New LP-Based Upper Bounds in the Rate-Vs.-Distance Problem for Binary Linear Codes

Elyassaf Loyfer and Nati Linial

***Abstract*— We develop a new family of linear programs, that yield upper bounds on the rate of binary linear codes of a given distance. Our bounds apply *only to linear codes.* Delsarte's LP is the weakest member of this family and our LP yields increasingly tighter upper bounds on the rate as its control parameter increases. Numerical experiments show significant improvement compared to Delsarte. These convincing numerical results, and the large variety of tools available for asymptotic analysis, give us hope that our work will lead to new improved asymptotic upper bounds on the possible rate of linear codes. A slightly prior work by Coregliano, Jeronimo and Jones offers a closely related family of linear programs which converges to the true bound. Here we provide a new proof of convergence for the same LPs.**

***Index Terms*— Error correction codes, linear codes, binary codes, linear programming.**

I. Introduction

IN THIS paper, we investigate the *rate vs. distance* problem. This fundamental question in coding theory seeks to estimate the largest possible size $A(n, d)$ of a length-n code of distance d. Our ultimate goal, however, is to make progress in the asymptotic version of the problem, which is to find $\mathcal{R}(\delta) := \limsup_{n\to\infty} \frac{1}{n} \log_2 A(n, \lfloor \delta n \rfloor)$.

The best lower bound that we have on $\mathcal{R}(\delta)$ is due to Gilbert [1] (for general codes) and Varshamov [2] (in the linear case) and is attained by random codes. The best upper bounds that we have are due to McEliece, Rodemich, Ramsey and Welch (MRRW) [3]. Based on Delsarte's linear program [4], these bounds are often called the first and second linear programming bounds. There is substantial empirical evidence [5] indicating that the MRRW bounds may be asymptotically all that Delsarte's LP yields.

We propose a new family of linear programs, which greatly strengthen Delsarte's LP. We stress that these new LPs apply *only* to linear codes. They come with a control parameter, an integer r. For $r = 1$ our LP coincides with Delsarte's, and as r increases the LP yields tighter upper bounds on the code's rate, at the cost of higher complexity.

Our numerical experiments (Fig. 1) show that even with $r = 2$ our LP is far stronger than Delsarte's, surpassing it in almost all instances that we were able to solve. The improved upper bound on the code's size is up to 2.5 times smaller than what Delsarte gives. Moreover, in all such instances where Delsarte's upper bound is known not to be tight, we improve it. Nevertheless, our results do not improve the best known upper bounds.

Our construction is based on the elementary fact that a linear code is closed under addition. Combined with Delsarte's LP this simple fact has considerable consequences. To actually derive them we use (i) The language of Boolean Fourier analysis (ii) Symmetry that is inherent in the problem. In analyzing Delsarte's LP, symmetrization reduces the problem size from exponential to polynomial in n, and brings Krawtchouk polynomials to the fore. Also here does symmetrization yield a dramatic reduction in size and reveals the role of multivariate Krawtchouk polynomials. There is a large body of work on these high-dimensional counterparts of univariate Krawtchouks, e.g., [8] and [9].

Although we are still unable to reach our main goal and derive better *asymptotic* bounds, there is good reason for hope. Over forty years since it was proved, the first MRRW bound is still the best upper bound that we have for a large range of parameters. Over the years this bound has been reproved using various tools and techniques. These include, properties of Krawtchouk polynomials [3], analysis of Boolean functions (e.g., [10], [11], [12], [13]), and spectral [14] as well as functional [15] analysis. We believe that it is a viable and promising direction to extend proofs of the first MRRW bound to our multivariate LP family. We are hopeful that this will lead to stronger bounds on the rate of linear codes. We focus here on binary codes, however our methods can be extended to q-ary codes as well.

A. Related Work

A prior work by Coregliano et al. [6] employs closely related ideas to produce a family of linear programs, which upper bound the size of linear codes. In comparison, our LP is stricter due to several conceptual new ideas that we introduce here. Numerical comparisons between our LP and

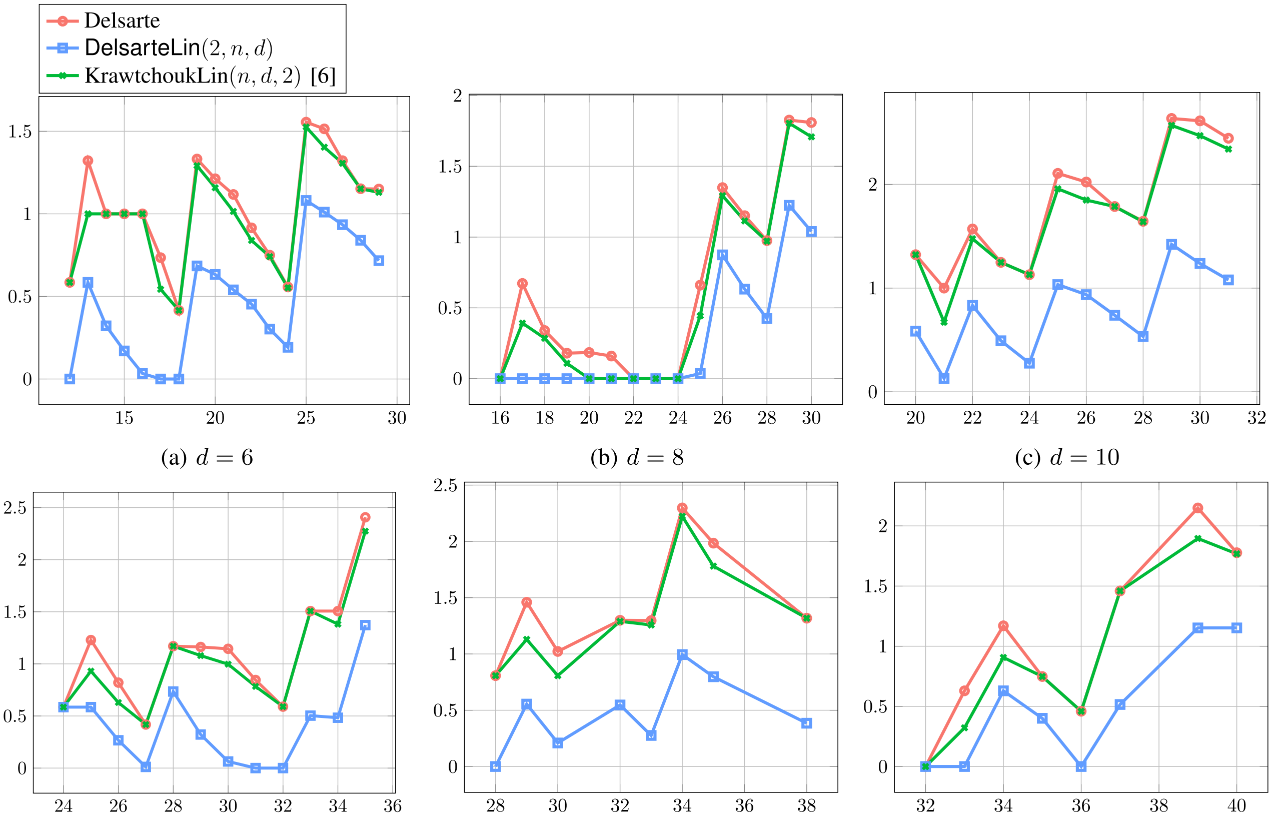
Manuscript received 16 May 2022; revised 27 September 2022; accepted 28 December 2022. Date of publication 16 January 2023; date of current version 21 April 2023. This work was supported in part by "High-Dimensional Combinatorics" of the Israel Science Foundation under Grant 659/18. *(Corresponding author: Elyassaf Loyfer.)*

The authors are with the School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel.

Communicated by I. Tamo, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3236660.

Digital Object Identifier 10.1109/TIT.2023.3236660



(a) $d = 6$ (b) $d = 8$ (c) $d = 10$

(d) $d = 12$ (e) $d = 14$ (f) $d = 16$

Fig. 1. (Lower is better). Numerical results comparing the bounds obtained by optimization problems, and the currently best known upper bounds, as reported in [7]. Each line is the ratio between the optimal value of the corresponding LP and the best known upper bound on $A^{\mathrm{Lin}}(n, d)$, scaled by $\log_2(\cdot)$. The x-axis in each plot varies over n. Experimental setup, as well as more detailed results, are given in Appendix B. For the bounds stated in terms of the codes' *dimension*, see Fig. 4.

that of [6] appear in Fig. 1 and Appendix B. We indicate the differences throughout the text where appropriate, in particular in Sections III-B and III-C.

In [6], they suggest two semi-definite programs (SDPs) which are equivalent to the LP family. One SDP is then used to prove that their program converges to the true bound as the control parameter grows. Here we suggest an alternative proof, which bypasses the use of SDPs.

It was Schrijver [16] who suggested to find an SDP that strengthens Delsarte's LP. His SDP improved the best upper bound for *general codes* in several finite instances, but there is still no known method to improve the asymptotic bounds using this SDP. Our LP yields tighter bounds than those of [16] on all of the instances that we were able to solve (see Appendix B). Our comparison with the results of Schrijver's work should be taken with a grain of salt due to the fact that his SDP does not account for linearity.

B. Organization of This Paper

The rest of the work is organized as follows. Section II provides preliminaries and notation. In Section III we develop our new LP family and discuss some of its properties. In particular, we prove its strength, examine its components, and suggest some variations that may prove useful in the asymptotic analysis. In Section III-D we provide an alternative proof for the convergence theorem of [6]. In Section IV we derive the symmetrized LP.

The derivation of our LPs motivates the definition of a new linear operator which we call *partial Fourier transform*. In Section V we explore some of its characteristics which are relevant to our LP. The main result of this section is an interesting equivalence between two properties of the code's indicator function.

Section VI connects our construction to the literature on multivariate Krawtchouk polynomials. These polynomials appear naturally when we symmetrize the LP. In addition, we develop the *partial* multivariate Krawtchouks, which are derived from the symmetrization of partial Fourier transform.

Appendix B shows results from numerical experiments on a wide range of parameters.

II. Notation and Preliminaries

A. General

We denote by $\mathbb{N}$ the set of nonnegative integers. For a positive integer r, $[r] := \{1, 2, \ldots, r\}$. Vectors are distinguished from scalars by boldface letters, e.g. $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$.

We consider two linear programs *equivalent* if their respective optimal values are equal. Likewise, relations between LPs, e.g. "=," "≤," refer to optimal values. We denote by $\mathrm{val}(\cdot)$ the optimal value of an LP.

B. The Boolean Hypercube

The n-dimensional Boolean hypercube, or simply *the cube* is, as usual, the linear space $\mathbb{F}_2^n$ or the set $\{0,1\}^n$. An element, or a vector, in the cube is denoted in bold, e.g. $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$. Addition $\boldsymbol{x} + \boldsymbol{y} \in \{0,1\}^n$ is bitwise "xor," or element-wise sum modulo 2. Inner product between vectors in the cube is done over $\mathbb{F}_2$: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^n x_i y_i \bmod 2$.

Let $f, g : \{0,1\}^n \to \mathbb{R}$ be two real functions on the cube. Their inner product is defined as

$$\langle f, g \rangle = 2^{-n} \sum_{\boldsymbol{x} \in \{0,1\}^n} f(\boldsymbol{x}) g(\boldsymbol{x})$$

and their convolution

$$(f * g)(\boldsymbol{x}) = 2^{-n} \sum_{\boldsymbol{y} \in \{0,1\}^n} f(\boldsymbol{y}) g(\boldsymbol{x} + \boldsymbol{y})$$

The tensor product of f and g is a function on $\{0,1\}^{2n}$:

$$(f \otimes g)(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) g(\boldsymbol{y}).$$

The Hamming weight of $\boldsymbol{x}$, denoted $|\boldsymbol{x}|$, is the number of non-zero bits, $|\boldsymbol{x}| = |\{1 \le i \le n : x_i \neq 0\}|$. For $i = 0, \ldots, n$, the i-th *level-set* is the set of all Boolean vectors of weight i. The indicator of the i-th level-set is called L_i:

$$L_i(\boldsymbol{x}) = \begin{cases} 1 & |\boldsymbol{x}| = i \\ 0 & \text{o/w}, \end{cases} \quad \boldsymbol{x} \in \{0,1\}^n$$

We denote Kronecker's delta function by $\delta_{\boldsymbol{x}}(\boldsymbol{y})$.

The *Fourier character* corresponding to $\boldsymbol{x} \in \{0,1\}^n$, denoted $\chi_{\boldsymbol{x}}$, is defined by

$$\chi_{\boldsymbol{x}}(\boldsymbol{y}) = (-1)^{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}, \quad \boldsymbol{y} \in \{0,1\}^n.$$

The set of characters $\{\chi_{\boldsymbol{x}}\}_{\boldsymbol{x} \in \{0,1\}^n}$ is an orthonormal basis for the space of real functions on the cube. The Fourier transform of a function $f : \{0,1\}^n \to \mathbb{R}$ is its projection over the characters, $\hat{f}(\boldsymbol{x}) = \langle f, \chi_{\boldsymbol{x}} \rangle = 2^{-n} \sum_{\boldsymbol{y}} \chi_{\boldsymbol{x}}(\boldsymbol{y}) f(\boldsymbol{y})$. In Fourier space, the inner product is not normalized: $\langle \hat{f}, \hat{g} \rangle_{\mathcal{F}} = \sum_{\boldsymbol{x}} \hat{f}(\boldsymbol{x}) \hat{g}(\boldsymbol{x})$.

We recall Parseval's identity: $\langle f, g \rangle = \langle \hat{f}, \hat{g} \rangle$; and the convolution theorem: $\widehat{(f * g)}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) \hat{g}(\boldsymbol{x})$.

The partial Fourier transform, denoted $\mathcal{F}_S$ that we introduce here plays an important role in our work, see Section III for details.

A comprehensive survey of harmonic analysis of Boolean functions can be found in [17].

C. Codes

A *binary code* of length n is a subset $\mathcal{C} \subset \{0,1\}^n$. Its *distance* is the smallest Hamming distance between pairs of words, $dist(\mathcal{C}) = \min_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}} |\boldsymbol{x} + \boldsymbol{y}|$. The largest cardinality of a code of length n and distance d is denoted by $A(n, d)$. The *rate* of $\mathcal{C}$ is defined as

$$R(\mathcal{C}) = n^{-1} \log_2(|\mathcal{C}|)$$

The rate-vs.-distance problem is to find, for every $n, d \in \mathbb{N}$, the quantity

$$R(n, d) := \max_{\mathcal{C}: dist(\mathcal{C}) \ge d} \log_2 |\mathcal{C}|/n$$

The asymptotic version of the problem is to find $\mathcal{R}(\delta) := \limsup_{n \to \infty} R(n, \delta n)$, for every $\delta \in (0, 1/2)$.

