A Study on Extension of Orthogonal Arrays and its Application to Experimental Designs and Unequal Error Protection Codes

直交配列の拡張及びその実験計画法と不均一誤り訂正符号への応用に関する研究

July 2010

Tomohiko Saito

A Study on Extension of Orthogonal Arrays and its Application to Experimental Designs and Unequal Error Protection Codes

直交配列の拡張及びその実験計画法と不均一誤り訂正符号への応用に関する研究

July 2010

Tomohiko Saito

Acknowledgments

I would like to thank my supervisor, Professor Toshiyasu Matsushima. He gave me the opportunity to begin this study and has taught me many things. I cannot thank him enough for his support. I would also like to thank my co-advisors, Professor Shigeichi Hirasawa, Professor Shin'ichi Oishi, and Professor Kiichiro Hashimoto, for their invaluable comments. In particular, Professor Shigeichi Hirasawa provided many suggestions and encouragement for this study. I would also like to thank Professor Shizuo Mawatari of the Aoyama Gakuin University. He gave me the opportunity to write this thesis and always encouraged me. I would also like to thank Professor Yoshifumi Ukita of the Yokohama College of Commerce, Dr. Ryo Nomura of Senshu University and Takahiro Yoshida of Chuo University for their invaluable comments and encouragement. I would also like to thank my colleagues, Dr. Yasunari Maeda of the Kitami Institute of Technology, Dr. Kazuhiko Minematsu of the NEC Corporation, Daiki Koizumi of Cyber University, Dr. Tota Suko and Shunsuke Horii of Waseda University, and Dr. Naoto Kobayashi of the Tokyo University of Technology for their advice. I would also like to thank all the members of the Matsushima and Hirasawa Laboratories at Waseda University. Finally, I would like to thank my parents for supporting me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Orthogonal Arrays (OAs) are essential in statistics and are used in fields such as computer science and cryptography [1, 7, 22, 23]. In statistics, they are primarily used in experimental designs, that is, they are immensely important in all areas of human investigation such as medicine, agriculture, and manufacturing. Further, the mathematical theory of OAs is beautiful and related to combinatorics, finite fields, geometry, and error-correcting codes [4, 7]. Therefore, many studies contributed on OAs from various points of view.

The definition of OAs is as follows: An OA is defined as an $N \times k$ array with entries from $GF(s)$ and *every* $N \times t$ *sub-array contains each* t*-tuple based on* $GF(s)$ *exactly the same number of times as a row.* This is called an OA with a *strength* t. Given below is an example of an OA of strength 2:

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$

Let us pick any two columns, say the first and the last:

$$\begin{array}{cc} 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{array}$$

We can observe that all 2-tuples, i.e.,

$$0\ 0, \quad 0\ 1, \quad 1\ 0, \text{ and } 1\ 1,$$

appear, and they all appear the same number of times (i.e., once).

On the basis of this observation, the construction problem for OAs can be formulated as follows:

- Find the OA with a minimum number of rows N, given the number of columns k, the order of the Galois field s, and the strength t.

It is important to find the lower bounds for OAs for this construction problem. The problem for finding the lower bounds for OAs is as follows:

- Find the lower bound for the number of rows N, given the number of columns k, the order of the Galois field s, and the strength t.

The lower bounds calculated for the number of rows are useful not only for evaluating OAs but also for the construction of OAs.

As stated above, the main applications of OAs are in experimental designs. Experimental designs are techniques that are employed to acquire more information using fewer experiments. For example, the following case is studied using an experimental design.

- In the manufacture of iron, the hardness of iron may be influenced by factors such as temperature, pressure, and the catalyst. Let us assume that certain manufacturers want to analyze how these factors and their interactions influence the hardness of iron. In this case, candidates, called *levels*, are set for each factor. For example, the temperatures can be of two levels 800 or 1000 degrees C, pressure can be 2 or 3 atmospheres and the catalyst can be from company A or B.

In this thesis, we consider only that case in which each factor has the same number of levels.

In experimental designs, it is important to design experiments that can estimate all the effects of different factors and various interactions that could affect the response variable of interest such as the hardness of iron, and where the number of experiments is kept as few as possible. For example, all the effects due to the factors and interactions can be estimated by conducting experiments all possible level combinations as given by the following.

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array}$$

In this table, each row represents one level combination. For example, the first row represents the experiment where the temperature is 800 degrees C, pressure is 2 atmospheres, and the catalyst is from company A. However, the number of experiments that need to be performed for all level combinations is too large; therefore, it is important to reduce the number of experiments. We know that the number of experiments that need to be conducted can be reduced by using OAs. When an OA is used in experimental designs, each row of the OA corresponds to one level combination. Therefore, the number of columns, the number of rows, and the order of the Galois field in the OA correspond to the number of factors, the number of experiments,

and the number of levels, respectively. Moreover, the strength of the OA corresponds to the interactions between factors. More specifically, an OA with the strength $2r$ is needed when all the interactions between r number of factors may affect the response variable.

OAs are also closely related to error-correcting codes. Error-correcting codes are coding techniques that can be used to correct errors in digital data transmission. In the study of error-correcting codes, the construction of codes is one of the most basic problems.

The definition of codes is as follows: A code is a subset of a k-dimension vector space over $GF(s)$, and when the size of the code is N and the Hamming distance of any two elements in the code is greater than d, it is called an s-nary code with the code length k, size N, and minimal distance d. The construction problem for codes can be formulated as follows:

- Find the code with the maximum size N, given the code length k, the order of the Galois field s, and the minimal distance d.

For the above construction problem, it is important to find the upper bounds for the codes using the following problem:

- Find the upper bound for the size N, given the code length k, the order of the Galois field s, and the minimal distance d.

In the past, OAs and codes had been developed in different fields, but the relationship between the two was clarified by Delsarte [4] and Hedayat and Sloane et al. [7]. In particular, the relationship between the strength of an OA and the dual distance of a code, which is one of the parameters used to characterize a code, is very important. Thus, some results from OAs can be applied to error-correcting codes and vice versa.

In previous work, Delsarte proposed the use of linear programming (LP) bounds to solve the problem of finding lower bounds for OAs (and upper bounds for codes) [4]. Consequently, the process of finding lower bounds for OAs (and upper bounds for codes) reduces to solving LP problems. Moreover, Hedayat and Sloane et al. actually solved these LP problems using a computer and compared the obtained LP bounds with other lower bounds [7]. From these results, it was found that the LP bounds are the tightest lower bounds.

In this thesis, we extend the definition of the strength t to the partial strength $T(\subseteq \{0,1\}^k)$ and introduce OAs with partial strength (POAs). In the study of POAs, the following construction problem is important.

- Find the POA with a minimum number of rows N, given the number of columns k, the order of the Galois field s, and the partial strength T.

Further, the following problem used to find the lower bound for POAs is also important.

- Find the lower bound for a number of rows N, given the number of columns k, the order of the Galois field s, and the partial strength T.

We address these problems in this thesis.

We introduce POAs for the following reasons:

1. POAs are more suitable for experimental designs than OAs because POAs can be used with a complicated model.

2. A subclass of POAs is related to unequal error protection (UEP) codes.

Let us discuss the first reason. As stated above, an OA with a strength $2r$ is needed if *all* interactions between r number of factors may affect the response variable. In an experimental design, it is, however, natural to assume a more complicated model, such that *some* interactions between the r factors may affect the response variable. POAs can be used with such a model, whereas OAs cannot be used. Next, we discuss the second reason. UEP codes were proposed by Masnick et al. [12]. UEP codes are useful for transmitting data having a different magnitude in each bit. In this thesis, we show that a subclass of POAs is related to UEP codes. From this relation, we can apply some results of POAs to UEP codes.

In this thesis, we define POAs and present the applications of POAs in experimental designs. Also, we describe the relationship between POAs and error-correcting codes. We propose finding the LP bounds for POAs by extending the method for finding LP bounds for OAs, as given by Delsarte. Consequently, the process of finding lower bounds for POAs reduces to solving LP problems. Therefore, we solve the LP problems using a computer and provide some numerical examples of LP bounds for POAs.

However, the number of variables or constrains in the LP problems is very high and the LP problems cannot be solved if the number of factors k is large. Thus, it is important to consider the subclasses of the POAs, such that the LP problems corresponding to the subclasses can be solved easily and such that the subclasses are important in applications involving experimental designs or error-correcting codes. Hence, we define OAs with different strengths in each column as a subclass of POAs. We then show that the LP problems corresponding to this subclass can be solved easily and that this subclass is important in applications involving experimental designs and error-correcting codes. We especially clarify the relation between this subclass and UEP codes and propose LP bounds for UEP codes. Moreover, we compare the obtained LP bounds for UEP codes with other bounds for UEP codes as proposed by Masnick et al. [12].

Lastly, we propose some construction methods for OAs with different strengths in each column. These methods use the relation between POAs and error-correcting codes and also use the construction methods for UEP codes as proposed by Masnick et al. [12] and Boyarinov et al. [2].

This thesis is organized as follows. In Chapter 2, we present some basic notations and provide previous studies as preliminaries. We first define some basic notations and OAs. Next, we discuss the application of OAs in experimental designs and the relation between OAs and error-correcting codes. Lastly, we present LP bounds for OAs (and error-correcting codes).

In Chapter 3, we discuss POAs. We first define POAs and present some basic properties of POAs. Next, we discuss the application of POAs in experimental designs and the relation between POAs and error-correcting codes. Lastly, we propose LP bounds for POAs and provide some numerical examples of LP bounds for POAs.

In Chapter 4, we describe OAs with different strengths in each column and also provide UEP codes. We first define OAs with different strengths in each column and provide UEP codes, and then propose their LP bounds. Further, we compare the LP bounds obtained for UEP codes with other upper bounds observed for UEP codes.

In Chapter 5, we present the construction methods for OAs with different strengths in each column. After proposing the construction methods for OAs with different strengths in each column, we provide some numerical examples of POAs constructed using the proposed methods.

In Chapter 6, we conclude this thesis and discuss our future studies.

Chapter 2

Preliminaries

In this chapter, we present some basic notions and provide previous studies as preliminaries. we first define some basic notations used in the whole of this thesis. Next, we define OAs and provide some basic properties of OAs. Next, we discuss the the applications of OAs. In particular, we provide details about the application of OAs in experimental designs. Next, we show the relation between OAs and error-correcting codes and provide some properties of OAs from the relation. Lastly, we present LP bounds for OAs (and error-correcting code).

