Minkowski sums (property analogue to disjoint orbits for minimum weight codewords). Finally, we give a formula for counting such codewords and test it for different polar codes. We retrieve well-known results for small-length codes as well as the formulas for Reed-Muller codes. We push a bit further our simulations and provide weight counting for large polar codes.

## II. PRELIMINARIES

### A. Basic Concepts in Coding Theory and Notations

We denote by  $\mathbb{F}_2$  the finite field with two elements and by  $\oplus$  the addition operator in this field. Also, subsets of consecutive integers are denoted by  $[\ell, u] \triangleq \{\ell, \ell+1, \dots, u\}$ , and by  $[n] \triangleq [0, n-1]$ . The *support* of a vector  $c = [c_0, \dots, c_{N-1}] \in \mathbb{F}_2^N$  is the set of indices where  $c$  has a nonzero coordinate, i.e.  $\text{supp}(c) \triangleq \{i \in [N] \mid c_i \neq 0\}$ . The cardinality of a set is denoted by  $|\cdot|$  and the set difference by  $\setminus$ . The *Hamming weight* of a vector  $c \in \mathbb{F}_2^N$  is  $w(c) \triangleq |\text{supp}(c)|$ . Given two vectors  $c = [c_0, c_1, \dots, c_{N-1}]$  and  $c' = [c'_0, c'_1, \dots, c'_{N-1}]$ , the *Hamming distance* between  $c$  and  $c'$  is defined to be the number of coordinates where  $c$  and  $c'$  differ, namely,  $d(c, c') = |\{i \in [N] \mid c_i \neq c'_i\}|$ .

A  $K$ -dimensional subspace  $\mathcal{C}$  of  $\mathbb{F}_2^N$  is called a linear  $(N, K, d)$  *code* over  $\mathbb{F}_2$  if the minimum distance of  $\mathcal{C}$ ,

$$d_{\min} = d(\mathcal{C}) \triangleq \min_{c, c' \in \mathcal{C}, c \neq c'} d(c, c') = d.$$

It is easy to see that the Hamming norm induces the Hamming distance and vice-versa. Hence, we have (see [3, Section 3.3])

$$w_{\min} \triangleq \min_{c \in \mathcal{C}, c \neq 0} w(c) = d(\mathcal{C}).$$

We usually use the short notation  $(N, K)$  for codes where we refer to  $N$  and  $K$  as the *length* and the *dimension* of the code. The vectors in  $\mathcal{C}$  are called *codewords*. We also collect all codewords of code  $\mathcal{C}$  with weight  $w$  in set  $W_w$  as

$$W_w(\mathcal{C}) = \{c \in \mathcal{C} \mid w(c) = w\}.$$

A *generator matrix*  $\mathbf{G}$  of an  $(N, K)$  code  $\mathcal{C}$  is a  $K \times N$  matrix in  $\mathbb{F}_2^{K \times N}$  whose rows are  $\mathbb{F}_2$ -linearly independent codewords of  $\mathcal{C}$ . Then  $\mathcal{C} = \{v\mathbf{G} : v \in \mathbb{F}_2^K\}$ .

Under a binary input additive white Gaussian noise (BI-AWGN) channel at a high signal-to-noise ratio (SNR) per information bit  $E_b/N_0$ , according to [3, Sect. 10.1], the

block error rate (BLER) of linear codes under soft-decision maximum likelihood (ML) decoding is upper bounded by

$$P_e^{ML} \leq \sum_{w=w_{\min}}^N A_w Q(\sqrt{2w \cdot R \cdot E_b/N_0}), \quad (1)$$

where  $A_w$  denotes the number of  $w$ -weight codewords (or equivalently  $A_w = |W_w(\mathcal{C})|$ ),  $Q(\cdot)$  is the tail probability of the normal distribution  $\mathcal{N}(0, 1)$  to find the pairwise error probability for a transmitted sequence and its corresponding distorted received one, and  $R \triangleq K/N$  is the code rate. In the literature,  $A_{w_{\min}}$  is known as *error coefficient* since it is the coefficient of the largest term for calculating the BLER upper bound. One can observe that a code with smaller  $A_{w_{\min}}$  is expected to provide a smaller BLER than a code with larger  $A_{w_{\min}}$ , assuming they both have identical  $w_{\min}$ . Note that this paper considers other dominant terms to give a more accurate measure of a code by finding  $A_w$  for all  $w < 2w_{\min}$ , in particular for  $w = 1.5w_{\min}$ .

### B. Monomial Codes

A monomial is a single-term algebraic expression indicating the product of any subset of variables in  $\mathbf{x} \triangleq (x_0, \dots, x_{m-1})$ . Assuming  $\mathbf{i} = (i_0, \dots, i_{m-1})$  where  $i_j \in \{0, 1\}$ , an  $m$ -variate monomial is denoted as

$$\mathbf{x}^{\mathbf{i}} = \prod_{j=0}^{m-1} x_j^{i_j} = x_0^{i_0} \cdots x_{m-1}^{i_{m-1}}.$$

For simplicity, we denote a monomial by  $f = \mathbf{x}^{\mathbf{i}}$ . Moreover, let us denote the degree of a monomial  $\mathbf{x}^{\mathbf{i}}$  as  $\deg(\mathbf{x}^{\mathbf{i}})$  (we have  $\deg(\mathbf{x}^{\mathbf{i}}) = w(\mathbf{i})$ ) and the set of all monomials by

$$\mathcal{M}_m \triangleq \{\mathbf{x}^{\mathbf{i}} \mid \mathbf{i} \in \mathbb{F}_2^m\}.$$

For any monomial  $f \in \mathcal{M}_m$  of degree  $1 \leq s \leq m$  denoted as  $f = x_{l_1} \dots x_{l_s}$  where  $0 \leq l_1 \leq l_2 \leq \dots \leq l_s \leq m-1$ , the *support of monomial* is denoted as  $\text{ind}(f) = \{l_1, \dots, l_s\}$ . Observe that the support of a monomial  $f = x_0^{f_0} \dots x_{m-1}^{f_{m-1}}$  includes all  $j$  where  $f_j = 1$ . The degree induces a ranking on any monomial set  $\mathcal{I} \subseteq \mathcal{M}_m$ , i.e.,  $\mathcal{I} = \bigcup_{j=0}^m \mathcal{I}_j$ , where  $\mathcal{I}_j = \{f \in \mathcal{I} \mid \deg(f) = j\}$ .

Since we are interested in evaluations of monomials over entries in  $\mathbb{F}_2^m$ , we will identify  $x_i$  with  $x_i^2$  and work in the ring  $\mathbf{R}_m = \mathbb{F}_2[x_0, \dots, x_{m-1}]/(x_0^2 - x_0, \dots, x_{m-1}^2 - x_{m-1})$ . The main reason of considering the aforementioned ideal is because we do not want to carry useless powers when multiplying polynomials or monomials. It will all become much clearer after explaining the connection between polynomials and codewords. Now, any subset  $\mathcal{I} \subseteq \mathcal{M}_m$  forms a *generating set* for a  $(n = 2^m, k = |\mathcal{I}|)$  monomial code  $\mathcal{C}$ . This monomial code  $\mathcal{C}$  as a linear code is the vector subspace  $\mathcal{C}(\mathcal{I}) \subseteq \mathbb{F}_2^n$  generated by the span of the row vectors as *basis* resulting from the evaluation of the monomials in  $\mathcal{I}$  [5]. That is,

$$\mathcal{C}(\mathcal{I}) = \text{span}(\{\text{ev}(f) \mid f \in \mathcal{I}\}),$$

where  $\text{ev}(f)$  is the binary vector obtained by evaluating of monomial  $f$  over all the binary entries in  $\mathbb{F}_2^m$ , i.e.,

$$\text{ev}(f) \triangleq (\text{ev}(f)(0, \dots, 0), \dots, \text{ev}(f)(1, \dots, 1)),$$

where the set  $\mathbb{F}_2^m$  is ordered in decreasing order (see Example 1 for  $m = 2$ ). More exactly, order the entries  $\mathbf{i} = (i_0, \dots, i_{m-1}) \in \mathbb{F}_2^m$  s.t. the left-most entry is the least significant bit (as the index of entries indicates) in the binary representation of  $\mathbf{i}$  and sort them in decreasing order. Since  $\text{ev}(f)$  is a binary vector its Hamming weight is  $w(\text{ev}(f))$  or simply  $|f|_m$  and sometimes called the weight of  $f$ . We do have for any  $f \in \mathcal{M}_m$ ,  $w(\text{ev}(f)) = 2^{m-\deg(f)}$ . Indeed, the points of evaluation where  $\text{ev}(f) = 1$  are defined by the subset  $\{(f_0, \dots, f_{m-1}) \in \mathbb{F}_2^m \mid \forall i \in \text{ind}(f) f_i = 1\}$ . This set has cardinality  $2^{m-|\text{ind}(f)|} = 2^{m-\deg(f)}$ , since there are  $m - \deg(f)$  free indices from  $\mathbb{F}_2$ .

**Example 1.** Let  $m = 2$ . The evaluation  $\text{ev}(f)$  of all monomials  $f \in \mathcal{M}_2 = \{\mathbf{1}, x_0, x_1, x_0x_1\}$  and their corresponding rows in  $\mathbf{G}_N$  is as follows:

	$(i_0 \ i_1) :$	11	01	10	00
$\mathbf{g}_0$	$\text{ev}(x_0x_1)$	1	0	0	0
$\mathbf{g}_1$	$\text{ev}(x_1)$	1	1	0	0
$\mathbf{g}_2$	$\text{ev}(x_0)$	1	0	1	0
$\mathbf{g}_3$	$\text{ev}(1)$	1	1	1	1

where, for instance, evaluation of  $f = x_0$  gives 1 for all  $(1, i_2)$ ,  $i_2 \in \{0, 1\}$  and 0 otherwise.

Since the function  $\text{ev}$  defines a vector space isomorphism between  $\mathbf{R}_m$  and  $\mathbb{F}_2^n$  (see Corollary 3,2,6 in [22]) then given a monomial code  $\mathcal{C}(\mathcal{I})$  with generator matrix  $\mathbf{G} = (\mathbf{g}_0 \cdots \mathbf{g}_k)^T$ , for every row  $\mathbf{g}_i$  where  $i \in [0, k)$  it exists a monomial  $f \in \mathcal{I}$  such that  $\text{ev}(f) = \mathbf{g}_i$ . The minimum distance of a monomial code is then [5]

$$\min_{\mathbf{c} \in \mathcal{C}(\mathcal{I})} w(\mathbf{c}) = \min_{f \in \mathcal{I}} w(\text{ev}(f)) = 2^{m-r^+(\mathcal{C})},$$

where  $r^+(\mathcal{C}) = \max_{f \in \mathcal{I}} \deg(f)$ .

With the formalism defined above, the Reed-Muller code  $\mathcal{R}(r, m)$  is a monomial code

$$\mathcal{R}(r, m) \triangleq \text{span}(\{\text{ev}(f) \mid f \in \mathcal{M}_m, \deg(f) \leq r\}).$$

We will also require to define the concept of sums/products of polynomial sets.

**Definition 1.** Given two polynomial sets  $\mathcal{S}, \mathcal{T} \in \mathbf{R}_m$  their *Minkowski sum* is  $\mathcal{S} + \mathcal{T} = \{s + t \mid s \in \mathcal{S}, t \in \mathcal{T}\}$ . Also, the product is defined as  $\mathcal{S} \cdot \mathcal{T} = \{s \cdot t \mid s \in \mathcal{S}, t \in \mathcal{T}\}$ , where  $+$  and  $\cdot$  stand for the polynomial addition and multiplication.

### III. DECREASING MONOMIAL CODES: THE ALGEBRAIC FORMALISM BEHIND POLAR CODES

#### A. Polar Codes

Polar codes of length  $N = 2^n$  are constructed based on the  $n$ -th Kronecker power, denoted by  $(\cdot)^{\otimes n}$ , of binary Walsh-Hadamard matrix  $\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ , that is,  $\mathbf{G}_N = \mathbf{G}_2^{\otimes n}$  which we

call it *polar transform* throughout this paper. We denote polar transform by rows  $\mathbf{g}_i, i = [N]$  as  $\mathbf{G}_N = [\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{N-1}]^T$  where operator  $T$  in  $[\cdot]^T$  denotes the transpose of the matrix. The generator matrix of a polar code is formed by selecting a set of rows of  $\mathbf{G}_N$ . We use  $\mathcal{A}$  to denote the set of indices of these rows and  $\mathcal{C}(\mathcal{A})$  to denote the linear code generated by the set of rows of  $\mathbf{G}_N$  indexed by  $\mathcal{A}$ . Note that  $\mathcal{A} \subseteq [N]$ . The characterization of the information set  $\mathcal{A}$  for polar codes relies on the channel polarization theorem [1] and the concept of *synthetic channel reliability*. A polar code of length  $N = 2^n$  is constructed by selecting a set  $\mathcal{A}$  of indices  $i \in [0, N - 1]$  with high reliability [1]. The indices in  $\mathcal{A}$  are dedicated to information bits, while the rest of the synthetic channels with indices in  $\mathcal{A}^c \triangleq [0, N - 1] \setminus \mathcal{A}$  are used to transmit a known value, '0' by default, which are called *frozen bits*. Regardless of the method we use for forming the set  $\mathcal{A}$  for a polar code, every synthetic channel represented by index  $i$ , denoted by  $W_n^i$  where  $i \in \mathcal{A}$  must be more reliable than any synthetic channels in  $\mathcal{A}^c$ .

