An Algebraic Framework for Extending Orthogonal Designs

Christos Koukouvinos[(1)], Dimitris E. Simos[(1)], Zafeirakis Zafeirakopoulos[(2)]
(1) Department of Mathematics, National Technical University of Athens
(2) Research Institute for Symbolic Computation (RISC) / DK-compmath
{ckoukouv,dsimos}@math.ntua.gr,zafeirakopoulos@risc.jku.at

Orthogonal designs (ODs) have numerous applications in Statistics, Telecommunications, Coding Theory and Cryptography, see [2]. An OD of order n and type $(s_1, s_2, \ldots, s_u)$ denoted $OD(n; s_1, s_2, \ldots, s_u)$ in the commuting variables $a_1, a_2, \ldots, a_u$, is a square matrix D of order n with entries from the set $\{0, \pm a_1, \pm a_2, \ldots, \pm a_u\}$ satisfying $DD^T = \sum_{i=1}^{u}(s_i a_i^2) I_n$, where I_n is the identity matrix of order n. A crucial lemma for manipulating ODs is the following.

Lemma 1 (Equating and Killing [2]). *If D is an orthogonal design $OD(n; s_1, s_2, \ldots, s_u)$ in the commuting variables $\{0, \pm a_1, \pm a_2, \ldots, \pm a_u\}$, then there exist orthogonal designs:*

(i) $OD(n; s_1, s_2, \ldots, s_i + s_j, \ldots, s_u)$ $(a_i = a_j)$ *(Equating)*

(ii) $OD(n; s_1, s_2, \ldots, s_{j-1}, s_{j+1}, \ldots, s_u)$ $(s_j = 0)$ *(Killing)*

on the $u-1$ commuting variables $\{0, \pm a_1, \pm a_2, \ldots, \pm a_{j-1}, \pm a_{j+1}, \ldots, \pm a_u\}$.

Sequences of zero autocorrelation give rise to orthogonal designs, for more details see [3]. Let $B = \{B_j : B_j = (b_{j1}, b_{j2}, ..., b_{jn}),\ j = 1, \ldots, \ell\}$, be a set of ℓ sequences of length n. The *autocorrelation function* $AF_B(s)$ is defined as

$$AF_B(s) = \sum_{j=1}^{\ell} \sum_{i=1}^{k} b_{ji} b_{j(i+s)}, \qquad s = 0, 1, ..., n-1. \tag{1}$$

We are interested in two types of AF, namely the periodic AF where $k = n$ and $i + s$ is computed modulo n, and the non-periodic AF where $k = n - s$. The set B has zero AF, if $AF_B(s) = 0$, for $s = 1, \ldots, n-1$.

Our main goal is to provide an algorithmic version for the reverse operations of Equating and Killing. We refer to the reverse of Equating as Splitting. The reverse of Killing can be interpreted in two ways; either as replacing zeros by an existing variable (Filling) or as replacing zeros by a new variable (Expanding). We note that Expanding can be performed by first Filling and then Splitting, but there is no guarantee that Filling-Splitting will give a result obtainable by Expanding the input.

In order to implement the reverse operations we work at the level of zero autocorrelation sequences. We derive algorithms for the manipulation of such sequences, so that from a given set of sequences we obtain another with some prescribed characteristics. To this end, we employ tools from symbolic computation.

1 The Reverse of Equating and Killing Lemma

In all three cases (Split, Fill, Expand), the goal is to obtain new sequences, which have zero AF and either one variable is split in two variables or some zeros and replaced. We give a solution for Split and Fill.

There is an intuitive way to express the conditions involved in the problems under consideration as an algebraic system. The algebraic systems we encounter, involve two essentially different sets of variables, $A = \{a_1, a_2, \ldots, a_u\}$ and $X = \{x_1, x_2, \ldots, x_r\}$. An extra variable t is used to describe the OD type. The polynomial ring we consider is $\mathbb{Q}[t, X, A]$ with a lexicographic order for which $t < x_1 < x_2 < \ldots < x_r < a_1 < a_2 < \ldots < a_u$.

1.1 The Algebraic System

Given B with zero AF we substitute the entries of the sequences to be replaced by elements from X. Namely, for Split we substitute with x_i the i-th occurrence of the variable to be split, denoted by x_m, while for Fill, we substitute with x_i the i-th zero in the sequences. The algebraic system modeling the problems should express the following restrictions.

1.1.1 Zero AF

We observe that the AF conditions (Eq. 1) are polynomials in $\mathbb{Q}[t, X, A]$. Since we require the AF to be zero, we add these polynomials in the algebraic system.

1.1.2 Binary Conditions

The variables x_i take values from a specific (finite) set V. For Split the set V is $\{-1, 1, -i, i\}$, while for Fill it is $\{-1, 0, 1\}$. Therefore, we add to the algebraic system the polynomials $\prod_{\alpha \in V} (x_i - \alpha), i = 1, 2, \ldots, r$

1.1.3 Type Conditions

The type of the desired OD implies the number of occurrences of each element of V as an X-coordinate in each root of the system. The variable t takes values in $\{1, 2, \ldots, k_1 = \lfloor \frac{s_m}{2} - 1 \rfloor\}$ for Split and in $\{1, 2, \ldots, k_2 = \ell n - \sum_{i=1}^{u} s_i\}$ for Fill. In Split we replace x_m with two variables, one appearing t and the other $s_m - t$ times while in Fill we replace t zeros. Thus, the polynomial conditions for Split and Fill are $\left\{ \prod_{i=1,2,\ldots,k_1} t - i, \left(\sum_{i=1,2,\ldots,r} x_i^2 \right) - s_m + 2t \right\}$ and $\left\{ \prod_{i=1,2,\ldots,k_2} t - i, \left(\sum_{i=1,2,\ldots,r} x_i^2 \right) - t \right\}$ respectively.

1.2 Computational Remarks

The A-variables are treated as parameters in the OD problem, thus we consider valid the solutions of the system for which all the A-variables are free. Although there are tools to deal with such problems (resultants, comprehensive Gröbner bases, cf. [1]), a simpler method suffices for the problem at hand. Due to the nature of the problem and the formulation described above we know that for the valid solutions the X-coordinates do not depend on A. Moreover, the number of solutions of Split and Fill is finite. Thus, we can substitute the A-variables by values in $\mathbb{Q}^u \setminus R$, where R is a finite subset of $\mathbb{Q}^u$. By substituting by random values, we get with probability 1 an algebraic system whose solutions are the same with the X-coordinates of the valid solutions of the initial system. Then we apply standard tools from computer algebra, in particular Gröbner bases [1]. The Gröbner basis of the (zero-dimensional) algebraic system provides a nice description, from which it is easy to enumerate all the solutions for Split and Fill, as well as to determine all possible types of ODs obtainable by the input sequences.

A naive implementation in the mathematical software Sage, indicates that the method is useful for the manipulation of sequences with zero AF. In particular, we could reproduce many existing results and arrive to a few new ones.

References

[1] D. Cox and J. Little and D. O'Shea, Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, Springer , 2005.

[2] A. V. Geramita and J. Seberry. Orthogonal designs. Quadratic forms and Hadamard matrices. *Lecture Notes in Pure and Applied Mathematics*, 45, New York, NY, Marcel Dekker, Inc., 1979

[3] C. Koukouvinos and J. Seberry. New weighing matrices and orthogonal designs constructed using two sequences with zero autocorrelation function - a review. *J. Statist. Plann. Inference*, 81:153–182, 1999.