2.1 Basic Notation

In this section, we define some basic notations used in the whole of this thesis. Let $GF(s)$ be the Galois field of order s. Let $\oplus$ be the exclusive-or operation, and $\cdot$ be the and operation. For any $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_k) \in \{0,1\}^k$, let $\boldsymbol{x} \oplus \boldsymbol{y} = (x_1 \oplus y_1, x_2 \oplus y_2, \ldots, x_n \oplus y_n)$, and $\boldsymbol{x} \cdot \boldsymbol{y} = x_1 \cdot y_1 \oplus x_2 \cdot y_2 \oplus \cdots \oplus x_n \cdot y_n$. Let $w(\boldsymbol{x})$ be the Hamming weight of $\boldsymbol{x} = (x_1, x_2, \ldots, x_k) \in \{0,1\}^k$, so defined by

$$w(\boldsymbol{x}) = \Big|\{i | x_i \neq 0\}\Big|. \tag{2.1}$$

The *Hamming distance* $dist(\boldsymbol{u}, \boldsymbol{v})$ between two vectors $\boldsymbol{u}, \boldsymbol{v} \in \{0,1\}^k$ is defined to be the number of positions where they differ, or in other words

$$dist(\boldsymbol{u}, \boldsymbol{v}) = w(\boldsymbol{u} \oplus \boldsymbol{v}). \tag{2.2}$$

For any set A, let $|A|$ be the number of elements of A. Let

$$\left(\begin{array}{c} x \\ m \end{array} \right) = \begin{cases} \frac{x(x-1)\cdots(x-m+1)}{m!} & \text{if } m \text{ is a positive integer ,} \\ 1 & \text{if } m = 0, \\ 0 & \text{otherwise ,} \end{cases} \tag{2.3}$$

where x is any nonnegative integer, and $m! = 1 \cdot 2 \cdot \cdots \cdot (m-1) \cdot m$, $0! = 1$.

2.2 Orthogonal Arrays

In this section we define OAs and provide some basic properties of OAs. At first, OAs can be defined as follows.

Definition 2.1 [7, Definition 1.1] An $N \times k$ array A with entries from $GF(s)$ is said to be an *Orthogonal Array with strength* t if every $N \times t$ sub-array of A contains each t-tuple based on $GF(s)$ exactly same times as row. We will denote such an array by $OA(N, k, s, t)$. ■

In the following, unless mentioned explicitly, we will consider the case that $s = 2$ for simplicity. Also, we will consider OAs whose rows are all distinct. These are called simple OAs. The next two examples give examples of OAs.

Example 2.1 The array in Table 2.2 is an $OA(8, 4, 2, 3)$. ■

Table 2.1: An $OA(8, 4, 2, 3)$

0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Example 2.2 The array in Table 2.2 is an $OA(4, 3, 2, 2)$. ■

Table 2.2: An OA with 2 levels and strength 2:$OA(4, 3, 2, 2)$

0	0	0
0	1	1
1	0	1
1	1	0

An $OA(N, k, 2, t)$ is said to be *linear* if the rows of $OA(N, k, 2, t)$ form a linear vector space. If an $OA(N, k, 2, t)$ is linear, the $OA(N, k, 2, t)$ has a basis for the linear vector space. This basis is usually given in the form of a $(\log_2 N) \times k$ matrix called a *generator matrix* whose rows are the basis.

Example 2.3 The $OA(8, 4, 2, 3)$ given in Table 2.2 is linear, and has the generator matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2.4}$$

■

Example 2.4 The $OA(4, 3, 2, 2)$ given in Table 2.2 is linear, and has the generator matrix

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

■

Moreover, the generator matrix of an $OA(N, k, 2, t)$ has the following properties.

Lemma 2.1 [7, Theorem 3.27 and 3.29] *Let A be an $N \times k$ linear array with 0,1 entries, and G be a generator matrix of A. A is an $OA(N, k, 2, t)$ if and only if any t columns of G are linearly independent over $\{0, 1\}$.* ■

The next lemma gives a necessary and sufficient condition for an array to be an OA, which does not assume linearity.

Lemma 2.2 [7, Theorem 3.30] *An $N \times k$ array A with $0, 1$ entries is an $OA(N, k, 2, t)$ if and only if*

$$\sum_{\boldsymbol{v}=\text{row of } A} (-1)^{\boldsymbol{u}\cdot\boldsymbol{v}} = 0, \tag{2.6}$$

for all $0, 1$ vectors $\boldsymbol{u}$ containing w 1's, for all w in the range $1 \leq w \leq t$, where the sum is over all rows $\boldsymbol{v}$ of A. ■

2.3 Applications of Orthogonal Arrays

2.3.1 Applications of Orthogonal Arrays

OAs are mainly used in experimental designs [1, 22]. This means that they are immensely important in all areas of human investigation such as medicine, agriculture, and manufacturing. Also, OAs are used in computer science and cryptography. For example, there are reports in which OAs are applied to secret sharing [19], authentication codes [20] and so on. In this study, the application in experimental designs is especially important, so we give a detailed explanation about experimental designs in Section 2.3.2.

Moreover, the mathematical theory of OAs is related to other combinatorics, such as error-correcting codes, difference schemes, Hadamard matrices, and Latin squares [4, 7]. In particular, OAs are closely related to error-correcting codes. Therefore, some results of OAs can be applied to these fields, and vice verse. In this study, the relation between OAs and error-correcting codes is especially important, so we provide details about the relation in Section 2.4.

2.3.2 Experimental Designs

Let $F_1, F_2, \ldots, F_k$ denote the k factors to be included in the experiment. In this thesis, we consider only the case that the number of levels for each factor is two. Therefore, the levels of each factor can be represented by 0 and 1, and the level combinations can be represented by the k-tuples $(x_1, x_2, \ldots, x_k) \in \{0,1\}^k$.

We use $y(\boldsymbol{x})$ to denote the response of the experiment with the level combination $\boldsymbol{x}$, and assume the following model.

$$y(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in I} f_{\boldsymbol{a}} \chi_{\boldsymbol{a}}(\boldsymbol{x}) + e(\boldsymbol{x}), \tag{2.7}$$

where,

- $I (\subseteq \{0,1\}^k)$: indexes of main and interactive factors included in the model (For example, $I = \{000, 100, 010, 001, 110\}$ suggests main factors of F_1, F_2, F_3 and an interactive factor of F_1F_2.),
- $f_{\boldsymbol{a}}$: an unknown parameter that represents the effect of a main or interactive factor $\underline{a} \in I$,
- $\chi_{\boldsymbol{a}}(\boldsymbol{x}) := (-1)^{\boldsymbol{a} \cdot \boldsymbol{x}^T}$,
- $e(\boldsymbol{x})$: a random error that has mean 0 and constant variance σ^2.

Moreover, we assume that the set I in (2.7) satisfies the following *monotonicity*.

$$\boldsymbol{a} \in I \Rightarrow \boldsymbol{b} \in I \quad \text{for } \forall\, \boldsymbol{b}\, (\boldsymbol{b} \sqsubseteq \boldsymbol{a}), \tag{2.8}$$

where $(b_1, b_2, \ldots, b_k) \sqsubseteq (a_1, a_2, \ldots, a_k)$ suggests that $b_i \leq a_i$, $i = 1, 2, \ldots k$.

In experimental designs, we are given a model of the experiment. This means that we are given an $I \subseteq \{0,1\}^k$ in (2.7). Then, we determine a set of $\boldsymbol{x} \in \{0,1\}^k$, which is called a *design* $X \subseteq \{0,1\}^k$. And we experiment according to the design X and estimate the effects $f_{\boldsymbol{a}}$, $\boldsymbol{a} \in I$ from the results of the experiments $\{(\boldsymbol{x}, y(\boldsymbol{x})) | \boldsymbol{x} \in X\}$.

In experimental designs, it is especially important to determine a design X so that all effects of main and interactive factors in the model can be estimated, where the number of experiments $|X|$ is as few as possible. For example, if we experiment with all level combination, that is $X = \{0,1\}^k$, we can get unbiased estimators for effects of all main and interactive factors by

$$\hat{f}_{\boldsymbol{a}} = \frac{1}{2^k} \sum_{\boldsymbol{x} \in \{0,1\}^k} y(\boldsymbol{x}) \chi_{\boldsymbol{a}}(\boldsymbol{x}). \tag{2.9}$$

Example 2.5 In a certain factory, materials(F_1), machines(F_2) and temperatures(F_3) are factors that may affect a ratio y of defective products. Each factor has two levels:

F_1 : F_0^1 (made in A company), F_1^1 (B company)

F_2 : F_0^2 (a new machine), F_1^2 (an old machine)

F_3 : F_0^3 (100 deg C), F_1^3 (200 deg C)

Moreover, all interactions between two factors, $F_1 \times F_2$, $F_1 \times F_3$, $F_2 \times F_3$, may affect y. Then, the model of y is as follows.

$$y(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in I} f_{\boldsymbol{a}} \chi_{\boldsymbol{a}}(\boldsymbol{x}) + e(\boldsymbol{x}), \tag{2.10}$$

where $y(\boldsymbol{x})$ is a ratio of defective products with the level combination $\boldsymbol{x} \in \{0,1\}^k$, and

$$I = \{000, 100, 010, 001, 110, 101, 011\}. \tag{2.11}$$

Then, if we experiment with all level combination as shown in Table 2.3, we can get unbiased estimators for all $f_{\boldsymbol{a}}$ using (2.9). For example, $\hat{f}_{100}$ is calculated by

$$\begin{aligned} \hat{f}_{100} &= \frac{1}{2^3} \sum_{\boldsymbol{x} \in \{0,1\}^3} y(\boldsymbol{x}) \chi_{100}(\boldsymbol{x}) && (2.12) \\ &= \frac{1}{8}\{y(000) + y(001) + y(010) + y(011) - y(100) - y(101) \\ &\qquad -y(110) - y(111)\}, && (2.13) \end{aligned}$$

so $\hat{f}_{100}$ is as follows:

$$\begin{array}{rl} y(000) = & f_{000} + f_{100} + f_{010} + f_{001} + f_{110} + f_{101} + f_{011} + e(000), \\ y(001) = & f_{000} + f_{100} + f_{010} - f_{001} + f_{110} - f_{101} - f_{011} + e(001), \\ y(010) = & f_{000} + f_{100} - f_{010} + f_{001} - f_{110} + f_{101} - f_{011} + e(010), \\ y(011) = & f_{000} + f_{100} - f_{010} - f_{001} - f_{110} - f_{101} + f_{011} + e(011), \\ -y(100) = & -f_{000} + f_{100} - f_{010} - f_{001} + f_{110} + f_{101} - f_{011} + e(100), \\ -y(101) = & -f_{000} + f_{100} - f_{010} + f_{001} + f_{110} - f_{101} + f_{011} + e(101), \\ -y(110) = & -f_{000} + f_{100} + f_{010} - f_{001} - f_{110} + f_{101} + f_{011} + e(110), \\ -y(111) = & -f_{000} + f_{100} + f_{010} + f_{001} - f_{110} - f_{101} - f_{011} + e(111), \\ \hline \hat{f}_{100} = & \qquad f_{100} \qquad\qquad\qquad\qquad\qquad\qquad + \bar{e}_{100}, \end{array}$$

where $\bar{e}_{100} = \frac{1}{8} \sum_{\boldsymbol{x} \in \{0,1\}^k} e(\boldsymbol{x})$.

■

If we experiment with all level combinations, we can get unbiased of all $f_{\boldsymbol{a}}$, but the number of the experiments is too large. Therefore, we reduce the number of experiments using OAs.