A linear code is a linear subspace. In the binary case, $\mathcal{C} \subset \{0,1\}^n$ is linear if and only if $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C} \Rightarrow \boldsymbol{x} + \boldsymbol{y} \in \mathcal{C}$, for every $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$. Consequently, in a linear code $dist(\mathcal{C}) = \min_{\boldsymbol{0} \neq \boldsymbol{x} \in \mathcal{C}} |\boldsymbol{x}|$. We denote by $A^{\mathrm{Lin}}(n, d)$ the maximal size of a binary linear code of length n and minimal distance d.

III. New Linear Programs

In this section we present a new family of linear programs, starting from Delsarte's LP. Later in this Section we discuss possible modifications to the LPs.

Let $\mathcal{C} \subset \{0,1\}^n$ be a code, not necessarily linear, with minimal distance d. Let $\mathbf{1}_{\mathcal{C}}$ be its indicator function, namely $\mathbf{1}_{\mathcal{C}}(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in \mathcal{C}$, and 0 otherwise. Define the function

$$f_{\mathcal{C}} = \frac{2^n}{|\mathcal{C}|} \mathbf{1}_{\mathcal{C}} * \mathbf{1}_{\mathcal{C}}$$

As we explain shortly, Fourier analysis of $f_{\mathcal{C}}$ yields Delsarte's LP for binary codes. Our new LP family is likewise obtained by considering the tensor product of copies of $f_{\mathcal{C}}$.

Indeed, it is easily verified that $f_{\mathcal{C}}(0) = 1$, and $f_{\mathcal{C}}(\boldsymbol{x}) = 0$ whenever $1 \le |\boldsymbol{x}| \le d - 1$. In addition, $f_{\mathcal{C}} \ge 0$ as a sum of indicator functions. Also, $\hat{f}_{\mathcal{C}} \ge 0$ because, by the convolution theorem, it is a squared function: $\hat{f}_{\mathcal{C}} = \frac{2^n}{|\mathcal{C}|} \hat{\mathbf{1}}_{\mathcal{C}}^2$. Lastly, summing $f_{\mathcal{C}}$ over the entire cube yields the cardinality of $\mathcal{C}$. This yields the following LP, whose optimal value is an upper bound on $A(n, d)$.

Definition 1: $\mathsf{Delsarte}_{cube}(n, d)$ is the following linear program:

$$\underset{f:\{0,1\}^n \to \mathbb{R}}{\text{maximize}} \quad \sum_{\boldsymbol{x} \in \{0,1\}^n} f(\boldsymbol{x}) \qquad (obj)$$

subject to:

$$f(0) = 1, \qquad (d1)$$

$$f \ge 0, \ \hat{f} \ge 0, \qquad (d2)$$

$$f(\boldsymbol{x}) = 0 \qquad \text{if } 1 \le |\boldsymbol{x}| \le d - 1 \qquad (d3)$$

Now let us assume further that $\mathcal{C}$ is linear. In this case, $\mathbf{1}_{\mathcal{C}}(\boldsymbol{x}) \mathbf{1}_{\mathcal{C}}(\boldsymbol{y}) = \mathbf{1}_{\mathcal{C}}(\boldsymbol{x}) \mathbf{1}_{\mathcal{C}}(\boldsymbol{x} + \boldsymbol{y})$, and consequently,

$$f_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \mathbf{1}_{\mathcal{C}} * \mathbf{1}_{\mathcal{C}} = \mathbf{1}_{\mathcal{C}}.$$

This implies a new set of constraints that hold for linear codes and can be added to the above LP:

$$f(\boldsymbol{x}) f(\boldsymbol{y}) = f(\boldsymbol{x}) f(\boldsymbol{x} + \boldsymbol{y})$$

However, these constraints are not linear in f, nor even convex.

Therefore, we consider instead tensor products of $f_{\mathcal{C}}$. Let $r \ge 1$ be an integer and define

$$f_{\mathcal{C}^r} = f_{\mathcal{C}} \otimes \cdots \otimes f_{\mathcal{C}} : \{0,1\}^{rn} \to \mathbb{R}.$$

The function $f_{\mathcal{C}^r}$ is defined on the rn-dimensional cube. We will view its argument as either a concatenation of r vectors in $\{0,1\}^n$, or an $r \times n$ matrix obtained by stacking the r vectors. For example, we write

$$f_{\mathcal{C}^r}(X) = f_{\mathcal{C}^r}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r)$$

where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \in \{0,1\}^n$ are the rows of the matrix $X \in \{0,1\}^{r \times n}$.

As suggested above, our LP family is derived from the linear properties of $f_{\mathcal{C}^r}$. Some of these properties apply even for non-linear $\mathcal{C}$ and are inherited from the properties of the original $f_{\mathcal{C}}$. The other type is properties that depend on the linearity of $\mathcal{C}$. We turn to describe both types.

We begin with the first type. It is clear that $f_{\mathcal{C}^r}(\mathbf{0}) = 1$, and $f_{\mathcal{C}^r}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = 0$ if any of the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r$ has weight between 1 and $d-1$. The non-negativity of $f_{\mathcal{C}}$ and $\hat{f}_{\mathcal{C}}$ imply the same for $f_{\mathcal{C}^r}$. But there is more: products of $f_{\mathcal{C}}$ and $\hat{f}_{\mathcal{C}}$ are also non-negative, e.g. $f_{\mathcal{C}}(\boldsymbol{x}_1)\hat{f}_{\mathcal{C}}(\boldsymbol{x}_2) \geq 0$ for every $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \{0,1\}^n$.

This motivates the definition of a new linear operator, which we name *partial Fourier transform.*

Definition 2: Let $S \subset [r]$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \in \{0,1\}^n$. The **partial Fourier character** $\Psi^S_{(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r)}$ is defined by

$$\Psi^S_{(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r)} := \psi^{(1)}_{\boldsymbol{x}_1} \otimes \psi^{(2)}_{\boldsymbol{x}_2} \otimes \cdots \otimes \psi^{(r)}_{\boldsymbol{x}_r}$$

where, given $\boldsymbol{x} \in \{0,1\}^n$,

$$\psi^{(i)}_{\boldsymbol{x}} := \begin{cases} \chi_{\boldsymbol{x}} & i \in S \\ \delta_{\boldsymbol{x}} & \text{o/w} \end{cases}$$

$\chi_{\boldsymbol{x}}$ is a Fourier character in $\{0,1\}^n$ and $\delta_{\boldsymbol{x}}$ is Kronecker's delta.

The **partial Fourier transform** is the linear projection of a function $g : \{0,1\}^{rn} \to \mathbb{R}$ on the partial characters,

$$\begin{aligned}\mathcal{F}_S(g)(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) &= 2^{(r-|S|)n} \langle g, \Psi^S_{(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r)} \rangle \\ &= 2^{-|S|n} \sum g(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r) \times \\ &\quad \times \prod_{i \in S} \chi_{\boldsymbol{x}_i}(\boldsymbol{y}_i) \prod_{i \in [r] \setminus S} \delta_{\boldsymbol{x}_i}(\boldsymbol{y}_i)\end{aligned}$$

the sum running over all $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r \in \{0,1\}^n$.
Observe that $\mathcal{F}_\emptyset(g) = g$, $\mathcal{F}_{[r]}(g) = \hat{g}$, and $\mathcal{F}_{\{i,j\}}(g) = \mathcal{F}_{\{i\}}(\mathcal{F}_{\{j\}}(g))$, for $1 \leq i, j \leq r$, $i \neq j$.

Using the new notation, we have $\mathcal{F}_S(f_{\mathcal{C}^r}) \geq 0$ for every $S \subset [r]$.

The last inherited property of $f_{\mathcal{C}^r}$ has to do with the cardinality of $\mathcal{C}$. Summing $f_{\mathcal{C}^r}$ over the entire rn-dimensional cube yields $|\mathcal{C}|^r$. Alternatively, one can obtain the value of $|\mathcal{C}|$ by summing one component over $\{0,1\}^n$, and fixing the other components at 0:

$$\sum_{\boldsymbol{x} \in \{0,1\}^n} f_{\mathcal{C}^r}(\boldsymbol{x}, 0, \ldots, 0) = \sum_{\boldsymbol{x} \in \{0,1\}^n} f_{\mathcal{C}}(\boldsymbol{x}) \left(f_{\mathcal{C}}(0)\right)^{r-1} = |C|$$

We turn to discuss the properties which depend on the linearity of the code $\mathcal{C}$. If $\mathcal{C}$ is a linear code and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \in \mathcal{C}$, then $\mathcal{C}$ contains their linear span. Hence

$$f_{\mathcal{C}^r}(X) = \prod_{i=1}^{r} \mathbf{1}_{\mathcal{C}}(\boldsymbol{x}_i) = \prod_{\boldsymbol{x} \in rowspan(X)} \mathbf{1}_{\mathcal{C}}(\boldsymbol{x})$$

which implies that $f_{\mathcal{C}^r}$ is invariant under the action of $\mathrm{GL}(r,2)$, the general linear group over $\mathbb{F}_2$.

$$f_{\mathcal{C}^r}(X) = f_{\mathcal{C}^r}(TX) \quad \forall T \in \mathrm{GL}(r,2),\ X \in \{0,1\}^{r \times n} \quad (1)$$

One more interesting property involves the dual code,

$$\mathcal{C}^\perp := \{\boldsymbol{x} \in \{0,1\}^n : \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{F}_2} = 0\ \forall \boldsymbol{y} \in \mathcal{C}\}.$$

If $\mathcal{C}$ is linear, then the Fourier transform of its indicator $\hat{\mathbf{1}}_{\mathcal{C}}$ is the indicator of the dual code, up to normalization (see e.g., [17], Proposition 3.11):

$$\hat{\mathbf{1}}_{\mathcal{C}} = \frac{1}{|\mathcal{C}^\perp|} \mathbf{1}_{\mathcal{C}^\perp}$$

This fact can be utilized through the partial Fourier transform as follows. Let $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$ such that $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{F}_2} \neq 0$, then either $\boldsymbol{x} \notin \mathcal{C}^\perp$ or $\boldsymbol{y} \notin \mathcal{C}$. Consequently, if $i \in S$ and $j \notin S$ for some $S \subseteq [r]$, and $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle_{\mathbb{F}_2} \neq 0$, then

$$\mathcal{F}_S(f_{\mathcal{C}^r})(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = \left(\prod_{k \in S} \frac{1}{|\mathcal{C}^\perp|} \mathbf{1}_{\mathcal{C}^\perp}(\boldsymbol{x}_k)\right) \left(\prod_{k \in [r] \setminus S} \mathbf{1}_{\mathcal{C}}(\boldsymbol{x}_k)\right) = 0 \quad (2)$$

Surprisingly perhaps, this adds no new information: properties (1) and (2) are equivalent, as we show in Section V.

This concludes our discussion on the linear properties of the tensor product $f_{\mathcal{C}^r}$. We are now ready to define the new LP family.

Definition 3: $\mathsf{DelsarteLin}(r, n, d)$:

$$\underset{f:\{0,1\}^{rn} \to \mathbb{R}}{\text{maximize}} \sum_{\boldsymbol{x} \in \{0,1\}^n} f(\boldsymbol{x}, 0, \ldots, 0) \quad (Obj)$$

subject to:

$$\begin{aligned}
&f(\mathbf{0}) = 1 && && (C1)\\
&\mathcal{F}_S(f) \geq 0 && \forall S \subset [r] && (C2)\\
&f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = 0 && \text{if } 1 \leq |\boldsymbol{x}_1| \leq d-1 && (C3)\\
&f(X) = f(TX) && \forall T \in \mathrm{GL}(r,2),\ X \in \{0,1\}^{r \times n} && (C4)
\end{aligned}$$

Here, $\mathcal{F}_S(f)$ is the partial Fourier transform defined above. Also, $\mathrm{GL}(r,2)$ is the general linear group over $\mathbb{F}_2$. Note also the parallels between conditions $(d1)$, $(d2)$, $(d3)$ resp. $(C1)$, $(C2)$, $(C3)$

In comparison, the LP family of [6] does not include the partial Fourier constraints, and differs in the objective function. It can be stated as follows.

$$\underset{f:\{0,1\}^{rn} \to \mathbb{R}}{\text{maximize}} \sum_{X \in \{0,1\}^{r \times n}} f(X)$$

subject to:

$$\begin{aligned}
&f(\mathbf{0}) = 1 \\
&f \geq 0,\ \hat{f} \geq 0 \\
&f(X) = 0 && \text{if } \exists\, \boldsymbol{x} \in \mathrm{rowspan}(X) \\
& && \text{s.t. } 1 \leq |\boldsymbol{x}| \leq d-1
\end{aligned}$$

Theorem 1: Let r, n, d be positive integers such that $d \leq n/2$.

1) $A^{\text{Lin}}(n,d) \leq \mathrm{val}\, \mathsf{DelsarteLin}(r, n, d)$
2) $\mathrm{val}\, \mathsf{DelsarteLin}(r+1, n, d) \leq \mathrm{val}\, \mathsf{DelsarteLin}(r, n, d)$
3) $\mathrm{val}\, \mathsf{DelsarteLin}(1, n, d) = \mathrm{val}\, \mathsf{Delsarte}_{cube}(n, d)$

We make a few comments before we turn to the proof. Already for $r = 2$, and in most instances, $\mathsf{DelsarteLin}$ is significantly stronger than Delsarte's. For more on this, see Figure 1 and Section B. We also note that $\mathsf{DelsarteLin}$ without $(C4)$ yields exactly the bounds as Delsarte's LP.