Polar codes with the information set  $\mathcal{A}$  can also be considered as monomial codes [22] where the relation between the generating set of monomials  $\mathcal{I} \subset \mathcal{M}_n$  discussed in Section II-B and the information set  $\mathcal{A}$  is as follows:

$$\forall f \in \mathcal{I}, \exists i \in \mathcal{A}, \text{ where } \text{supp}(i) = [n] \setminus \text{ind}(f). \quad (2)$$

Accordingly, we can define set  $\mathcal{A}_{m-r}$  equivalent to  $\mathcal{I}_r$ , collecting the row indices with  $|\text{supp}(i)| = m - r$  for  $i \in \mathcal{A}_{m-r}$  where we have  $m = n$  for polar codes while  $m$  is used for decreasing monomial codes in general throughout this paper. The rows of matrix  $\mathcal{G}_N = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{\otimes n}$  represent all possible evaluations of monomials over  $\mathbb{F}_2^n$ . The relation between  $i$  and  $f$  is hence defined as

$$i = \sum_{j \in [n] \setminus \text{ind}(f)} 2^j. \quad (3)$$

Following (3), let  $\bar{f}$  defined by  $\text{ind}(\bar{f}) = [n] \setminus \text{ind}(f)$  represent a row of  $\mathbf{G}_N$ . For instance, row  $\bar{f} = x_4 x_3 x_1 \rightarrow (00101)_2 = 5$  is equivalent to monomial  $f = x_2 x_0 \rightarrow (11010)_2 = 26$ . We may use  $\bar{f}$  and its decimal equivalent interchangeably. Note that due to the advantage of simplifying the polynomial formalism, we slightly depart from the usual convention for polar codes which is to use in the Kronecker product of  $\mathbf{G}_2$ . Instead, we use  $\mathcal{G}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ . It is easy to see that the two definitions (ours and the conventional one) are equivalent, they just amount to rearranging the code positions.

### B. Decreasing Monomial Codes

For the first time, it was shown in [4] that polar codes, similar to Reed-Muller codes, belong to a large family of codes called Decreasing Monomial Codes. Furthermore, the monomial order was partially characterized in [5] and [23] which is called (universal) partial order in the context of sub-channels relation in terms of reliability. The fundamental algebraic relation between Reed-Muller codes and polar codes goes beyond their monomial structure. It is well known that

both polar and Reed-Muller monomial sets ( $\mathcal{I}$ ) obey a certain order relation denoted  $\preceq$ , as it was shown in [5] and [23].

Let us define the partial order over a set of monomials. We establish the relation  $\preceq$  and  $\preceq_w$  between two monomials  $f$  and  $g$  with the same degree and different degrees in the following. Note that the “ $w$ ” in  $\preceq_w$  stands for “weak” (as in weak order) compared to  $\preceq$  in the sense that any pair of monomials  $f, g$  that satisfy the relation  $f \preceq_w g$ , also satisfy the relation  $f \preceq g$ , by definition. We will use the “|” symbol to denote division between monomials, i.e.,  $f|g$  iff  $\text{ind}(f) \subseteq \text{ind}(g)$ . Naturally, we have the greatest common division of two monomials  $\text{gcd}(f, g) = h$  with  $\text{ind}(h) = \text{ind}(f) \cap \text{ind}(g)$ .

**Definition 2.** Let  $m$  be a positive integer and  $f, g \in \mathcal{M}_m$ . Then  $f \preceq_w g$  if and only if  $f|g$ . When  $\text{deg}(f) = \text{deg}(g) = s$  say that  $f \preceq_{sh} g$  if  $\forall 1 \leq \ell \leq s \quad i_\ell \leq j_\ell$ , where  $f = x_{i_1} \dots x_{i_s}, g = x_{j_1} \dots x_{j_s}$ .

Define  $f \preceq g$  iff  $\exists g^* \in \mathcal{M}_m$  s.t.  $f \preceq_{sh} g^* \preceq_w g$ .

**Remark 1.** One might have  $g^* = g$  in the definition of  $\preceq$ . In such cases if  $f \preceq g$  and  $\text{deg}(f) = \text{deg}(g)$  we have  $\preceq = \preceq_{sh}$ .

The notation  $f \preceq_{sh} g$  comes the fact that one could obtain  $g$  from  $f$  by positively shifting some of the variables in  $f$ . For example  $x_2 x_3 \preceq_{sh} x_2 x_6$  since  $x_6$  is a shift by 3 positions of  $x_3$ .

Remark that there is a chain relation on the variables, i.e.,  $x_0 \preceq x_1 \preceq \dots \preceq x_{m-1}$ . Also, the  $\preceq$  is a order relation that is partial, e.g.,  $x_3 x_4$  and  $x_1 x_5$  are not comparable with respect to  $\preceq$ . The monomial sets studied in this paper are all decreasing monomial sets.

**Definition 3.** A set  $\mathcal{I} \subseteq \mathcal{M}_m$  is *decreasing* if and only if ( $f \in \mathcal{I}$  and  $g \preceq f$ ) implies  $g \in \mathcal{I}$ .

If we map every monomial  $f \in \mathcal{M}_n$  to the corresponding synthetic channel denoted by  $W_n^f$  in the context of polar codes, the relation between every pair of synthetic channels in terms of channel reliability can be established as follows: Let  $f$  and  $g$  be two monomials such that  $f \preceq_w g$ , then according to [22, Proposition 3.3.29], we have  $W_n^g \preceq_d W_n^f$ . Here, ‘ $d$ ’ in  $\preceq_d$  indicates that the channel  $W_n^g$  is a degradation of  $W_n^f$ . Now, as a polar code is generated by the set of monomials  $\mathcal{I} \subset \mathcal{M}_n$ , if  $g \in \mathcal{I}$ , then it implies that  $f$  also belongs to  $\mathcal{I}$ .

### C. Permutation Group

The set of permutations that map codewords of a code  $\mathcal{C}$  to other codewords, i.e., leave the code invariant, forms the *automorphism group* of the code  $\mathcal{C}$ , which is denoted by  $\text{Aut}(\mathcal{C})$ . Hence, A permutation  $\pi$  is an automorphism of code  $\mathcal{C}$  if and only if for every  $c \in \mathcal{C}$ , we have  $\pi(c) \in \mathcal{C}$ .

A bijective affine transformation over  $\mathbb{F}_2^m$  is represented by a pair  $(\mathbf{B}, \varepsilon)$  where  $\mathbf{B} = (b_{i,j})$  is an invertible matrix lying in the general linear group  $\text{GL}(m, 2)$  and  $\varepsilon$  in  $\mathbb{F}_2^m$ . The action of  $(\mathbf{B}, \varepsilon)$  on a monomial  $g = \prod_{i \in \text{ind}(g)} x_i$  denoted by  $(\mathbf{B}, \varepsilon) \cdot g$  replaces each variable  $x_i$  of  $g$  by a variable  $y_i$  as

$$y_i = x_i + \sum_{j=0}^{i-1} b_{i,j} x_j + \varepsilon_i,$$

where  $b_{i,j}$  and  $\varepsilon_i$  are in  $\mathbb{F}_2$ . This new variable  $y_i$ , is in fact a linear form (a polynomial in which all terms have a degree at most 1). Also, the maximum variable of this linear form is  $x_i$  as others are smaller than  $x_i$  w.r.t. the order relation  $\preceq$ .

For decreasing monomial codes, a lower triangular affine transformation denoted by  $LTA(m, 2)$  is employed where  $\mathbf{B} \in GL(m, 2)$  is a lower triangular binary matrix with  $b_{i,i} = 1$  and  $b_{i,j} = 0$  whenever  $j > i$ . Hence, the lower triangular affine group  $LTA(m, 2)$  can be expressed as the following mapping from  $\mathbb{F}_2^m$  to itself.

$$\mathbf{x} \rightarrow \mathbf{B}\mathbf{x} + \boldsymbol{\varepsilon},$$

where the matrix multiplication represents linear maps, and vector addition represents translations. For special sub-classes of decreasing monomial code, such as the Reed-Muller codes the complete permutation group is known. Indeed, the general affine group is the complete permutation group of any  $\mathcal{R}(r, m)$  where  $1 \leq r < m - 1$ .

**Example 2.** Let  $g = x_1x_4$  for  $m = 5$ . Then we have the mapping  $\mathbf{x} \rightarrow \mathbf{B}\mathbf{x} + \boldsymbol{\varepsilon}$  as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ b_{1,0} & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ b_{4,0} & b_{4,1} & b_{4,2} & b_{4,3} & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon_1 \\ 0 \\ 0 \\ \varepsilon_4 \end{bmatrix}$$

The set of polynomials resulting from the action of  $LTA(m, 2)$  on a monomial is collected in a set named orbit.

**Definition 4.** The orbit of a monomial  $f$  under the action of  $LTA(m, 2)$  is defined as the set of polynomials

$$LTA(m, 2) \cdot f = \{(\mathbf{B}, \boldsymbol{\varepsilon}) \cdot f \mid (\mathbf{B}, \boldsymbol{\varepsilon}) \in LTA(m, 2)\}.$$

Since  $LTA(m, 2)$  acts as a permutation on  $ev(f)$ , all the elements in  $LTA(m, 2) \cdot f$  have the same Hamming weight. This fact will be useful when estimating the number of minimum weight codewords of a decreasing monomial code.

#### D. Minimum Weight Codewords

Before focusing on larger weights, let us review how the minimum weight codewords are counted. In essence, the authors in [5] use the action of  $LTA(m, 2)$  on the subset  $\mathcal{I}_r$  where  $r$  is the maximum degree of monomials in a decreasing monomial set  $\mathcal{I}$ . The problem is that when we consider the complete group  $LTA(m, 2)$  one needs to determine the stabilizer subgroup for each coset leader. To achieve this goal a particular subgroup of  $LTA(m, 2)$  was defined.

**Definition 5** ([5], [22]). For any  $g \in \mathcal{M}_m$  define  $LTA(m, 2)_g$  as the subgroup of  $(\mathbf{B}, \boldsymbol{\varepsilon}) \in LTA(m, 2)$  by

$$\varepsilon_i = 0 \text{ if } i \notin \text{ind}(g) \quad \text{and} \quad b_{ij} = \begin{cases} 0 & \text{if } i \notin \text{ind}(g) \\ 0 & \text{if } j \in \text{ind}(g). \end{cases}$$

**Example 3.** Let  $g = x_0x_1$ , then by definition we have  $LTA(m, 2) \cdot g = \{(x_0 + \varepsilon_0)(x_1 + a_{1,0}x_0 + \varepsilon_1) \mid \varepsilon_0, a_{1,0}, \varepsilon_1 \in \mathbb{F}_2\}$ . The complete group has 4 distinct elements although as

$\varepsilon_0, a_{1,0}, \varepsilon_1$  can take two values in  $\{0, 1\}$ , we expected  $2^3 = 8$  elements. Four repeated elements leave  $g$  invariant, namely

$$(x_0 + 1)(x_1 + x_0 + 1) = (x_0 + 1)(x_1 + 1) = x_0x_1 + x_1 + x_0 + 1,$$

$$(x_0 + 1)(x_1 + x_0) = (x_0 + 1)x_1 = x_0x_1 + x_1,$$

$$x_0(x_1 + x_0) = x_0(x_1 + 1) = x_0x_1 + x_0,$$

$$x_0(x_1 + x_0 + 1) = x_0x_1.$$

As can be seen, since  $0 \in \text{ind}(g)$ , when  $a_{1,0} = 1$  as highlighted in blue above, the obtained polynomials are identical with another one in the group. Recall that  $x_j^2 = x_j$  in  $\mathbb{F}_2$ . On the other hand, the subgroup  $LTA(m, 2)_g$  gives only  $2^2$  distinct elements since by definition  $a_{1,0} = 0$ . The elements of the subgroup are  $(x_0 + 1)(x_1 + 1)$ ,  $(x_0 + 1)x_1$ ,  $x_0(x_1 + 1)$ , and  $x_0x_1$ .

Notice that for any monomial  $g \in \mathcal{M}_m$  the subgroup action of  $LTA(m, 2)_g$  on any monomial  $f \in \mathcal{M}_m$  is well defined. Also, by definition of  $LTA(m, 2)_g$  we observe that only the variables in  $g$  that are smaller, w.r.t.  $\preceq$ , than the variables in  $f$  are making a contribution in the group action. Let us give an example.

**Example 4.** Take  $g = x_4$  and  $f = x_0x_2$ . Then

$$LTA(m, 2)_g \cdot f = LTA(m, 2) \cdot f.$$

If we let  $g = x_0x_4$  and  $f = x_1x_2x_5$ , then any polynomial from  $LTA(m, 2)_g \cdot f$  can be written as follows  $(x_1 + \varepsilon_1)(x_2 + b_{2,1}x_1 + \varepsilon_2)(x_5 + b_{5,3}x_3 + b_{5,2}x_2 + b_{5,1}x_1 + \varepsilon_5)$ .