Let $A' (\subseteq \{0,1\}^k)$ be a set whose elements are rows of an $OA(N, k, 2, 2r)$, so $|A'| = N$. Suppose that we can assume that at most r interactive factors are included in the model, that is

$$I = \{\boldsymbol{e} \in \{0,1\}^k | w(\boldsymbol{e}) \leq r\}, \tag{2.14}$$

in (2.7). Then, if we experiment according to the design A', we can get unbiased estimators for $f_{\boldsymbol{a}}$, $\boldsymbol{a} \in I$ using the following calculation:

$$\hat{f}_{\boldsymbol{a}} = \frac{1}{|A'|} \sum_{\boldsymbol{x} \in A'} y(\boldsymbol{x}) \chi_{\boldsymbol{a}}(\boldsymbol{x}). \tag{2.15}$$

Table 2.3: Experiment conditions and data

Experiment no.	F_1	F_2	F_3	y [%]
1	0	0	0	0.5
2	0	0	1	0.4
3	0	1	0	0.1
4	0	1	1	0.1
5	1	0	0	1.2
6	1	0	1	1.5
7	1	1	0	0.7
8	1	1	1	0.6

Example 2.6 In Example 2.5, suppose that we know that no interaction of factors affect y. Then, we can assume the following model

$$y(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in I} f_{\boldsymbol{a}} \chi_{\boldsymbol{a}}(\boldsymbol{x}) + e(\boldsymbol{x}), \tag{2.16}$$

where

$$I = \{000, 100, 010, 001\}. \tag{2.17}$$

Then, we experiment according to $A' = \{000, 011, 101, 110\}$, as shown in Table 2.4. This design A' is from the $OA(4, 3, 2, 2)$ in Table 2.2 . Then, we can get the unbiased estimators for $f_{\boldsymbol{a}}$, $f_{\boldsymbol{a}} \in I$ by (2.15). For example, $\hat{f}_{100}$ is calculated by

$$\hat{f}_{100} = \frac{1}{4} \sum_{\boldsymbol{x} \in A'} y(\boldsymbol{x}) \chi_{100}(\boldsymbol{x}) \tag{2.18}$$

$$= \frac{1}{4} \{y(000) + y(011) - y(101) - y(110)\}, \tag{2.19}$$

so $\hat{f}_{100}$ is as follows:

$$\begin{aligned}
y(000) &= f_{000} + f_{100} + f_{010} + f_{001} + e(000), \\
y(011) &= f_{000} + f_{100} - f_{010} - f_{001} + e(011), \\
-y(101) &= -f_{000} + f_{100} - f_{010} + f_{001} + e(101), \\
-y(110) &= -f_{000} + f_{100} + f_{010} - f_{001} + e(110), \\
\hline
\hat{f}_{100} &= f_{100} + \bar{e}_{100},
\end{aligned}$$

where $\bar{e}_{100} = \frac{1}{4} \sum_{\boldsymbol{x} \in A'} e(\boldsymbol{x})$.

■

Table 2.4: Experiment conditions and data using an OA

Experimental no.	F_1	F_2	F_3	y(%)
1	0	0	0	0.5
2	0	1	1	0.1
3	1	0	1	1.5
4	1	1	0	0.7

2.4 Properties of Orthogonal Arrays from Relations with Error-Correcting Codes

In this section, we introduce error-correcting codes and their relation with OAs. Moreover, we provide some properties of OAs from the relation. These properties are useful for leading to LP bounds in Section 2.5.

2.4.1 Error-Correcting Codes

An *error-correcting code* or simply *code* is any collection C of vectors in $GF(s)^k$. The vectors in C are called *codewords*. In this thesis, we consider only the case that $s = 2$ as well as OAs. We define the *minimal distance* d of a code C to be the minimal Hamming distance between distinct codewords:

$$d = \min_{\boldsymbol{u},\boldsymbol{v}\in C,\boldsymbol{u}\neq\boldsymbol{v}} dist(\boldsymbol{u},\boldsymbol{v}).$$

If $C(\subseteq \{0,1\}^k)$ contains N codewords and the minimal distance of C is d, then we say that it is a code of the length k, size N, and minimal distance d over $GF(2)$ or simply $(k, N, d)_2$ code.

Example 2.7 $C_1 = \{000, 011, 101, 110\}$ is an $(3, 4, 2)_2$ code. ■

Example 2.8

$$\begin{aligned} C_2 = \{&0000000, 0110100, 1110010, 1000110, 1010001, 1100101, 0100011, 0010111, \\ &1101000, 1011100, 0011010, 0101110, 0111001, 0001101, 1001011, 1111111, \} \end{aligned} \tag{2.20}$$

is a $(7, 16, 3)_2$ code. This code is a member of the class of codes called Hamming codes. ■

C is said to be linear if C is a linear vector subspace. As well as linear OAs, a linear code is specified by a basis for linear vector space, given in the form of $(\log_2 N) \times k$ generator matrix. Moreover, a linear code C is also specified by a *parity check matrix*, which is $(k - \log_2 N) \times k$ matrix H defined by

$$H\boldsymbol{x}^T = 0, \tag{2.21}$$

for any $\boldsymbol{x} \in C$.

Example 2.9 The code C_1 in Example 2.7 is linear and has the generator matrix

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \tag{2.22}$$

and has the parity check matrix

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}. \tag{2.23}$$

■

Example 2.10 The code C_2 in Example 2.8 is linear and has the generator matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \tag{2.24}$$

and has the parity check matrix

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{2.25}$$

■

Further, if C is linear, then its dual code $C^{\perp}$ is defined by the set of vectors which are orthogonal to all codewords of C:

$$C^{\perp} := \{\boldsymbol{u} | \boldsymbol{u} \cdot \boldsymbol{v} = 0 \text{ for all } \boldsymbol{v} \in C\}. \tag{2.26}$$

We note that a generator matrix of $C^{\perp}$ is a parity check matrix of C. Let $d^{\perp}$ be the minimal distance of $C^{\perp}$. Then $d^{\perp}$ is said to be the *dual distance* of C.

Further, the minimal distance can be defined using the *distance distribution*. Next, we define the distance distribution. In order to define the distance distribution, we introduce the next notation.

Definition 2.2 For any positive integer k, $W_i^{(k)} (= W_i)$, $i = 0, 1, \ldots k$, are defined by

$$W_i^{(k)} := \{\boldsymbol{w} \in \{0,1\}^k | w(\boldsymbol{w}) = i\}. \tag{2.27}$$

If there is no danger of confusion we omit the k. ■

Then, the *distance distribution of a code* $C (\subseteq \{0,1\}^k)$ is defined as $(k+1)$-tuple $(A_0, A_1, \ldots, A_k)$, where

$$A_i = \frac{1}{|C|} \sum_{\boldsymbol{x} \in C} \Big| \{\boldsymbol{y} \in C | \boldsymbol{x} \oplus \boldsymbol{y} \in W_i\} \Big|, i = 0, 1, \ldots, k. \tag{2.28}$$

Moreover, the minimal distance of a code C is the largest positive integer d such that

$$A_1 = A_2 = \cdots = A_{d-1} = 0. \tag{2.29}$$

Further, if C is a linear code, the distance distribution A_i is equal to the *weight distribution* A'_i, that is

$$A_i = A'_i \left(= \left| \{ \boldsymbol{y} \in C | \boldsymbol{y} \in W_i \} \right| \right), i = 0, 1, \ldots, k. \tag{2.30}$$

2.4.2 Delsarte Theorem

Next, we provide some properties of the distance distribution. For this, we define the Krawtchouk polynomial in the next Definition 2.3.

Definition 2.3 For any positive integer k, the *Krawtchouk polynomial* $P_i(z; k)(= P_i(z))$ is defined by

$$P_i(z; k) := \sum_{r=0}^{i} (-1)^r \begin{pmatrix} z \\ r \end{pmatrix} \begin{pmatrix} k-z \\ i-r \end{pmatrix}, i = 0, 1, \ldots k, \tag{2.31}$$

where z is an indeterminate. If there is no danger of confusion we omit the k. ■

The Krawtchouk polynomial satisfy the following Lemma 2.3.

Lemma 2.3 [7, Theorem 4.10] *If* $\boldsymbol{v} \in W_j$, *then*

$$\sum_{\boldsymbol{u} \in W_i} (-1)^{\boldsymbol{u} \cdot \boldsymbol{v}} = P_i(j), \tag{2.32}$$

where $i, j \in \{0, 1, \ldots, k\}$. ■

Lemma 2.3 is useful for proving Theorem 2.1 and Theorem 2.2.

The next Theorem 2.1 is called MacWilliams Theorem. MacWilliams Theorem provides an important property of the distance distribution.

Theorem 2.1 [11, Ch.5 Theorem 1] *Let* $C(\subseteq \{0, 1\}^k)$ *be a linear code, and* $C^\perp$ *be the dual code of* C. *Let* A_i *and* $A_i^\perp$, $i = 0, 1, \ldots, k$, *be the distance distribution of* C *and* $C^\perp$. *Then,*

$$A_i^\perp = \frac{1}{|C|} \sum_{j=0}^{k} A_j P_i(j), i = 0, 1, \ldots, k. \tag{2.33}$$

■

The C is constrained to a linear code in Theorem 2.1, but the MacWilliams Theorem holds if C is a nonlinear code [11]. However, we do not use these results directly, and we use (2.33) to define the dual distance in the same way as [7]. We give a detailed explanation about this in Section 2.4.3.

The next Theorem 2.2 is called Delsarte Theorem. Delsarte Theorem also provides an important property of the distance distribution.

Theorem 2.2 [11, Ch.5 Theorem 6] *Let $C(\subseteq \{0,1\}^n)$ be a code, and $A_i, i = 0, 1, \ldots, k$ be the distance distribution of C. Then,*

$$\frac{1}{|C|}\sum_{j=0}^{k} A_j P_i(j) \geq 0, \quad i = 0, 1, \ldots, k. \tag{2.34}$$

■

Delsarte Theorem is a strong linear constraint of the distance distribution. Therefore, Delsarte Theorem is useful for leading to LP bounds.

2.4.3 Relations between Orthogonal Arrays and Error-Correcting Codes

For a nonlinear code, we still define the dual distance distribution by (2.33), calling the numbers $(A_0^\perp, A_1^\perp, \ldots, A_k^\perp)$ the *MacWilliams transform* of the distance distribution. Then, it is still ture that $A_i^\perp \geq 0$ for all i from (2.34), and we define the dual distance to be the largest positive integer $d^\perp$ such that

$$A_1^\perp = A_2^\perp = \cdots = A_{d^\perp - 1}^\perp = 0. \tag{2.35}$$

Thus, if $A_1^\perp = A_2^\perp = \cdots = A_k^\perp = 0$, we define $d^\perp$ to be $k+1$. It also follows that $A_0^\perp = 1$ from (2.33).

