Finding Orthogonal Designs with Gröbner bases

by

Fei Wang

A research paper
presented to the University of Waterloo
in partial fulfillment of the
requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Prof. Mark Giesbrecht, Prof. Ilias Kotsireas

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

Baumert and Hall [2] specify how to construct the Williamson array based on quaternions. A Williamson array H is a 4×4 symbolic matrix which satisfies $HH^T = (A^2 + B^2 + C^2 + D^2)I_4$. Here A, B, C, D are the entries of H as indeterminates. A Williamson array is an orthogonal design of order 4. Quaternions can be seen as the generalization of complex numbers and can be extended to larger dimension by the Cayley-Dickson process. The algebras associated with the process are called Cayley-Dickson algebras. By analogy to Baumert and Hall's method, we extend the construction of Williamson arrays to higher order using matrix representations of Cayley-Dickson algebras. In this paper, we give an example about how to construct orthogonal designs of order 128 using matrix representations of order 128 Cayley-Dickson algebra. Using Gröbner basis, we find the solutions of a polynomial system which give us orthogonal designs of order 128. We further construct Hadamard matrices based on the orthogonal designs we find.

Acknowledgements

I would like to thank my supervisors, Mark Giesbrecht and Ilias Kotsireas, for all their help and guidance this summer, and my second reader, George Labahn, for reading this paper and providing valuable feedback. I would also like to thank all members of the Centre for Computational Mathematics in Industry and Commerce for their support throughout this year.

Dedication

For my family, who offered me unconditional love and support throughout the course of this research paper.

Table of Contents

Chapter 1

Introduction

An *orthogonal design* is an orthogonal matrix whose entries are indeterminates. An orthogonal matrix has the property that the product of itself and its transpose is an identity. In particular, by assigning ± 1 to the indeterminate entries, one can get various Hadamard matrices from an orthogonal design. More formally, We make the following definition:

Definition 1 *Let $x_1, \ldots, x_t$ be commuting indeterminates. An* orthogonal design *X of order n and type $(s_1, \ldots, s_t)$, denoted by $OD(n; s_1, \ldots, s_t)$, where $s_1, \ldots, s_t$ are positive integers, is a matrix of order n with entries from $\{0, \pm x_1, \ldots, \pm x_t\}$, such that*

$$XX^{Tr} = (\sum_{i=1}^{t} s_i x_i^2) I_n,$$

where X^{Tr} denotes the transpose of X and I_n denotes the identity matrix of order n.[12]

An example of an orthogonal design is given as below:

$$H_4 = \begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix}$$

Here, $H_4 H_4^{Tr} = (A^2+B^2+C^2+D^2) I_4$ and H_4 is an orthogonal design of type $OD(4; 1, 1, 1, 1)$.

Orthogonal designs are used in combinatorics, statistics, coding theory, telecommunications, and other areas. For more details on orthogonal designs, see[10, 14] and on Hadamard matrices see [6].

In this article, we will explore the existence of the orthogonal designs of order 128. We will use the matrix representation of the so-called Cayley-Dickson algebra to construct an order 128 symbolic matrix which is almost an orthogonal design and then use Gröbner bases to find the solutions which make this "Near-orthogonal" symbolic matrix satisfy the definition of an orthogonal design.

Given the basis elements of a Cayley-Dickson algebra, we can associate a multiplication table about these basis elements. An example of a multiplication table for quaternions is given by the four elements $1, i, j, k$, having the properties

$$i^2 = -1, j^2 = -1, k^2 = -1, ij = k, ji = -k, ik = -j, ki = j, jk = i, kj = -i.$$

The multiplication table is as follows:

$\times$	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

.

Suppose we have a Cayley-Dickson algebra of order $t = 2^n$ with basis elements being $1, e_1, \ldots, e_{t-1}$. Define $v = [1, e_1, \ldots, e_{t-1}]^t$. Then by right multiplications $v \cdot 1, v \cdot e_1, \ldots, v \cdot e_{t-1}$, we can associate a matrix to each basis element (the right multiplications for quaternions are detailed in Chapter 3). In this article, we consider the Cayley-Dickson algebra of dimension 128 (the details of this algebra are described in Chapter 3), associating matrices to the basis elements by the right multiplication. We add all these matrices together linearly with indeterminate coefficients and get a large symbolic matrix. We then try to find the solutions which make the matrix an orthogonal design.

We show how to apply Gröbner bases to this problem through a substitution technique in order to find the solutions to a system of thousands of equations. The equations are just the entries of the order 128 symbolic matrix multiplied by its transpose. We show that the Gröbner bases has some nice properties which make it easier to find the solutions. The substitution technique can reduce the number of equations and help us find more solutions. Each solution gives exactly one orthogonal design, and we then use these orthogonal designs to later construct Hadamard matrices.

In this article, we find new orthogonal designs of order 128 with different types. They are $OD(128; 1, 1, 63, 63)$, $OD(128; 1, 1, 2, 62, 62)$, $OD(128; 1, 1, 2, 4, 60, 60)$, $OD(128; 1, 1, 2, 4, 8, 56, 56)$, and $OD(128; 1, 1, 2, 4, 8, 16, \ldots, 16)$. We use orthogonal design of $OD(128; 1, 1, 2, 4, 8, 16, \ldots, 16)$ to construct Hadamard matrices of order 384 and 640.