Proof:

1) By the preceding discussion, for every binary linear code $\mathcal{C}$ of length n and minimal distance d, $f_{\mathcal{C}^r}$ is a feasible solution with value $|\mathcal{C}|$.
2) Let $f : \{0,1\}^{(r+1)n} \to \mathbb{R}$ be a feasible solution to $\mathsf{DelsarteLin}(r+1, n, d)$. We construct a feasible solution to $\mathsf{DelsarteLin}(r, n, d)$ with value at least $\mathrm{val}(f)$. Let

$$g : \{0,1\}^{rn} \to \mathbb{R}, \quad g(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r, \mathbf{0})$$

It is easy to verify that g is feasible for $\mathsf{DelsarteLin}(r, n, d)$, and it is clear that $\mathrm{val}(g) = \mathrm{val}(f)$.
3) Obvious, $\mathsf{DelsarteLin}(1, n, d)$ and $\mathsf{Delsarte}_{cube}(n, d)$ are identical.

□

In the rest of this section, we examine the strength and consequences of some components of $\mathsf{DelsarteLin}$. We also discuss two modifications that may be helpful in the search for asymptotic results.

A. On the Significance of $(C4)$

As mentioned above, $(C4)$ is equivalent to a constraint that uses the dual code:

$$\mathcal{F}_S(f)(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = 0 \quad \text{if } \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle_{\mathbb{F}_2} = 1 \text{ for some } i \in S,\ j \notin S \qquad (C5)$$

We prove the equivalence below, in Lemma 1. As $(C4)$ and $(C5)$ are the only constraints that rely on the code's linearity, without them the LP is equivalent to Delsarte's LP, for every r.

An obvious consequence of $(C4)$ is that $(C3)$ is equivalent to

$$f(X) = 0 \text{ if } 1 \le |\boldsymbol{u}^\intercal X| \le d - 1 \text{ for some } \boldsymbol{u} \in \{0,1\}^r \qquad (C3')$$

$\{\boldsymbol{u}^\intercal X\}_{\boldsymbol{u} \in \{0,1\}^r}$ is the row span of X. Similarly, $(C4)$ renders the objective (Obj) equivalent to

$$\text{maximize } (2^r - 1)^{-1} \sum_{0 \neq \boldsymbol{u} \in \{0,1\}^r} \sum_{\boldsymbol{x} \in \{0,1\}^n} f(u_1 \cdot \boldsymbol{x}, \ldots, u_r \cdot \boldsymbol{x}) \qquad (Obj'')$$

To numerically test the significance of $(C4)$, we removed it but kept its immediate consequences. Namely, we replaced (Obj) and $(C3)$ with (Obj'') and $(C3')$. A sample from our numerical experiments is shown in Figure 2. It confirms that this change does weaken the LP, though not significantly. However, we only experimented with $r = 2$, and it is possible that for larger values of r the difference becomes more substantial.

Lastly, $(C4)$ implies other symmetries for $\mathcal{F}_S(f)$. While these do not strengthen the LP, they provide an exponential in r reduction in the number of constraints. For proof, see Lemma 1.

$$\mathcal{F}_S(f)(X) = \mathcal{F}_S(f)(T_1 T_2 X), \qquad (C6)$$

for every $T_1, T_2 \in \mathrm{GL}(r, 2)$, such that $T_1 e_i = e_i\ \forall i \in S$ and $T_2 e_i = e_i\ \forall i \in [r] \setminus S$.

$$\mathcal{F}_S(f)(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = \mathcal{F}_{\pi^{-1}(S)}(f)(\boldsymbol{x}_{\pi(1)}, \ldots, \boldsymbol{x}_{\pi(r)}), \quad (C7)$$

for every $\pi \in \mathfrak{S}_r$ – permutation on r elements.

n	d \ r	1 Delsarte	2 $\mathsf{DelsarteLin}$	2 $(Obj''), (C3'), \cancel{(C4)}$
16	4	2048	2048	2048
	6	256	131.72	156.44
	8	32	32	32
17	4	3640.89	3072.96	3075
	6	425.56	256	264.88
	8	50.72	32	32.31

Fig. 2. Numerical experiments on the significance of $(C4)$. The first column is Delsarte's LP. The second column is $\mathsf{DelsarteLin}(2, n, d)$. The third column is a modification of $\mathsf{DelsarteLin}(2, n, d)$, where $(C3)$ is replaced by $(C3')$; the objective function is replaced by (Obj''); and $(C4)$ is removed.

n	d \ r	1 Delsarte	2 $(C2)$	2 $(C2')$	3 $(C2')$
13	6	40	24.26	32	23.07
30	8	114816	71094.5	107044	-
30	10	12525.4	5928.52	11340.4	-
30	12	1131.79	582.09	1026.28	-
30	14	129.68	80.08	112	-

Fig. 3. Comparison between $(C2)$, $(C2')$ and Delsarte. Each column shows the optimal value a of different LP. The LPs from left to right: Delsarte's LP; Our LP with $r = 2$; Our modified LP with $(C2')$ instead of $(C2)$, with $r = 2$; and again the modified LP, with $r = 3$. This exhausts the results that we have for $r = 3$.

B. On the Significance of $(C2)$

A weaker, simpler LP is obtained from $\mathsf{DelsarteLin}(r, n, d)$ by replacing $(C2)$ with

$$f \ge 0,\ \hat{f} \ge 0 \qquad (C2')$$

This modification restores the feasible region of the LP developed by [6]. The modified LP is still stronger than Delsarte's LP, and it becomes stronger with growing r, as we prove in Theorem 2. Its simplicity might make it more suitable for asymptotic analysis.

We observed empirically that this modification greatly weakens the LP. A small sample is given here in Figure 3, and more can be found in Section B and in Figure 1.

Theorem 2: Let r, n, d be positive integers such that $d \le n/2$. For every binary linear code $\mathcal{C}$ with length n and distance d,

$$|\mathcal{C}| \le \mathrm{val}\,\mathsf{DL}_{(C2')}(r+1, n, d) \le \mathrm{val}\,\mathsf{DL}_{(C2')}(r, n, d)$$

where $\mathsf{DL}_{(C2')}(r,n,d)$ is the variant of $\mathsf{DelsarteLin}(r,n,d)$ in which $(C2)$ is replaced by $(C2')$.

Proof: The first inequality follows from Theorem 1, by noting that every feasible solution to $\mathsf{DelsarteLin}(r,n,d)$ is a feasible solution to the modified version.

For the second inequality, let $f:\{0,1\}^{(r+1)n}\to\mathbb{R}$ be a feasible solution to $\mathsf{DL}_{(C2')}(r+1,n,d)$. Define

$$g:\{0,1\}^{rn}\to\mathbb{R},\quad g(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r)=f(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r,\boldsymbol{0})$$

It is obvious that $g(\boldsymbol{0})=1$; $g\geq 0$; $g(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r)=0$ if $1\leq|\boldsymbol{x}_1|\leq d-1$; and that $g(X)=g(TX)$ for every $T\in\mathrm{GL}(r,2)$. To prove that g is feasible, it remains to show that $\hat{g}\geq 0$. Observe that

$$\begin{aligned}\hat{g}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r)&=\mathcal{F}_{\{1,\dots,r\}}(f)(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r,\boldsymbol{0})\\&=2^n\mathcal{F}_{\{r+1\}}(\hat{f})(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r,\boldsymbol{0})\\&=\sum_{\boldsymbol{y}\in\{0,1\}^n}\chi_{\boldsymbol{0}}(\boldsymbol{y})\hat{f}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r,\boldsymbol{y})\\&=\sum_{\boldsymbol{y}\in\{0,1\}^n}\hat{f}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r,\boldsymbol{y})\end{aligned}$$

which is non-negative since $\hat{f}\geq 0$. The value of f equals the value of g, which is at most $\operatorname{val}\mathsf{DL}_{(C2')}(r,n,d)$.

$\square$

C. On the Objective Function

As discussed above, an alternative objective function can be used, which bounds $\left(A^{\mathrm{Lin}}(n,d)\right)^r$ instead of $A^{\mathrm{Lin}}(n,d)$:

$$\text{maximize}\sum_{\boldsymbol{x}_1,\dots,\boldsymbol{x}_r\in\{0,1\}^n}f(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r)\qquad(Obj')$$

This is the objective function used in [6].

Our numerical calculations reveal rather minor differences between the two objectives, with no consistent advantage to one over the other. See detailed results in Section B.

We state:

Conjecture 1: Let r,n,d be positive integers such that $d\leq n/2$. Then

$$\left(\operatorname{val}\mathsf{DL}_{(Obj')}(r+1,n,d)\right)^{1/(r+1)}\leq\left(\operatorname{val}\mathsf{DL}_{(Obj')}(r,n,d)\right)^{1/r}$$

Here, $\mathsf{DL}_{(Obj')}(r,n,d)$ is obtained from $\mathsf{DelsarteLin}(r,n,d)$ by replacing the objective function with (Obj').

Due to the non-linear relation between the two objective functions we are presently only able to prove the following. A similar Theorem can likewise be proved for the variant where $(C2')$ replaces $(C2)$.

Theorem 3: Let r,n,d be positive integers such that $d\leq n/2$. Then

$$\left(\operatorname{val}\mathsf{DL}_{(Obj')}(r+1,n,d)\right)^{1/(r+1)}\leq\max\begin{cases}\left(\operatorname{val}\mathsf{DL}_{(Obj')}(r,n,d)\right)^{1/r},\\\operatorname{val}\mathsf{DelsarteLin}(r+1,n,d)\end{cases}$$

Proof: Let $f:\{0,1\}^{(r+1)n}\to\mathbb{R}$ be a feasible solution to $\mathsf{DL}_{(Obj')}(r+1,n,d)$. Then f is also a feasible solution to $\mathsf{DelsarteLin}(r+1,n,d)$. Let

$$\begin{aligned}v_1&=\left(\sum f(\boldsymbol{x}_1,\dots,\boldsymbol{x}_{r+1})\right)^{1/(r+1)}\\v_2&=\sum f(\boldsymbol{x},\boldsymbol{0},\dots,\boldsymbol{0})\end{aligned}$$

where the sums are over $\boldsymbol{x}_1,\dots,\boldsymbol{x}_{r+1}\in\{0,1\}^n$ and over $\boldsymbol{x}\in\{0,1\}^n$, respectively.

If $v_1\leq v_2$ then we are done, because v_2 is not greater than the optimum of $\mathsf{DelsarteLin}(r+1,n,d)$.

Otherwise, $v_1>v_2$. Define $g:\{0,1\}^{rn}\to\mathbb{R}$ as

$$g(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r)=\frac{1}{v_2}\sum_{\boldsymbol{y}\in\{0,1\}^n}f(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r,\boldsymbol{y})$$

It is not hard to verify that g is a feasible solution to $\mathsf{DL}_{(Obj')}(r,n,d)$. Now consider its value:

$$\left(\sum g(\boldsymbol{x}_1,\dots,\boldsymbol{x}_r)\right)^{1/r}=\left(\frac{v_1^{r+1}}{v_2}\right)^{1/r}\geq v_1$$

and the value of g is at most the optimal value of $\mathsf{DL}_{(Obj')}(r,n,d)$.

$\square$

D. Approximate Completeness

Coregliano et al. [6] prove that for r large enough, the LP family with the objective function (Obj') converges to $A^{\mathrm{Lin}}(n,d)^r$. For binary linear codes, it can be stated as follows:

Theorem 4 (Approximate Completeness): Let $\varepsilon\in(0,1)$ and $r\geq 2n^2/\log_2(1+\varepsilon)$. Then

$$\left(\operatorname{val}\mathsf{DL}_{(Obj')}(r,n,d)\right)^{1/r}\leq(1+\varepsilon)A^{\mathrm{Lin}}(n,d)$$

The proof in [6] is based on an SDP formulation which is equivalent to the LP family. The idea of the proof is to upper-bound the variables, and then count the non-zero variables. Our proof follows the same idea, without using an SDP. The following proposition provides upper bounds on the variables, which is followed by a count of the non-zero variables.

Proposition 1: Let $f:\{0,1\}^n\to\mathbb{R}$ such that $f(0)=1$ and $\hat{f}\geq 0$. Then $f\leq 1$.

Proof: Let $0\neq\boldsymbol{x}\in\{0,1\}^n$. Since $\hat{f}\geq 0$, we have

$$0\leq\sum_{\boldsymbol{y}:\langle\boldsymbol{y},\boldsymbol{x}\rangle_{\mathbb{F}_2}=1}\hat{f}(\boldsymbol{y})=\sum_{\boldsymbol{y}:\langle\boldsymbol{y},\boldsymbol{x}\rangle_{\mathbb{F}_2}=1}\sum_{\boldsymbol{z}\in\{0,1\}^n}\chi_{\boldsymbol{y}}(\boldsymbol{z})f(\boldsymbol{z})$$

For every $\boldsymbol{y}$ in the sum, there holds $\chi_{\boldsymbol{y}}(\boldsymbol{x})=-1$ and $\chi_{\boldsymbol{y}}(0)=1$. Hence,

$$0\leq 2^{n-1}f(0)-2^{n-1}f(\boldsymbol{x})+\sum_{\boldsymbol{z}\neq 0,\boldsymbol{x}}f(\boldsymbol{z})\sum_{\boldsymbol{y}:\langle\boldsymbol{y},\boldsymbol{x}\rangle_{\mathbb{F}_2}=1}\chi_{\boldsymbol{y}}(\boldsymbol{z})$$

We complete the proof by showing that the last term vanishes. So, let $[\boldsymbol{x},\boldsymbol{z}]$ be the $2\times n$ matrix whose rows are $\boldsymbol{x}$ and $\boldsymbol{z}$. The action of multiplying $[\boldsymbol{x},\boldsymbol{z}]$ by $\boldsymbol{y}\in\{0,1\}^n$ divides the n-dimensional cube into cosets in $\mathbb{F}_2^2$. If $\boldsymbol{x}\neq\boldsymbol{z}$ and both are non-zero, then each coset has cardinality 2^{n-2}. The inner sum is over the cosets $(1,0)$ and $(1,1)$. If $\boldsymbol{y}$ is in the first coset, then $\chi_{\boldsymbol{z}}(\boldsymbol{y}) = 1$, and if it is in the second then $\chi_{\boldsymbol{z}}(\boldsymbol{y}) = -1$. In total, the sum vanishes. $\square$

Proof of Theorem 4: Let f be a solution to $\mathsf{DL}_{(Obj')}(r,n,d)$. By Proposition 1, $f \le 1$.

Let k_0 be the largest possible dimension of a binary linear code with length n and distance d, namely $2^{k_0} = A^{\mathrm{Lin}}(n,d)$. Then f vanishes of every $r \times n$ binary matrix of $\mathbb{F}_2$-rank larger than k_0. Then, the value of the LP corresponding to f is at most $\sum_{k=0}^{k_0} \gamma_{n,r,k}$, where $\gamma_{n,r,k}$ is the number of such matrices of rank exactly k.