Theorem 2.3 [7, Theorem 4.9] *If C is a $(k, N, d)_2$ code over $\{0,1\}$ with dual distance $d^\perp$, then the codewords of C form the rows of an $OA(N, k, 2, d^\perp - 1)$ with entries from $\{0,1\}$. Conversely, the rows of an $OA(N, k, 2, t)$ over $\{0,1\}$ form a $(k, N, d)_2$ linear code over $\{0,1\}$ with dual distance $d^\perp \geq t+1$. If the OA has strength t but not $t+1$, $d^\perp = t+1$.* ■

Example 2.11 Let $C = \{000, 011, 101, 110\}$. This is a $(3, 4, 2)_2$ code. Then $C^\perp = \{000, 111\}$, so the dual distance of C is 3. Therefore, the OA corresponding to the code C, that is in Table 2.2, is an $OA(4, 3, 2, 2)$. ■

2.4.4 Properties of Orthogonal Arrays from the Relations with Error-Correcting Codes

From Theorem 2.3, an OA can be regarded as a code. So, let $\bar{C}$ be an $OA(N, k, 2, t)$ and C be a code formed by $\bar{C}$ in the same way as Theorem 2.3. Then, the distance distribution $(A_0, A_1, \ldots, A_k)$ of the $\bar{C}$ is defined by (2.28), and satisfy

$$N = A_0 + A_1 + \cdots + A_k. \tag{2.36}$$

Moreover, the distance distribution of the $\bar{C}$ satisfy

$$A_0 = 1, \tag{2.37}$$

$$A_i \geq 0, i = 1, 2, \ldots, k, \tag{2.38}$$

$$\sum_{j=0}^{k} A_j P_i(j) \geq 0, i = 0, 1, \ldots k, \tag{2.39}$$

$$\sum_{j=0}^{k} A_j P_i(j) = 0, i = 1, 2, \ldots t, \tag{2.40}$$

where (2.39) is from Theorem 2.2 and (2.40) is from Theorem 2.3. These properties are useful for leading to LP bounds in Section 2.5.

2.5 Main Problems in Orthogonal Arrays

2.5.1 Main Problems in Orthogonal Arrays

Construction problem for OAs and problem for finding lower bounds for OAs can be formulated as follows.

- Find the OA with a minimum number of rows N, given the number of columns k and the strength t.
- Find the lower bound for the number of rows N, given the number of columns k and the strength t.

These are two main problems in the study of OAs, so there are many studies for the two problems [7]. Delsarte proposed linear programming (LP) bounds to solve the problem of finding lower bounds [4]. And now, it is known that LP bounds are very good lower bounds for OAs. We give a detailed explanation of the LP bounds in the next Section 2.5.2.

2.5.2 Linear Programming Bounds for Orthogonal Arrays

Next, we present the LP bounds as proposed by Delsarte [4]. The next Theorem 2.4 provides the LP bounds for OAs.

Theorem 2.4 [7, Theorem 4.15] *Let $N_{LP}(k; d^{\perp})$ be the solution to the following linear programming problem: choose real numbers $A_0, A_1, \ldots A_k$ so as to*

$$\textit{minimize} \quad \sum_{i=0}^{k} A_i \tag{2.41}$$

subject to the constraints

$$A_0 = 1, \tag{2.42}$$

$$A_i \geq 0, i = 1, 2, \ldots, k, \tag{2.43}$$

$$\sum_{j=0}^{k} A_j P_i(j) \geq 0, i = 0, 1, \ldots k, \tag{2.44}$$

$$\sum_{j=0}^{k} A_j P_i(j) = 0, i = 1, 2, \ldots t, \tag{2.45}$$

where $t = d^{\perp} - 1$. *Then the size of any orthogonal array* $OA(N, k, 2, t)$ *satisfies*

$$N \geq N_{LP}(k; d^{\perp}). \tag{2.46}$$

■

We can obtain these LP bounds for OAs from the properties in Section 2.4.4. Further, we can obtain analogous result for codes as follows.

Theorem 2.5 *Let* $M_{LP}(k; d)$ *be the solution to the following linear programming problem: choose real numbers* $A_0, A_1, \ldots, A_k$ *so as to*

$$maximize \quad \sum_{i=0}^{k} A_i \tag{2.47}$$

subject to the constraints

$$A_0 = 1, \tag{2.48}$$

$$A_i = 0, i = 1, 2, \ldots, d-1, \tag{2.49}$$

$$A_i \geq 0, i = d, d+1, \ldots, k, \tag{2.50}$$

$$\sum_{j=0}^{k} A_j P_i(j) \geq 0, i = 0, 1, \ldots k. \tag{2.51}$$

Then the size N *of any* $(k, N, d)_2$ *code satisfies*

$$N \leq M_{LP}(k; d). \tag{2.52}$$

■

Theorem 2.4 and 2.5 have the drawback that one usually needs a computer to apply it, or to verify bounds that someone else has obtained from it. Furthermore, since the coefficients in (2.33) are large and alternate in sign, one must always worry about the reliability of the computer's answer.

The following Theorem 2.6 removes some of these drawback. In particular, bounds obtained from it can be verified with much less effort than those obtained from Theorem 2.4 and 2.5. (Instead of running the simplex algorithm, one need only check that certain numbers have the correct sign) We state this result for both OAs and codes.

Theorem 2.6 [7, Theorem 4.17] *(i) Orthogonal arrays. Suppose we can find a polynomial $f(x)$ of the form*

$$f(x) = 1 + \sum_{i=1}^{k} f_i P_i(x) \tag{2.53}$$

such that the following conditions hold:

$$f_i = 0 \text{ for } i = 1, 2, \ldots, t, \tag{2.54}$$

$$f_i \leq 0 \text{ for } i = t+1, t+2, \ldots, k, \tag{2.55}$$

$$f(j) \geq 0 \text{ for } j = 0, 1, \ldots, k. \tag{2.56}$$

Then the size of any $OA(N, k, 2, t)$ satisfies

$$N \geq f(0). \tag{2.57}$$

(ii) Codes. Suppose we can find a polynomial $f(x)$ of the form (2.53) such that the following conditions hold:

$$f_i \geq 0 \text{ for } i = 1, 2, \ldots, k, \tag{2.58}$$

$$f(j) \leq 0 \text{ for } j = d, d+1, \ldots, k. \tag{2.59}$$

Then the number of distinct codewords in any $(k, N, d)_2$ code satisfies

$$N \leq f(0). \tag{2.60}$$

■

Chapter 3

Orthogonal Arrays with Partial Strength

3.1 Introduction

In this Chapter 3, we introduce orthogonal arrays with partial strength (POAs), which are extended from OAs. POAs are more suitable for the application in experimental designs. Moreover, a subclass of POAs is related to UEP codes as proposed by Masnick et al. [12]. Therefore, some results in POAs can be applied to UEP codes.

In this chapter, we first define POAs and discuss the application of POAs in experimental designs. Next, we extend error-correcting codes and clarify the relation between POAs and the extended error-correcting codes. From the relations, we derive some properties of POAs, which are useful for leading to LP bounds for POAs. Next, we propose LP bounds for POAs and provide some numerical examples of these bounds. In Chapter 4, we provide details about the application of POAs to UEP codes.

This chapter is organized as follows. In Section 3.2, we define POAs and provide some basic properties of POAs. In Section 3.3, we discuss the application of POAs, especially in experimental designs. In Section 3.4, we extend error-correcting codes and clarify the relation between POAs and the extended error-correcting codes. Further, we derive some properties of POAs from the relation. In Section 3.5, construction problem for POAs and problem for finding lower bounds for POAs are formulated and we propose LP bounds for POAs to solve the problem for finding lower bounds for POAs. Moreover, we provide some numerical examples of the proposed bounds.

3.2 Orthogonal Arrays with Partial Strength

In this section, we define POAs and provide some basic properties of POAs. Let $v(\boldsymbol{a}) := \{i | a_i \neq 0\}$, where $\boldsymbol{a} = (a_1, a_2, \ldots, a_k)$. POAs are defined as follows.

Definition 3.1 Let k be any positive integer. Let $T \subseteq \{0,1\}^k$ and $T' = \{v(\boldsymbol{a}) | \boldsymbol{a} \in T\}$. Then an $N \times k$ array A with entries from $\{0,1\}$ is said to be an *Orthogonal Array with partial strength* T if the A has the following property; for any $\{i_1, i_2, \ldots, i_l\} \in T'$, $N \times l$ sub-array formed from the i_1, i_2, $\ldots$, i_l-th columns of A contains each l-tuple based on $\{0,1\}$ exactly same times as row. We will denote such an array by $POA(N, k, 2, T)$. ■

We note that an $POA(N, k, 2, T)$ is equal to an $OA(N, k, 2, t)$ if $T = \{\boldsymbol{a} \in \{0,1\}^k | w(\boldsymbol{a}) \leq t\}$.

Example 3.1 The array in Table 3.1 is an $POA(8, 4, 2, T)$, where

$$\begin{aligned} T = \{&0000000, 100000, 010000, 001000, 000100, 000010, 000001, 110000, 101000, \\ &100100, 100010, 100001, 011000, 010100, 010010, 010001, 001100, 001010, \\ &001001, 000110, 000101, 000011, 111000, 110100, 110010, 110001\}. \end{aligned} \tag{3.1}$$

■

Table 3.1: An OA with the strength T defined by (3.1)

0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

As well as OAs, if the rows of a $POA(N, k, 2, T)$ form a linear vector space, this POA is said to be linear and has a $(\log_2 N \times k)$ generator matrix.

Example 3.2 The POA given in Table 3.1 is linear and has the generator matrix

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

■

Moreover, the generator matrix of an $POA(N, k, 2, T)$ has the following properties.

Lemma 3.1 *Let A be an $N \times k$ linear array with 0,1 entries, and $G = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k]$ be a generator matrix of A. A is an $POA(N, k, 2, T)$ if and only if G satisfy*

$$\boldsymbol{g}_{i_1} + \boldsymbol{g}_{i_2} + \cdots + \boldsymbol{g}_{i_l} \neq \boldsymbol{0}, \tag{3.3}$$

for any $\{i_1, i_2, \ldots, i_l\} (\in \{v(\boldsymbol{a}) | \boldsymbol{a} \in T\})$. ■

The next Lemma gives a necessary and sufficient condition for an array to be an POA, which does not assume linearity.

Lemma 3.2 *An $N \times k$ array A with $0, 1$ entries is an $POA(N, k, 2, T)$ if and only if*

$$\sum_{\boldsymbol{v}=\text{row of } A} (-1)^{\boldsymbol{u}\cdot\boldsymbol{v}} = 0, \tag{3.4}$$

for all $\boldsymbol{u} \in T$, where the sum is over all rows $\boldsymbol{v}$ of A.