What is particular about this subgroup is that it poses two major properties stated in the following theorem.

**Theorem 1** ([5], [22]). Let  $f \in \mathcal{M}_m$ . Then we have

$$LTA(m, 2) \cdot f = LTA(m, 2)_f \cdot f. \quad (4)$$

Moreover, there are no polynomials in  $LTA(m, 2)_f \cdot f$  that are fixed by more than one group element (the identity).

Now, we look at the size of the orbit  $LTA(m, 2)_f \cdot f$  denoted by  $|LTA(m, 2)_f \cdot f|$ . We can break down the action of  $LTA(m, 2)_f$  on  $f$  into two operations:

- Translation: every variable  $x_i$  in the monomial can be translated by a scalar  $\varepsilon_i \in \mathbb{F}_2$  as  $x_i + \varepsilon_i$ . Hence, a monomial  $f$  of degree  $\text{deg}(f)$ , admits as many translations as possibilities for all its variables, which equals  $2^{\text{deg}(f)}$ .
- Linear Mapping: every variable  $x_i$  can be mapped into a "new variable" ( $y_i$ ), which is a linear combination of variable(s)  $x_j$  where  $j \in [i] \setminus \text{ind}(f)$  as  $y_i = x_i + \sum_{j=0, j \notin \text{ind}(f)}^{i-1} b_{i,j}x_j$ . The extra variables considered in  $y_i$  express the degree of freedom we have on  $x_i$ . This will be denoted by  $\lambda_f(i) = |\{j \in [i] \mid j \notin \text{ind}(f)\}|$  and represents the maximum number of variables in the group action on the variable  $x_i$ . The total number of free variables on all  $x_i$  in the support of  $f$  will be  $|\lambda_f(f)| = \sum_{i \in \text{ind}(f)} \lambda_f(i)$ . Then, the total possible

actions for all variables in the support of  $f$  is  $2^{|\lambda_f|}$  (since we are defined over  $\mathbb{F}_2$ ).

In general, we can have  $\text{LTA}(m, 2)_f \cdot g$ , where  $f$  and  $g$  might be different. In such cases, we have

$$|\lambda_f(g)| = \sum_{i \in \text{ind}(g)} \lambda_f(i). \quad (5)$$

When  $g = f$  we shall simplify the notations and use  $\lambda_f$ . Therefore, the cardinality of the orbit of monomial  $f$  under the action of  $\text{LTA}(m, 2)$  is

$$|\text{LTA}(m, 2)_f \cdot f| = 2^{\deg(f) + |\lambda_f|} \quad (6)$$

Example 5 illustrates how this counting procedure works. The previous theorem has a direct consequence, an efficient formula for counting the number of minimum weight codewords. Let us suppose we have a decreasing monomial set  $\mathcal{I}$  with maximum degree monomials  $r$ , i.e.,  $\mathcal{I}_r$  is not trivial. We know that any minimum weight codeword of  $\mathcal{C}(\mathcal{I})$  is of weight  $2^{m-r}$  and more significant any  $c \in \mathcal{C}(\mathcal{I})$  with  $w(c) = 2^{m-r}$  has the following form  $c = \text{ev}(y_1 \dots y_r)$  where  $y_i$  are linear independent forms, i.e.,  $y_i \in \mathbf{R}_m$  with  $\deg(y_i) = 1$  and for any index  $i$  the equation  $y_i = \sum_{1 \leq j \leq r, j \neq i} \varepsilon_j y_j$  admits a single solution over  $\mathbb{F}_2$ , which is the zero vector  $\varepsilon_j = 0, \forall j$ . The counting method relies on the following.

- 1) Any monomial  $f \in \mathcal{I}_r$  will define an orbit under  $\text{LTA}(m, 2)_f$  in which all polynomials evaluate to a minimum weight codeword;
- 2) We know how to count the cardinal of the orbit  $|\text{LTA}(m, 2)_f \cdot f| = |\text{LTA}(m, 2)_f|$  (by Theorem 1);
- 3) For any pair of monomials  $f, g \in \mathcal{I}_r$  the orbits  $\text{LTA}(m, 2)_f \cdot f$  and  $\text{LTA}(m, 2)_g \cdot g$  are disjoint.
- 4) Finally, sum over all monomials  $f \in \mathcal{I}_r$ .

One key ingredient used to demonstrate that the aforementioned procedure retrieves all minimum weight codewords, is the following lemma, demonstrated in Proposition 3.7.12 in [22].

**Lemma 1.** Let  $P = \prod_{j=1}^l y_j$  be a product of  $l$  independent linear forms  $y_i$  each having maximum variables  $x_{i_j}$ . Then  $P$  can be written as  $P = \prod_{j=1}^l y_j^*$  where all maximum variables  $x_{i_j^*}$  in  $y_j$  are pairwise distinct.

Observe that every  $f \in \mathcal{I}_r$  represents a coset leader for coset  $\mathcal{C}_{\bar{f}}$  and the orbit represents the set of core rows where their row combinations along with the balancing rows result in minimum weight codewords [24].

**Example 5.** Let  $m = 8$  and  $\mathcal{I}$  be a decreasing monomial set with  $r = 4$ . Suppose  $f \in \mathcal{I}_r = \{x_1 x_0 (x_3 x_2, x_4 x_2, x_5 x_2, x_4 x_3)\}$ , then the table below illustrates the procedure of finding the total number of codewords with weight  $w_{\min}$ , i.e.,  $|W_{w_{\min}}|$ .

#### IV. STRUCTURAL PROPERTIES OF CODEWORDS WITH WEIGHT $1.5 w_{\min}$

We shall begin this section by recaling a classification result on Reed-Muller codes, which will represent the foundation of

ind( $f$ )		{3, 2, 1, 0}		{4, 2, 1, 0}		{5, 2, 1, 0}		{4, 3, 1, 0}
$(\lambda_f(3), \dots, \lambda_f(0))$		(0, 0, 0, 0)		(1, 0, 0, 0)		(2, 0, 0, 0)		(1, 1, 0, 0)
$ \text{LTA}(m, 2)_f \cdot f $		$2^4$		$2^5$		$2^6$		$2^6$
$ W_{w_{\min}} $		$176$						

our weight enumeration. Kasami et al. characterized codewords of Reed-Muller codes with weights less than twice the minimum weight in [15] and derived explicit formulas for the enumeration of these weights.

**Theorem 2.** [25],[15, Theorem 1] Let  $r < m$  and  $P \in \mathbf{R}_m$  be such that  $\deg(P) \leq r$  with  $0 < w(\text{ev}(P)) < 2^{m+1-r}$ . Then  $P$  is affine equivalent (it can be transformed using an affine transformation) to one of the forms

- 1)  $P = y_1 \dots y_{r-\mu} (y_{r-\mu+1} \dots y_r + y_{r+1} \dots y_{r+\mu})$  where  $m \geq r + \mu, r \geq \mu \geq 3$
- 2)  $P = y_1 \dots y_{r-2} (y_{r-1} y_r + \dots + y_{r+2\mu-3} y_{r+2\mu-2})$  where  $m - r + 2 \geq 2\mu, \mu \geq 2$ .

In both cases  $y_i$  are linear independent forms and  $w(\text{ev}(P)) = 2^{m+1-r} - 2^{m+1-r-\mu}$ .

**Example 6.** Let  $m = 9$  and  $r = 3$ . From the conditions in Thm. 2 we notice that  $3 \leq \mu \leq 3$  (case 1)) and  $4 \leq 2\mu \leq m - r + 2 = 8$  which implies  $\mu \leq 4$  (case 2)). Hence, have

- $P = y_1 y_2 y_3$  which gives  $w(\text{ev}(P)) = w_{\min} = 2^{9-3} = 64$ ;
- $\mu = 2$  (case 2) in Thm. 2),  $P = y_1 (y_2 y_3 + y_4 y_5)$  and we have  $w(\text{ev}(P)) = 2^7 - 2^{7-2} = 128 - 32 = 96$ ;
- $\mu = 3$  (case 1) in Thm. 2),  $P = y_1 y_2 y_3 + y_4 y_5 y_6$  and we have  $w(\text{ev}(P)) = 2^7 - 2^{7-3} = 128 - 16 = 112$ ;
- $\mu = 4$  (case 2) in Thm. 2),  $P = y_1 (y_2 y_3 + y_4 y_5 + y_6 y_7)$  and we have  $w(\text{ev}(P)) = 2^7 - 2^{7-4} = 128 - 8 = 120$ ;

**Remark 2.** While for Reed-Muller codes any affine transformation globally preserves the code, in the general case of decreasing monomial codes, this fact is no longer true. Hence, for decreasing monomial codes applying directly Theorem 2 for counting such codewords is not possible.

In this article we will restrict our analysis to the case  $\mu = 2$  which is equivalent to codewords of weight  $1.5 w_{\min}$ .

**Corollary 1.** Let  $r < m$  be positive integers s.t.  $r = \max_{\text{ev}(P) \in \mathcal{C}} \deg(P)$ . Then any codeword of weight  $1.5 w_{\min}$ , up to an affine transformation, is equal to  $\text{ev}(y_1 \dots y_{r-2} (y_{r-1} y_r + y_{r+1} y_{r+2}))$ .

Tacking a closer look at the shape of the polynomial in Corollary 1 we deduce that any codeword of weight  $1.5 w_{\min}$  is equal to the sum of two minimum weight codewords. Indeed, both  $y_1 \dots y_r$  and  $y_1 \dots y_{r-2} y_{r+1} y_{r+2}$  define minimum weight codewords (as product of  $r$  independent linear forms), and thus they belong to two distinct orbits  $\text{LTA}(m, 2) \cdot f, \text{LTA}(m, 2) \cdot g$ . Hence, our first task is to understand what is the structure of the sum of these two sets.

Recall that the subgroup  $\text{LTA}(m, 2)_f$  on  $f$  generates the exact same orbit as the complete group action  $\text{LTA}(m, 2)$  on

any monomial  $f$ . Since here we are dealing with sums and product of orbits we would like to know how the subgroup property is preserved in this case.

**Lemma 2.** Let  $f, g \in \mathcal{I}_r$  and  $h = \gcd(f, g) \in \mathcal{M}_m$ . Then

$$\begin{aligned} & \text{LTA}(m, 2) \cdot h \cdot \left( \text{LTA}(m, 2) \cdot \frac{f}{h} + \text{LTA}(m, 2) \cdot \frac{g}{h} \right) \\ &= \text{LTA}(m, 2)_h \cdot h \cdot \left( \text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h} \right). \end{aligned}$$

The proof of our result can be found in Appendix B. Let us give an example on how this decomposition works.

**Example 7.** Let  $h = x_0x_6$ ,  $f/h = x_3x_2$  and  $g/h = x_5x_1$ . The polynomial  $P = x_0(x_6 + x_2)((x_3 + x_0)x_2 + x_5(x_1 + x_0))$  belongs to  $\text{LTA}(m, 2) \cdot h \cdot \left( \text{LTA}(m, 2) \cdot \frac{f}{h} + \text{LTA}(m, 2) \cdot \frac{g}{h} \right)$ . By expanding the product we have

$$\begin{aligned} P &= x_0(x_6 + x_2)(x_3 + x_0)x_2 + x_0(x_6 + x_2)x_5(x_1 + x_0) \\ &= x_0(x_6 + x_2)(x_3 + 1)x_2 + x_0(x_6 + x_2)x_5(x_1 + 1) \\ &= x_0(x_6 + x_2)((x_3 + 1)x_2 + x_5(x_1 + 1)) \end{aligned}$$

So,  $P$  is an element of the set  $\text{LTA}(m, 2)_h \cdot h \cdot \left( \text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h} \right)$ .

Moving forward, we propose a classification theorem for codewords of weight  $1.5 w_{\min}$  for decreasing monomial codes. This result is somehow the equivalent of Kasami and Tokuras's classification theorem where affine transformations are allowed, to a classification where transformations from the  $\text{LTA}(m, 2)$  are allowed.

**Theorem 3.** Let  $\mathcal{C}(\mathcal{I})$  be a decreasing monomial code and  $r = \max_{f \in \mathcal{I}} \deg(f)$ . Then any codeword of weight  $1.5 w_{\min}$ , say  $\text{ev}(P)$  is s.t.  $\exists f, g \in \mathcal{I}_r$  with  $P \in \text{LTA}(m, 2)_h \cdot h \cdot \left( \text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h} \right)$ , where  $h = \gcd(f, g)$ , and  $\deg(h) = r - 2$ .

Notice that this result is constructive, i.e., given a codeword  $\mathbf{c} = \text{ev}(P)$  of weight  $1.5 w_{\min}$ , one can compute two monomials  $f, g$  such that  $P$  belongs to the Minkowski sum of their orbits under  $\text{LTA}(m, 2)$ . In the following example we will illustrate how to compute the monomials  $f, g$  given  $P$  (for more details see the proof of the theorem in Appendix C).