C. Koukouvinos and D. E. Simos [13] listed a table of orthogonal designs they found. They constructed a orthogonal design of type $OD(128; 16, 16, 16, 16, 16, 16, 16, 16)$. Here, we provide a new orthogonal design of type $OD(128; 16, 16, 16, 16, 16, 16, 16, 16)$ by equating variables in $OD(128; 1, 1, 2, 4, 8, 16, \ldots, 16)$.

Chapter 2

Orthogonal Designs

Recall the definition of orthogonal design:

Let $x_1, \ldots, x_t$ be commuting indeterminates. An orthogonal design X of order n and type $(s_1, \ldots, s_t)$ denoted by $OD(n; s_1, \ldots, s_t)$, where $s_1, \ldots, s_t$ are positive integers, is a matrix of order n with entries from $\{0, \pm x_1, \ldots, \pm x_t\}$, such that

$$XX^{Tr} = (\sum_{i=1}^{t} s_i x_i^2) I_n,$$

where X^{Tr} denotes the transpose of X and I_n denotes the identity matrix of order n [12].

An orthogonal design of type $OD(n; s_1, \ldots, s_t)$ is called a *full orthogonal design*, if $n = s_1 + s_2 + \cdots + s_t$.

Recall the example of orthogonal design given before:

$$H_4 = \begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix}.$$

Here, $H_4 H_4^{Tr} = (A^2 + B^2 + C^2 + D^2) I_4$, if we assign the entries A, B, C, D the values (x_1, x_2, x_3, x_4), then $H_4 H_4^{Tr} = (\sum_{i=1}^{4} x_i^2) I_4$. So H_4 is an orthogonal design of type $OD(4; 1, 1, 1, 1)$. Furthermore, it is also a full orthogonal design.

Definition 2 *The Radon function ρ is defined by $\rho(n) := 8q + 2^r$ when $n = 2^k \cdot p$, where $p \in Z^+$ is odd, $k = 4q + r$, and $0 \leq r < 4$. see [5]*

The *Radon function* gives an upper bound of the number of different variables. That is, given a full orthogonal design of type $OD(n; s_1, \ldots, s_t)$, we have $t \leq \rho(n)$.

For $n = 128$, we have $128 = 2^7$, so $k = 7$, and $q = 1, r = 3$. Therefore, we have $\rho(128) = 16$ meaning the upper bound for orthogonal design of order 128 is 16.

There are various methods to construct orthogonal designs. For more the details, see [5].

Some important conjecture about the orthogonal designs are as follows:

Conjecture 1 *There exists an $OD(8t; t, t, t, t, t, t, t, t)$ for every positive integer t. [5]*

Conjecture 2 *There exists an $OD(128, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8)$. [5]*

Orthogonal designs provide some of the most important constructions for Hadamard matrices. These conjectures can help us better understand the existence and constructions of Hadamard matrices.

Chapter 3

Duplication Technique, Cayley-Dickson Algebras

In mathematics, the Cayley-Dickson construction produces a sequence of algebras over the field of real numbers, each with twice the dimension of the previous one [12]. The algebras produced by this process are called Cayley-Dickson algebras. In this paper, we will use these algebras to construct orthogonal designs.

Starting from real numbers $\mathbb{R}$, we get complex numbers $\mathbb{C}$ by applying Cayley-Dickson construction to $\mathbb{R}$. Repeatedly applying this process to complex numbers $\mathbb{C}$, we get the well-known Cayley-Dickson algebra which is *Quaternions*. Quaternions were first introduced by Irish mathematician Sir William Hamilton in 1843. Quaternions can be seen as an extension of complex numbers, whose operation is defined by two rules below:

(1) $(a, b)^* = (a^*, -b)$,

(2) $(a, b)(c, d) = (ac - d^*b, da + bc^*)$.

Here, a, b, c, d are all complex numbers, so a quaternion can be seen as a pair of complex numbers with $*$ denoting the conjugation operation. The conjugation operation is defined recursively by (1). The conjugation of a real number is just itself. As we can see, the first rule defines the conjugation rule of quaternions which is just an extension of the case in complex numbers, and the second rule defines the multiplication rule of quaternions. Quaternions have dimension 4 over the real numbers.

Having quaternions already defined, we then form the ordered pairs of quaternions, which are called the *Octonions*, with multiplication and conjugation rules defined as before for quaternions. Octonions have dimension 8 over the real numbers.

Continuing with this duplication technique, we can construct algebras of dimension 2^n. This is the so-called Cayley-Dickson construction, and the associated algebras are called Cayley-Dickson algebras.

3.1 Examples

A basis for quaternions is given by the four elements 1,i,j,k, having the properties: $i^2 = -1, j^2 = -1, k^2 = -1, ij = k, ji = -k, ik = -j, ki = j, jk = i, kj = -i$. Then we associate a 4×4 matrix to each basis element.

Let $v = [1, i, j, k]^{Tr}$ and consider the right multiplications $v \cdot 1, v \cdot i, v \cdot j, v \cdot k$, $v \cdot 1 = [1, i, j, k]^{Tr}$. We obtain a matrix $q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Continuing $v \cdot i = [i, -1, -k, j]^{Tr}$ and we obtain a matrix $q_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. Similarly, $v \cdot j = [j, k, -1, -i]^{Tr}$ and this gives $q_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$. $v \cdot k = [k, -j, i, -1]^{Tr}$ and this gives $q_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$.