Proof: If A is an $POA(N, k, 2, T)$ then it is easy to see that (3.4) hold. Next, we will prove the converse statement. For any $\{i_1, i_2, \ldots, i_l\} \in T$, let $n(i_1, i_2, \ldots, i_l)$ denote the number of occurrences of the l-tuple $(i_1, i_2, \ldots i_l)$ in l columns under consideration, where each $i_r \in \{0, 1\}$. By choosing the vector $\boldsymbol{v}$ to have all possible 2^l different values in these l coordinates, and to be zero elsewhere, we obtain 2^l equations for the 2^l unknown $n(i_1, i_2, \ldots, i_l)$. If $\boldsymbol{v}$ identically zero the right-hand side of the equation is N, otherwise it is 0. Certainly setting all $n(i_1, i_2, \ldots, i_l)$ equal to $N/2^l$ is a solution. The coefficient matrix is the character table of elementary abelian group of type 2^l, which (by the orthogonality of characters) is an invertible matrix. Therefore the solution is unique, and the proof is complete. ■

3.3 Applications of Orthogonal Arrays with Partial Strength

We introduce POAs for the following reasons:

1. POAs are more suitable for experimental designs than OAs because POAs can be used with a complicated model.

2. A subclass of POAs is related to UEP codes.

We give detailed explanations about the first reason in the remain of this section and about the second reason in Chapter 4.

In Chapter 2, we showed that an $OA(M, n, 2, t)$ can be used if indexes of main and interactive factors in the model can be described by (2.14). For example, an $OA(M, 6, 2, 4)$ can be used if indexes of main and interactive factors in the model can be described by

$$I = \{\boldsymbol{e} \in \{0, 1\}^6 | w(\boldsymbol{e}) \leq 2\}, \tag{3.5}$$

$$\begin{aligned} = \{&000000, 100000, 010000, 001000, 000100, 000010, 000001, 110000, \\ &101000, 100100, 100010, 100001, 011000, 010100, 010010, \\ &010001, 001100, 001010, 001001, 000110, 000101, 000011, \}. \end{aligned} \tag{3.6}$$

However, more complicated interactive factors are often assumed in experimental designs, for example

$$I = \{000000, 100000, 010000, 001000, 000100, 000010, 000001, 110000\}. \tag{3.7}$$

This means that six main factors $F_1, F_2, \ldots, F_6$ and one interactive factor $F_1 \times F_2$ are included in the model. Of course, all effects of factors can be estimated by using an $OA(M, 6, 4, 2)$ in this case. However, by using an POA, all effects of factors can be estimated with fewer number of experiments.

Example 3.3 Suppose that six main factors $F_1, F_2, \ldots, F_6$ and one interactive factors $F_1 \times F_2$ are included in the model, that is the following model can be assumed.

$$y(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in I} f_{\boldsymbol{a}} \chi_{\boldsymbol{a}}(\boldsymbol{x}) + e(\boldsymbol{x}), \tag{3.8}$$

where

$$I = \{000000, 100000, 010000, 001000, 000100, 000010, 000001, 110000\}. \tag{3.9}$$

In this case, all effects of factors, $f_{\boldsymbol{a}}$, $\boldsymbol{a} \in I$, can be estimated by using the POA in Table 3.1. For example, $\hat{f}_{100000}$ is calculated by

$$\begin{aligned} \hat{f}_{100000} &= \frac{1}{2^3} \sum_{\boldsymbol{x} \in \{0,1\}^3} y(\boldsymbol{x}) \chi_{100000}(\boldsymbol{x}) && (3.10) \\ &= \frac{1}{8}\{y(000000) + y(001111) + y(010011) + y(011100) \\ &\quad -y(100101) - y(101010) - y(110110) - y(111001)\}, && (3.11) \end{aligned}$$

so $\hat{f}_{100}$ is as follows.

$$\begin{array}{r} f_{000000} + f_{100000} + f_{010000} + f_{001000} + f_{000100} + f_{000010} + f_{000001} + f_{110000} + e(000000), \\ f_{000000} + f_{100000} + f_{010000} - f_{001000} - f_{000100} - f_{000010} - f_{000001} + f_{110000} + e(001111), \\ f_{000000} + f_{100000} - f_{010000} + f_{001000} + f_{000100} - f_{000010} - f_{000001} - f_{110000} + e(010011), \\ f_{000000} + f_{100000} - f_{010000} - f_{001000} - f_{000100} + f_{000010} + f_{000001} - f_{110000} + e(011100), \\ -f_{000000} + f_{100000} - f_{010000} - f_{001000} + f_{000100} - f_{000010} + f_{000001} + f_{110000} + e(100101), \\ -f_{000000} + f_{100000} - f_{010000} + f_{001000} - f_{000100} + f_{000010} - f_{000001} + f_{110000} + e(101010), \\ -f_{000000} + f_{100000} + f_{010000} - f_{001000} + f_{000100} + f_{000010} - f_{000001} - f_{110000} + e(110110), \\ -f_{000000} + f_{100000} + f_{010000} + f_{001000} - f_{000100} - f_{000010} + f_{000001} - f_{110000} + e(111001), \\ \hline f_{100000} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad + \bar{e}_{100000}, \end{array}$$

where $\bar{e}_{100000} = \sum_{\boldsymbol{x} \in A''} e(\boldsymbol{x})$, and A'' is the design from the POA in Table 3.1. On the other hand, the best OA with the column number 6 and strength 4 is an $OA(32, 6, 2, 4)$ [7]. Thus, the number of experiments by the POA is fewer than OAs. ■

In general, the following can be said. Suppose that the following model can be assumed as the response variable.

$$y(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in I} f_{\boldsymbol{a}} \chi_{\boldsymbol{a}}(\boldsymbol{x}) + e(\boldsymbol{x}), \tag{3.12}$$

where the I only satisfy the monotonicity. Let $A''(\subseteq \{0,1\}^k)$ be a set whose elements are rows of an $POA(N,k,2,T)$, where $T(\subseteq \{0,1\}^k)$ satisfy

$$\{\boldsymbol{e}_1 \oplus \boldsymbol{e}_2 | \forall \boldsymbol{e}_1, \boldsymbol{e}_2 \in I\} \subseteq T. \tag{3.13}$$

In this case, if we experiment according to the design A'', we can obtain unbiased estimators of $f_{\boldsymbol{a}}$, $\boldsymbol{a} \in I$ using the following calculation:

$$\hat{f}_{\boldsymbol{a}} = \frac{1}{|A''|} \sum_{\boldsymbol{x} \in A''} y(\boldsymbol{x}) \chi_{\boldsymbol{a}}(\boldsymbol{x}). \tag{3.14}$$

3.4 Properties of Orthogonal Arrays with Partial Strength from Relations with Extended Error-Correcting Codes

In this section, we extend error-correcting codes and clarify the relation between POAs and the extended error-correcting codes. Further, we derive some properties of POAs from the relation. These properties are extensions of the results in Section 2.4.3 and are useful for leading to LP bounds for POAs.

3.4.1 Extension of Error-Correcting Codes

We extend $(k,N,d)_2$ codes and define extended codes as follows.

Definition 3.2 Let k be any positive integer and $D \subseteq \{0,1\}^k$. If a code $C \subseteq \{0,1\}^k$ satisfies

$$|C| = N, \tag{3.15}$$

$$\forall \boldsymbol{x}, \boldsymbol{y} \in C(\boldsymbol{x} \neq \boldsymbol{y}), \forall \boldsymbol{z} \in D, \quad \boldsymbol{x} \oplus \boldsymbol{y} \neq \boldsymbol{z}, \tag{3.16}$$

then C is called a $(k,N,D)_2$ *extended code.* ■

We note that a $(k,N,D)_2$ extended code is equal to a $(k,N,d)_2$ code if $D = \{\boldsymbol{a} \in \{0,1\}^k | w(\boldsymbol{a}) \leq d-1\}$.

Next, we introduce m split distance distribution, which is extended from the distance distribution. In order to introduce m split distance distribution, we define the following notation.

Definition 3.3 Let k, m, κ_1, κ_2, $\ldots$, κ_m be any positive integers, where $1 \leq m \leq k$ and $k = \kappa_1 + \kappa_2 + \cdots + \kappa_m$. Then $W^{(\kappa_1,\kappa_2,\ldots,\kappa_m)}_{i_1,i_2,\ldots,i_m} (= W_{i_1,i_2,\ldots,i_m})$, $i_j = 0,1,\ldots,\kappa_j$, $1 \leq j \leq m$ are defined by

$$W^{(\kappa_1,\kappa_2,\ldots,\kappa_m)}_{i_1,i_2,\ldots,i_m} := \{\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m) \in \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2} \times \cdots \times \{0,1\}^{\kappa_m} \\ | wt(\boldsymbol{w}_1) = i_1, wt(\boldsymbol{w}_2) = i_2, \ldots, wt(\boldsymbol{w}_m) = i_m\}. \tag{3.17}$$

If there is no danger of confusion we omit the $\kappa_1, \kappa_2, \ldots \kappa_m$. ■

Then the m split distance distribution is defined as follows.

Definition 3.4 Let k, m, κ_1, κ_2, $\ldots$, κ_m be any positive integers, where $1 \leq m \leq k$ and $k = \kappa_1 + \kappa_2 + \cdots + \kappa_m$. Then, the m *split distance distribution* $A_{i_1,i_2,\ldots,i_m}$, $i_j \in \{0, 1, \ldots, \kappa_j\}$, $1 \leq j \leq m$ *of a code* $C(\subseteq \{0,1\}^k)$ is defined by

$$A_{i_1,i_2,\ldots,i_m} = \frac{1}{|C|} \sum_{\boldsymbol{x} \in C} \left| \{\boldsymbol{y} \in C | \boldsymbol{x} \oplus \boldsymbol{y} \in W_{i_1,i_2,\ldots,i_m}\} \right|,$$
$$i_1 = 0, 1, \ldots, \kappa_1, i_2 = 0, 1, \ldots, \kappa_2, \ldots, i_m = 0, 1, \ldots, \kappa_m. \tag{3.18}$$

■

Like the distance distribution, if C is a linear code, the m split distance distribution $A_{i_1,i_2,\ldots,i_m}$ is equal to the m *split weight distribution* $A'_{i_1,i_2,\ldots,i_m}$, that is

$$A_{i_1,i_2,\ldots,i_m} = A'_{i_1,i_2,\ldots,i_m} \left(= \left| \{\boldsymbol{y} \in C | \boldsymbol{y} \in W_{i_1,i_2,\ldots,i_m}\} \right| \right),$$
$$i_1 = 0, 1, \ldots, \kappa_1, i_2 = 0, 1, \ldots, \kappa_2, \ldots, i_m = 0, 1, \ldots, \kappa_m. \tag{3.19}$$

Moreover, the $D(\subseteq \{0,1\}^k)$ of a (k, N, D) extended code is defined using k split distance distribution. If the k split distance distribution of a extended code C ($C \subseteq \{0,1\}^k$, $|C| = N$) satisfy

$$A_{i_1,i_2,\ldots,i_k} = 0 \text{ for } \forall (i_1, i_2, \ldots, i_k) \in D, \tag{3.20}$$

then C is $(k, N, D)_2$ extended code. Thus, D is called k split distance in what follows.

3.4.2 Extension of Delsarte Theorem

Next, we provide some properties of the m split distance distribution. The results in this section are extended from theorems in Section 2.4.2.

The next definition is extended from the Krawtchouk polynomial as defined in Definition 2.3.