We next derive an upper bound on $\gamma_{n,r,k}$. There are exactly $\prod_{i=1}^{k}(2^n - 2^i) \le 2^{nk}$ ordered bases of k-dimensional subspaces of $\mathbb{F}_2^n$. There are $r(r-1)\cdots(r-k+1) \le r^k$ possible ways to place the chosen ordered base in an $r \times n$ matrix, and then 2^k options to choose each of the remaining rows without increasing the rank. Hence, $\gamma_k \le 2^{nk} r^k 2^{k(r-k)}$, and

$$\begin{aligned}
\operatorname{val}\mathsf{DL}_{(Obj')}(r,n,d) &\le \sum_{k=0}^{k_0} 2^{nk} r^k 2^{kr} \\
&\le (k_0+1)2^{k_0(n+r+\log_2(r))} \\
&\le 2^{n^2 + n\log_2(r) + \log_2(n+1)} A^{\mathrm{Lin}}(n,d)^r \\
&\le (1+\varepsilon)^r A^{\mathrm{Lin}}(n,d)^r
\end{aligned}$$

in the last inequality we use the assumption that $r \ge 2n^2/\log_2(1+\varepsilon)$. $\square$

IV. Symmetrized Linear Programs

Due to the inherent symmetries of the LPs from section III they can be symmetrized without affecting the objective function. The advantage is that the symmetrized LP is significantly smaller than the original form. This is what we consider in this section.

Let $\mathfrak{S}_n$ be the symmetric group on n elements. It acts on $\{0,1\}^{r\times n}$ by column permutations:

$$\sigma \cdot X = [\boldsymbol{\xi}_{\sigma(1)}, \dots, \boldsymbol{\xi}_{\sigma(n)}]$$

where $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n$ are the columns of $X \in \{0,1\}^{r\times n}$, and $\sigma \in \mathfrak{S}_n$. It also acts on functions $f : \{0,1\}^{r\times n} \to \mathbb{R}$ via $(\sigma \circ f)(X) = f(\sigma \cdot X)$.

We say that a solution f to $\mathsf{DelsarteLin}(r,n,d)$ is *symmetric* if it is constant on $\mathfrak{S}_n$-orbits, i.e., if $f = \sigma \circ f$ for every $\sigma \in \mathfrak{S}_n$. Symmetric solutions can clearly be described more concisely, and as we observe below, there exist optimal symmetric solutions.

Generally speaking, suppose that the group G acts on the variables of a linear program $\mathcal{P}$. We say that f, a feasible solution of $\mathcal{P}$ is *G-invariant* if $g \circ f$ is feasible and $\operatorname{val}(g\circ f) = \operatorname{val}(f)$, for every $g \in G$. If every feasible solution is invariant, we say that $\mathcal{P}$ is G-invariant. An invariant solution f need not be symmetric, but averaging can yield a symmetric solution via

$$\overline{f} := |G|^{-1} \sum_{g\in G} g \circ f$$

By linearity and convexity, $\overline{f}$ is feasible and has the same value as f. Consequently, a G-invariant LP has a symmetric optimal solution.

Let us verify that $\mathsf{DelsarteLin}(r,n,d)$ is $\mathfrak{S}_n$-invariant. Let f be a feasible solution and $\sigma \in \mathfrak{S}_n$.

- $(C1)$ $f(\sigma \cdot \mathbf{0}) = f(\mathbf{0}) = 1$.
- $(C2)$ By Proposition 4 from Section V below, if $\mathcal{F}_S(f) \ge 0$ then also $\mathcal{F}_S(\sigma \circ f) \ge 0$.
- $(C3)$ Row weights are invariant under column permutations.
- $(C4)$ Permuting of the columns of X is equivalent to multiplication from the right by a permutation matrix P. Since matrix multiplication is associative,
$$(\sigma \circ f)(TX) = f(T(XP)) = f(XP) = (\sigma \circ f)(X)$$
for every $T \in \mathrm{GL}(r,2)$.
- (Obj) (also (Obj')) Permutation only affects the order of summation, but not the total sum.

Hence, $\sigma \circ f$ is a feasible solution with the same value as f.

Therefore, there is no loss in restricting to symmetric solutions of $\mathsf{DelsarteLin}$, i.e., to solutions f that are constant on the orbits $\{0,1\}^{r\times n}/\mathfrak{S}_n$. Such solutions can be expressed as a linear combination of orbit indicators:

$$f(X) = \sum_{Orb \in \{0,1\}^{r\times n}/\mathfrak{S}_n} \varphi_{Orb} \cdot \mathbf{1}_{Orb}(X)$$

where $\mathbf{1}_{Orb} : \{0,1\}^{r\times n} \to \{0,1\}$ is the indicator function of the set $Orb \in \{0,1\}^{r\times n}/\mathfrak{S}_n$, and (φ_{Orb}) are real numbers. To exploit this symmetry we reformulate the LP in terms of (φ_{Orb}).

The following definition will be useful in depicting the set of orbits.

Definition 4: Let $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n \in \{0,1\}^r$ be the columns of $X \in \{0,1\}^{r\times n}$. The **column enumerator** of X counts how many times each vector in $\{0,1\}^r$ appears as a column in X:

$$\Gamma_X \in \mathbb{N}^{2^r}, \quad \Gamma_X(\boldsymbol{u}) = |\{1 \le i \le n : \boldsymbol{\xi}_i = \boldsymbol{u}\}|$$

Observe that when $r = 1$, $\Gamma_{\boldsymbol{x}}(1) = |\boldsymbol{x}|$ and $\Gamma_{\boldsymbol{x}}(0) = n - |\boldsymbol{x}|$.

The column enumerator of a matrix clearly determines its orbit, i.e., $\mathfrak{S}_n \cdot X = \mathfrak{S}_n \cdot Y \iff \Gamma_X = \Gamma_Y$. The set of orbits $\{0,1\}^{r\times n}/\mathfrak{S}_n$ is therefore isomorphic to the set of all possible column enumerators, which we denote by $\mathcal{I}_{r,n}$,

$$\mathcal{I}_{r,n} := \{\boldsymbol{\alpha} = (\alpha_0, \dots, \alpha_{2^r-1}) : \alpha_i \in \mathbb{N}, \sum_{i=0}^{2^r-1} \alpha_i = n\} \quad (3)$$

Equivalently, it is the set of ordered partitions of n into 2^r parts. In the sequel, we will introduce a different equivalent way of looking at $\mathcal{I}_{r,n}$.

The level-set indicator function of $\boldsymbol{\alpha} \in \mathcal{I}_{r,n}$ is defined via

$$L_{\boldsymbol{\alpha}} : \{0,1\}^{r\times n} \to \{0,1\}, \quad L_{\boldsymbol{\alpha}}(X) = \begin{cases} 1 & \Gamma_X = \boldsymbol{\alpha} \\ 0 & \text{o/w} \end{cases}$$

This allows us to express any symmetric solution to $\mathsf{DelsarteLin}(r,n,d)$ as follows:

$$f = \sum_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}} \varphi_{\boldsymbol{\alpha}} L_{\boldsymbol{\alpha}}$$

We need to introduce some more notation. Let $\epsilon_{\boldsymbol{u}} : \{0,1\}^r \to \mathbb{R}$ be the indicator of $\boldsymbol{u}$. Namely, $\epsilon_{\boldsymbol{u}}(\boldsymbol{v}) = 1$ if $\boldsymbol{v} = \boldsymbol{u}$ and 0 otherwise, for $\boldsymbol{v} \in \{0,1\}^r$. Note the distinction between indicators in $\mathbb{R}^{\{0,1\}^r}$, and those in $\{0,1\}^r$, which we denote by $\boldsymbol{e}_i$, for $i = 1, \dots, r$. We write, for example,

$$\boldsymbol{\alpha} = (n-k)\epsilon_{\mathbf{0}} + k\epsilon_{\boldsymbol{e}_i} \in \mathcal{I}_{r,n}$$

Here, $\mathbf{0}, \boldsymbol{e}_i \in \{0,1\}^r$, and k is an integer between 0 and n.

Every $\boldsymbol{\alpha} \in \mathcal{I}_{r,n}$ is also considered as a real function on $\{0,1\}^r$. Namely, $\alpha_{\boldsymbol{u}}$ is synonymous with α_i, where $\boldsymbol{u} \in \{0,1\}^r$ is the binary representation of $i \in \mathbb{N}$. As a Boolean function, we apply Fourier transform to $\boldsymbol{\alpha}$: $\hat{\boldsymbol{\alpha}}_{\boldsymbol{u}} = \langle \chi_{\boldsymbol{u}}, \boldsymbol{\alpha} \rangle$, for every $\boldsymbol{u} \in \{0,1\}^r$.

Let us now rewrite $\mathsf{DelsarteLin}(r,n,d)$ in terms of $(\varphi_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}}$.

$(C1)$ The orbit of $\mathbf{0} \in \{0,1\}^{r\times n}$ contains only the element $\mathbf{0}$, so $f(\mathbf{0}) = 1$ implies $\varphi_{n\epsilon_{\mathbf{0}}} = 1$.

$(C2)$ By linearity of (partial) Fourier transform,

$$\mathcal{F}_S(f)(X) = \sum_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}} \varphi_{\boldsymbol{\alpha}} \cdot \mathcal{F}_S(L_{\boldsymbol{\alpha}})(X)$$

for every $S \subset [r]$ and $X \in \{0,1\}^{r\times n}$.

In Section V below, we show that $\mathcal{F}_S(L_{\boldsymbol{\alpha}})(X)$ depends only on the column enumerator of X.

When $S = [r]$, namely for $\hat{L}_{\boldsymbol{\alpha}}(X)$, it turns out that it is a multivariate polynomial in Γ_X. In Section VI we denote $\hat{L}_{\boldsymbol{\alpha}}(X) := K_{\boldsymbol{\alpha}}(\Gamma_X)$, and show that $\{K_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}}$ is a set of polynomials over $\mathbb{R}^{2^r}$ orthogonal w.r.t. the multinomial distribution. These polynomials are called *multivariate Krawtchouks*.

For $S \neq [r]$, we denote $\mathcal{F}_S(L_{\boldsymbol{\alpha}})(X) := K^S_{\boldsymbol{\alpha}}(\Gamma_X)$. We call the set $\{K^S_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}}$ *partial Krawtchouks*. These are orthogonal functions w.r.t. an appropriate measure, though not polynomials. In Section VI we describe these functions as products of multivariate Krawtchouks.

Constraint $(C2)$ implies

$$\sum_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}} \varphi_{\boldsymbol{\alpha}} K^S_{\boldsymbol{\alpha}} \geq 0$$

for every $S \subset [r]$.

$(C3)$ The following proposition expresses the weights of the row space of X in terms of its column enumerator.

Proposition 2: For every $X = (x_{i,j}) \in \{0,1\}^{r\times n}$ and $\boldsymbol{u} \in \{0,1\}^r$,

$$|\boldsymbol{u}^\intercal X| = \frac{1}{2}\left(n - 2^r \widehat{\Gamma}_X(\boldsymbol{u})\right)$$

where $\widehat{\Gamma}_X(\boldsymbol{u})$ is the Fourier transform of Γ_X at $\boldsymbol{u}$.

Proof: By definition, $\boldsymbol{u}^\intercal X \in \{0,1\}^n$ and $|\boldsymbol{u}^\intercal X| = \sum_{j=1}^n (\boldsymbol{u}^\intercal X)_j$, where $(\boldsymbol{u}^\intercal X)_j$ is the j-th bit and the sum is over the integers. Concretely, for $j = 1, \dots, n$:

$$\begin{aligned}(\boldsymbol{u}^\intercal X)_j &= \sum_{i:u_i=1} x_{i,j} \bmod 2 \\ &= \frac{1}{2}\left(1 - (-1)^{\sum_{i:u_i=1} x_{i,j}}\right) \\ &= \frac{1}{2}\left(1 - (-1)^{\langle \boldsymbol{u}, \boldsymbol{\xi}_j\rangle}\right)\end{aligned}$$

where $\boldsymbol{\xi}_j = (x_{i,j})_{i=1}^r$ is the j-th column of X. Thus

$$|\boldsymbol{u}^\intercal X| = \frac{1}{2}\sum_{j=1}^n \left(1 - (-1)^{\langle \boldsymbol{u}, \boldsymbol{\xi}_j\rangle}\right)$$

But $\boldsymbol{\xi}_j$ appears $\Gamma_X(\boldsymbol{\xi}_j)$ times in X, so grouping the summands by column, we have

$$|\boldsymbol{u}^\intercal X| = \frac{1}{2}\sum_{\boldsymbol{v} \in \{0,1\}^r} \Gamma_X(\boldsymbol{v})(1 - (-1)^{\langle \boldsymbol{u}, \boldsymbol{v}\rangle}) = \frac{1}{2}\left(n - \chi^\intercal_{\boldsymbol{u}}\,\Gamma_X\right)$$

□

Thus, we require that $\varphi_{\boldsymbol{\alpha}} = 0$ whenever $1 \leq \frac{1}{2}\left(n - 2^r \widehat{\boldsymbol{\alpha}}_{\boldsymbol{u}}\right) \leq d-1$ for some $\boldsymbol{u} \in \{0,1\}^r$.

$(C4)$ When $X \in \{0,1\}^{r\times n}$ gets multiplied on the left by $T \in \mathrm{GL}(r,2)$, its column enumerator, Γ_X gets modified. Here we need to define the action of T on $\mathcal{I}_{r,n}$, in a way that is consistent with this modification. Indeed, define

$$(T \cdot \boldsymbol{\alpha})_{\boldsymbol{u}} = \alpha_{T^{-1}\boldsymbol{u}}$$

This ensures $T \cdot \Gamma_X = \Gamma_{TX}$.