The matrices q_1, q_2, q_3, q_4 have the same properties as quaternions: $q_2 \cdot q_3 = q_4, q_2 \cdot q_4 = -q_3, q_3 \cdot q_2 = -q_4, q_3 \cdot q_4 = q_2, q_4 \cdot q_2 = q_3, q_4 \cdot q_3 = -q_2, q_2^2 = q_3^2 = q_4^2 = -q_1$. Let $H_4 = Aq_1 + Bq_2 + Cq_3 + Dq_4$. Then it is easy to check that $H_4^{Tr} = Aq_1 - Bq_2 - Cq_3 - Dq_4$ and so we have $H_4 H_4^{Tr} = (A^2 + B^2 + C^2 + D^2)q_1 = (A^2 + B^2 + C^2 + D^2)I_4$. Thus H_4 is an orthogonal design of type $OD(4; 1, 1, 1, 1)$. This method was first described by Baumert and Hall [2, 3].

3.2 Our approach

By applying the Cayley-Dickson construction repeatedly, one obtains a Cayley-Dickson algebra of dimension 128 with basis $e_0 = 1, e_1, e_2, \cdots, e_{127}$. We do not present the multiplication table here because it is too large.

To associate a 128×128 matrix to each basis element, we use the right multiplication operator, on the column vector

$$v = [1, e_1, \cdots e_{127}]^{Tr}.$$

Then the 128 right multiplications $v \cdot e_0, v \cdot e_1, \cdots v \cdot e_{127}$ give rise to one hundred and twenty-eight 128×128 matrices $q_0, q_1, \cdots, q_{127}$. Let $A_1, A_2, \cdots, A_{128}$ be commuting indeterminates. The sum $H = \sum_{i=0}^{127} A_{i+1} q_i$ is equal to a 128×128 matrix with the property that the diagonal elements of HH^{Tr} are all equal to $\sum_{i=1}^{128} A_i^2$, but whose other elements are not necessarily all zero. By requiring that all elements of HH^{Tr} (except the diagonal elements) equal to zeros, we obtain 5088 equations in 126 variables (all variables except A_1, A_{65}, these two variables are cancelled).

In the 4×4 case, we use Cayley-Dickson algebra of dimension 4 with bases $1, e_1, e_2, e_3$. We associate four matrices q_1, q_2, q_3, q_4 to the four basis elements. We sum them and get $H_4 = Aq_1 + Bq_2 + Cq_3 + Dq_4$. In this case $H_4 H_4^{Tr} = (A^2 + B^2 + C^2 + D^2) I_4$ which is already a diagonal matrix. Therefore, we do not need to require all non-diagonal elements of $H_4 H_4^{Tr}$ equal to zeros. This also happens in 8×8 case. However, when the dimension goes higher, we will lose this nice property.

Finding the solutions of the 5088 equations is a very difficult computational problem. However, for each of those polynomials, there seems to be another polynomial which is equal to it up to sign (we do not say "equation" here to avoid confusion, if we require the polynomials to be zeros, then we get the equations). In other words, the structure of these 5088 polynomials is actually like this: $\{f_1, -f_1, f_2, -f_2, \dots\}$ if we ignore the order. Therefore, we can reduce the 5088 equations to 2544 in the best case. This is a process of "cleaning up" or simplifying the equations. Theoretically, exempting all the redundant equations is expensive, but in practice, our "cleaning up" algorithm works very well, and it produces only 2546 equations after cleaning up (it may still have redundant polynomials, but only 2 extra in the worst case).

Once the "cleaning up" is done, we need to find the possible substitutions which make the 2546 equations become zero. We will use Gröbner bases to find good substitutions. These are introduced in Chapter 4 and Chapter 5.

Chapter 4

Gröbner Bases Computations

In this section, we introduce some basic concepts about Gröbner bases, which are a fundamental object of study in computational algebra and its applications. Let F be a field and $I = \langle p_1, \ldots, p_k \rangle$ an ideal in the multivariate polynomial ring $F[x_1, \ldots, x_n]$. A Gröbner basis for I is a basis that can be computed from the polynomials $p_1, \ldots, p_k$ with many properties that make it effective in computations within that ideal.

Definition 3 *Given a term order (lexicographic order, graded lexicographic order,..etc), the leading term of a polynomial f is the maximal term of the polynomial (up to the given term order), which is denoted by $Lp(f)$. The leading coefficient of a polynomial is the coefficient of the leading term, denoted by $Lc(f)$.*

A Gröbner basis G of an ideal I in a polynomial ring R over a field is characterized by the following property, relative to some term order:

The leading term of any polynomial in I is divisible by the leading term of some polynomial in the basis G. [4]

A Gröbner basis is called *reduced* if the leading coefficient of each element of the basis is 1 and no monomial in any element of the basis is in the ideal generated by the leading terms of the other elements of the basis (that is, no monomial in any element of the basis can be divided by the leading term of any other element of the basis) [4]. Gröbner bases can be computed by the classical Buchberger algorithm and also can be computed by the much more efficient algorithms F_4 and F_5 by Faugère [7, 8] (but which still require exponential time). These algorithms are all implemented in many computer algebra systems. For this paper, we will use the implementation of Gröbner bases in Maple and Singular (using

Sage). These are both well-known computer algebra systems at the current time with good implementations of up-to-date methods.