Definition 3.5 For any positive integers κ_1, κ_2, $\ldots$,κ_m a polynomial $P_{i_1,i_2,\ldots,i_m}(z_1, z_2, \ldots, z_m; \kappa_1, \kappa_2, \ldots, \kappa_m)$ $(= P_{i_1,i_2,\ldots,i_m}(z_1, z_2, \ldots, z_m))$ is defined by

$$P_{i_1,i_2,\ldots,i_m}(z_1, z_2, \ldots, z_m; \kappa_1, \kappa_2, \ldots, \kappa_m) := P_{i_1}(z_1; \kappa_1) P_{i_2}(z_2; \kappa_2) \ldots P_{i_m}(z_m; \kappa_m),$$
$$i_1 = 0, 1, \ldots, \kappa_1, i_2 = 0, 1, \ldots, \kappa_2, \ldots, i_m = 0, 1, \ldots, \kappa_m. \tag{3.21}$$

where z_1, z_2, $\ldots$, z_m are indeterminate and $P_{i_1}(z_1; \kappa_1), P_{i_2}(z_2; \kappa_2), \ldots, P_{i_m}(z_m; \kappa_m)$ are the Krawtchouk polynomials. If there is no danger of confusion we omit the $\kappa_1, \kappa_2, \ldots, \kappa_m$. ■

Also, the extended Krawtchouk polynomial satisfy the following property, which corresponds to Lemma 2.3.

Lemma 3.3 *If* $\boldsymbol{v} \in W_{j_1,j_2,\ldots,j_m}$, *then*

$$\sum_{\boldsymbol{u}\in W_{i_1,i_2,\ldots i_m}} (-1)^{\boldsymbol{u}\cdot\boldsymbol{v}} = P_{i_1,i_2,\ldots,i_m}(j_1,j_2,\ldots,j_m), \tag{3.22}$$

where $i_1, j_1 \in \{0,1,\ldots,\kappa_1\}$, $i_2, j_2 \in \{0,1,\ldots,\kappa_2\}$, $\ldots$, $i_m, j_m \in \{0,1,\ldots,\kappa_m\}$.

Proof: Let $\boldsymbol{v} \in W_{j_1,j_2,\ldots,j_m}$. Then we can write $v = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m) \in W_{j_1}^{(\kappa_1)} \times W_{j_2}^{(\kappa_2)} \times \cdots \times W_{j_m}^{(\kappa_m)}$. Therefore,

$$\begin{aligned}
&\sum_{\boldsymbol{u}\in W_{i_1,i_2,\ldots,i_m}} (-1)^{\boldsymbol{u}\cdot\boldsymbol{v}} \\
&= \sum_{(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_m)\in W_{i_1}^{(\kappa_1)}\times W_{i_2}^{(\kappa_2)}\times\cdots\times W_{i_m}^{(\kappa_m)}} (-1)^{(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_m)\cdot(\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_m)} && (3.23)\\
&= \sum_{\boldsymbol{u}_1\in W_{i_1}^{(\kappa_1)}} \sum_{\boldsymbol{u}_2\in W_{i_2}^{(\kappa_2)}} \sum_{\boldsymbol{u}_m\in W_{i_m}^{(\kappa_m)}} (-1)^{(\boldsymbol{u}_1\cdot\boldsymbol{v}_1)\oplus(\boldsymbol{u}_2\cdot\boldsymbol{v}_2)\oplus\cdots\oplus(\boldsymbol{u}_m\cdot\boldsymbol{v}_m)} && (3.24)\\
&= \sum_{\boldsymbol{u}_1\in W_{i_1}^{(\kappa_1)}} (-1)^{\boldsymbol{u}_1\cdot\boldsymbol{v}_1} \sum_{\boldsymbol{u}_2\in W_{i_2}^{(\kappa_2)}} (-1)^{\boldsymbol{u}_2\cdot\boldsymbol{v}_2} \cdots \sum_{\boldsymbol{u}_m\in W_{i_m}^{(\kappa_m)}} (-1)^{\boldsymbol{u}_m\cdot\boldsymbol{v}_m} && (3.25)\\
&= P_{i_1}(j_1;\kappa_1)P_{i_2}(j_2;\kappa_2)\ldots P_{i_m}(j_m;\kappa_m) && (3.26)\\
&= P_{i_1,i_2,\ldots,i_m}(j_1,j_2,\ldots,j_m). && (3.27)
\end{aligned}$$

where (3.26) is from Lemma 2.3. ■

The next Theorem 3.1 is extended from Macwilliams Theorem as shown in Theorem 2.1. Theorem 3.1 provides an important property of the m split distance distribution.

Theorem 3.1 *Let* $C(\subseteq \{0,1\}^k)$ *be a linear code, and* $C^\perp$ *be the dual code of* C. *Let* $A_{i_1,i_2,\ldots,i_m}$ *and* $A^\perp_{i_1,i_2,\ldots,i_m}$, $i_j = 0,1,\ldots,\kappa_j$, $1 \le j \le m$, *be the* m *split distribution of* C *and* $C^\perp$. *Then,*

$$\begin{aligned}
A^\perp_{i_1,i_2,\ldots,i_m} = \frac{1}{|C|}\sum_{j_1=0}^{\kappa_1}\sum_{j_2=0}^{\kappa_2}\cdots\sum_{j_m=0}^{\kappa_m} A_{j_1,j_2,\ldots,j_m}P_{i_1,i_2,\ldots,i_m}(j_1,j_2,\ldots,j_m),\\
i_1 = 0,1,\ldots,\kappa_1, i_2 = 0,1,\ldots,\kappa_2,\ldots,i_m = 0,1,\ldots,\kappa_m. \qquad (3.28)
\end{aligned}$$

■

Before the proof of Theorem 3.1, we present the next Lemma 3.4. In Lemma 3.4, let f be any mapping defined on $\{0,1\}^n$, and $\hat{f}$ be the Hadamard transform of f, that is

$$\hat{f}(\boldsymbol{u}) = \sum_{\boldsymbol{v}\in\{0,1\}^n} (-1)^{\boldsymbol{u}\cdot\boldsymbol{v}} f(\boldsymbol{v}), \boldsymbol{u} \in \{0,1\}^n. \tag{3.29}$$

Lemma 3.4 [11, Ch.5 Lemma 2] *Let* C *be a linear code, and* $C^\perp$ *be the dual code of* C. *Then*

$$\sum_{\boldsymbol{u}\in C^\perp} f(\boldsymbol{u}) = \frac{1}{|C|}\sum_{\boldsymbol{u}\in C} \hat{f}(\boldsymbol{u}). \tag{3.30}$$

■

Proof of Theorem 3.1: Let the mapping $f_{W_{i_1,i_2,\ldots,i_m}}$ be

$$f_{W_{i_1,i_2,\ldots,i_m}}(\boldsymbol{u}) = \begin{cases} 0, & \boldsymbol{u} \notin W_{i_1,i_2,\ldots,i_m}, \\ 1, & \boldsymbol{u} \in W_{i_1,i_2,\ldots,i_m}, \end{cases}$$
$$i_1 = 0, 1, \ldots \kappa_1, i_2 = 0, 1, \ldots \kappa_2, \ldots, i_m = 0, 1, \ldots \kappa_m. \tag{3.31}$$

If (3.31) is used as the mapping f in (3.30), the left-hand side of (3.30) is

$$\sum_{\boldsymbol{u} \in C^\perp} f_{W_{i_1,i_2,\ldots,i_m}}(\boldsymbol{u}) = A^\perp_{i_1,i_2,\ldots,i_m}, \tag{3.32}$$

because $C^\perp$ is a linear code.

Further, the right-hand side of (3.30) is

$$\frac{1}{|C|} \sum_{\boldsymbol{u} \in C} \hat{f}_{W_{i_1,i_2,\ldots,i_m}}(\boldsymbol{u})$$
$$= \frac{1}{|C|} \sum_{\boldsymbol{u} \in C} \sum_{\boldsymbol{v} \in W_{i_1,i_2,\ldots,i_m}} (-1)^{\boldsymbol{u}\cdot\boldsymbol{v}} \tag{3.33}$$
$$= \frac{1}{|C|} \sum_{j_1=0}^{\kappa_1} \sum_{j_2=0}^{\kappa_2} \cdots \sum_{j_m=0}^{\kappa_m} \sum_{\boldsymbol{u} \in C \cap W_{j_1,j_2,\ldots,j_m}} \sum_{\boldsymbol{v} \in W_{i_1,i_2,\ldots,i_m}} (-1)^{\boldsymbol{u}\cdot\boldsymbol{v}} \tag{3.34}$$
$$= \frac{1}{|C|} \sum_{j_1=0}^{\kappa_1} \sum_{j_2=0}^{\kappa_2} \cdots \sum_{j_m=0}^{\kappa_m} A_{j_1,j_2,\ldots,j_m} P_{i_1,i_2,\ldots,i_m}(j_1, j_2, \ldots, j_m), \tag{3.35}$$

where (3.35) is from Lemma 3.3. Thus, we can obtain (3.28). ■

As well as MacWilliams Theorem, we use Theorem 3.1 to define m split dual distance of linear and nonlinear codes.

The next Theorem 3.2 is extended from Delsarte Theorem as shown in Theorem 2.2. Theorem 3.2 provides an important property of the m split distance distribution.

Theorem 3.2 *Let $C(\subseteq \{0,1\}^k)$ be a code and $A_{i_1,i_2,\ldots,i_m}$, $i_j = 0, 1, \ldots, \kappa_j$, $1 \le j \le m$, be the m split distribution of C. Then,*

$$\frac{1}{|C|} \sum_{j_1=0}^{\kappa_1} \sum_{j_2=0}^{\kappa_2} \cdots \sum_{j_m=0}^{\kappa_m} A_{j_1,j_2,\ldots,j_m} P_{i_1,i_2,\ldots,i_m}(j_1, j_2, \ldots, j_m) \ge 0,$$
$$i_1 = 0, 1, \ldots, \kappa_1, i_2 = 0, 1, \ldots, \kappa_2, \ldots, i_m = 0, 1, \ldots, \kappa_m. \tag{3.36}$$