(Obj) The vector $(\boldsymbol{x}, 0, \dots, 0) \in \{0,1\}^{rn}$ corresponds to the matrix $\boldsymbol{e}_1\boldsymbol{x}^\intercal \in \{0,1\}^{r\times n}$. Say $|\boldsymbol{x}| = k$. Then, its column enumerator is $\Gamma_{\boldsymbol{e}_1\boldsymbol{x}^\intercal} = (n-k)\epsilon_{\mathbf{0}} + k\epsilon_{\boldsymbol{e}_1}$. The orbit of $\boldsymbol{e}_1\boldsymbol{x}^\intercal$ has cardinality $\binom{n}{k}$. Hence, the objective function becomes

$$\text{maximize} \quad \sum_{k=0}^n \binom{n}{k} \varphi_{(n-k)\epsilon_{\mathbf{0}} + k\epsilon_{\boldsymbol{e}_1}}$$

(Obj') Summing over the entire set $\mathcal{I}_{r,n}$ with multiplicites,

$$\text{maximize} \quad \sum_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}} \binom{n}{\boldsymbol{\alpha}} \varphi_{\boldsymbol{\alpha}}$$

where $\binom{n}{\boldsymbol{\alpha}}$ is the multinomial coefficient.

Let us now define the symmetrized version of $\mathsf{DelsarteLin}$.

Definition 5: $\mathsf{DelsarteLin}_{/\mathfrak{S}_n}(r,n,d)$:

$$\underset{\varphi:\mathcal{I}_{r,n}\to\mathbb{R}}{\text{maximize}} \quad \sum_{k=0}^n \binom{n}{k} \varphi_{(n-k)\epsilon_{\mathbf{0}} + k\epsilon_{\boldsymbol{e}_1}} \qquad (Obj_{/\mathfrak{S}_n})$$

subject to:

$$\varphi_{n\epsilon_{\mathbf{0}}} = 1 \qquad (C1_{/\mathfrak{S}_n})$$

$$\sum_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}} \varphi_{\boldsymbol{\alpha}} K^S_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) \geq 0 \qquad \forall S \subset [r],\ \boldsymbol{\beta} \in \mathcal{I}_{r,n} \qquad (C2_{/\mathfrak{S}_n})$$

$$\varphi_{\boldsymbol{\alpha}} = 0 \qquad \text{if } 1 \leq \frac{1}{2}(n - 2^r \hat{\boldsymbol{\alpha}}_{\boldsymbol{e}_1}) \leq d-1 \qquad (C3_{/\mathfrak{S}_n})$$

$$\varphi_{\boldsymbol{\alpha}} = \varphi_{T\cdot\boldsymbol{\alpha}} \qquad \forall T \in \mathrm{GL}(r,2) \qquad (C4_{/\mathfrak{S}_n})$$

We also mention two important variations, $(C2')$ and (Obj'):

$$\text{maximize} \quad \sum_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}} \varphi_{\boldsymbol{\alpha}} \qquad (Obj'_{/\mathfrak{S}_n})$$

$$\varphi \geq 0; \quad \sum_{\boldsymbol{\alpha} \in \mathcal{I}_{r,n}} \varphi_{\boldsymbol{\alpha}} K_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) \geq 0 \qquad \forall \boldsymbol{\beta} \in \mathcal{I}_{r,n} \qquad (C2'_{/\mathfrak{S}_n})$$

By the comments from the beginning of this section, we have the following equivalence.

Proposition 3: For every positive integers r, n, d, such that $d \leq n/2$,

$$\operatorname{val}\mathsf{DelsarteLin}_{/\mathfrak{S}_n}(r,n,d) = \operatorname{val}\mathsf{DelsarteLin}(r,n,d)$$

Note that $\mathsf{DelsarteLin}_{/\mathfrak{S}_n}(1,n,d)$ is identical to Delsarte's LP. Observe that $\mathcal{I}_{1,n}$ is isomorphic to the set $\{0,1,\ldots,n\}$. Rewrite the LP with a new set of variables, $a_k := \binom{n}{k}\varphi_{(n-k)\epsilon_0+k\epsilon_1}$, for $k = 0,1,\ldots,n$. Using the Krawtchouk symmetry identity, $\binom{n}{j}K_i(j) = \binom{n}{i}K_j(i)$, transform the Krawtchouk constraint $(C2_{/\mathfrak{S}_n})$ as follows:

$$\sum_{j=0}^{n}\binom{n}{j}^{-1}a_jK_j(i) = \binom{n}{i}^{-1}\sum_{j=0}^{n}a_jK_i(j)$$

The result is Delsarte's LP:

Definition 6: $\mathsf{Delsarte}(n,d)$:

$$\underset{a_0,\ldots,a_n\in\mathbb{R}}{\text{maximize}} \quad \sum_{i=0}^{n} a_i \qquad (obj_{/\mathfrak{S}_n})$$

subject to:

$$a_0 = 1 \qquad (d1_{/\mathfrak{S}_n})$$

$$a_i \geq 0; \quad \sum_{j=0}^{n} a_jK_i(j) \geq 0, \quad 0 \leq i \leq n \qquad (d2_{/\mathfrak{S}_n})$$

$$a_i = 0 \quad \text{if } 1 \leq i \leq d-1 \qquad (d3_{/\mathfrak{S}_n})$$

V. On Partial Fourier Transform

In this section we explore interactions between the groups $\mathfrak{S}_n$ and $\mathrm{GL}(r,2)$ and the partial Fourier transform. The former, $\mathfrak{S}_n$ acts on $\{0,1\}^{r\times n}$ by permuting columns. The latter, $\mathrm{GL}(r,2)$ acts on $\{0,1\}^{r\times n}$ by matrix multiplication from the left. The group of order-r permutation matrices is a subgroup of $\mathrm{GL}(r,2)$ which acts on $\{0,1\}^{r\times n}$ by permuting rows.

The main result of this section is Lemma 1. It shows that the constraints implied by the dual code are equivalent to those that follow the symmetry w.r.t. the general linear group (see Section III-A). The easy propositions of this section are useful also for computational purposes.

We recall our dual view of $\{0,1\}^{rn}$, once as a concatenation of r vectors $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r \in \{0,1\}^n$, and once as a matrix $X \in \{0,1\}^{r\times n}$ whose rows are the above vectors. If the group G acts on $\{0,1\}^{r\times n}$, and $g \in G$, we denote $(f\circ g)(X) = f(g\cdot X)$ for any $X \in \{0,1\}^{r\times n}$ and $f : \{0,1\}^{rn} \to \mathbb{R}$.

The proofs for some of the following propositions appear in the appendix.

Proposition 4: Let $\sigma \in \mathfrak{S}_n$, $X \in \{0,1\}^{r\times n}$, $S \subset [r]$, and $f : \{0,1\}^{r\times n} \to \mathbb{R}$. Then,

$$\mathcal{F}_S(f\circ\sigma) = \mathcal{F}_S(f)\circ\sigma$$

Proposition 5: Let $\pi \in \mathfrak{S}_r$ act on the set $\{0,1\}^{r\times n}$ by row permutation. Let $X \in \{0,1\}^{r\times n}$, $S \subset [r]$, and $f : \{0,1\}^{r\times n} \to \mathbb{R}$. Then,

$$\mathcal{F}_S(f\circ\pi) = \mathcal{F}_{\pi^{-1}(S)}(f)\circ\pi$$

Proposition 6: Let $T \in \mathrm{GL}(r,2)$ be the elementary matrix of row addition, mapping $\boldsymbol{e}_i \mapsto \boldsymbol{e}_i + \boldsymbol{e}_j$, for some $i,j \in [r]$, $i \neq j$, and $\boldsymbol{e}_k \mapsto \boldsymbol{e}_k$ for $k \neq i$, where $\boldsymbol{e}_k \in \{0,1\}^r$ is the k-th standard basis vector. Let $X \in \{0,1\}^{r\times n}$, $S \subset [r]$, and $f : \{0,1\}^{r\times n} \to \mathbb{R}$. Then,

- if $i,j \in S$:
$$\mathcal{F}_S(f\circ T) = \mathcal{F}_S(f)\circ T^{\mathsf{T}}$$
- if $i,j \notin S$:
$$\mathcal{F}_S(f\circ T) = \mathcal{F}_S(f)\circ T$$
- if $i \in S, j \notin S$:
$$\mathcal{F}_S(f\circ T)(X) = \chi_{\boldsymbol{x}_i}(\boldsymbol{x}_j)\mathcal{F}_S(f)(X)$$

Note that do not consider the case $i \notin S, j \in S$, since the expression does not simplify in that case.

Lemma 1: Let $f : \{0,1\}^{r\times n} \to \mathbb{R}$. The following are equivalent:

1) For every $T \in \mathrm{GL}(r,2)$,
$$f = f\circ T.$$
2) For every $S \subset [r]$,
$$\mathcal{F}_S(f)(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r) = 0,$$
if $\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle = 1 \bmod 2$ for some $i \in S$ and $j \in [r]\setminus S$.
3) For every $S \subset [r]$,
$$\mathcal{F}_S(f) = \mathcal{F}_S(f)\circ(T_1T_2)$$
if $T_1, T_2 \in \mathrm{GL}(r,2)$, and $T_1\boldsymbol{e}_i = \boldsymbol{e}_i$ for every $i \in S$, $T_2\boldsymbol{e}_i = \boldsymbol{e}_i$ for every $i \in [r]\setminus S$.

Proof:

- (1) ⇒ (2): Let $S \subsetneq [r]$, $S \neq \emptyset$. Let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r \in \{0,1\}^n$ and $i \in S$, $j \in [r]\setminus S$ s.t. $\langle \boldsymbol{x}_i,\boldsymbol{x}_j\rangle = 1 \bmod 2$. Let $T \in \mathrm{GL}(r,2)$ be the mapping $\boldsymbol{x}_i \mapsto \boldsymbol{x}_i + \boldsymbol{x}_j$ and $\boldsymbol{x}_k \mapsto \boldsymbol{x}_k$ for $k \neq i$. By assumption and by proposition 6,
$$\mathcal{F}_S(f) = \mathcal{F}_S(f\circ T) = \chi_{\boldsymbol{x}_i}(\boldsymbol{x}_j)\mathcal{F}_S(f) = -\mathcal{F}_S(f)$$
Hence $\mathcal{F}_S(f) = 0$.
- (2) ⇒ (3): It is enough to show that $\mathcal{F}_S(f)$ is invariant under the mapping $\boldsymbol{x}_i \mapsto \boldsymbol{x}_i + \boldsymbol{x}_j$, where $i \neq j$ and i,j are either both in S or both in $[r]\setminus S$. The rest follows by composition of such operators.
If $|S| \leq 1$ the claim holds trivially. Otherwise, let $i \neq j$, $i,j \in S$, and let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r \in \{0,1\}^n$. Observe that $\mathcal{F}_S(f) = \mathcal{F}_{\{i\}}\mathcal{F}_{S\setminus\{i\}}(f)$. Hence
$$\begin{aligned}&\mathcal{F}_S(f)(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r)\\ &= 2^{-n}\sum_{\boldsymbol{y}\in\{0,1\}^r}\chi_{\boldsymbol{x}_i}(\boldsymbol{y})\mathcal{F}_{S\setminus\{i\}}(f)(\ldots,\boldsymbol{x}_{i-1},\boldsymbol{y},\boldsymbol{x}_{i+1},\ldots)\end{aligned}$$
by assumption, $\mathcal{F}_{S\setminus\{i\}}(f)(\ldots,\boldsymbol{x}_{i-1},\boldsymbol{y},\boldsymbol{x}_{i+1},\ldots) = 0$ if $\langle \boldsymbol{y},\boldsymbol{x}_j\rangle = 1$, hence $\chi_{\boldsymbol{x}_j}(\boldsymbol{y}) = 1$ for every non-zero element of the sum. So
$$\begin{aligned}&\mathcal{F}_S(f)(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r)\\ &\quad= 2^{-n}\sum_{\boldsymbol{y}\in\{0,1\}^r}\chi_{\boldsymbol{x}_i}(\boldsymbol{y})\chi_{\boldsymbol{x}_j}(\boldsymbol{y})\times\\ &\qquad\times\mathcal{F}_{S\setminus\{i\}}(f)(\ldots,\boldsymbol{x}_{i-1},\boldsymbol{y},\boldsymbol{x}_{i+1},\ldots)\\ &\quad= \mathcal{F}_S(f)(\ldots,\boldsymbol{x}_{i-1},\boldsymbol{x}_i+\boldsymbol{x}_j,\boldsymbol{x}_{i+1},\ldots)\end{aligned}$$
To see that the same applies if $i,j \in [r]\setminus S$, observe that $\mathcal{F}_S(f) = 2^n\mathcal{F}_{\{i\}}\mathcal{F}_{S\cup\{i\}}(f)$ and repeat the same steps.
- (3) ⇒ (1): Take $S = \emptyset$.

□

VI. On Multivariate Krawtchouk Polynomials

The multivariate Krawtchouk polynomials are orthogonal polynomials on the multinomial distribution. Univariate Krawtchouk polynomials are the Fourier transform of the level sets in the Boolean cube, and as we show in this section, these polynomials are the Fourier transform of the level-set indicators $\{L_\alpha\}$.

We borrow the terminology of [8]. The multinomial distribution $m(\boldsymbol{\alpha}, \boldsymbol{p})$ arises in the stochastic process where n identical balls are independently dropped into d bins, where the probability of falling into the i-th bin is p_i. The probability that α_i balls end up in bin i is

$$m(\boldsymbol{\alpha}, \boldsymbol{p}) = \binom{n}{\alpha_0, \ldots, \alpha_{d-1}} \prod_{i=0}^{d-1} p_i^{\alpha_i} = \binom{n}{\boldsymbol{\alpha}} \boldsymbol{p}^{\boldsymbol{\alpha}}$$

Here $\boldsymbol{p} = (p_0, \ldots, p_{d-1})$, all α_i are nonnegative integers and their sum is n. We use the shorthand $m(\boldsymbol{\alpha})$ when $\boldsymbol{p}$ is uniform.