One of the characteristics of a Gröbner basis is that it can answer the question: does a system of equations have a solution. Also, a reduced Gröbner basis for a system of polynomials is an equivalent and hopefully simpler form of that system relative to an ordering, from which information about the solutions of the system can be found.

4.1 Over the Rationals

In the previous section, we described a system of 2546 equations with 126 variables (A_1 and A_{65} are missing). Our primary goal is to compute the Gröbner basis over the polynomial ring of rationals. In Maple, however, our methods seem to take very long time and makes it unfeasible to compute directly over the rationals. We can still solve a small case directly over rational field. As an example, see [12], where the Gröbner basis of 42 equations is a set of 21 simpler equations. The memory cost is very expensive over $\mathbb{Q}$ in Maple due to the growth of coefficients. As an alternative, we will replace the rational field by a finite field (modulo some prime) as the ground field of the polynomial ring. This will keep the size of intermediate coefficients constant and so save considerable time.

In this article, we will have to compute Gröbner bases of systems with different number of equations. Basically, we have found that if the number of equations is over 500, then the computation of Gröbner basis over rational field is an extremely time and space consuming process which makes it almost unfeasible and unpractical. In Chapter 5, we will compute Gröbner basis of systems with 2546, 591, 126 equations. For the 126 equations, we will still compute it over rational field, while for the 2546 and 591 equations, we will replace the rational field by a finite field.

4.2 Over $\mathbb{F}_p$ (mod p case)

Intermediate coefficient swell is a notorious difficulty of Buchberger's algorithm for computing Gröbner basis over the rational numbers. During the execution of the algorithm, many intermediate polynomials are computed. Unfortunately, the coefficients of these intermediate polynomials can grow up to enormous size, even if the final Gröbner basis has very small coefficients [1]. Regarding this, the modular approach can be very useful to reduce the cost of the algorithm (especially for saving the memory).

To implement the modular approach, we need to introduce the definition of "lucky" primes.

Definition 4 *A prime integer, p, is called* lucky *for I if and only if $Lp(G) = Lp(G_p)$. Here $I = \langle f_1, \ldots, f_r \rangle$ is an ideal in $\mathbb{Q}[X]$. G is the reduced Gröbner basis for I in $\mathbb{Q}[X]$ and G_p the reduced Gröbner basis for $I_p = \langle \bar{f}_1, \ldots, \bar{f}_r \rangle \in \mathbb{Z}_p[X]$. [1]*

Recall the definition of $Lp(*)$, for example, $Lp(5xy + 4x) = xy$. The Gröbner basis modular a lucky prime is the one that doesn't lose algebraic information about the correct Gröbner basis. More specifically, the leading terms of the G_p are same as the leading terms of G . Once we get G_p for different lucky primes p, we can use techniques like the *Chinese remainder theorem* [1] or p-adic to lift it back to $\tilde{G}$ and check if $\tilde{G} = G$. There are many papers about these techniques, see, e.g. [1, 9]. In this paper, we just choose a large prime p, and simply make $\tilde{G} = G_p$.

Once we get $\tilde{G}$, we need to check and make sure that the result is correct. The following theorem from [1] is helpful:

Theorem 1 *Let $\tilde{G} \in \mathbb{Q}[X]$ be a set of polynomials such that $Lp(\tilde{G}) = Lp(G_p)$, $\tilde{G}$ is a Gröbner basis for the ideal that it generates, which is denoted by $\langle \tilde{G} \rangle$, and $I \subseteq \langle \tilde{G} \rangle$. Then $I = \langle \tilde{G} \rangle$. [1]*

Note that Theorem 1 does not require p to be a lucky prime. This theorem tell us that once we get $\tilde{G}$ by making $\tilde{G} = G_p$, we need to do two things to check whether $\tilde{G}$ is the correct candidate: The first is to check whether $\tilde{G}$ is a reduced Gröbner basis for the ideal that it generates in $\mathbb{Q}[X]$. This is obviously true since no monomial term of a polynomial in $\tilde{G}$ can be divided by the leading term of any other polynomial in $\tilde{G}$. Next, we must show that $I \subseteq \langle \tilde{G} \rangle$. This can simply be done by showing that the generators of I, $f_1, \ldots, f_r$, reduced to zero using $\tilde{G}$.

We will use this modular approach to compute the reduced Gröbner basis of the large system (more than 500 equations) in the following content of this paper. First, we choose a prime $p = 32771$, which is the next prime of 2^{15} and compute G_p in the polynomial ring $\mathbb{Z}_p[X]$. After we get the candidate $\tilde{G} = G_p$, we then further check whether $\tilde{G}$ is actually equal to G by checking all the polynomials of the system reduced to zero modular $\tilde{G}$. If so, then we are done. If not, we choose another prime, compute the reduced Gröbner basis G_p and let $\tilde{G} = G_p$, then check it again. We can simply use Maple command "`NormalForm()`" to do this check.

Chapter 5

Substitution Techniques and Structure of Associated Gröbner Bases

Given a system of equations with many indeterminates, finding the substitutions which provide the solutions of the system is an interesting problem. We describe a method to solve this problem in this chapter:

1) If the number of equations is small, we simply compute the reduced Gröbner basis of the system and find the substitutions, since in this case the reduced Gröbner basis usually has a very simple and nice form.