Proof: For $i_1 = 0, 1, \ldots, \kappa_1, i_2 = 0, 1, \ldots, \kappa_2, \ldots, i_m = 0, 1, \ldots, \kappa_m$,

$$\frac{1}{|C|}\sum_{j_1=0}^{\kappa_1}\sum_{j_2=0}^{\kappa_2}\cdots\sum_{j_m=0}^{\kappa_m} A_{j_1,j_2,\ldots,j_m} P_{i_1,i_2,\ldots,i_m}(j_1, j_2, \ldots, j_m)$$

$$= \frac{1}{|C|^2}\sum_{j_1=0}^{\kappa_1}\sum_{j_2=0}^{\kappa_2}\cdots\sum_{j_m=0}^{\kappa_m} \sum_{\substack{(\boldsymbol{x},\boldsymbol{y})\in C^2,\\ \boldsymbol{x}\oplus\boldsymbol{y}\in W_{j_1,j_2,\ldots,j_m}}} P_{i_1,i_2,\ldots,i_m}(j_1, j_2, \ldots, j_m) \tag{3.37}$$

$$= \frac{1}{|C|^2}\sum_{j_1=0}^{\kappa_1}\sum_{j_2=0}^{\kappa_2}\cdots\sum_{j_m=0}^{\kappa_m} \sum_{\substack{(\boldsymbol{x},\boldsymbol{y})\in C^2,\\ \boldsymbol{x}\oplus\boldsymbol{y}\in W_{j_1,j_2,\ldots,j_m}}} \sum_{\boldsymbol{u}\in W_{i_1,i_2,\ldots,i_m}} (-1)^{\boldsymbol{u}\cdot(\boldsymbol{x}\oplus\boldsymbol{y})} \tag{3.38}$$

$$= \frac{1}{|C|^2}\sum_{\boldsymbol{x}\in C}\sum_{\boldsymbol{y}\in C}\sum_{\boldsymbol{u}\in W_{i_1,i_2,\ldots,i_m}} (-1)^{\boldsymbol{u}\cdot\boldsymbol{x}}(-1)^{\boldsymbol{u}\cdot\boldsymbol{y}} \tag{3.39}$$

$$= \sum_{\boldsymbol{u}\in W_{i_1,i_2,\ldots,i_m}} \left(\frac{1}{|C|}\sum_{\boldsymbol{x}\in C}(-1)^{\boldsymbol{u}\cdot\boldsymbol{x}}\right)^2 \tag{3.40}$$

$$\geq 0, \tag{3.41}$$

where (3.38) is from Lemma 3.3. ■

3.4.3 Relations between Orthogonal Arrays with Partial Strength and Extended Error-Correcting Codes

For a nonlinear, we still define the m split dual distance distribution by (3.28), calling the number $A^{\perp}_{i_1,i_2,\ldots,i_m}$, $i_j \in \{0, 1, \ldots, \kappa_j\}$, $1 \leq j \leq m$ the MacWilliams transform of the m split distance distribution. Then it is still true that $A^{\perp}_{i_1,i_2,\ldots,i_m} \geq 0$ for all $i_1, i_2, \ldots, i_m$ from (3.36). Here, we use the k split dual distance distribution for a code $C \subseteq \{0,1\}^k$, that is

$$A^{\perp}_{i_1,i_2,\ldots,i_k} = \frac{1}{|C|}\sum_{j_1=0}^{1}\sum_{j_2=0}^{1}\cdots\sum_{j_k=0}^{1} A_{j_1,j_2,\ldots,j_k} P_{i_1,i_2,\ldots,i_k}(j_1, j_2, \ldots, j_k),$$
$$i_j \in \{0,1\}, 1 \leq j \leq k. \tag{3.42}$$

Further, we define the k *split dual distance* to be a set $D^{\perp}$ such that

$$A^{\perp}_{i_1,i_2,\ldots,i_k} = 0, \text{ for } \forall(i_1, i_2, \ldots, i_k) \in D^{\perp}. \tag{3.43}$$

Theorem 3.3 *If C is a $(k, N, D)_2$ extended code over $\{0,1\}$ with the dual k split distance $D^{\perp}$, then the codewords of C form the rows of an $POA(N, k, 2, D^{\perp})$ with entries from $\{0,1\}$. Conversely, the rows of an $POA(N, k, 2, T)$ over $\{0,1\}$ form a $(k, N, D)_2$ extended code over $\{0,1\}$ with the dual k split distance T.*

Proof: If C has the dual k split distance $D^{\perp}$, then

$$A^{\perp}_{i_1,i_2,\ldots,i_k} = 0, \text{ for } \forall(i_1, i_2, \ldots, i_k) \in D^{\perp}. \tag{3.44}$$

From (3.40), this implies

$$\sum_{\boldsymbol{x}\in C}(-1)^{\boldsymbol{x}\cdot\boldsymbol{u}}=0 \tag{3.45}$$

for all $\boldsymbol{u}\in D$. Therefore, by Lemma 3.2, the matrix of codewords of C forms an orthogonal array with the partial strength $D^{\perp}$. Conversely, if the orthogonal array has the partial strength T, from Lemma 3.2 and (3.40) we have

$$A^{\perp}_{i_1,i_2,\ldots,i_k}=0,\ \text{for}\ \forall(i_1,i_2,\ldots,i_k)\in T. \tag{3.46}$$

■

3.4.4 Properties of Orthogonal Arrays with Partial Strength from Relations with Extended Error-Correcting Codes

From Theorem 3.3, a POAs can be regarded as an extended code. Thus, let $\bar{C}$ be a $POA(N,k,2,T)$ and C be an extended code formed by $\bar{C}$ in the same way as Theorem 3.3. Then, the k distance distribution $A_{i_1,i_2,\ldots,i_k}$, $(i_1,i_2,\ldots,i_k)\in\{0,1\}^k$ of the $\bar{C}$ is defined by (3.18), and the k split distance distribution satisfy

$$N=\sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\cdots\sum_{i_k=0}^{1}A_{i_1,i_2,\ldots,i_k}. \tag{3.47}$$

Moreover, the k split distance distribution of the $\bar{C}$ satisfy

$$A_{0,0,\ldots,0}=1, \tag{3.48}$$

$$A_{i_1,i_2,\ldots,i_k}\geq 0,(i_1,i_2,\ldots,i_k)\in\{0,1\}^k, \tag{3.49}$$

$$\sum_{j_1=0}^{1}\sum_{j_2=0}^{1}\cdots\sum_{j_k=0}^{1}A_{j_1,j_2,\ldots,j_k}P_{i_1,i_2,\ldots,i_k}(j_1,j_2,\ldots,j_k)\geq 0,(i_1,i_2,\ldots,i_k)\in\{0,1\}^k, \tag{3.50}$$

$$\sum_{j_1=0}^{1}\sum_{j_2=0}^{1}\cdots\sum_{j_k=0}^{1}A_{j_1,j_2,\ldots,j_k}P_{i_1,i_2,\ldots,i_k}(j_1,j_2,\ldots,j_k)=0,(i_1,i_2,\ldots i_k)\in T. \tag{3.51}$$

where (3.50) is from Theorem 3.2 and (3.51) is from Theorem 3.3. These properties are useful for leading to LP bounds for POAs in Section 3.5.

3.5 Main Problems in Orthogonal Arrays with Partial Strength

3.5.1 Main Problems in Orthogonal Arrays with Partial Strength

Construction problem for POAs and problem for finding lower bounds for POAs can be formulated as follows.

- Find the POA with a minimum number of rows N, given the number of columns k, and the partial strength $T(\subseteq \{0,1\}^k)$.
- Find the lower bound for a number of rows N, given the number of columns k, and the partial strength $T(\subseteq \{0,1\}^k)$.

As well as OAs, these are main problems in the study of POAs. In the past, many construct methods for POAs were already proposed [1, 3, 13, 22]. However, there are no studies for the problem for finding lower bounds for POAs. Therefore, we propose LP bounds for POAs, which are extended from LP bounds for OAs, as given in Section 2.5. We show the LP bounds for POAs in the next section 3.5.2.

3.5.2 Linear Programming Bounds for Orthogonal Arrays with Partial Strength

We propose LP bounds for POAs in the next Theorem 3.4.

Theorem 3.4 *For any $T \subseteq \{0,1\}^k$, let $N_{LP}(k;T)$ be the solution to the following linear programming problem: choose real numbers $A_{i_1,i_2,\ldots,i_k}$, $(i_1,i_2,\ldots,i_k) \in \{0,1\}^k$, so as to*

$$\text{minimize} \quad \sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\cdots\sum_{i_k=0}^{1} A_{i_1,i_2,\ldots,i_k} \tag{3.52}$$

subject to the constraints

$$A_{0,0,\ldots,0} = 1, \tag{3.53}$$

$$A_{i_1,i_2,\ldots,i_k} \geq 0, (i_1,i_2,\ldots,i_k) \in \{0,1\}^k, \tag{3.54}$$

$$\sum_{j_1=0}^{1}\sum_{j_2=0}^{1}\cdots\sum_{j_k=0}^{1} A_{j_1,j_2,\ldots,j_k} P_{i_1,i_2,\ldots,i_k}(j_1,j_2,\ldots,j_k) \geq 0, (i_1,i_2,\ldots,i_k) \in \{0,1\}^k, \tag{3.55}$$

$$\sum_{j_1=0}^{1}\sum_{j_2=0}^{1}\cdots\sum_{j_k=0}^{1} A_{j_1,j_2,\ldots,j_k} P_{i_1,i_2,\ldots,i_k}(j_1,j_2,\ldots,j_k) = 0, (i_1,i_2,\ldots i_k) \in T. \tag{3.56}$$

Then the size of any orthogonal array $POA(N,k,2,T)$ satisfies

$$N \geq N_{LP}(k;T). \tag{3.57}$$

■

We can obtain the LP bounds for POAs from the properties in Section 3.4.4. Further, we can obtain analogous result for extended codes as follows.

Theorem 3.5 *For any $D \subseteq \{0,1\}^k$, let $M_{LP}(k;D)$ be the solution to the following linear programming problem: choose real numbers $A_{i_1,i_2,\ldots,i_k}$, $i_j \in \{0,1\}$, $1 \leq j \leq k$, so as to*

$$\text{maximize} \quad \sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\cdots\sum_{i_k=0}^{1} A_{i_1,i_2,\ldots,i_k} \tag{3.58}$$

subject to the constraints

$$A_{0,0,\ldots,0} = 1, \tag{3.59}$$

$$A_{i_1,i_2,\ldots,i_k} = 0, (i_1, i_2, \ldots, i_k) \in D, \tag{3.60}$$

$$A_{i_1,i_2,\ldots,i_k} \geq 0, (i_1, i_2, \ldots, i_k) \notin D, \tag{3.61}$$

$$\sum_{j_1=0}^{1} \sum_{j_2=0}^{1} \cdots \sum_{j_k=0}^{1} A_{j_1,j_2,\ldots,j_k} P_{i_1,i_2,\ldots,i_k}(j_1, j_2, \ldots, j_k) \geq 0, (i_1, i_2, \ldots, i_k) \in \{0,1\}^k. \tag{3.62}$$

Then the size N of any $(k, N, D)_2$ code satisfies

$$N \leq M_{LP}(k; D). \tag{3.63}$$

■

Moreover, we can obtain the following Theorem as well as Theorem 2.6.