Orthogonal systems of univariate polynomials are constructed by applying a Gram-Schmidt process to the polynomials $1, x, x^2, \ldots$ e.g., [18]. The result depends only on a measure that we fix on the underlying set. However, as mentioned e.g., in [19], in the process of defining an orthogonal multivariate family of polynomials, there is another choice to make, and this choice affects the resulting family. Namely, we need to choose the order in which we go over the monomials of a given degree. In [8], this freedom is mitigated by choosing a basis of orthogonal functions on $\{0, 1, \ldots, d-1\}$. Every such basis leads to a unique set of orthogonal polynomials, as follows. Let $\boldsymbol{h} = \{h^l\}_{l=0}^{d-1}$ be a complete set of orthogonal functions w.r.t. $\boldsymbol{p}$, with $h^0 \equiv 1$. Namely,

$$\sum_{i=0}^{d-1} h^l(i) h^k(i) p_i = \delta_{lk} a_k, \quad 0 \le k, l \le d$$

The Krawtchouks are defined in terms of a generating function. Fix $\boldsymbol{\alpha}$ and $\boldsymbol{h}$ as above. For every choice of nonnegative reals $\xi_0, \ldots, \xi_{d-1}$ whose sum is n, we define

$$Q(\boldsymbol{\xi}) = Q_{\boldsymbol{\alpha}}(\boldsymbol{\xi}, \boldsymbol{h}) = \underset{\prod_{i=1}^{d-1} w_i^{\alpha_i}}{\operatorname{coef}} \prod_{j=0}^{d-1} \left\{ 1 + \sum_{l=1}^{d-1} w_l h^l(j) \right\}^{\xi_j}$$

where $\boldsymbol{w} = (w_0 \ldots, w_{d-1})$ are formal variables. The total degree of $Q_{\boldsymbol{\alpha}}$ is $\sum_{i=1}^{d-1} \alpha_i$. Note that α_0, w_0 do not appear in the definition. An equivalent definition that does include α_0, w_0 is:

$$Q_{\boldsymbol{\alpha}}(\boldsymbol{\xi}, \boldsymbol{h}) = \underset{\boldsymbol{w}^{\boldsymbol{\alpha}}}{\operatorname{coef}} \prod_{j=0}^{d-1} \left\{ \sum_{l=0}^{d-1} w_l h^l(j) \right\}^{\xi_j} \tag{4}$$

It is easy to see the equivalence by expanding each factor with the multinomial expansion. We will be using all of this with $d = 2^r$, uniform $\boldsymbol{p} \equiv 2^{-r}$ and with the orthonormal functions that are the characters of $\{0,1\}^r$: $\boldsymbol{h} = \{\chi_{\boldsymbol{u}}\}_{\boldsymbol{u} \in \{0,1\}^r}$.

Recall the definition of level-set indicators, $\{L_{\boldsymbol{\alpha}}\}$:

$$L_{\boldsymbol{\alpha}}(X) = \mathbf{1}_{[\Gamma_X = \boldsymbol{\alpha}]}, \quad X \in \{0,1\}^{r \times n}$$

We also defined $\mathcal{I}_{r,n}$ the set of all ordered partitions of $[n]$ into 2^r parts.

$$\mathcal{I}_{r,n} = \{\Gamma_X : X \in \{0,1\}^{r \times n}\}$$

Also, $\mathcal{I}_{r,n}$ is the support of the multinomial distribution with n balls, 2^r bins, where $\boldsymbol{p}$ is uniform.

Let $X \in \{0,1\}^{r \times n}$ be a random matrix that results by sampling n columns independently and uniformly from $\{0,1\}^r$. The probability that $L_{\boldsymbol{\alpha}}(X) = 1$ is $m(\boldsymbol{\alpha}) = 2^{-rn}\binom{n}{\alpha}$. It is clear that $L_{\boldsymbol{\alpha}}(X)$ depends only on Γ_X, and by proposition 4 this is true for $\hat{L}_{\boldsymbol{\alpha}}(X)$ as well. Define

$$K_{\boldsymbol{\alpha}}(\Gamma_X) = 2^{rn} \hat{L}_{\boldsymbol{\alpha}}(X), \quad X \in \{0,1\}^{r \times n}$$

It is easy to see that $\{K_{\boldsymbol{\alpha}}\}$ are orthogonal with respect to $m(\boldsymbol{\alpha})$, using Parseval's identity:

$$\begin{aligned}
&\sum_{\boldsymbol{\gamma}} m(\boldsymbol{\gamma}) K_{\boldsymbol{\alpha}}(\boldsymbol{\gamma}) K_{\boldsymbol{\beta}}(\boldsymbol{\gamma}) \\
&\quad = \sum_{\boldsymbol{\gamma}} 2^{-rn} \sum_{X : \Gamma_X = \boldsymbol{\gamma}} 2^{2rn} \hat{L}_{\boldsymbol{\alpha}}(X) \hat{L}_{\boldsymbol{\beta}}(X) \\
&\quad = \sum_{X \in \{0,1\}^{r \times n}} L_{\boldsymbol{\alpha}}(X) L_{\boldsymbol{\beta}}(X) \\
&\quad = \binom{n}{\boldsymbol{\alpha}} \delta_{\boldsymbol{\alpha}, \boldsymbol{\beta}}
\end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathcal{I}_{r,n}$. The extra 2^{-rn} is there because inner product is normalized in the non-Fourier space.

The following proposition shows that $\{K_{\boldsymbol{\alpha}}\}$ are Krawtchouk polynomials.

Proposition 7: $K_{\boldsymbol{\alpha}}$ is the Krawtchouk polynomial $Q_{\boldsymbol{\alpha}}(\cdot, \boldsymbol{h})$ with $d = 2^r$, $\boldsymbol{h}$ are the Fourier characters $\{\chi_{\boldsymbol{u}}\}_{\boldsymbol{u} \in \{0,1\}^r}$, and $\boldsymbol{p} \equiv 2^{-r}$ is the uniform distribution.

Proof: Let $X \in \{0,1\}^{r \times n}$ and $\Gamma_X = \boldsymbol{\beta} = (\beta_{\boldsymbol{u}})_{\boldsymbol{u} \in \{0,1\}^r}$. We show that $K_{\boldsymbol{\alpha}}(\boldsymbol{\beta})$ coincides with the definition of $Q_{\boldsymbol{\alpha}}(\boldsymbol{\beta}, \{\chi_{\boldsymbol{u}}\}_{\boldsymbol{u} \in \{0,1\}^r})$ in (4).

By definition,

$$K_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) = \hat{L}_{\boldsymbol{\alpha}}(X) = 2^{-rn} \sum_{Y \in \{0,1\}^{r \times n}} (-1)^{\langle X, Y \rangle} L_{\boldsymbol{\alpha}}(Y)$$

The inner product between X and Y can be expressed column-wise,

$$\langle X, Y \rangle = \sum_{j=1}^{n} \langle (X^\intercal)_j, (Y^\intercal)_j \rangle = \sum_{\boldsymbol{u}, \boldsymbol{v} \in \{0,1\}^r} \langle \boldsymbol{u}, \boldsymbol{v} \rangle \, \Gamma_{[X,Y]}(\boldsymbol{u}, \boldsymbol{v})$$

where $[X, Y] \in \{0,1\}^{2r \times n}$ is the stacking of X on top of Y, and $\Gamma_{[X,Y]}(\boldsymbol{u}, \boldsymbol{v})$ is the number of times the column $[\boldsymbol{u}, \boldsymbol{v}] \in \{0,1\}^{2r}$ appears in the matrix $[X, Y]$. We consider $\Gamma_{[X,Y]}(\boldsymbol{u}, \boldsymbol{v})$ as a matrix indexed by $\{0,1\}^r \times \{0,1\}^r$. Its $\boldsymbol{u}$-th row sums to $\beta_{\boldsymbol{u}}$ and its $\boldsymbol{v}$-th column sums to $\Gamma_Y(\boldsymbol{v})$. Hence,

$$\hat{L}_{\boldsymbol{\alpha}}(X) = 2^{-rn} \sum_{A} \sum_{\substack{Y \in \{0,1\}^{r \times n}: \\ \Gamma_{[X,Y]} = A}} \prod_{\boldsymbol{u}, \boldsymbol{v} \in \{0,1\}^r} (-1)^{\langle \boldsymbol{u}, \boldsymbol{v} \rangle A_{\boldsymbol{u}, \boldsymbol{v}}} L_{\boldsymbol{\alpha}}(Y)$$

where the outer sum is over all matrices $A \in \mathbb{N}^{2^r \times 2^r}$ with $A \cdot \mathbf{1} = \boldsymbol{\beta}$.

If $\Gamma_Y = \boldsymbol{\alpha}$ then $\mathbf{1}^\intercal \cdot A = \boldsymbol{\alpha}$. In particular, A uniquely determines $L_{\boldsymbol{\alpha}}(Y)$, so the product does not depend on Y. The sum over Y evaluates to the size of the set $\{Y \in \{0,1\}^{r\times n} : \Gamma_{[X,Y]} = A\}$, which we now compute. For every $\boldsymbol{u} \in \{0,1\}^r$, $[X,Y]$ contains $\Gamma_X(\boldsymbol{u}) = \beta_{\boldsymbol{u}}$ columns whose prefix is $\boldsymbol{u}$. For every $\boldsymbol{v} \in \{0,1\}^r$, the column $[\boldsymbol{u},\boldsymbol{v}]$ appears $A_{\boldsymbol{u},\boldsymbol{v}}$ times in $[X,Y]$. Since the position of the $\boldsymbol{u}$'s is fixed, it is left to position the $\boldsymbol{v}$'s with respect to each $\boldsymbol{u}$. Hence

$$|\{Y \in \{0,1\}^{r\times n} : \Gamma_{[X,Y]} = A\}| = \prod_{\boldsymbol{u}\in\{0,1\}^r} \binom{\beta_{\boldsymbol{u}}}{A_{\boldsymbol{u}}}$$

where $A_{\boldsymbol{u}}$ is the row of A that is indexed by $\boldsymbol{u}$.

Let $\boldsymbol{w} = (w_{\boldsymbol{v}})_{\boldsymbol{v}\in\{0,1\}^r}$ be formal variables. If A is such that $\mathbf{1}^\intercal A = \boldsymbol{\alpha}$ then

$$\prod_{\boldsymbol{u},\boldsymbol{v}} w_{\boldsymbol{v}}^{A_{\boldsymbol{u},\boldsymbol{v}}} = \prod_{\boldsymbol{v}} w_{\boldsymbol{v}}^{\sum_{\boldsymbol{u}} A_{\boldsymbol{u},\boldsymbol{v}}} = \prod_{\boldsymbol{v}} w_{\boldsymbol{v}}^{\alpha_{\boldsymbol{v}}}$$

hence $\hat{L}_{\boldsymbol{\alpha}}(X)$ equals

$$\operatorname*{coef}_{\boldsymbol{w}^{\boldsymbol{\alpha}}} 2^{-rn} \sum_A \prod_{\boldsymbol{u}\in\{0,1\}^r} \binom{\beta_{\boldsymbol{u}}}{A_{\boldsymbol{u}}} \prod_{\boldsymbol{v}\in\{0,1\}^r} \left((-1)^{\langle \boldsymbol{u},\boldsymbol{v}\rangle} w_{\boldsymbol{v}}\right)^{A_{\boldsymbol{u},\boldsymbol{v}}}$$

The sum over A can be expanded to nested sums over its rows,

$$\sum_A = \sum_{A_0}\sum_{A_1}\cdots\sum_{A_{2^r-1}}$$

where $A_{\boldsymbol{u}} \in \mathbb{N}^{2^r}$, $A_{\boldsymbol{u}} \cdot \mathbf{1} = \beta_{\boldsymbol{u}}$. Every factor in the product depends on a single row of A, so the product and the sum can be transposed.

Then, by the multinomial theorem:

$$\hat{L}_{\boldsymbol{\alpha}}(X) = \operatorname*{coef}_{\boldsymbol{w}^{\boldsymbol{\alpha}}} 2^{-rn} \prod_{\boldsymbol{u}\in\{0,1\}^r} \left\{ \sum_{\boldsymbol{v}\in\{0,1\}^r} (-1)^{\langle \boldsymbol{u},\boldsymbol{v}\rangle} w_v \right\}^{\beta_{\boldsymbol{u}}}$$
$$= 2^{-rn} Q_{\boldsymbol{\alpha}}(\boldsymbol{\beta}, \{\chi_{\boldsymbol{v}}\}_{\boldsymbol{v}\in\{0,1\}^r})$$

□

We turn to deal with the partial Fourier transform of the level-set indicators. Let $S \subsetneq \{1,\ldots,r\}$ be non-empty. Denote by X', X'' be the sub-matrix of $X \in \{0,1\}^{r\times n}$ with row set S and $[r]\setminus S$, respectively. Similarly, $\boldsymbol{u}', \boldsymbol{u}''$ are obtained from $\boldsymbol{u} \in \{0,1\}^r$ by restricting to $S, [r]\setminus S$, respectively. For $\boldsymbol{\alpha} \in \mathcal{I}_{r,n}$ define the rearrangement of $\boldsymbol{\alpha}$ into a matrix $\boldsymbol{\alpha}^S = (\alpha^S_{\boldsymbol{u}'',\boldsymbol{u}'})_{\boldsymbol{u}''\in\{0,1\}^{r-|S|},\boldsymbol{u}'\in\{0,1\}^{|S|}}$ by

$$\alpha^S_{\boldsymbol{u}'',\boldsymbol{u}'} = \alpha_{\boldsymbol{u}} \quad \boldsymbol{u} \in \{0,1\}^r$$

For $X \in \{0,1\}^{r\times n}$ we define $K^S_{\boldsymbol{\alpha}}$ as follows:

$$K^S_{\boldsymbol{\alpha}}(\Gamma_X) := 2^{|S|n}\mathcal{F}_S(L_{\boldsymbol{\alpha}})(X)$$

The next proposition says that $K^S_{\boldsymbol{\alpha}}$ is a sparse product of lower-order Krawtchouks.

Proposition 8: For every $\boldsymbol{\beta} \in \mathcal{I}_{r,n}$ there holds

$$K^S_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) = \begin{cases} \prod K_{\boldsymbol{\alpha}^S_{\boldsymbol{v}}}(\boldsymbol{\beta}^S_{\boldsymbol{v}}) & \text{if } \boldsymbol{\alpha}^S_{\boldsymbol{v}}\cdot\mathbf{1} = \boldsymbol{\beta}^S_{\boldsymbol{v}}\cdot\mathbf{1}\ \forall \boldsymbol{v} \\ 0 & \text{otherwise.} \end{cases}$$

where the product is over all $\boldsymbol{v} \in \{0,1\}^{r-|S|}$. Here $\boldsymbol{\alpha}^S_v$ is the row of $\boldsymbol{\alpha}^S$ at index $\boldsymbol{v}$, and $\boldsymbol{\alpha}^S_{\boldsymbol{v}}\cdot\mathbf{1} = \sum_{\boldsymbol{u}'\in\{0,1\}^{|S|}} \alpha^S_{\boldsymbol{v},\boldsymbol{u}'}$.