2) If the number of equations is large, the reduced Gröbner basis will also usually be a large set. In this case, we may not find interesting substitutions directly. However, usually we can find a substitution which can reduce the number of equations significantly with the help of the reduced Gröbner basis. Once we find it, we use the substitution to obtain a system with much fewer equations. We compute a reduced Gröbner basis of this smaller system again and find the substitution to reduce the size. We repeat this procedure until the reduced Gröbner basis becomes "simple and nice".

We give two examples to explain this method:

Example 1 is from [12], where the reduced Gröbner basis of a set of 42 equations with 14 variables has 21 equations. Each equation is a binomial. The solutions of it can be easily found.

Example 2 is the system of 2546 equations with 126 variables we get in Chapter 3. It fits

case (2) in our method above. We first compute the reduced Gröbner basis here, expecting that it can provide us more clear guidance. Also, the solutions of Gröbner basis are same as its generators (i.e, the Gröbner basis of a set of polynomials is equivalent to the set itself). After we computed the Gröbner basis, as we expected, the reduced Gröbner basis of the 2546 equations has some nice properties. The binomials of the reduced Gröbner basis (there are still lots of polynomials with many terms) have a very simple and nice form:

$$A_j A_{i+64} - A_i A_{j+64}, \forall i, j = 2, \ldots, 64. \tag{5.1}$$

Since the number of equations is huge, we will employ some substitution techniques here to reduce the number of equations. The binomials (5.1) of the Gröbner basis above give us insight about how to substitute (equating variables). Once we do the substitution, the number of equations will become much less than that it was before. We then compute the Gröbner basis again and do further substitution. We repeat this procedure until all the elements of Gröbner basis are binomials (this will happen when the size of the system is small).

As we know from Chapter 3, A_1 and A_{65} are cancelled and they are not in the system, which means we don't need to relate them to other indeterminant (these two can be independent to other indeterminates in our substitutions). Also, if we fix i, and assume $A_{i+64} \neq A_i$, then due to the structure of (5.1), we must first require that

$$A_{i+64} = A_{j+64}, \forall j = 2, \ldots, 64;$$
$$A_i = A_j, \forall j = 2, \ldots, 64.$$

We substitute $A_j = A_i, j = 2, \ldots, 64, A_{j+64} = A_{i+64}, j = 2, \ldots, 64$ into the 2546 equations, and find all the equations become zeros eventually. Therefore this substitution gives us a solution of the system: $A_2 = A_3 = \cdots = A_{63} = A_{64}$ and $A_{66} = A_{67} = \cdots = A_{127} = A_{128}$. From the substitution, we get a orthogonal design which is $OD(128; 1, 1, 63, 63)$ (A_1, A_{65} are two independent variables, the other 126 variables are partitioned into two groups as above).

Since there is no limitation on i, then for any i between 2 and 64, $A_{i+64} \neq A_i$ will lead to the same substitution: $A_2 = A_3 = \cdots = A_{63} = A_{64}$ and $A_{66} = A_{67} = \cdots = A_{127} = A_{128}$. Therefore, we only need to consider two cases:

$$A_{i+64} \neq A_i, \forall i = 2, \ldots 64 \quad \text{and the resulting solution:}$$
$$A_2 = A_3 = \cdots = A_{63} = A_{64}, \ A_{66} = A_{67} = \cdots = A_{127} = A_{128}, \tag{5.2}$$

$$A_{i+64} = A_i, \forall i = 2, \ldots 64. \tag{5.3}$$

We substitute (5.3) into the system, and get another system of 591 equations of 62 variables (A_{33} is cancelled, A_1 has not been in the system from the beginning). Then we try to compute the reduced Gröbner basis of these 591 equations to find the solution as we did before. The interesting thing is that the reduced Göbner basis of these 591 equations has the same nice properties as the case of the 2546 equations. That is: the binomials of the reduced Gröbner basis also has a very simple form:

$$A_j A_{i+32} - A_i A_{j+32}, \forall i, j = 2, \ldots, 32.$$

As we have discussed before, we consider two cases:

$$\begin{aligned} &A_{i+32} \neq A_i, \forall i = 2, \ldots 32 \quad \text{and the resulting solution:} \\ &A_2 = A_3 = \cdots = A_{31} = A_{32}, \ A_{34} = A_{35} = \cdots = A_{63} = A_{64} \end{aligned} \tag{5.4}$$

$$A_{i+32} = A_i, \forall i = 2, \ldots 32. \tag{5.5}$$

Since we assume $A_{i+64} = A_i, \ \forall i = 2, \ldots, 64$, (5.5) will give us another orthogonal design of type $OD(128; 1, 1, 2, 62, 62)$. That is $A_1, A_{65}, A_{33} = A_{97}$, and the other 124 variable are separated into two parts.