**Example 8.** Let  $m = 9, r = 3$ . Any  $P \in \mathbf{R}_m$  with  $w(\text{ev}(P)) = 2^{9+1-3} - 2^{9+1-5} = 96 = 1.5 \times 64 = 1.5 w_{\min}$ . Consider the following cases.

1) one factor, all distinct maximum variables:

$P = (x_4 + x_0)x_2x_1 + (x_5 + x_1)(x_4 + x_0)x_3$ . By Thm. 2  $\text{ev}(P)$  is a  $1.5 w_{\min}$  since once  $P$  is factored we obtain  $P = (x_4 + x_0)(x_2x_1 + (x_5 + x_1)x_3)$ , which has the form  $y_1(y_2y_3 + y_4y_5)$ . By simply tacking the maximum variables in each  $y_i$  we can set  $f = x_4x_2x_1, g = x_5x_4x_3$  and hence  $h = \gcd(f, g) = x_4, f/h = x_2x_1, g/h = x_5x_3$ .

2) one factor, four distinct maximum variables:

$P = (x_4 + x_0)(x_5 + x_2)x_1 + (x_5 + x_1)(x_4 + x_0)x_3$ , once

factored, as in the previous case, it leads to the required form in Theorem 2,  $P = y_1(y_2y_3 + y_4y_5)$ . However, simply setting  $f, g$  using the maximum variables does not work. Indeed, it would lead to  $f = x_5x_4x_1, g = x_5x_4x_1$ , which is not the form stated in Thm. 3. However, reshaping the terms in  $P$  as in the proof of Thm. 3, one gets  $P = (x_4 + x_0)((x_5 + x_1)(x_3 + x_1) + (x_2 + x_1)x_1)$ . Now, taking the maximum variables in each linear independent form leads to  $f = x_5x_4x_3, g = x_4x_2x_1$ , as required in Thm. 3.

3) one factor, three distinct maximum variables:

$P = (x_6 + x_0)(x_5 + x_4)(x_3 + x_1 + 1) + (x_5 + x_0)(x_6 + x_0)(x_3 + 1)$ . Indeed, by Theorem 2, this corresponds to a  $1.5 w_{\min}$ -weight codeword, since one can write  $P = y_1(y_2y_3 + y_4y_5)$  where  $y_i$  are all linear independent forms. If we simply select the maximum variables (in blue) in each  $y_1$  we would get  $f = g = x_6x_5x_3$  which is not really helpful. Cleverly manipulating the polynomial, one obtains  $P = (x_6 + x_0)((x_5 + x_0)x_1 + (x_4 + x_0)(x_3 + x_1 + 1))$ , from which  $f = x_6x_5x_1, g = x_6x_4x_3$ , exactly as in Thm. 3.

For Reed-Muller codes Theorem 2 combined with the general affine group were the two main ingredients for characterizing and counting codewords of weight smaller than  $2 w_{\min}$ . Decreasing monomial codes do not admit the complete general affine group. Hence, our result provides the first step towards understanding how such codewords are formed when we deal with decreasing monomial codes.

Going further, we will determine the cardinality of  $\text{LTA}(m, 2)_h \cdot h \cdot \left( \text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h} \right)$  from Theorem 3. The major challenge here is to estimate  $|\text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h}|$ . It is obvious that the following upper bound [26] holds  $|\text{LTA}(m, 2) \cdot f + \text{LTA}(m, 2) \cdot g| \leq |\text{LTA}(m, 2) \cdot f| |\text{LTA}(m, 2) \cdot g|$ . However, reaching the upper bound is not always the case. Let us start by giving an example.

**Example 9.** Let  $f = x_4x_2x_0$  and  $g = x_3x_2x_1$ . Then, any polynomial  $P \in \text{LTA}(m, 2) \cdot f$  and  $Q \in \text{LTA}(m, 2) \cdot g$  can be written as  $P = (\mathbf{A}, \varepsilon) \cdot f, Q = (\mathbf{B}, \gamma) \cdot g$ , or equivalently

$$\begin{aligned} P &= (x_4 + b_{4,3}x_3 + b_{4,1}x_1 + \varepsilon_4)(x_2 + b_{2,1}x_1 + \varepsilon_2)(x_0 + \varepsilon_0) \\ Q &= (x_3 + a_{3,0}x_0 + \gamma_3)(x_2 + a_{2,0}x_0 + \gamma_2)(x_1 + a_{1,0}x_0 + \gamma_0) \end{aligned}$$

where the scalars  $a_{i,j}, b_{i,j}, \varepsilon_i, \gamma_i \in \mathbb{F}_2$ .

The cardinality of the orbits are  $|\text{LTA}(m, 2) \cdot f| = 2^{3+(2+1+0)} = 2^6$  and  $|\text{LTA}(m, 2) \cdot g| = 2^{3+(1+1+1)} = 2^6$ . Hence, we have

$$|\text{LTA}(m, 2) \cdot f + \text{LTA}(m, 2) \cdot g| \leq 2^{12}.$$

In this example, however, the upper bound is not achieved due to the existence of *collisions*. Let us define the term collision as follows:

**Definition 6.** Collision: Any pair of distinct polynomials  $P, P^* \in \text{LTA}(m, 2) \cdot f$  and  $Q, Q^* \in \text{LTA}(m, 2) \cdot g$  with  $P \neq P^*, Q \neq Q^*$  such that they produce the same sum, that is,  $P + Q = P^* + Q^*$ , results in a collision.

An example for collision is  $P = (x_4 + x_3)x_2x_0$ ,  $Q = x_3x_2x_1$  and  $P^* = x_4x_2x_0$ ,  $Q^* = x_3x_2(x_1 + x_0)$ . Notice that in Example 9, we were able to generate an invariant for  $x_4x_2x_0 + x_3x_2x_1$  by simply inserting in each monomial a variable from the other monomial. It is not always possible to do so.

We need to define the number of collision for two monomials.

**Definition 7.** Let  $f = x_{i_1}x_{i_2}$  and  $g = x_{j_1}x_{j_2}$  with  $\gcd(f, g) = 1$  and  $i_2 > j_2$ . The *degree of collision* of  $f$  and  $g$  is

$$\alpha_{f,g} = \begin{cases} 0 & i_2 > i_1 > j_2 > j_1 \\ 1 & i_2 > j_2 > i_1 > j_1 \\ 2 & i_2 > j_2 > j_1 > i_1 \end{cases}.$$

For any given pair  $(P, Q)$  with  $P \in \text{LTA}(m, 2) \cdot f$  and  $Q \in \text{LTA}(m, 2) \cdot g$ , the quantity  $\alpha_{f,g}$  will allow us to count how many collisions we get for any fixed pair of polynomials. Indeed, as we will demonstrate, this parameter in nothing more than the number of independent linear equations one needs to satisfy for collisions given a pair of monomials  $f, g$ . Hence, it is normal that there is a connection between the structure of  $f$  and  $g$  and this parameter.

**Example 10.** Let's consider two cases separately

- $i_2 > j_2 > j_1 > i_1$  with  $f = x_6x_2, g = x_5x_3$ , which imply  $\alpha = 2$ . For  $P = (x_6 + x_4 + 1)(x_2 + x_1) \in \text{LTA}(m, 2) \cdot f$  and  $Q = (x_5 + x_4 + x_2)(x_3 + x_2 + x_0 + 1) \in \text{LTA}(m, 2) \cdot g$ . We can create two non-trivial distinct pairs  $P^*, Q^*$ 
  - $P^* = (x_6 + x_4 + x_3 + x_2 + x_0)(x_2 + x_1)$  and  $Q^* = (x_5 + x_4 + x_1)(x_3 + x_2 + x_0 + 1)$
  - $P^* = (x_6 + x_5 + x_3 + x_2 + x_1 + x_0)(x_2 + x_1)$  and  $Q^* = (x_5 + x_4 + x_1)(x_3 + x_1 + x_0 + 1)$
- $i_2 > j_2 > i_1 > j_1$  with  $f = x_4x_2, g = x_3x_0$ . We obtain that  $f + g = (x_4 + x_0)x_2 + (x_3 + x_2)x_0$ .

Why is the parameter  $\alpha_{f,g}$  important? Mainly because it will help us count how many polynomials are overcounted.

**Proposition 1.** Let  $\mathcal{I} \subseteq \mathcal{M}_m$  be a decreasing monomial set and  $f = x_{i_1}x_{i_2}$  and  $g = x_{j_1}x_{j_2}$  with  $\gcd(f, g) = 1$  and  $i_2 > j_2$ . Then

$$|\text{LTA}(m, 2) \cdot f + \text{LTA}(m, 2) \cdot g| = \frac{|\text{LTA}(m, 2) \cdot f| \times |\text{LTA}(m, 2) \cdot g|}{2^{\alpha_{f,g}}}, \quad (7)$$

The proof of this result is provided in the appendix D.

While up to this point we have characterized the cardinality of a Minkowski sum of distinct orbits, there is still a last question left unanswered. What is the cardinality of the product  $\text{LTA}(m, 2)_h \cdot h \cdot (\text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h})$ . The answer here is rather obvious, the cardinal of the product set is equal to the product of the cardinal of each set.

**Lemma 3.** Let  $\mathcal{I}$  be a decreasing set with  $r = \max_{f \in \mathcal{I}} \deg(f)$  and  $f, g \in \mathcal{I}_r$  with  $h = \gcd(f, g)$  s.t.  $\deg(h) = r - 2$ . Then

$$\begin{aligned} & |\text{LTA}(m, 2)_h \cdot h \cdot (\text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h})| = \\ & |\text{LTA}(m, 2)_h \cdot h| \times |(\text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h})| \end{aligned} \quad (8)$$

## V. COUNTING FORMULA FOR $1.5 w_{\min}$ WEIGHT CODEWORDS

We have previously demonstrated that any decreasing monomial code with monomial set  $\mathcal{I}$  and maximum degree monomials  $r = \max_{f \in \mathcal{I}} \deg(f)$  will contain codewords of weight  $1.5 w_{\min}$  if and only if there are monomials of degree  $r$  (denote a pair of such monomials  $(f, g)$ ) such that they share  $r - 2$  variables, i.e.,  $\deg(\gcd(f, g)) = r - 2$ . For example, a decreasing monomial code with maximum degree monomials  $x_0x_1x_2, x_0x_1x_3$  will not have codewords of weight  $1.5 w_{\min}$ , while the code with maximum degree variables  $x_0x_1x_2, x_0x_1x_3, x_0x_2x_3, x_0x_1x_4$  will have codewords of weight  $1.5 w_{\min}$  since the pair  $(x_0x_1x_4, x_0x_2x_3)$  has the common factor  $x_0$  of degree  $r - 2 = 1$ .

Hence, from Theorem 3 we deduce the following.

**Corollary 2.** Let  $\mathcal{I}$  be a decreasing monomial set with  $r = \max_{f \in \mathcal{I}} (\deg(f))$ . Then we have

$$\begin{aligned} W_{1.5 w_{\min}} = & \bigcup_{\substack{f, g \in \mathcal{I}_r \\ h = \gcd(f, g) \in \mathcal{I}_{r-2}}} \text{LTA}(m, 2)_h \cdot h \\ & \cdot \left( \text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h} \right) \end{aligned} \quad (9)$$

Next, we will demonstrate that for distinct monomial pairs  $(f, g), (f^*, g^*)$  define disjoint polynomial sets. In a sense, we transpose the property of disjoint orbits from minimum weight codewords to  $1.5 w_{\min}$ .

**Proposition 2.** Let  $\mathcal{I}$  be a decreasing monomial set and let  $(f, g), (f^*, g^*) \in \mathcal{I}_r \times \mathcal{I}_r$  with  $(f, g) \neq (f^*, g^*)$  and  $\deg(h) = \deg(h^*) = r - 2$ , where  $h = \gcd(f, g)$  and  $h^* = \gcd(f^*, g^*)$ . Then the sets  $\text{LTA}(m, 2)_h \cdot h \cdot (\text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h})$  and  $\text{LTA}(m, 2)_{h^*} \cdot h^* \cdot (\text{LTA}(m, 2)_{f^*} \cdot \frac{f^*}{h^*} + \text{LTA}(m, 2)_{g^*} \cdot \frac{g^*}{h^*})$  are disjoint.

We are now in possession of three ingredients that mixed together will give us a closed formula for  $1.5 w_{\min}$  weight codewords of any decreasing monomial code, and thus, implicitly of polar and ReedMuller codes. Let us recall them and then state our result.

- any  $1.5 w_{\min}$  weight codeword is the evaluation of a polynomial that belongs to a Minkowski sum of two orbits (Theorem 3)
- two distinct pair of degree  $r$  monomials generate distinct orbits (Proposition 2)
- we know how to count the cardinality of Minkowski sums of orbits (Proposition 1)



**Theorem 4.** Let  $\mathcal{I}$  be a decreasing monomial set and  $r = \max_{f \in \mathcal{I}} \deg(f)$ .

$$|W_{1.5 w_{\min}}| = \sum_{\substack{f, g \in \mathcal{I}_r \\ h = \gcd(f, g) \in \mathcal{I}_{r-2}}} 2^{r+2+|\lambda_h|+|\lambda_f(\frac{f}{h})|+|\lambda_g(\frac{g}{h})|-\alpha_{\frac{f}{h}, \frac{g}{h}}} \quad (10)$$

where  $\frac{f}{h} = x_{i_1} x_{i_2}$ ,  $\frac{g}{h} = x_{j_1} x_{j_2}$  with  $i_2 > j_2$ .