Proof: Let $X \in \{0,1\}^{r\times n}$ such that $\Gamma_X = \boldsymbol{\beta}$. By definition,

$$K^S_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) = \mathcal{F}_S(L_{\boldsymbol{\alpha}})(X)$$
$$= 2^{-|S|n} \sum_{Y\in\{0,1\}^{r\times n}} (-1)^{\langle X',Y'\rangle}\delta_{X''}(Y'')L_{\boldsymbol{\alpha}}(Y)$$

We express $\delta_{X''}(Y'')L_{\boldsymbol{\alpha}}(Y)$ in terms of Y', $\boldsymbol{\alpha}^S$ and $\boldsymbol{\beta}^S$.

Let $Y \in \{0,1\}^{r\times n}$ such that $\Gamma_Y = \boldsymbol{\alpha}$ and $Y'' = X''$. The number of times $\boldsymbol{u}'' \in \{0,1\}^{r-|S|}$ occurs in Y'' is $\boldsymbol{\alpha}^S_{\boldsymbol{u}''}\cdot\mathbf{1} = \sum_{\boldsymbol{u}'} \alpha^S_{\boldsymbol{u}'',\boldsymbol{u}'}$. But this is equal $\boldsymbol{\beta}^S_{\boldsymbol{u}''}\cdot\mathbf{1}$ because $Y'' = X''$.

Let $Y'|_{Y''=\boldsymbol{u}''}$ be the subset of columns from Y' for which the corresponding column in Y'' is $\boldsymbol{u}''$. Then $\Gamma_{Y'|_{Y''=\boldsymbol{u}''}} = \boldsymbol{\alpha}^S_{\boldsymbol{u}''}$.

$$L_{\boldsymbol{\alpha}}(Y)\delta_{X''}(Y'')$$
$$= \prod_{\boldsymbol{u}''\in\{0,1\}^{r-|S|}} \mathbf{1}_{[\boldsymbol{\alpha}^S_{\boldsymbol{u}''}\cdot\mathbf{1}=\boldsymbol{\beta}^S_{\boldsymbol{u}''}\cdot\mathbf{1}]} L_{\boldsymbol{\alpha}^S_{\boldsymbol{u}''}}(Y'|_{Y''=\boldsymbol{u}''})$$

The sum over Y can be broken into nested sums over $\{Y'|_{Y''=\boldsymbol{u}''}\}_{\boldsymbol{u}''\in\{0,1\}^{r-|S|}}$,

$$\sum_{Y\in\{0,1\}^{r\times n}} = \sum_{Y'_0}\sum_{Y'_1}\cdots\sum_{Y'_{2^{r-|S|}-1}}$$

where $Y'_{\boldsymbol{u}''} \in \{0,1\}^{|S|\times n}$ are mutually independent. Each factor in the product depends on a single $Y'_{\boldsymbol{u}''}$ so the order of summations and products can be reversed,

$$\mathcal{F}_S(L_{\boldsymbol{\alpha}})(X) = 2^{-|S|n} \prod_{\boldsymbol{u}''\in\{0,1\}^{r-|S|}} \Bigg[\mathbf{1}_{[\boldsymbol{\alpha}^S_{\boldsymbol{u}''}\cdot\mathbf{1}=\boldsymbol{\beta}^S_{\boldsymbol{u}''}\cdot\mathbf{1}]} \times$$
$$\times \sum_{Y'_{\boldsymbol{u}''}\in\{0,1\}^{|S|\times n}} (-1)^{\langle X'|_{X''=\boldsymbol{u}''}, Y'_{\boldsymbol{u}''}\rangle} L_{\boldsymbol{\alpha}^S_{\boldsymbol{u}''}}(Y'|_{Y''=\boldsymbol{u}''}) \Bigg]$$

Observe that the inner sum is simply

$$\hat{L}_{\boldsymbol{\alpha}^S_{\boldsymbol{u}''}}(X'|_{X''=\boldsymbol{u}''}) = K^S_{\boldsymbol{\alpha}^S_{\boldsymbol{u}''}}(\boldsymbol{\beta}^S_{\boldsymbol{u}''}).$$

□

In the last part of this section we consider $K^S_{\boldsymbol{\alpha}}$ under the action of the general linear group $\mathrm{GL}(r,2)$.

Fix $S \subset [r]$ and $\boldsymbol{\alpha} \in \mathcal{I}_{r,n}$. It is easy to verify that

$$K^S_{T\cdot\boldsymbol{\alpha}}(\Gamma_X) = \mathcal{F}_S(L_{\boldsymbol{\alpha}} \circ T)(X)$$

for every $X \in \{0,1\}^{r\times n}$ and $T \in \mathrm{GL}(r,2)$. Propositions 9 and 10 below are immediate consequences of 5 and 6.

Proposition 9: Let $T \in \mathrm{GL}(r,2)$ be a permutation matrix, $T\boldsymbol{e}_i = \boldsymbol{e}_{\pi(i)}$ for some $\pi \in \mathfrak{S}_r$. Then

$$K^S_{T\cdot\boldsymbol{\alpha}} = K^{\pi^{-1}(S)}_{\boldsymbol{\alpha}} \circ T$$

Proposition 10: Let $T \in \mathrm{GL}(r,2)$ be the mapping $\boldsymbol{e}_i \mapsto \boldsymbol{e}_i + \boldsymbol{e}_j$ for some $i,j \in [r]$, and $\boldsymbol{e}_k \mapsto \boldsymbol{e}_k$ for every $k \neq i$.

- if $i,j \in S$:
$$K^S_{T\cdot\boldsymbol{\alpha}} = K^S_{\boldsymbol{\alpha}} \circ T^\intercal$$
- if $i,j \notin S$:
$$K^S_{T\cdot\boldsymbol{\alpha}} = K^S_{\boldsymbol{\alpha}} \circ T$$
- if $i \in S, j \notin S$:
$$K^S_{T\cdot\boldsymbol{\alpha}}(\boldsymbol{\beta}) = (-1)^{\sum_{\boldsymbol{u}\in\{0,1\}^r}\beta_{\boldsymbol{u}}u_iu_j} K^S_{\boldsymbol{\alpha}}(\boldsymbol{\beta})$$

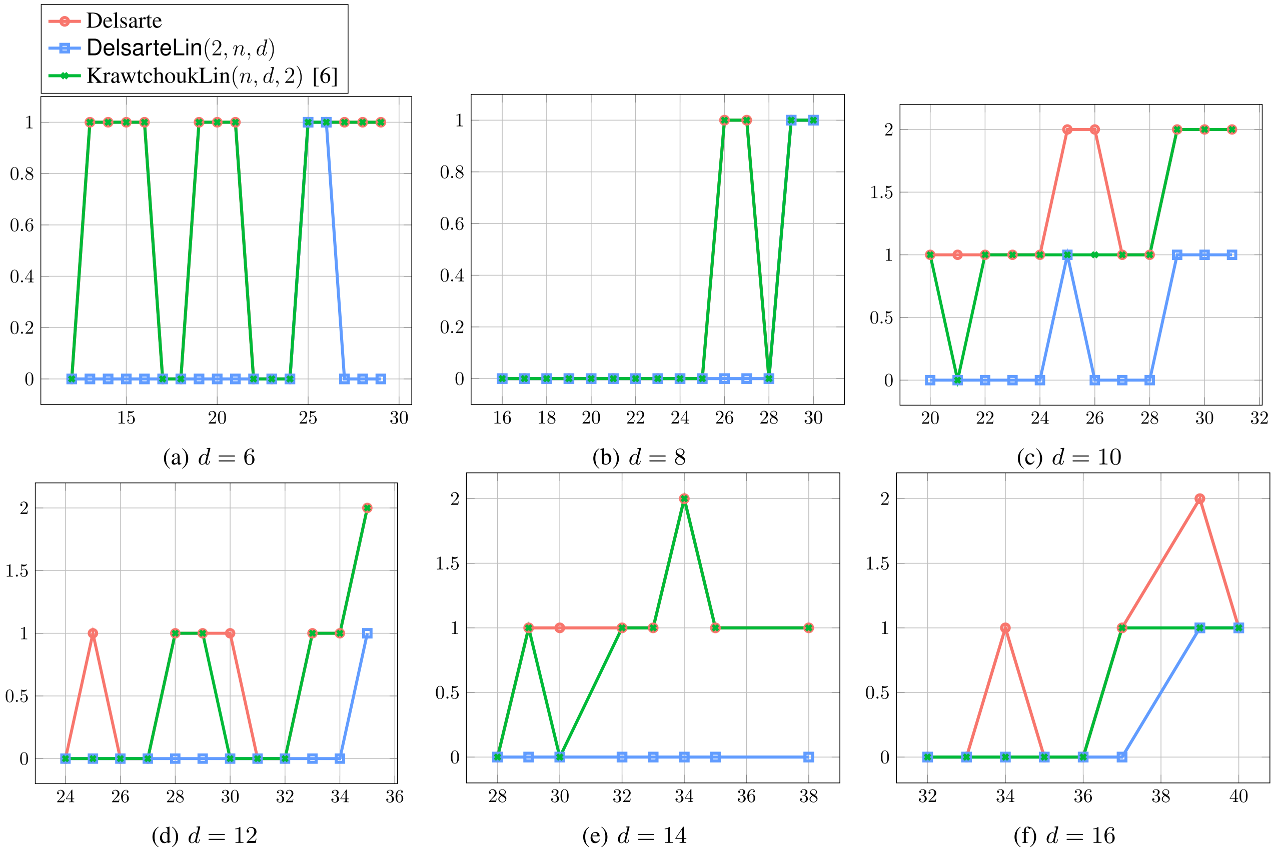


Fig. 4. (Lower is better). Similar to Figure 1, stated in terms of the code's *dimension*. Each point represents $\lfloor \log_2(LP(n,d)) \rfloor - \text{bestKnown}(n,d)$, where LP is one of: Delsarte(n,d), DelsarteLin$(2,n,d)$, and KrawtchoukLin$(n,d,2)$, and bestKnown(n,d) is the best known upper bound on $\log_2(A^{\text{Lin}}(n,d))$.

Appendix A
Proofs for Section V

Proof of Proposition 4: For $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$,

$$\begin{aligned}\chi_{\boldsymbol{x}}(\sigma \cdot \boldsymbol{y}) &= (-1)^{\sum_{i=1}^n x_i y_{\sigma(i)}} \\ &= (-1)^{\sum_{i=1}^n x_{\sigma^{-1}(i)} y_i} \\ &= \chi_{\sigma^{-1} \cdot \boldsymbol{x}}(\boldsymbol{y})\end{aligned}$$

and

$$\delta_{\boldsymbol{x}}(\sigma \cdot \boldsymbol{y}) = \prod_{i=1}^n \delta_{x_i}(y_{\sigma(i)}) = \prod_{i=1}^n \delta_{x_{\sigma^{-1}(i)}}(y_i) = \delta_{\sigma^{-1} \cdot \boldsymbol{x}}(\boldsymbol{y})$$

Hence, for $X, Y \in \{0,1\}^{r \times n}$,

$$\chi_X^S(\sigma \cdot Y) = \chi_{\sigma^{-1} \cdot X}^S(Y)$$

Finally, letting $Y' = \sigma \cdot Y$,

$$\begin{aligned}\mathcal{F}_S(f \circ \sigma)(X) &= \sum_{Y \in \{0,1\}^{r \times n}} \chi_X(Y) f(\sigma \cdot Y) \\ &= \sum_{Y' \in \{0,1\}^{r \times n}} \chi_X(\sigma^{-1} \cdot Y') f(Y') \\ &= \sum_{Y' \in \{0,1\}^{r \times n}} \chi_{\sigma \cdot X}(Y') f(Y') \\ &= \mathcal{F}_S(f)(\sigma \cdot X)\end{aligned}$$

□

Proof of Proposition 5:

$$\begin{aligned}&\mathcal{F}_S(f \circ \pi)(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) \\ &= 2^{-rn} \sum_{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r} \chi^S_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r) f(\boldsymbol{y}_{\pi(1)}, \ldots, \boldsymbol{y}_{\pi(r)}) \\ &= 2^{-rn} \sum_{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r} \chi^S_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r}(\boldsymbol{y}_{\pi^{-1}(1)}, \ldots, \boldsymbol{y}_{\pi^{-1}(r)}) \times \\ &\times f(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r) \\ &= 2^{-rn} \sum_{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r} \prod_{i \in S} \chi_{\boldsymbol{x}_i}(\boldsymbol{y}_{\pi^{-1}(i)}) \times \\ &\times \prod_{i \in [r] \setminus S} \delta_{\boldsymbol{x}_i}(\boldsymbol{y}_{\pi^{-1}(i)}) f(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r) \\ &= 2^{-rn} \sum_{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r} \prod_{j \in \pi^{-1}(S)} \chi_{\boldsymbol{x}_{\pi(j)}}(\boldsymbol{y}_j) \times \\ &\times \prod_{j \in [r] \setminus \pi^{-1}(S)} \delta_{\boldsymbol{x}_{\pi(j)}}(\boldsymbol{y}_j) f(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r) \\ &= \mathcal{F}_{\pi^{-1}(S)}(f)(\boldsymbol{x}_{\pi(1)}, \ldots, \boldsymbol{x}_{\pi(r)})\end{aligned}$$

□

Proof of Proposition 6: Consider f as a function of r vectors, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \in \{0,1\}^n$. Then T maps $\boldsymbol{x}_i \mapsto \boldsymbol{x}_i + \boldsymbol{x}_j$, and $\boldsymbol{x}_k \mapsto \boldsymbol{x}_k$ for $k \neq i$.

For $k = 1, \ldots, r$, and $\boldsymbol{x} \in \{0,1\}^n$, define a set of functions $\{\psi^{(k)}_{\boldsymbol{x}}\}_{k \in [r]}$, where $\psi^{(k)}_{\boldsymbol{x}} = \chi_{\boldsymbol{x}}$ if $k \in S$ and

$\psi^{(k)}_{\boldsymbol{x}} = \delta_{\boldsymbol{x}}$ otherwise.

$$\begin{aligned}&\mathcal{F}_S(f\circ T)(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r)\\&= 2^{-rn}\sum_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_r}\prod_{k=1}^{r}\psi^{(k)}_{\boldsymbol{y}_k}(\boldsymbol{y}_k)f(\boldsymbol{y}_1,\ldots,\underbrace{\boldsymbol{y}_i+\boldsymbol{y}_j}_{\text{index } i},\ldots,\boldsymbol{y}_r)\end{aligned}$$

replacing the sum over $\boldsymbol{y}_i$ by a sum over $\boldsymbol{y}_i+\boldsymbol{y}_j$, we get

$$\begin{aligned}=&2^{-rn}\sum_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_r}\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\psi^{(j)}_{\boldsymbol{y}_j}(\boldsymbol{y}_j)\times\\&\times\prod_{k\in[r]\setminus\{i,j\}}\psi^{(k)}_{\boldsymbol{y}_k}(\boldsymbol{y}_k)f(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_i,\ldots,\boldsymbol{y}_r)\end{aligned}\tag{5}$$

We now examine the expression $\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\psi^{(j)}_{\boldsymbol{y}_j}(\boldsymbol{y}_j)$ in the different cases of the proposition.