We continue to move on to substitute (5.5) into the system of 591 equations and then get a system of 126 equations with 30 variables (A_{17} is cancelled). We then compute the reduced Gröbner basis of these 126 equations and as we expected, we get the nice binomials with the simple form:

$$A_j A_{i+16} - A_i A_{j+16}, \forall i, j = 2, \ldots, 16.$$

We then consider two cases similar to what we did before:

$$\begin{aligned} &A_{i+16} \neq A_i, \forall i = 2, \ldots 16 \quad \text{and the resulting solution:} \\ &A_2 = A_3 = \cdots = A_{15} = A_{16}, A_{18} = A_{19} = \cdots = A_{31} = A_{32} \end{aligned} \tag{5.6}$$

$$A_{i+16} = A_i, \forall i = 2, \ldots 16. \tag{5.7}$$

Because we already assume $A_{i+32} = A_i, \forall i = 2, \ldots, 32$, and also the first substitution $A_{i+64} = A_i, \forall i = 2, \ldots, 64$, case (5.7) will give us another orthogonal design of type $OD(128; 1, 1, 2, 4, 60, 60)$. That is $A_1, A_{65}, A_{33} = A_{97}, A_{17} = A_{49} = A_{81} = A_{113}$ and the other 120 variables are separated equally into two parts.

We continue to substitute (5.7) into the system of 126 equations and get another system of 21 equations with 14 different variables (A_1 and A_9 are missed). The Gröbner basis of the 21 equations has exactly the form as below:

$$A_jA_{i+8} - A_iA_{j+8}, \forall i, j = 2, \ldots, 8. \tag{5.8}$$

Equation (5.8) has two solutions:

$$A_1 = A_2 = \cdots = A_7 = A_8, \ A_{10} = A_{11} = \cdots = A_{15} = A_{16}; \tag{5.9}$$

$$A_{i+8} = A_i, i = 2, \ldots, 8. \tag{5.10}$$

Solution (5.10) leads to an orthogonal design of type $OD(128; 1, 2, 4, 8, 56, 56)$. Solution (5.10) leads to an orthogonal design of type $OD(128; 1, 2, 4, 8, 16, \ldots, 16)$.

Chapter 6

Results on Orthogonal Designs

In the previous chapter we found all the solutions of the system of 2546 equations using the Gröbner bases and the substitution techniques. Each solution gives us one orthogonal design. We now list all of the orthogonal designs of order 128 we have found here:

(1). $OD(128; 1, 1, 63, 63)$:

- A_1;
- A_{65};
- $A_2 = A_3 = \cdots = A_{63} = A_{64}$;
- $A_{66} = A_{67} = \cdots = A_{127} = A_{128}$.

(2). $OD(128; 1, 1, 2, 62, 62)$:

- A_1;
- A_{65};
- $A_{33} = A_{97}$;
- $A_2 = A_3 = \cdots = A_{32} = A_{66} = A_{67} = \cdots = A_{96}$;
- $A_{34} = A_{35} = \cdots = A_{64} = \cdots = A_{98} = A_{99} = A_{128}$.

(3). $OD(128; 1, 1, 2, 4, 60, 60)$:

- A_1;
- A_{65};
- $A_{33} = A_{97}$;
- $A_{17} = A_{49} = A_{81} = A_{113}$;
- $A_2 = \cdots = A_{16} = A_{34} = \cdots = A_{48} = A_{66} = \cdots = A_{80} = A_{98} = \cdots = A_{112}$;
- $A_{18} = \cdots = A_{32} = A_{50} = \cdots = A_{64} = A_{82} = \cdots = A_{96} = A_{114} = \cdots = A_{128}$.

(4). $OD(128; 1, 1, 2, 4, 8, 56, 56)$:

- A_1;
- A_{65};
- $A_{33} = A_{97}$;
- $A_{17} = A_{49} = A_{81} = A_{113}$;
- $A_9 = A_{25} = A_{41} = A_{57} = A_{73} = A_{89} = A_{105} = A_{121}$;
- $A_2 = \cdots = A_8 = A_{18} = \cdots = A_{24} = A_{34} = \cdots = A_{40} = A_{50} = \cdots = A_{56} = A_{66} = \cdots = A_{72} = A_{82} = \cdots = A_{88} = A_{98} = \cdots = A_{104} = A_{114} = \cdots = A_{120}$;
- $A_{10} = \cdots = A_{16} = A_{26} = \cdots = A_{32} = A_{42} = \cdots = A_{48} = A_{58} = \cdots = A_{64} = A_{74} = \cdots = A_{80} = A_{90} = \cdots = A_{96} = A_{106} = \cdots = A_{112} = A_{122} = \cdots = A_{128}$.

(5). $OD(128; 1, 1, 2, 4, 8, 16, \ldots, 16)$:

- A_1;
- A_{65};
- $A_{33} = A_{97}$;
- $A_{17} = A_{49} = A_{81} = A_{113}$;
- $A_9 = A_{25} = A_{41} = A_{57} = A_{73} = A_{89} = A_{105} = A_{121}$;