The closed form expression proposed in Theorem 4 for counting the codewords with weight  $1.5 w_{\min}$  can be implemented by a simple procedure. A MATLAB script realising equation (10) can be found in Appendix A. Observe that the procedure has the complexity of order  $O(|\mathcal{I}_r|^2 \cdot r)$  due to the nested loops. A MATLAB script is available in [27].

**Example 11.** Let us consider the polar code (128,64) as decreasing monomial code with  $m = 7$  and  $r = 4$ . Let the set  $\mathcal{I}_r$  of monomials of degree 4 be defined by

$$x_0 x_1 x_2 x_3, x_0 x_1 x_2 x_4, x_0 x_1 x_3 x_4,$$

$$x_0 x_2 x_3 x_4, x_1 x_2 x_3 x_4, x_0 x_1 x_2 x_5, x_0 x_1 x_3 x_5$$

equivalent to  $\mathcal{A}_{m-r} = \{112, 104, 100, 98, 97, 88, 84\}$  as row indices of  $\mathbf{G}_N$ . Table I shows all the possible combinations of monomials that admit a degree  $r - 2 = 2$  common factor  $h$  and their associated cardinalities. The penalties account for the collisions, i.e., the identical codewords that should not be counted. The column "Total" results from the multiplication of the cardinalities of all subgroups and the penalties.

**Example 12.** Let us take the polar code (128,64) in Example 11. Since the construction of polar codes is channel dependent, one can improve the code by reducing the size of set  $\mathcal{I}_r$  (or equivalently by adjusting the design-SNR using any construction method [28]). Table II shows the sets  $\mathcal{A}_{m-r}$  corresponding to design-SNR = 0, 3, 6, and 8, from top to bottom. Note that when  $\mathcal{A}_{m-r} = \{112, 104\}$ , or equivalently  $\mathcal{I}_r = \{x_0 x_1 x_2 x_3, x_0 x_1 x_2 x_4\}$ , the common factor  $x_0 x_1 x_2$  is of degree  $3 > r - 2$  that does not satisfy the condition to have  $1.5 w_{\min}$ -weight codewords. Hence, no codewords with  $1.5 w_{\min}$  exist. Moreover, by changing the design-SNR, we are moving from polar codes to Reed-Muller codes.

As an example for longer codes, let us take the polar code (2048,1024) where  $m = 11$ . The last two rows of Table II show the results for the enumeration of this code constructed with design-SNR = 2 and 3 (bottom). Since the enumeration of codewords with weight  $w_{\min}$  and  $1.5 w_{\min}$  depends on set  $\mathcal{I}_r$ , a reduction in the cardinality of this set can reduce both. This can be observed in Table II.

## VI. ON THE FORMATION OF $1.5 w_{\min}$ -WEIGHT CODEWORDS VIA $\mathbf{G}_N$ -ROW COMBINATIONS

In this section, we show the implications of discussions in the previous sections on the formation of  $1.5 w_{\min}$ -weight codewords via combining particular pairs of  $w_{\min}$ -weight codewords. As shown in [31], [32],  $w_{\min}$ -weight codewords of a polar code  $\mathcal{C}(\mathcal{A})$  are formed in the cosets led by a row

$\mathbf{g}_i, i \in \mathcal{A}_{m-r}$ . However, the  $1.5 w_{\min}$ -weight codewords are generated by combining  $w_{\min}$ -weight codewords of distinct cosets. Hence, let us first define the cosets as following:

**Definition 8.** Cosets: we can partition a polar code into  $|\mathcal{A}|$  disjoint cosets  $\mathcal{C}_i(\mathcal{A}) = \mathbf{g}_i + \mathcal{C}(\mathcal{A} \setminus [0, i])$  of its subcodes  $\mathcal{C}(\mathcal{A} \setminus [0, i])$  for  $i \in \mathcal{A}$  where  $\mathbf{g}_i$  is the  $i$ -th row of the polar transform  $\mathbf{G}_N$ , that is

$$\mathcal{C}_i(\mathcal{A}) \triangleq \left\{ \mathbf{g}_i \oplus \bigoplus_{h \in \mathcal{H}_i} \mathbf{g}_h : \mathcal{H}_i \subseteq \mathcal{A} \setminus [0, i] \right\} \subseteq \mathcal{C}(\mathcal{A}). \quad (11)$$

Note that a coset led by row  $\bar{f}$  can be considered as the action of permutation group  $\text{LTA}(m, 2)_f$  on  $f$ . As the weight of the codewords in every coset  $\mathcal{C}_i(\mathcal{I})$  is [24]

$$w(\mathbf{g}_i \oplus \bigoplus_{j \in \mathcal{H}_i} \mathbf{g}_j) \geq w(\mathbf{g}_i), \quad (12)$$

where  $\mathcal{H}_i \subseteq [i+1, N-1]$ , the weight of codewords of the code  $\mathcal{C}(\mathcal{A})$  lying in the coset  $\mathcal{C}_i(\mathcal{A})$  for any  $i \notin \mathcal{A}_{m-r}$  will be larger than  $w_{\min}$ . Hence, we only consider the cosets  $\mathcal{C}_i, i \in \mathcal{A}_{m-r}$  for  $w_{\min}$ -weight codewords.

The indices of core rows in  $\mathbf{G}_N$  forming  $w_{\min}$ -weight codewords in coset  $\mathcal{C}_{\bar{f}}$  are collected in set  $\mathcal{K}_{\bar{f}}$  defined as

$$\mathcal{K}_{\bar{f}} = \{i \in \mathcal{A}_{m-r} \setminus [0, \bar{f}] : |\text{ind}(i) \setminus \text{ind}(\bar{f})| = 1\},$$

and  $|\mathcal{K}_{\bar{f}}| = \deg(f) + |\lambda_f|$ . Recall the relation between  $\bar{f}$  and  $f$  as the binary representation of  $\bar{f}$  is 1's complement of the binary representation of  $f$ . Suppose we have two  $w_{\min}$ -weight codewords  $\mathbf{c}_{\bar{f}}$  and  $\mathbf{c}_{\bar{g}}$ , equivalent to  $\text{ev}(P)$  and  $\text{ev}(Q)$  where  $P \in \text{LTA}(m, 2)_f \cdot f$  and  $Q \in \text{LTA}(m, 2)_g \cdot g$ . The codeword  $\mathbf{c}_{\bar{f}}$  is generated by

$$\mathbf{c}_{\bar{f}} = \mathbf{g}_{\bar{f}} \oplus \bigoplus_{j \in \mathcal{J}} \mathbf{g}_j \oplus \bigoplus_{m \in \mathcal{M}(\mathcal{J})} \mathbf{g}_m. \quad (13)$$

where  $\mathcal{J} \subseteq \mathcal{K}_{\bar{f}}$  and  $\mathcal{M}(\mathcal{J})$  is defined as a set of additional rows to get  $w(\mathbf{c}_{\bar{f}}) = w_{\min}$  if  $w(\mathbf{g}_{\bar{f}} \oplus \bigoplus_{j \in \mathcal{J}} \mathbf{g}_j) > w_{\min}$  (see [32]). The codeword  $\mathbf{c}_{\bar{g}}$  is similarly formed.

To get a  $1.5 w_{\min}$ -weight codeword by adding  $\mathbf{c}_{\bar{f}}$  and  $\mathbf{c}_{\bar{g}}$ , i.e.,  $w(\mathbf{c}_{\bar{f}} \oplus \mathbf{c}_{\bar{g}}) = 1.5 w_{\min}$ , codewords  $\mathbf{c}_{\bar{f}}, \mathbf{c}_{\bar{g}}$  must satisfy the following condition:

$$|\text{ind}(\mathbf{c}_{\bar{f}}) \cap \text{ind}(\mathbf{c}_{\bar{g}})| = \frac{1}{4} w_{\min}.$$

These codewords exist in cosets  $\mathcal{C}_{\bar{f}}$  and  $\mathcal{C}_{\bar{g}}$  where

$$|\text{ind}(\bar{g}) \setminus \text{ind}(\bar{f})| = 2. \quad (14)$$

This is equivalent to Corollary 1 where there exist two distinct variables in each term, i.e.,  $y_{r-1} y_r$  and  $y_{r+1} y_{r+2}$ . To keep this condition satisfied for every pair of codewords in distinct cosets, we need to limit the choice of codewords from each coset to the ones matching in  $|\text{ind}(f) \cap \text{ind}(g)| = r - 2$  shared variables. This reduces the choice of codewords from  $2^{|\mathcal{K}_{\bar{f}}|}$  to  $2^{|\mathcal{K}_{\bar{f}}| - (r-2)}$  in  $\mathcal{C}_{\bar{f}}$  or alternatively in  $\mathcal{C}_{\bar{g}}$ . That is, we do not have the freedom to combine any pair of  $w_{\min}$ -codewords from the cosets but they should match. In terms of row combinations in each coset, this means the values of coordinates  $\text{ind}(f) \cap$

$\bar{f}, \bar{g}$	$\text{ind}(f), \text{ind}(g)$	$\text{ind}(h)$	$\text{ind}(\frac{f}{h})$	$\text{ind}(\frac{g}{h})$	$ \text{LTA}(m, 2)_h \cdot h $	$ \text{LTA}(m, 2)_f \cdot \frac{f}{h} $	$ \text{LTA}(m, 2)_g \cdot \frac{g}{h} $	Penalties	Total
104, 84	[0, 1, 2, 4], [0, 1, 3, 5]	[0, 1]	[2, 4]	[3, 5]	$2^{2+0+0}$	$2^{2+0+1}$	$2^{2+1+2}$	$2^{-1}$	512
100, 88	[0, 1, 3, 4], [0, 1, 2, 5]	[0, 1]	[3, 4]	[2, 5]	$2^{2+0+0}$	$2^{2+1+1}$	$2^{2+0+2}$	$2^{-2}$	256
98, 88	[0, 2, 3, 4], [0, 1, 2, 5]	[0, 2]	[3, 4]	[1, 5]	$2^{2+0+1}$	$2^{2+1+1}$	$2^{2+0+2}$	$2^{-2}$	512
98, 84	[0, 2, 3, 4], [0, 1, 3, 5]	[0, 3]	[2, 4]	[1, 5]	$2^{2+0+2}$	$2^{2+1+1}$	$2^{2+0+2}$	$2^{-2}$	1024
97, 88	[1, 2, 3, 4], [0, 1, 2, 5]	[1, 2]	[3, 4]	[0, 5]	$2^{2+1+1}$	$2^{2+1+1}$	$2^{2+0+2}$	$2^{-2}$	1024
97, 84	[1, 2, 3, 4], [0, 1, 3, 5]	[1, 3]	[2, 4]	[0, 5]	$2^{2+1+2}$	$2^{2+1+1}$	$2^{2+0+2}$	$2^{-2}$	2048

**Table I** Illustration of enumerating the polar code (128,64) given in Example 11, where  $A_{1.5 w_{\min}} = |W_{1.5 w_{\min}}| = 5376$ .

$\mathcal{A}_{m-r}$	$m$	$r$	$w_{\min}$	$A_{w_{\min}}$	$A_{1.5 w_{\min}}$	Ref.
{112, 104, 100, 98, 97, 88, 84}	7	4	8	688	5376	
{112, 104, 100, 98, 88}	7	4	8	304	768	
{112, 104}	7	4	8	48	0	[14]
{120, 116, ..., 23, 15}	7	3	16	94488	74078592	[29]
{1920, 1856, ..., 1680}	11	7	16	11648	215040	[30]
{1920, 1856}	11	7	16	384	0	

**Table II** The Impact of code constructions, i.e., set  $\mathcal{A}$ , on  $A_{1.5 w_{\min}}$  for codes (128,64) and (2048,1024) discussed in Example 12. Our results match the results in the provided references.

$\text{ind}(g)$  in the binary representation of row indices in each coset should match the rows in the other cosets.

**Example 13.** Let us take cosets  $\mathcal{C}_{84}$  and  $\mathcal{C}_{104}$  in the polar code (128,64) where  $m = 7, r = 4$ . Observe that rows  $\mathbf{g}_{104}$  and  $\mathbf{g}_{84}$  are individually considered  $w_{\min}$ -weight codewords and  $\text{ind}(f) \cap \text{ind}(g) = \{0, 1\}$ . Since the condition (14) is satisfied for  $\bar{f} = 104, \bar{g} = 84$ , we have  $w(\mathbf{g}_{104} \oplus \mathbf{g}_{84}) = 1.5 w_{\min}$ . Now, we add row  $85 = (1010101)_2 \in \mathcal{K}_{84}$  to  $\mathbf{g}_{84}$  to form another  $w_{\min}$ -weight codeword in coset  $\mathcal{C}_{84}$ . However, to form  $1.5 w_{\min}$ -weight codeword in combination with  $\mathbf{g}_{104}$ , we have to include row  $105 = (1101001)_2 \in \mathcal{K}_{104}$  in coset  $\mathcal{C}_{104}$  as well. Observe that the binary digits at coordinates 0,1 (highlighted in blue) for the rows of two cosets are matched and as a result  $w((\mathbf{g}_{104} \oplus \mathbf{g}_{105}) \oplus (\mathbf{g}_{84} \oplus \mathbf{g}_{85})) = 1.5 w_{\min}$ .