- $i,j\in S$:

$$\begin{aligned}\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\psi^{(j)}_{\boldsymbol{x}_j}(\boldsymbol{y}_j) &= \chi_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\chi_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\\&=\chi_{\boldsymbol{x}_i}(\boldsymbol{y}_i)\chi_{\boldsymbol{x}_i+\boldsymbol{x}_j}(\boldsymbol{y}_j)\\&=\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i)\psi^{(j)}_{\boldsymbol{x}_i+\boldsymbol{x}_j}(\boldsymbol{y}_j)\end{aligned}$$

  which is the same as applying the mapping $\boldsymbol{x}_j\mapsto\boldsymbol{x}_i+\boldsymbol{x}_j$, or equivalently $T^{\intercal}$, to $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r$.
- $i,j\notin S$:

$$\begin{aligned}\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\psi^{(j)}_{\boldsymbol{x}_j}(\boldsymbol{y}_j) &= \delta_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\delta_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\\&=\delta_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{x}_j)\delta_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\\&=\delta_{\boldsymbol{x}_i+\boldsymbol{x}_j}(\boldsymbol{y}_i)\delta_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\\&=\psi^{(i)}_{\boldsymbol{x}_i+\boldsymbol{x}_j}(\boldsymbol{y}_i)\psi^{(j)}_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\end{aligned}$$

  which equivalent to applying T to $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r$.
- $i\in S, j\notin S$:

$$\begin{aligned}\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\psi^{(j)}_{\boldsymbol{x}_j}(\boldsymbol{y}_j) &= \chi_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\delta_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\\&=\chi_{\boldsymbol{x}_i}(\boldsymbol{x}_j)\chi_{\boldsymbol{x}_i}(\boldsymbol{y}_i)\delta_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\\&=\chi_{\boldsymbol{x}_i}(\boldsymbol{x}_j)\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i)\psi^{(j)}_{\boldsymbol{x}_j}(\boldsymbol{y}_j)\end{aligned}$$

  observe that $\chi_{\boldsymbol{x}_i}(\boldsymbol{x}_j)$ is constant with respect to the sum in (5).
- $i\notin S, j\in S$:

$$\psi^{(i)}_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\psi^{(j)}_{\boldsymbol{x}_j}(\boldsymbol{y}_j) = \delta_{\boldsymbol{x}_i}(\boldsymbol{y}_i+\boldsymbol{y}_j)\chi_{\boldsymbol{x}_j}(\boldsymbol{y}_j)$$

  Here there is no obvious way to rewrite the functions so as to separate $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$.

□

Appendix B
Numerical Results

We have experimented with several variants of $\mathsf{DelsarteLin}_{/\mathfrak{S}_n}(2,n,d)$ with n ranging between 10 and 40 and $d\le n/2$ is even. In those variants of the LP - we replace $(C2)$ with $(C2')$, and (Obj) with (Obj'). The table below only shows the results for $n\ge 20$.

The number of variables is $|\mathcal{I}_{r,n}| = \binom{n+2^r-1}{2^r-1}$ and, if we consider r as a constant, there are $O_n(|\mathcal{I}_{r,n}|)$ constraints. In practice, symmetrization w.r.t. $\mathrm{GL}(r,2)$ reduces the problem size (variables $\times$ constraints) by a factor of $2^{\Omega(r)}$, which is significant. Since we have not yet developed the necessary theoretical tools for such symmetrization, it was carried out algorithmically. We intend to develop such theory so as to solve instances of $\mathsf{DelsarteLin}_{/\mathfrak{S}_n}(r,n,d)$ with larger values of r.

The number of variables is further reduced using the well-known fact, that if d is even then an even code attains $A(n,d)$. Namely, we set $\varphi_{\boldsymbol{\alpha}}=0$ if $(n-\chi_{\boldsymbol{u}}^{\intercal}\boldsymbol{\alpha})/2$ is odd, for some $\boldsymbol{u}\in\{0,1\}^r$.

The Krawtchouk polynomials were computed with a recurrence formula, e.g. (16) in [8]. The partial Krawtchouks $K^S_{\boldsymbol{\alpha}}$ were computed using proposition 8. We used two exact solvers: SoPlex [20], [21], [22] and QSoptEx [23], with up to 128GB of RAM and at most 3 days of runtime. Some instances were solved by one solver and not the other. Missing entries were solved by neither.

The best in each row is marked with boldface. Entries are marked with a "$*$" if $\lfloor\log_2(\text{entry})\rfloor$ equals the best known upper bound, as reported in [7].

Variant		$(C2')$ (Obj')	(Obj)	$(C2)$ (Obj')	(Obj)	Delsarte	Schrijver
n	dist.						
20	4	26214*	26214*	**21845***	**21845***	26214*	-
	6	2328	2285	**1588***	1593*	2373	-
	8	268*	**256***	**256***	**256***	291*	274*
	10	40	40	**24***	**24***	40	-
21	4	**43691***	**43691***	**43691***	**43691***	47663*	-
	6	4197	4138	3010*	**2977***	4443	-
	8	**512***	**512***	**512***	**512***	572*	-
	10	51*	52*	**35***	36*	64	-
22	4	**87381***	**87381***	**87381***	**87381***	**87381***	-
	6	7380*	7327*	5770*	**5608***	7724*	-
	8	**1024***	**1024***	**1024***	**1024***	**1024***	-
	10	92	89	**57***	61*	95	87
23	4	**174763***	**174763***	**174763***	**174763***	**174763***	-
	6	13703*	13690*	10447*	**10102***	13776*	13766*
	8	**2048***	**2048***	**2048***	**2048***	**2048***	-
	10	152	152	**90***	93*	152	-
24	4	**349525***	**349525***	**349525***	**349525***	**349525***	-
	6	24054*	24018*	18786*	**18715***	24108*	-
	8	**4096***	**4096***	**4096***	**4096***	**4096***	-
	10	280	280	**155***	160*	280	-
	12	**48***	**48***	**48***	**48***	**48***	-
25	4	599186*	599186*	582826*	**579701***	645278*	-
	6	47481	47176	**34657**	34729	48149	47998
	8	5666*	5571*	4450*	**4200***	6475*	5477*
	10	511	497	**262**	284	551	503
	12	61*	62*	49*	**48***	75	-
26	4	1198373*	1198373*	1126532*	**1121065***	1198373*	-
	6	86693	86847	**66014**	66638	93623	-
	8	10099	10031	7516*	**7508***	10435	-
	10	930	922	**490***	533	1040	886
	12	105*	99*	**77***	79*	113*	-

27	4	2396745*	2396745*	**2097152***	**2097152***	2396745*	-
	6	162180	162027	**125238***	126201*	163840	-
	8	17803	17727	**12698***	12774*	18190	17768
	10	1766	1766	**854***	954*	1766	-
	12	171*	171*	132*	**129***	171*	-
28	4	4793490*	4793490*	**4194304***	-	4793490*	-
	6	291202	291173	234649*	**234626***	291271	-
	8	32126*	32119*	**21989***	22120*	32206*	32151*
	10	3194	3189	**1482***	1646*	3200	-
	12	288	288	**213***	**213***	288	-
	14	56*	56*	**32***	**32***	56*	-
29	4	**8388608***	**8388608***	**8388608***	**8388608***	8947849*	-
	6	574493	573756	**430773***	432499*	581827	-
	8	57247	57217	**38276**	39205	58097	-
	10	6155	6074	**2743**	3181	6363	-
	12	550	541	**320***	323*	573	-
	14	70	72	**47***	49*	88	-
30	8	108267	107044	**67353**	71095	114816	-
	10	11517	11340	**4827**	5929	12525	-
	12	1022*	1026	**535***	582*	1132	-
	14	114*	112*	**74***	80*	130	-
31	10	20838	20738	**8651**	-	22296	-
	12	1781*	1763*	**1024***	**1024***	1840*	-
	14	196	196	**110***	115*	196	-
32	12	3082*	3082*	**2048***	**2048***	3082*	-
	14	314	313	**187***	191*	315	-
	16	**64***	**64***	**64***	**64***	**64***	-
33	12	5821	5821	**2903***	3037*	5821	-
	14	617	612	**310***	324*	629	-
	16	80*	82*	65*	**64***	99*	-
34	12	10878	10671	-	**5726***	11641	-
	14	1195	1203	**510***	568	1258	-
	16	124*	120*	**99***	103*	144	-
35	12	20011	19810	-	**10603**	21727	-
	14	1774	1759	**890***	989*	2026	-
	16	215*	215*	172*	**169***	215*	-
36	14	-	-	**1664***	1786*	3177	-
	16	352*	352*	**256***	**256***	352*	-
	18	72	72	**40**	**40**	72	-
37	16	704	704	**366***	-	704	-
38	14	-	10211	-	**5348***	10211	-
39	16	1905	1918	**1138**	**1138**	2271	-
40	16	3493	3488	-	**2276**	3510	-

REFERENCES

[1] E. N. Gilbert, "A comparison of signalling alphabets," *Bell Syst. Tech. J.*, vol. 31, no. 3, pp. 504–522, May 1952.

[2] R. R. Varshamov, "Estimate of the number of signals in error correcting codes," *Doklady Akademii Nauk SSSR*, vol. 117, no. 5, pp. 739–741, 1957.

[3] R. J. McEliece, E. R. Rodemich, H. Rumsey, and L. Welch, "New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 2, pp. 157–166, Mar. 1977.

[4] P. Delsarte, "An algebraic approach to the association schemes of coding theory," *Philips Res. Rep. Suppl.*, vol. 10, pp. 6–97, Mar. 1973.

[5] A. Barg and D. B. Jaffe, "Numerical results on the asymptotic rate of binary codes," in *Codes and Association Schemes*, vol. 56. Providence, RI, USA: AMS, 2001, pp. 25–32.

[6] L. N. Coregliano, F. G. Jeronimo, and C. Jones, "A complete linear programming hierarchy for linear codes," in *Proc. 13th Innov. Theor. Comput. Sci. Conf. (ITCS)*, M. Braverman, Ed., 2022, pp. 51:1–51:22, Art. no. 51.

[7] M. Grassl. (2007). *Bounds on the Minimum Distance of Linear Codes and Quantum Codes*. Accessed: Apr. 18, 2022. [Online]. Available: http://wwwcodetables.de accessed on 2022-04-18.

[8] P. Diaconis and R. Griffiths, "An introduction to multivariate Krawtchouk polynomials and their applications," *J. Stat. Planning Inference*, vol. 154, pp. 39–53, Nov. 2014.

[9] F. A. Grünbaum, "A system of multivariable Krawtchouk polynomials and a probabilistic application," *Symmetry, Integrability Geometry: Methods Appl.*, vol. 7, p. 118, Dec. 2011.

[10] J. Friedman and J.-P. Tillich, "Generalized Alon-Boppana theorems and error-correcting codes," *SIAM J. Discrete Math.*, vol. 19, no. 3, pp. 700–718, Jan. 2005.

[11] M. Navon and A. Samorodnitsky, "On Delsarte's linear programming bounds for binary codes," in *Proc. 46th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2005, pp. 327–336.

[12] M. Navon and A. Samorodnitsky, "Linear programming bounds for codes via a covering argument," *Discrete Comput. Geometry*, vol. 41, no. 2, pp. 199–207, Mar. 2009.

[13] A. Samorodnitsky, "One more proof of the first linear programming bound for binary codes and two conjectures," 2021, *arXiv:2104.14587*.

[14] A. M. Barg and D. Y. Nogin, "Spectral approach to linear programming bounds on codes," *Problems Inf. Transmiss.*, vol. 42, no. 2, pp. 77–89, Jun. 2006.

[15] A. Barg and D. Nogin, "A functional view of upper bounds on codes," in *Coding and Cryptology*. Singapore: World Scientific, 2008, pp. 15–24.

[16] A. Schrijver, "New code upper bounds from the terwilliger algebra and semidefinite programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2859–2866, Aug. 2005.

[17] R. O'Donnell, *Analysis of Boolean Functions*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[18] G. Szegö, *Orthogonal Polynomials*, vol. 23. Providence, RI, USA: American Mathematical Society, 1939.

[19] C. F. Dunkl and Y. Xu, *Orthogonal Polynomials of Several Variables*, no. 155. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[20] A. M. Gleixner, D. E. Steffy, and K. Wolter, "Improving the accuracy of linear programming solvers with iterative refinement," in *Proc. 37th Int. Symp. Symbolic Algebr. Comput. (ISSAC)*, 2012, pp. 187–194.

[21] A. M. Gleixner, D. E. Steffy, and K. Wolter, "Iterative refinement for linear programming," *INFORMS J. Comput.*, vol. 28, no. 3, pp. 449–464, Jul. 2016.

[22] G. Gamrath et al., "The SCIP optimization suite 7.0," Zuse Inst. Berlin, Berlin, Germany, ZIB-Rep., 2020.

[23] D. L. Applegate, W. Cook, S. Dash, and D. G. Espinoza, "Exact solutions to linear programming problems," *Oper. Res. Lett.*, vol. 35, no. 6, pp. 693–699, Nov. 2007.

Elyassaf Loyfer is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, The Hebrew University of Jerusalem.

Nati Linial is currently a Full Professor at the School of Computer Science and Engineering and the Institute of Mathematics, The Hebrew University of Jerusalem. He is one of the main builders of the bridge between mathematics and computer science. His honors include the Inaugural FOCS "Test of Time" Award in 2019, Rothschild Prize in Mathematics and Computer Science in 2016, Dijkstra Prize in 2013, a fellow of The American Mathematical Society—The Inaugural Class in 2012, and Conant Prize of the American Mathematical Society in 2008.