- $A_2 = A_{10} = A_{18} = A_{26} = A_{34} = A_{42} = A_{50} = A_{58} = A_{66} = A_{74} = A_{82} = A_{90} = A_{98} = A_{106} = A_{114} = A_{122}$;
- $A_3 = A_{11} = A_{19} = A_{27} = A_{35} = A_{43} = A_{51} = A_{59} = A_{67} = A_{75} = A_{83} = A_{91} = A_{99} = A_{107} = A_{115} = A_{123}$;
- $A_4 = A_{12} = A_{19} = A_{28} = A_{36} = A_{44} = A_{52} = A_{60} = A_{68} = A_{76} = A_{84} = A_{92} = A_{100} = A_{108} = A_{116} = A_{124}$;
- $A_5 = A_{13} = A_{20} = A_{29} = A_{37} = A_{45} = A_{53} = A_{61} = A_{69} = A_{77} = A_{85} = A_{93} = A_{101} = A_{109} = A_{117} = A_{125}$;
- $A_6 = A_{14} = A_{21} = A_{30} = A_{38} = A_{46} = A_{54} = A_{62} = A_{70} = A_{78} = A_{86} = A_{94} = A_{102} = A_{110} = A_{118} = A_{126}$;
- $A_7 = A_{15} = A_{22} = A_{31} = A_{39} = A_{47} = A_{55} = A_{63} = A_{71} = A_{79} = A_{87} = A_{95} = A_{103} = A_{111} = A_{119} = A_{127}$;
- $A_8 = A_{16} = A_{23} = A_{32} = A_{40} = A_{48} = A_{56} = A_{64} = A_{72} = A_{80} = A_{88} = A_{96} = A_{104} = A_{112} = A_{120} = A_{128}$.

Chapter 7

Constructing Hadamard Matrices

We will use the orthogonal designs from the last chapter to construct Hadamard matrices of order $128n$. For the orthogonal design of $H_{128} = OD(128; 16, \ldots, 16)$ which can be obtained by equating variables of corresponding $(1, 1, 2, 4, 8)$ from $OD(128; 1, 1, 2, 4, 8, 16, \ldots, 16)$ found in the previous chapter, we have eight indeterminates $A_1, A_2, \cdots, A_8$. We replace $A_1, A_2, \cdots, A_8$ by matrices, and let n denote the order of $n \times n$ matrices $A_1, A_2, \cdots, A_8$. Imitating the classical Williamson construction (see [11]), we take the eight matrices $A_1, A_2, \cdots, A_8$ to be symmetric circulant matrices of order n each, defined via the matrix

$$U = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Define the eight matrix $A_1, A_2, \cdots, A_8$ to be polynomials in U, so that they commute with each other (can be checked easily):

$$\begin{aligned} A_1 &= a_0 I_n + a_1 U + \cdots + a_{n-1} U^{n-1}, \\ A_2 &= b_0 I_n + b_1 U + \cdots + b_{n-1} U^{n-1}, \\ &\vdots \\ A_7 &= g_0 I_n + g_1 U + \cdots + g_{n-1} U^{n-1}, \\ A_8 &= h_0 I_n + h_1 U + \cdots + h_{n-1} U^{n-1}. \end{aligned}$$

Also, the eight matrix $A_1, A_2, \cdots, A_8$ will be symmetric if we require:

$$a_{n-i} = a_i, b_{n-i} = b_i, c_{n-i} = c_i, d_{n-i} = d_i,$$

$$e_{n-i} = e_i, f_{n-i} = f_i, g_{n-i} = g_i, h_{n-i} = h_i.$$

for $i = 1, \ldots, n-1$.

Now let H_{128n} denote the matrix obtained by replacing $A_1, A_2, \cdots, A_8$ with circulant matrices of order n defined above. Then H_{128n} is of order $128n$ and by block matrix multiplication, we have the property:

$$H_{128n} H_{128n}^{Tr} = I_{128} \otimes (16A_1^2 + 16A_2^2 + 16A_3^2 + 16A_4^2 + 16A_5^2 + 16A_6^2 + 16A_7^2 + 16A_8^2), \quad (7.1)$$

where $\otimes$ denotes the Kronecker product. By requiring $16A_1^2+16A_2^2+16A_3^2+16A_4^2+16A_5^2+16A_6^2+16A_7^2+16A_8^2$ be a diagonal matrix, we can find the solutions which make H_{128n} be Hadamard matrices.

7.1 Hadamard matrices for $n = 3, 5$

For $n = 3$, we obtain only one equation from (7.1):

$$a_0a_1 + b_0b_1 + c_0c_1 + d_0d_1 + e_0e_1 + f_0f_1 + g_0g_1 + h_0h_1 + 4 = 0.$$

This equation has exactly 3584 solutions when all 16 variables take ± 1 values. These solutions give rise to Hadamard matrices of order 384.

For $n = 5$, we obtain two equations:

$$\begin{aligned} &a_1a_0 + a_1a_2 + b_1b_0 + b_1b_2 + c_1c_0 + c_1c_2 + d_1d_0 + d_1d_2 + e_1e_0 + e_1e_2 + f_1f_0 \\ &+f_1f_2 + g_1g_0 + g_1g_2 + h_1h_0 + h_1h_2 + 4 = 0, \end{aligned}$$

and

$$\begin{aligned} &a_0a_2 + a_1a_2 + b_0b_2 + b_1b_2 + c_0c_2 + c_1c_2 + d_0d_2 + d_1d_2 + e_0e_2 + e_1e_2 + f_0f_2 \\ &+f_1f_2 + g_0g_2 + g_1g_2 + h_0h_2 + h_1h_2 + 4 = 0. \end{aligned}$$

The solutions of these two equations give rise to Hadamard Matrices of order 640.

If we use $OD(128; 1, 1, 2, 4, 8, 16, \ldots, 16)$ instead of the simplified $OD(128; 16, \ldots, 16)$, we can get many more solutions because we have more variables.