Hence, the number of  $1.5 w_{\min}$ -weight codewords resulting from the combination of every pair of  $w_{\min}$ -weight codewords, one from  $\mathcal{C}_{\bar{f}}$  and the other from  $\mathcal{C}_{\bar{g}}$ , can be at most  $A_{1.5 w_{\min}}\{\bar{f}, \bar{g}\} \leq 2^{|\mathcal{K}_{\bar{f}}|} \times 2^{|\mathcal{K}_{\bar{g}}|} \times 2^{-(r-2)}$ . Moreover, when  $\mathcal{K}_{\bar{f}} \cap \mathcal{K}_{\bar{g}} \neq \emptyset$ , we will have multiple identical codewords, equivalent to collisions in Definition 6.

**Example 14.** Let us take  $\bar{f}, \bar{g}$  as 22, 25  $\in \mathcal{A}_{m-r}$  for the polar code (32,16) where  $m = 5, r = 2$ . Since  $\mathcal{K}_{\bar{f}} \cap \mathcal{K}_{\bar{g}} = \{26, 28\}$ , the rows  $\mathbf{g}_{26}$  and  $\mathbf{g}_{28}$  involved in the formation of  $w_{\min}$ -weight codewords in both cosets  $\mathcal{C}_{22}$  and  $\mathcal{C}_{25}$ , e.g.,  $w(\mathbf{g}_{22} \oplus \mathbf{g}_{26}) = w_{\min}$  in  $\mathcal{C}_{22}$  and  $w(\mathbf{g}_{25} \oplus \mathbf{g}_{26}) = w_{\min}$  in  $\mathcal{C}_{25}$ . As a result, the overcounting occurs if we consider the codewords  $(\mathbf{g}_{22} \oplus \mathbf{g}_{26}) \oplus \mathbf{g}_{25}$  and  $\mathbf{g}_{22} \oplus (\mathbf{g}_{25} \oplus \mathbf{g}_{26})$  distinct while they are not.

Hence, there will exist  $2^{|\mathcal{K}_{\bar{f}} \cap \mathcal{K}_{\bar{g}}|}$  redundant codewords and

we need to avoid collisions by discounting them as

$$A_{1.5 w_{\min}}\{\bar{f}, \bar{g}\} = 2^{|\mathcal{K}_{\bar{f}}| + |\mathcal{K}_{\bar{g}}| - (r-2) - |\mathcal{K}_{\bar{f}} \cap \mathcal{K}_{\bar{g}}|}. \quad (15)$$

Note that (15) is equivalent to the summation terms in (10).

**Example 15.** Let us take  $\bar{f}, \bar{g}$  as 22, 25  $\in \mathcal{A}_{m-r}$  for the polar code (32,16) where  $m = 5, r = 2$ . Observe that  $|\text{ind}(\bar{g}) \setminus \text{ind}(\bar{f})| = |\{0, 3, 4\} \setminus \{1, 2, 4\}| = 2$ ,  $\mathcal{K}_{22} = \{23, 26, 28, 30\}$ , and  $\mathcal{K}_{25} = \{26, 27, 28, 29\}$ . As  $\text{ind}(g) \cap \text{ind}(f) = \{0, 3\} \cap \{1, 2\} = \emptyset$ , then we have  $A_{1.5 w_{\min}}\{\bar{f}, \bar{g}\} = 2^4 \times 2^4 \times 2^{-2} = 64$ . Observe that since we have  $r = 2$  and  $|\text{ind}(\bar{g}) \setminus \text{ind}(\bar{f})| = 2$  for every pair in  $\mathcal{A}_{m-r}$ , the condition  $\text{ind}(h) = \text{ind}(g) \cap \text{ind}(f) = \emptyset$  is always satisfied. As another example, let us take  $\bar{f}, \bar{g}$  as 104, 84  $\in \mathcal{A}_{m-r}$  for the polar code (128,64) where  $m = 7, r = 4$  (see 1st row of Table I). Since  $|\mathcal{K}_{104}| = 5, |\mathcal{K}_{84}| = 7$ , and  $\mathcal{K}_{104} \cap \mathcal{K}_{84} = \{112\}$ , then  $A_{1.5 w_{\min}}\{104, 84\} = 2^5 \times 2^7 \times 2^{-2} \times 2^{-1} = 512$ .

## VII. CONCLUSION

This paper presents a framework for the characterization of codewords of decreasing monomial codes with weights less than  $2 w_{\min}$ . Specifically, we provide a closed-form expression for the enumeration of  $1.5 w_{\min}$ -weight codewords. We also demonstrate how  $1.5 w_{\min}$ -weight codewords are formed by combining a pair of  $w_{\min}$ -codewords. The results show that the number of  $1.5 w_{\min}$ -weight codewords depends solely on the maximum-degree monomials.

## REFERENCES

- [1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels", *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
- [2] 3GPP, "Final report of 3GPP TSG RAN WG1 #87 v1.0.0", Tech. Rep., Nov. 2016, Reno, USA. [Online]. Available: <http://www.3gpp.org/ftp/tsg%20ran/WG1%20R1/TSGR1%2087/Report/Final%20Minutes%20report%20RAN1%5C%2387%20v100.zip>.
- [3] S. Lin and D. J. Costello, "Error Control Coding", in *2nd Edition*, Pearson, Upper Saddle River: Prentice Hall, 2004, pp. 395–400.
- [4] M. Bardet, J. Chȃulet, V. Dragoi, A. Otmani, and J.-P. Tillich, "Cryptanalysis of the McEliece Public Key Cryptosystem Based on Polar Codes", in *Post-Quantum Cryptography*, vol. 9606, 2016, pp. 118–143.
- [5] M. Bardet, V. Dragoi, A. Otmani, and J.-P. Tillich, "Algebraic properties of polar codes from a new polynomial formalism", in *2016 IEEE Int. Symp. Inf. Theory (ISIT)*, 2016, pp. 230–234.
- [6] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "Construction of Polar Codes With Sublinear Complexity", *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 2782–2791, 2019.

[7] Y. Li, H. Zhang, R. Li, J. Wang, W. Tong, G. Yan, and Z. Ma, “The Complete Affine Automorphism Group of Polar Codes”, in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06.

[8] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer, and S. t Brink, “Automorphism ensemble decoding of Reed-Muller codes”, *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6424–6438, Oct. 2021.

[9] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, “On the automorphism group of polar codes”, 2021 IEEE Int. Symp. Inf. Theory (ISIT), Jul. 2021, pp. 1230–1235.

[10] C. Pillet, V. Bioglio, and I. Land, “Polar codes for automorphism ensemble decoding”, *IEEE Information Theory Workshop (ITW)*, vol. 2021, pp. 1–6, Oct. 2021.

[11] —, “Classification of automorphisms for the decoding of polar codes”, in *ICC 2022 - IEEE International Conference on Communications*, May 2022, pp. 110–115.

[12] K. Ivanov and R. L. Urbanke, “On the efficiency of polar-like decoding for symmetric codes”, *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 163–170, Jan. 2022.

[13] K. Ivanov and R. Urbanke, “Polar codes do not have many affine automorphisms”, in *2022 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2022, pp. 2374–2378.

[14] H. Yao, A. Fazeli, and A. Vardy, “A Deterministic Algorithm for Computing the Weight Distribution of Polar Codes”, arXiv, preprint, 2020. arXiv: 2102.07362v1.

[15] T. Kasami and N. Tokura, “On the weight structure of Reed-Muller codes”, *Trans. Inf. Theory*, vol. 16, no. 6, pp. 752–759, Nov. 1970.

[16] T. Kasami, N. Tokura, and S. Azumi, “On the weight enumeration of weights less than 2.5d of Reed–Muller codes”, *Information and Control*, vol. 30, no. 4, pp. 380–395, 1976.

[17] Z. Liu, K. Chen, K. Niu, and Z. He, “Distance spectrum analysis of polar codes”, in *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2014, pp. 490–495, 2014.

[18] M. Valipour and S. Yousefi, “On probabilistic weight distribution of polar codes”, *IEEE commun. lett.*, vol. 17, no. 11, pp. 2120–2123, 2013.

[19] Q. Zhang, A. Liu, and X. Pan, “An enhanced probabilistic computation method for the weight distribution of polar codes”, *IEEE Communications Letters*, vol. 21, no. 12, pp. 2562–2565, 2017.

[20] M. P. C. Fossorier and S. Lin, “Weight distribution for closest coset decoding of  $-u-u+v-$  constructed codes”, in *IEEE Trans. on Information Theory*, vol. 43, no. 3, pp. 1028–1030, May 1997.

[21] R. Polyanskaya, M. Davletshin, and N. Polyanski, “Weight Distributions for Successive Cancellation Decoding of Polar Codes”, *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7328–7336, Dec. 2020.

[22] V. F. Dragoi, “An algebraic approach for the resolution of algorithmic problems raised by cryptography and coding theory”, Ph.D. dissertation, Normandie Université, 2017.

[23] C. Schürch, “A partial order for the synthesized channels of a polar code”, in *2016 IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, 2016, pp. 220–224.

[24] M. Rowshan, S. H. Dau, and E. Viterbo, “Error Coefficient-reduced Polar/PAC Codes”, preprint, 2021. arXiv: 2111.088435.

[25] N. J. Sloane and E. R. Berlekamp, “The weight enumerator for second-order Reed-Muller codes”, *IEEE Trans. Information Theory*, vol. 16, no. 6, pp. 745–751, 1970.

[26] T. Tao and V. H. Vu, *Additive Combinatorics*, 1st. Cambridge pp. 112-178: Cambridge University Press, 2006.

[27] [Online]. Available: <https://github.com/mohammad-rowshan/closed-form-weight-enumeration-of-polar-codes>.

[28] H. Vangala, E. Viterbo, and Y. Hong, “A Comparative Study of Polar Code Constructions for the AWGN Channel”, 2015. arXiv: 1501.02473.

[29] M. Sugino, Y. Ienaga, N. Tokura, and T. Kasami, “Weight distribution of (128, 64) Reed-Muller code (Corresp.)”, *IEEE Transactions on Information Theory*, vol. 17, no. 5, pp. 627–628, Sep. 1971.

[30] B. Li, H. Shen, and D. Tse, “An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check”, *IEEE Communications Letters*, vol. 16, no. 12, pp. 2044–2047, 2012.

[31] M. Rowshan, S. Hoang Dau, and E. Viterbo, “Improving the Error Coefficient of Polar Codes”, in *2022 IEEE Information Theory Workshop (ITW)*, 2022, pp. 249–254. DOI: 10.1109/ITW54588.2022.9965852.

[32] M. Rowshan and J. Yuan, “Fast Enumeration of Minimum Weight Codewords of PAC Codes”, 2022, pp. 255–260. DOI: 10.1109/ITW54588.2022.9965901.

## APPENDIX A

### MATLAB SCRIPT FOR ENUMERATION OF CODEWORDS WITH WEIGHT $w_{\min}$ AND $1.5 w_{\min}$

The following listing presents the MATLAB™ function *weight\_enum*, which serves to enumerate the first two minimum weight codewords. The inputs required are the set  $\mathcal{A}$  of coordinates of the non-frozen bits and  $n = \log_2(N)$  where  $N$  is the code length. It is worth noting that the elements of set  $\mathcal{A}$  must be sorted in ascending order. This is crucial, as the conditions for determining  $\alpha_{\frac{f}{h}, \frac{g}{h}}$  are dependent on the order of  $f$  and  $g$ , as detailed in Theorem 10. The function begins by extracting the maximum-degree monomials from set  $\mathcal{A}$  and storing them in set  $\mathcal{A}_{m-r}$  (lines 2 and 3). Next, in the two inner for-loops, the function finds all distinct pairs of monomials in set  $\mathcal{A}_{m-r}$  named  $f$  and  $g$ , whose greatest common divisor (GCD) has a degree of  $r - 2$ . Utilizing (6), (10), and the function *lambda* representing (5),  $A_w$  for  $w \in \{w_{\min}, 1.5 w_{\min}\}$  is computed and returned.

```
function [w,A_w] = weight_enum(A,n)
    r = max(sum(~(dec2bin(A)-'0')),2);
    Amr = A(find(sum(~(dec2bin(sort(A))-'0')),2)==r)
    Amr_sub = Amr; w=zeros(1,2); A_w=zeros(1,2); ...
        w(1)=2^(n-r); w(2)=1.5*w(1);
    for i = Amr
        f = find(~(reverse(dec2bin(i,n))-'0'));
        A_w(1) = A_w(1) + 2^(r+lambda(f,f));
        Amr_sub(Amr_sub==i) = [];
        for j = Amr_sub
            g = find(~(reverse(dec2bin(j,n))-'0'));
            h = intersect(f,g);
            if length(h)==r-2
                foh = setdiff(f,h); goh = ...
                    setdiff(g,h);
                alpha = 1*(foh(2)>goh(2) & ...
                    goh(2)>foh(1)) + ...
                    1*(goh(1)>foh(1));
                A_w(2) = A_w(2) + 2^(r+2 + ...
                    lambda(h,h) + lambda(f,foh) ...
                    + lambda(g,goh) - alpha);
            end
        end
    end
end
function orbit = lambda(f,g)
    orbit = 0;
    for i = g
        orbit = orbit + length(setdiff([1:i-1],f));
    end
end
```

## APPENDIX B

### PROOF OF LEMMA 2

*Proof:* Consider an element  $P \in LTA(m, 2)_h \cdot h \cdot \left( LTA(m, 2)_f \cdot \frac{f}{h} + LTA(m, 2)_g \cdot \frac{g}{h} \right)$ . By definition of  $LTA(m, 2)$  and  $LTA(m, 2)_h \cdot h$  we have  $P \in LTA(m, 2) \cdot h \cdot \left( LTA(m, 2) \cdot \frac{f}{h} + LTA(m, 2) \cdot \frac{g}{h} \right)$ . Conversely, choose an el-

ement  $P \in \text{LTA}(m, 2) \cdot h \cdot \left( \text{LTA}(m, 2) \cdot \frac{f}{h} + \text{LTA}(m, 2) \cdot \frac{g}{h} \right)$ .