In fact, for $n = 3$, we obtain one equation:

$$a_1a_0 + l_1l_0 + 2k_1k_0 + 4j_1j_0 + 8i_1i_0 + 16b_1b_0 + 16c_1c_0 + 16d_1d_0 \\ + 16e_1e_0 + 16f_1f_0 + 16g_1g_0 + 16h_0h_1 + 64 = 0.$$

This equation has exactly 43904 solutions when all 24 variables take ± 1 values. Each solution give rise to a Hadamard matrix of order 384.

For $n = 5$ we have two equations:

$$64 + 16e_1e_0 + 16e_1e_2 + 16f_1f_0 + 16f_1f_2 + 16g_1g_0 + 16g_1g_2 + 16h_1h_0 + 16h_1h_2 \\ +8i_1i_0 + 8i_1i_2 + 4j_1j_0 + 4j_1j_2 + 2k_1k_0 + 2k_1k_2 + l_1l_0 + l_1l_2 + a_1a_0 \\ +a_1a_2 + 16b_1b_0 + 16b_1b_2 + 16c_1c_0 + 16c_1c_2 + 16d_1d_0 + 16d_1d_2 = 0,$$

and

$$64 + 16e_1e_2 + 16e_0e_2 + 16f_1f_2 + 16f_0f_2 + 16g_1g_2 + 16g_0g_2 + 16h_1h_2 + 16h_0h_2 \\ +8i_1i_2 + 8i_0i_2 + 4j_1j_2 + 4j_0j_2 + 2k_1k_2 + 2k_0k_2 + l_1l_2 + l_0l_2 + a_1a_2 \\ +a_0a_2 + 16b_1b_2 + 16b_0b_2 + 16c_1c_2 + 16c_0c_2 + 16d_1d_2 + 16d_0d_2 = 0.$$

The solutions of these two equations give rise to Hadamard matrices of order 640.

Chapter 8

Conclusion and Future Work

In this article, we have found new orthogonal designs of order 128 and also constructed Hadamard matrices using orthogonal designs we have found. This method can be generalized to higher dimension and can be used to construct higher order orthogonal design and Hadamard Matrix.

Recall Conjecture 1 in Chapter 2. From what we have constructed, we have shown that this conjecture holds for $t = 16$.

Recall the Radon function in Chapter 1. We have shown that the maximum number of different variables for orthogonal design of order 128 can't exceed 16. Also, since we have constructed $OD(128; 1, 1, 2, 4, 8, 16, \ldots, 16)$, we know that the maximum number is not less than 12. Therefore, the maximum number of different variables for orthogonal design of order 128 is between 12 and 16.

For lower order cases, Kotsireas and Koukouvinos [12] has found $OD(32; 4, 4, 4, 4, 4, 4, 4, 4)$ and $OD(64; 8, 8, 8, 8, 8, 8, 8, 8)$ using this techniques. We constructed $OD(128; 16, 16, 16, 16, 16, 16, 16, 16)$, so it is reasonable to conjecture that for every integer n, this method can be used to construct $OD(2^n; s, \ldots, s)$ where $s = 2^{n-3}$. This conjecture should be studied in the future.

References

[1] Elizabeth Arnold. Modular algorithms for computing Gröbner bases. *Journal of Symbolic Computation*, 35(4):403–419, 2003.

[2] L. D. Baumert and M. Hall. Hadamard matrices of the Williamson type. *Math Comput*, 19:442–447, 1965.

[3] L. D. Baumert and M. Hall. A new construction for Hadamard matrices. *Bull Amer Math Soc*, 71:169–170, 1965.

[4] T. Becker and V. Weispfenning. *Gröbner bases. A computational approach to commutative algebra*. Graduate Texts in Mathematics141. Springer-Verlag, New York, 1993.

[5] Charles J. Colbourn and Jeffrey H. Dinitz, editors. *The CRC handbook of combinatorial designs*. CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1996.

[6] R. Craigen. Hadamard matrices and designs. In C. J. Colbourn and J. H. Dinitz, editors, *The CRC Handbook of Combinatorial Designs*, pages 370–377. CRC Press, Boca Raton, Florida, 1996.

[7] Jean-Charles Faugére. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).

[8] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 75–83 (electronic), New York, 2002. ACM.

[9] Pauer Franz. On lucky ideals for Gröbner basis computation. *Journal of Symbolic Computation*, 14:471–482, 1992.

[10] Anthony V. Geramita and Jennifer Seberry. *Orthogonal designs*, volume 45 of *Lecture Notes in Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 1979. Quadratic forms and Hadamard matrices.

[11] M. Hall Jr. *Combinatorial Theory*. Reprint of 1986 second edition. Wiley Classics Library, Wiley, New York, 1998.

[12] Ilias S. Kotsireas and Christos Koukouvinos. Orthogonal designs via computational algebra. *J. Combin. Des.*, 14(5):351–362, 2006.

[13] Christos Koukouvinos and Dimitris E. Simos. Improving the lower bounds on inequivalent Hadamard matrices through orthogonal designs and meta-programming techniques. *Applied Numerical Mathematics*, 60:370–377, 2010.

[14] J. Seberry and R. Craigen. Orthogonal designs. In C. J. Colbourn and J. H. Dinitz, editors, *The CRC Handbook of Combinatorial Designs*, pages 400–406. CRC Press, Boca Raton, Florida, 1996.