We thus have

$$P = \prod_{i \in \text{ind}(h)} (x_i + \sum_{j < i, j \notin \text{ind}(h)} b_{i,j} x_j + \varepsilon_i) \times \\ \prod_{i \in \text{ind}(\frac{f}{h})} (x_i + \sum_{j < i, j \notin \text{ind}(\frac{f}{h})} b_{i,j} x_j + \varepsilon_i) \\ + \prod_{i \in \text{ind}(h)} (x_i + \sum_{j < i, j \notin \text{ind}(h)} b_{i,j} x_j + \varepsilon_i) \times \\ \prod_{i \in \text{ind}(\frac{g}{h})} (x_i + \sum_{j < i, j \notin \text{ind}(\frac{g}{h})} b_{i,j} x_j + \varepsilon_i)$$

$$P = \prod_{i \in \text{ind}(h)} (x_i + \sum_{j < i, j \notin \text{ind}(h)} b_{i,j} x_j + \varepsilon_i) \times \\ \prod_{i \in \text{ind}(\frac{f}{h})} (x_i + \sum_{j < i, j \notin \text{ind}(\frac{f}{h})} b_{i,j}^* x_j + \varepsilon_i^*) \\ + \prod_{i \in \text{ind}(h)} (x_i + \sum_{j < i, j \notin \text{ind}(h)} b_{i,j} x_j + \varepsilon_i) \times \\ \prod_{i \in \text{ind}(\frac{g}{h})} (x_i + \sum_{j < i, j \notin \text{ind}(\frac{g}{h})} b_{i,j}^* x_j + \varepsilon_i^*)$$

where  $b_{i,j}^*$  are obtained as in the proof of Proposition 3.7.3 from [22] (see Lemma 1).  $\blacksquare$

### APPENDIX C PROOF OF THEOREM 3

*Proof:* By Theorem 2, any codeword  $\text{ev}(P)$  with  $w(\text{ev}(P)) = 1.5 w_{\min}$  is up to an affine transformation  $P = y_1 \dots y_{r-2} (y_{r-1} y_r + y_{r+1} y_{r+2})$ .

The direct implication is a consequence of the definition of  $\text{LTA}(m, 2)$  and Theorem 2.

For the converse implication consider a codeword  $\text{ev}(P)$  with  $P = y_1 \dots y_{r-2} (y_{r-1} y_r + \dots + y_{r+2\mu-3} y_{r+2\mu-2})$ . The maximum variables in  $[y_1 \dots y_{r-2} y_{r-1} y_r], \dots, [y_1 \dots y_{r-2} y_{r+2\mu-3} y_{r+2\mu-2}]$  are all distinct (see Lemma 1). Hence, we can construct  $h = x_{i_1} \dots x_{i_{r-2}}$  and  $f/h = x_{i_{r-1}} x_{i_r}, \dots, g/h = x_{i_{r+1}} x_{i_{r+2}}$ . Notice that while  $i_1, \dots, i_{r-2}$  are all pairwise distinct, the indices  $i_{r-1}, i_r, i_{r+2}$  do not have to be pairwise distinct. Let us demonstrate that there is a affine transformation such that these indices are also pairwise distinct. Suppose  $f/h$  and  $g/h$  share in common at least 1 maximum variable. Formally, we can write  $P_1 \in \text{LTA}(m, 2)_f \cdot f/h$  as  $P_1 = (x_{i_1} + v_{i_1})(x_{i_2} + v_{i_2})$  and  $P_2 \in \text{LTA}(m, 2)_g \cdot g/h$  as  $P_2 = (x_{j_1} + z_{j_1})(x_{j_2} + z_{j_2})$  where  $v_{i_1}, v_{i_2}, z_{j_1}, z_{j_2}$  are linear function with maximum variables strictly smaller than the index, and there is at least one index  $i_l$  equal to one index  $j_l$ . Also, we have  $i_1 < i_2$  and  $j_1 < j_2$ . Then we have the following cases

- $i_1 = j_1$  and  $i_2 > j_2$ . Then

$$P_1 + P_2 = (x_{i_1} + v_{i_1})(x_{i_2} + v_{i_2}) + (x_{i_1} + z_{i_1})(x_{j_2} + z_{j_2}) \\ = (x_{i_1} + v_{i_1})(x_{i_2} + v_{i_2} + x_{j_2} + z_{j_2}) \\ + (x_{j_2} + z_{j_2})(v_{i_1} + z_{i_1}) \\ \in \text{LTA}(m, 2)_f \cdot f/h + \text{LTA}(m, 2)_{g^*} \cdot g^*/h,$$

with  $g^*/h = x_{j_1^*} x_{j_2}$  where  $j_1^* = \max\{l \mid x_l \in v_{i_1} + z_{i_1}\}$ . Since  $j_1^* < i_1$  this implies that all four indices  $i_1, i_2, j_1^*, j_2$  are distinct. Also,  $j_1^*$  exists since  $v_{i_1} \neq z_{i_1}$  and  $v_{i_1} \neq 1 + z_{i_1}$ . If  $v_{i_1} = z_{i_1}$  then we would have that  $P_1 + P_2$  is a product of two linear forms and hence  $\text{ev}(P)$  is a minimum weight codeword, which contradicts our hypothesis. If  $v_{i_1} = 1 + z_{i_1}$  then the weight of  $P_1 + P_2$  would be 2 times the weight of  $P_1$ , and this would imply that the weight of  $P$  is strictly bigger than  $1.5 w_{\min}$  which contradicts the hypothesis.

- $i_1 = j_2$ . Then

$$P_1 + P_2 = (x_{i_1} + v_{i_1})(x_{i_2} + v_{i_2}) + (x_{j_1} + z_{j_1})(x_{i_1} + z_{i_1}) \\ = (x_{i_1} + v_{i_1})(x_{i_2} + v_{i_2} + x_{j_1} + z_{j_1}) \\ + (x_{j_1} + z_{j_1})(v_{i_1} + z_{i_1}) \\ \in \text{LTA}(m, 2)_f \cdot f/h + \text{LTA}(m, 2)_{g^*} \cdot g^*/h,$$

with  $g^*/h = x_{j_1} x_{j_2^*}$  where  $j_2^* = \max\{l \mid x_l \in v_{i_1} + z_{i_1}\}$ . Here, we might have  $j_2^* = j_1$  but in that case since  $(x_{j_1} + z_{j_1})(v_{i_1} + z_{i_1})$  is a product of linear forms, by Lemma 1 it can be rewritten such that maximum variables are distinct, fact that ends the proof for this case. Also, as in the previous case  $j_2^*$  exists.

- $i_1 = j_1$  and  $i_2 = j_2$ . Then

$$P_1 + P_2 = (x_{i_1} + v_{i_1})(x_{i_2} + v_{i_2}) + (x_{i_1} + z_{i_1})(x_{i_2} + z_{i_2})$$

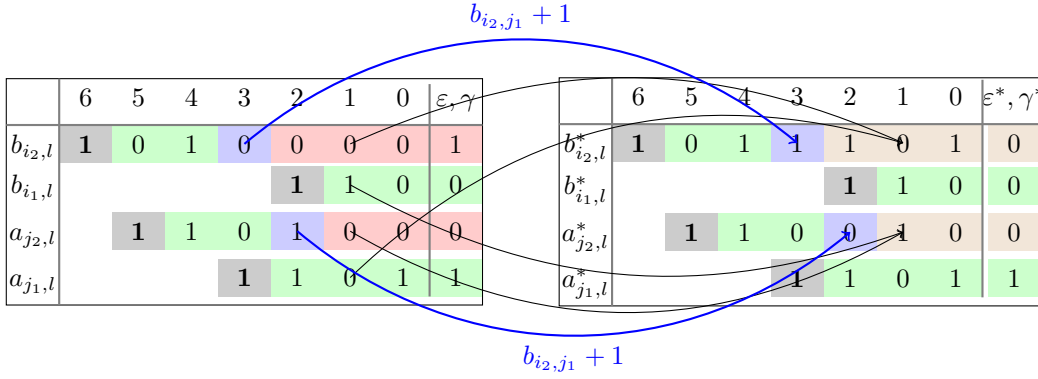
Since  $v_{i_1} \neq z_{i_1}$  (or equivalently  $\text{ev}(P)$  is not a minimum weight codeword) and  $v_{i_2} \neq z_{i_2}$  we have

$$P_1 + P_2 = (x_{i_2} + z_{i_2})(z_{i_1} + v_{i_1}) + (z_{i_2} + v_{i_2})(x_{i_1} + v_{i_1}).$$

The two terms in the sum have maximum variables  $x_{i_2} x_{j_1^*}$  and  $x_{j_2^*} x_{i_1}$  with  $j_1^* = \max\{l \mid x_l \in v_{i_1} + z_{i_1}\}$  and  $j_2^* = \max\{l \mid x_l \in v_{i_2} + z_{i_2}\}$ . If all four indices are distinct the proof is finished. If not, go to the previous items.  $\blacksquare$

### APPENDIX D PROOF OF PROPOSITION 1

Let  $f = x_{i_1} x_{i_2}$  and  $g = x_{j_1} x_{j_2}$  with  $\text{gcd}(f, g) = 1$  and  $i_2 > j_2$  and  $P = (\mathbf{B}, \varepsilon) \cdot f, P = (\mathbf{B}^*, \varepsilon^*) \cdot f, Q = (\mathbf{A}, \gamma)$ .



**Fig. 1** Application of Proposition 1 in the case  $i_2 > j_2 > j_1 > i_1$  with  $f = x_6x_2, g = x_5x_3$ . For  $P = (x_6 + x_4 + 1)(x_2 + x_1) \in \text{LTA}(m, 2) \cdot f$  and  $Q = (x_5 + x_4 + x_2)(x_3 + x_2 + x_0 + 1) \in \text{LTA}(m, 2) \cdot g$  we can create a non-trivial distinct pair  $P^*, Q^*$  with  $P^* = (x_6 + x_4 + x_3 + x_2 + x_0)(x_2 + x_1)$  and  $Q^* = (x_5 + x_4 + x_1)(x_3 + x_2 + x_0 + 1)$ . The meaning of the colors is: gray cells are for indices of variables in the support, green cells are for  $b_{i, j}^* = b_{i, j}$  or  $a_{i, j}^* = a_{i, j}$ , blue cells are denoting  $b_{i, j}^* = 1 + b_{i, j}$  or  $a_{i, j}^* = 1 + a_{i, j}$ , and brown cells are obtained by the combination of red cells and their green counterparts; that is,  $b_{i_2, l}^* = b_{i_2, l} + a_{j_1, l}$  and  $a_{j_2, l}^* = a_{j_2, l} + b_{i_1, l}$ .

$g, Q^* = (\mathbf{A}^*, \boldsymbol{\gamma}^*) \cdot g$ . By definition of  $\text{LTA}(m, 2)$  and of a collision we have

$$\begin{aligned}
& (x_{i_2} + \sum_{l < i_2, l \neq i_1} b_{i_2, l} x_l + \varepsilon_{i_2})(x_{i_1} + \sum_{l < i_1} b_{i_1, l} x_l + \varepsilon_{i_1}) \\
& - (x_{i_2} + \sum_{l < i_2, l \neq i_1} b_{i_2, l}^* x_l + \varepsilon_{i_2}^*)(x_{i_1} + \sum_{l < i_1} b_{i_1, l}^* x_l + \varepsilon_{i_1}^*) \\
& = (x_{j_2} + \sum_{l < j_2, l \neq j_1} a_{j_2, l} x_l + \gamma_{j_2})(x_{j_1} + \sum_{l < j_1} a_{j_1, l} x_l + \gamma_{j_1}) \\
& - (x_{j_2} + \sum_{l < j_2, l \neq j_1} a_{j_2, l}^* x_l + \gamma_{j_2}^*)(x_{j_1} + \sum_{l < j_1} a_{j_1, l}^* x_l + \gamma_{j_1}^*)
\end{aligned} \tag{16}$$

Since  $x_{i_2}$  is the maximum variable, extracting the coefficient of  $x_{i_2}$  gives

$$x_{i_1} + \sum_{l < i_1} b_{i_1, l} x_l + \varepsilon_{i_1} = x_{i_1} + \sum_{l < i_1} b_{i_1, l}^* x_l + \varepsilon_{i_1}^*.$$

This implies

$$\begin{aligned}
& \left( \sum_{l < i_2, l \neq i_1} (b_{i_2, l} - b_{i_2, l}^*) x_l + \varepsilon_{i_2} - \varepsilon_{i_2}^* \right) (x_{i_1} + \sum_{l < i_1} b_{i_1, l} x_l + \varepsilon_{i_1}) \\
& = Q - Q^*. \tag{17}
\end{aligned}$$

If we are in the case  $i_2 > i_1 > j_2$ . Then either  $x_l$  is the maximum variable (in the first factor of the first term) or  $x_{i_1}$  is the maximum variable (in the second factor of the first term). Either ways we deduce  $Q - Q^* = 0$  which implies  $\mathbf{A} = \mathbf{A}^*$  and further no possible collisions. Hence, in this case the Minkowski sum  $\text{LTA}(m, 2) \cdot f + \text{LTA}(m, 2) \cdot g$  has maximum cardinality.

If we are in the case  $i_2 > j_2 > i_1$ . If we consider that the next maximum variable is  $x_l$  (present in the first term), with

$j_2 < l < i_2$  we obtain  $b_{i_2, l} = b_{i_2, l}^*$  for all such indices. This means that we have

$$\begin{aligned}
& \left( \sum_{l < j_2, l \neq i_1} (b_{i_2, l} - b_{i_2, l}^*) x_l + \varepsilon_{i_2} - \varepsilon_{i_2}^* \right) (x_{i_1} + \sum_{l < i_1} b_{i_1, l} x_l + \varepsilon_{i_1}) \\
& = Q - Q^*. \tag{18}
\end{aligned}$$

If the next maximum variable is  $x_{j_2}$  we have two cases,  $b_{i_2, j_2} - b_{i_2, j_2}^* = 0$  and  $b_{i_2, j_2} - b_{i_2, j_2}^* = 1$ .

**A) The case  $b_{i_2, j_2} - b_{i_2, j_2}^* = 0$ .** The last equation becomes

$$\begin{aligned}
& \left( \sum_{l < j_2, l \neq i_1} (b_{i_2, l} - b_{i_2, l}^*) x_l + \varepsilon_{i_2} - \varepsilon_{i_2}^* \right) (x_{i_1} + \sum_{l < i_1} b_{i_1, l} x_l + \varepsilon_{i_1}) \\
& = Q - Q^*. \tag{19}
\end{aligned}$$

Extracting the coefficient of  $x_{j_2}$  gives

$$x_{j_1} + \sum_{l < j_1} a_{j_1, l} x_l + \gamma_{j_1} = x_{j_1} + \sum_{l < j_1} a_{j_1, l}^* x_l + \gamma_{j_1}^*.$$

which implies

$$\begin{aligned}
& \left( \sum_{l < j_2, l \neq i_1} (b_{i_2, l} - b_{i_2, l}^*) x_l + \varepsilon_{i_2} - \varepsilon_{i_2}^* \right) (x_{i_1} + \sum_{l < i_1} b_{i_1, l} x_l + \varepsilon_{i_1}) \\
& = \left( \sum_{l < j_2, l \neq j_1} (a_{j_2, l} - a_{j_2, l}^*) x_l + \gamma_{j_2} - \gamma_{j_2}^* \right) (x_{j_1} + \sum_{l < j_1} a_{j_1, l} x_l + \gamma_{j_1}).
\end{aligned} \tag{20}$$

In order to have equation (20) valid we need

- $b_{i_2, l} = b_{i_2, l}^*$  for  $j_1 < l < j_2$
- $a_{j_2, l} = a_{j_2, l}^*$  for  $i_1 < l < j_2$
- $b_{i_2, j_1} - b_{i_2, j_1}^* = 1$
- $a_{j_2, i_1} - a_{j_2, i_1}^* = 1$
- $b_{i_2, l} - b_{i_2, l}^* = a_{j_1, l}$  for  $l < j_1$
- $a_{j_2, l} - a_{j_2, l}^* = b_{i_1, l}$  for  $l < i_1$
- $\varepsilon_{i_2} - \varepsilon_{i_2}^* = \gamma_{j_1}$
- $\gamma_{j_2} - \gamma_{j_2}^* = \varepsilon_{i_1}$

Since these are all the possible coefficients this case ends here.

**B) The case**  $b_{i_2, j_2} - b_{i_2, j_2}^* = 1$ . We have

$$\begin{aligned} & (x_{j_2} + \sum_{l < j_2, l \neq i_1} (b_{i_2, l} - b_{i_2, l}^*)x_l + \varepsilon_{i_2} - \varepsilon_{i_2}^*)(x_{i_1} + \sum_{l < i_1} b_{i_1, l}x_l + \varepsilon_{i_1}) \\ & = Q - Q^*. \quad (21) \end{aligned}$$

Extracting the coefficient of  $x_{j_2}$  gives

$$x_{i_1} + \sum_{l < i_1} b_{i_1, l}x_l + \varepsilon_{i_1} = \sum_{l < j_1} (a_{j_1, l} - a_{j_1, l}^*)x_l + \gamma_{j_1} - \gamma_{j_1}^*.$$

This equation can hold only if  $i_1 < j_1$ . If this is the case then we deduce

$$\begin{aligned} a_{j_1, l} - a_{j_1, l}^* &= 0, \text{ for } i_1 < l < j_1 \\ a_{j_1, i_1} - a_{j_1, i_1}^* &= 1, \\ a_{j_1, l} - a_{j_1, l}^* &= b_{i_1, l}, \text{ for } l < i_1 \\ \gamma_{j_1} - \gamma_{j_1}^* &= \varepsilon_{i_1}. \end{aligned}$$

Also we deduce

$$\begin{aligned} & (x_{j_2} + \sum_{l < j_2, l \neq i_1} (b_{i_2, l} - b_{i_2, l}^*)x_l + \varepsilon_{i_2} - \varepsilon_{i_2}^*)(x_{i_1} + \sum_{l < i_1} b_{i_1, l}x_l + \varepsilon_{i_1}) \\ & = (x_{j_2} + \sum_{l < j_2, l \neq j_1} a_{j_2, l}x_l + \gamma_{j_2})(x_{j_1} + \sum_{l \leq i_1} a_{j_1, l}x_l + \gamma_{j_1}) \\ & - (x_{j_2} + \sum_{l < j_2, l \neq j_1} a_{j_2, l}^*x_l + \gamma_{j_2})(x_{j_1} + \sum_{l \leq i_1} a_{j_1, l}x_l + \gamma_{j_1}) \\ & - (x_{j_2} + \sum_{l < j_2, l \neq j_1} a_{j_2, l}^*x_l + \gamma_{j_2})(x_{i_1} + \sum_{l < i_1} b_{i_1, l}x_l + \varepsilon_{i_1}) \end{aligned} \quad (22)$$

which leads to

$$\begin{aligned} & \left( \sum_{l < j_2, l \neq i_1} (b_{i_2, l} - b_{i_2, l}^*)x_l + \varepsilon_{i_2} - \varepsilon_{i_2}^* - \sum_{l < j_2, l \neq j_1} a_{j_2, l}^*x_l - \gamma_{j_2} \right) \\ & \quad \times \left( x_{i_1} + \sum_{l < i_1} b_{i_1, l}x_l + \varepsilon_{i_1} \right) \\ & = \left( \sum_{l < j_2, l \neq j_1} (a_{j_2, l} - a_{j_2, l}^*)x_l + \gamma_{j_2} - \gamma_{j_2}^* \right) \left( x_{j_1} + \sum_{l \leq i_1} a_{j_1, l}x_l + \gamma_{j_1} \right) \end{aligned} \quad (23)$$

Notice that one can not have  $x_l$  with  $l > j_1$  as maximum variable in the previous equation, i.e., we need to have  $a_{j_2, l} = a_{j_2, l}^*$  for  $l > i_1$  and  $b_{i_2, l} - b_{i_2, l}^* - a_{j_2, l}^* = 0$  for  $l > j_1$ . And as in the previous case we can determine a set of conditions under which equation (23) is valid.

- $b_{i_2, j_1} - b_{i_2, j_1}^* = 1$
- $a_{j_2, i_1} - a_{j_2, i_1}^* = 1$
- $b_{i_2, l} - b_{i_2, l}^* = a_{j_1, l} - a_{j_2, l}^*$  for  $l < j_1$
- $a_{j_2, l} - a_{j_2, l}^* = b_{i_1, l}$  for  $l < i_1$
- $\varepsilon_{i_2} - \varepsilon_{i_2}^* = \gamma_{j_1} - \gamma_{j_2}$
- $\gamma_{j_2} - \gamma_{j_2}^* = \varepsilon_{i_1}$

Since, all possible coefficients are present, this case ends here.

Resuming the cases, when  $i_2 > i_1 > j_2$  no collisions, when  $i_2 > j_2 > j_1 > i_2$  we have two restrictions, and when  $i_2 > j_2 > i_1 > j_1$  we have a single restriction. Each restriction is an equation between two free variables  $(b_{i_2, j_2}, b_{i_2, j_2}^*)$  over

$\mathbb{F}_2$ , hence, each restriction generates 2 possible solutions, from which we deduce the wanted result. In other words, the variable  $b_{i_2, j_2}$  is a free variable in **B** that allows us to count for each element the number of invariants.

#### APPENDIX E PROOF OF LEMMA 3

Let us suppose by absurd that there are polynomials  $H, H^* \in \text{LTA}(m, 2)_h \cdot h$  and  $P, P^* \in \text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h}$  with  $H \neq H^*$  and  $P \neq P^*$  s.t.  $HP = H^*P^*$ . Since  $h$  is a product of variables that are not present in  $P$  or  $P^*$  extracting the coefficient of  $h$  from  $HP$  and  $H^*P^*$  implies  $P = P^*$ . But this would contradict our hypothesis, and ends the proof.

#### APPENDIX F PROOF OF PROPOSITION 2

*Proof:* Suppose by absurd that there is a common polynomial  $P$  between the two sets. Since  $P$  belongs to  $\text{LTA}(m, 2)_h \cdot h \cdot (\text{LTA}(m, 2)_f \cdot \frac{f}{h} + \text{LTA}(m, 2)_g \cdot \frac{g}{h})$  this implies that the maximum monomials in  $P$  are  $f + g$  (by Definition of the  $\text{LTA}(m, 2)$ ). Also since  $P$  belongs to  $\text{LTA}(m, 2)_{h^*} \cdot h^* \cdot (\text{LTA}(m, 2)_{f^*} \cdot \frac{f^*}{h^*} + \text{LTA}(m, 2)_{g^*} \cdot \frac{g^*}{h^*})$  this implies that the maximum monomials in  $P$  are  $f^* + g^*$ . Hence,  $f + g = f^* + g^*$ . Both sums can not be equal to zero since  $f \neq g$  and  $f^* \neq g^*$ . Hence, we need to have either  $(f, g) = (f^*, g^*)$  or  $(f, g) = (g^*, f^*)$  which ends the proof. ■

#### APPENDIX G PROOF OF THEOREM 4

*Proof:* First, the cardinality of  $W_{1.5 \text{ w}_{\min}}$  can be computed as the sum of  $|\text{LTA}(m, 2)_h \cdot h| \times |(\text{LTA}(m, 2)_f \cdot x_{i_1}x_{i_2} + \text{LTA}(m, 2)_g \cdot x_{j_1}x_{j_2})|$  for all possible  $h$  with  $\deg(h) = r - 2$  and  $f = hx_{i_1}x_{i_2}, g = hx_{j_1}x_{j_2}$ . Second, recall that  $|\text{LTA}(m, 2)_h \cdot g| = 2^{\deg(g) + |\lambda_h(g)|}$  and  $|\text{LTA}(m, 2)_h \cdot h| = 2^{\deg(h) + |\lambda_h|}$ . Combined with Proposition 1 we have that  $|(\text{LTA}(m, 2)_f \cdot x_{i_1}x_{i_2} + \text{LTA}(m, 2)_g \cdot x_{j_1}x_{j_2})| = |(\text{LTA}(m, 2)_f \cdot x_{i_1}x_{i_2})| \times |\text{LTA}(m, 2)_g \cdot x_{j_1}x_{j_2}| \times 2^{\alpha \frac{f}{h}, \frac{g}{h}}$ . Hence, each orbit has cardinality  $2^{r-2+|\lambda_h|} \times 2^{2+|\lambda_f(\frac{f}{h})|+2+|\lambda_g(\frac{g}{h})|-\alpha \frac{f}{h}, \frac{g}{h}}$ , which ends the proof. ■