Theorem 3.6 *(i) Orthogonal arrays. Suppose we can find a polynomial $f(x_1, x_2, \ldots x_k)$ of the form*

$$f(x_1, x_2, \ldots, x_k) = 1 + \sum_{\substack{(i_1, i_2, \ldots, i_k) \in \{0,1\}^k, \\ (i_1, i_2, \ldots, i_k) \neq \mathbf{0}}} f_{i_1,i_2,\ldots,i_k} P_{i_1,i_2,\ldots,i_k}(x_1, x_2, \ldots, x_k) \tag{3.64}$$

such that the following conditions hold:

$$f_{i_1,i_2,\ldots,i_k} \leq 0 \text{ for } (i_1, i_2, \ldots, i_k) \notin T, \tag{3.65}$$

$$f(j_1, j_2, \ldots, j_k) \geq 0 \text{ for } (j_1, j_2, \ldots, j_k) \in \{0,1\}^k. \tag{3.66}$$

Then the size of any $POA(N, k, 2, T)$ satisfies

$$N \geq f(0, 0, \ldots, 0). \tag{3.67}$$

(ii) Codes. Suppose we can find a polynomial $f(x_1, x_2, \ldots, x_k)$ of the form (3.64) such that the following conditions hold:

$$f_{i_1,i_2,\ldots,i_k} \geq 0 \text{ for } (i_1, i_2, \ldots, i_k) \in \{0,1\}^k, \tag{3.68}$$

$$f(j_1, j_2, \ldots, j_k) \leq 0 \text{ for } (j_1, j_2, \ldots, j_k) \notin D. \tag{3.69}$$

Then the number of distinct codewords in any $(k, N, D)_2$ code satisfies

$$N \leq f(0, 0, \ldots, 0). \tag{3.70}$$

Proof: The dual linear program to the linear program defined by (3.52)-(3.56) is to choose real numbers $f_{i_1,i_2,\ldots,i_k}$, $(i_1, i_2, \ldots i_k) \in \{0,1\}^k$, $(i_1, i_2, \ldots i_k) \neq \mathbf{0}$ so as to

$$maximize \;\; 1 + \sum_{\substack{(i_1, i_2, \ldots, i_k) \in \{0,1\}^k, \\ (i_1, i_2, \ldots, i_k) \neq \mathbf{0}}} f_{i_1,i_2,\ldots,i_k} P_{i_1,i_2,\ldots,i_k}(0, 0, \ldots, 0) \tag{3.71}$$

subject to the constraints

$$f_{i_1,i_2,\ldots,i_k} \le 0 \text{ for } (i_1,i_2,\ldots,i_k) \notin T, \tag{3.72}$$

$$\sum_{\substack{(i_1,i_2,\ldots,i_k)\in\{0,1\}^k,\\ (i_1,i_2,\ldots,i_k)\neq \mathbf{0}}} f_{i_1,i_2,\ldots,i_k} P_{i_1,i_2,\ldots,i_k}(j_1,j_2,\ldots,j_k) \ge -1$$

$$\text{for } (j_1,j_2,\ldots,j_k)\in\{0,1\}^k. \tag{3.73}$$

It follows from the duality theorem of linear programming that any feasible solution to the dual problem gives a lower bound on the optimal solution to the primal problem. So if $f_{i_1,i_2,\ldots,i_k}$ satisfy (3.72) and (3.73), then

$$N_{LP} \ge 1 + \sum_{\substack{(i_1,i_2,\ldots,i_k)\in\{0,1\}^k,\\ (i_1,i_2,\ldots,i_k)\neq \mathbf{0}}} f_{i_1,i_2,\ldots,i_k} P_{i_1,i_2,\ldots,i_k}(0,0,\ldots,0) \tag{3.74}$$

With $f(x_1,x_2,\ldots,x_k)$ as in (3.64), the first assertion of the theorem follows. The proof for the codes case is analogous. ■

Next, we provide two examples to illustrate Theorem 3.4.

Example 3.4 We consider the case that that $k = 4$ and

$$T_1 = \{0000, 1000, 0100, 0010, 0001, 1100, 1010, 1001, 0110, 0101, 0011, 1110, 1101\}. \tag{3.75}$$

In this case, we can obtain the optimal solution $N_{LP}(k;T_1) = 8$ using a computer. In fact, the POA in Table 3.2 is an $POA(4,8,2,T_1)$, and the generator matrix of this POA is

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.76}$$

■

Table 3.2: An $POA(8,4,2,T_1)$

0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Example 3.5 We consider the case that that $k = 5$ and

$$\begin{aligned} T_2 = \{&00000, 10000, 01000, 00100, 00010, 00001, 11000, 10100, \\ &10010, 10001, 01100, 01010, 01001, 00110, 00101, 00011, \\ &11100, 11010, 11001, 10110, 01110, 00111, 11110\}. \end{aligned} \tag{3.77}$$

In this case, we can obtain the optimal solution $N_{LP}(k; T_2) = 16$ using a computer. In fact, the POA in Table 3.3 is an $POA(4, 8, 2, T_2)$, and the generator matrix of this POA is

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.78}$$

■

Table 3.3: An $POA(8, 4, 2, T_2)$

0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

3.6 Concluding Remarks

In this Chapter, we defined POAs and proposed LP bounds for POAs. Moreover, we provided numerical examples of the proposed bounds.

We first defined POAs and provided some basic properties of POAs. Next, we discussed the application of POAs, especially in experimental designs. Next, we defined extended error-correcting codes and clarified the relation between POAs and the extended error-correcting codes. Further, we derived some properties of POAs from the relation. In the derivation, we used the m split distance distribution and derived some properties of the m split distance distribution, which are extended from MacWilliams Theorem and Delsarte Theorem. Lastly, we proposed LP bounds for POAs and provided some numerical examples of the proposed bounds.

Chapter 4

A Subclass of Orthogonal Arrays with Partial Strength and its Applications to Unequal Error Protection Codes

4.1 Introduction

In Chapter 3, we proposed LP bounds for POAs. Consequently, the process of finding lower bounds for POAs reduces to solving LP problems. However, these LP problems has 2^k variables or constraints, so these LP problem cannot be solved if k is large. Therefore, it is important to consider the subclasses of the POAs such that the LP problems corresponding to the subclasses can be solved easily and such that the subclasses are important in applications involving experimental designs and error-correcting codes.

In this Chapter 4, we introduce *OAs with different strengths in each column* as a subclass of POAs. The LP problems corresponding to this subclass can be solved easily. Further, this subclass is important in applications in experimental designs and error-correcting codes. In particular, the application in error-correcting codes are important because this subclass is related to UEP codes [12, 2].

In this chapter, we first define OAs with different strengths in each column and propose its LP bounds. Next, we propose LP bounds for UEP code using the result. Next, we verify the effectiveness of the proposed LP bounds for UEP codes. More specifically, we compare the proposed bounds with the modified Hamming bound for UEP codes as proposed by Masnick et al. [12], and provide some numerical examples of the proposed bounds.

This chapter is organized as follows. In Section 4.2, we define OAs with different strengths in each column, and propose its LP bounds. In Section 4.3, we define UEP codes and propose LP bounds for UEP codes. In Section 4.4, we verify the effectiveness of the LP bounds for UEP codes.

4.2 A Subclass of Orthogonal Arrays with Partial Strength and its Linear Programming Bounds

At first, we define OAs with different strengths in each column as a subclass of POAs. The definition of OAs with different strengths in each column is as follows.

Definition 4.1 Let $m, k, \kappa_1, \kappa_2, \ldots, \kappa_m, t_1, t_2, \ldots, t_m$ be positive integers, where $k = \kappa_1 + \kappa_2 + \cdots + \kappa_m$. If an array A is an $POA(N, k, 2, T)$, where

$$\begin{aligned} T = \{\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m) \in \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2} \times \cdots \times \{0,1\}^{\kappa_m} | \\ (\boldsymbol{z}_1 \neq \boldsymbol{0}, w(\boldsymbol{z}) \leq t_1) \text{ or } (\boldsymbol{z}_2 \neq \boldsymbol{0}, w(\boldsymbol{z}) \leq t_2) \text{ or } \ldots \\ \text{or } (\boldsymbol{z}_m \neq \boldsymbol{0}, w(\boldsymbol{z}) \leq t_m)\}, \end{aligned} \tag{4.1}$$

then A is called an *OAs with different strengths in each column* and denoted by

$$POA(N, (\kappa_1, \kappa_2, \ldots, \kappa_m), 2, (t_1, t_2, \ldots, t_m)). \tag{4.2}$$

■

In applications in experimental designs, an $POA(N, (\kappa_1, \kappa_2, \ldots, \kappa_m), 2, (t_1, t_2, \ldots, t_m))$ can be used when the following model is assumed.

$$y(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in I} f_{\boldsymbol{a}} \chi_{\boldsymbol{a}}(\boldsymbol{x}) + e(\boldsymbol{x}), \tag{4.3}$$

where

$$\begin{aligned} I = \{\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m) \in \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2} \times \cdots \times \{0,1\}^{\kappa_m} | \\ (\boldsymbol{e} = \boldsymbol{0}) \text{ or } (\boldsymbol{e}_1 \neq \boldsymbol{0}, w(\boldsymbol{e}) \leq \lfloor \frac{t_1}{2} \rfloor) \text{ or } (\boldsymbol{e}_2 \neq \boldsymbol{0}, w(\boldsymbol{e}) \leq \lfloor \frac{t_2}{2} \rfloor) \\ \text{or } \ldots \text{ or } (\boldsymbol{e}_m \neq \boldsymbol{0}, w(\boldsymbol{e}) \leq \lfloor \frac{t_m}{2} \rfloor)\}. \end{aligned} \tag{4.4}$$

For example, if $m = 2$, $\kappa_1 = 1$, $\kappa_2 = 4$, $t_1 = 4$, and $t_2 = 2$, then

$$I = \{00000, 10000, 01000, 00100, 00010, 00001, 11000, 10100, 10010, 10001\}. \tag{4.5}$$

This means that there are five factors $F_1, F_2, \ldots F_5$ and all interactive factors of order two including F_1, that is $F_1 \times F_2$, $F_1 \times F_3$, $F_1 \times F_4$ and $F_1 \times F_5$.

Next, we propose LP bounds for $POA(N, (\kappa_1, \kappa_2, \ldots, \kappa_m), 2, (t_1, t_2, \ldots, t_m))$. In what follows, the notation $\mathcal{T}$ is defined by

$$\begin{aligned} \mathcal{T} := \{(i_1, i_2, \ldots, i_m) \in \{0, 1, \ldots, \kappa_1\} \times \{0, 1, \ldots, \kappa_2\} \times \ldots \{0, 1, \ldots, \kappa_m\} | \\ (i_1 \neq 0, \sum_{j=1}^{m} i_j \leq t_1) \text{ or } (i_2 \neq 0, \sum_{j=1}^{m} i_j \leq t_2) \text{ or } \\ \ldots \text{ or } (i_m \neq 0, \sum_{j=1}^{m} i_j \leq t_m)\}. \end{aligned} \tag{4.6}$$

Then, LP bounds for $POA(N, (\kappa_1, \kappa_2, \ldots, \kappa_m), 2, (t_1, t_2, \ldots, t_m))$ are as follows.

Theorem 4.1 *Let $N_{LP}(\kappa_1, \kappa_2, \ldots, \kappa_m; t_1, t_2, \ldots, t_m)$ be the solution to the following linear programming problem: choose real numbers $A_{i_1,i_2,\ldots,i_m}$, $i_j = 0, 1, \ldots, \kappa_j$, $1 \le j \le m$, so as to*

$$\textit{minimize} \quad \sum_{i_1=0}^{\kappa_1} \sum_{i_2=0}^{\kappa_2} \cdots \sum_{i_m=0}^{\kappa_m} A_{i_1,i_2,\ldots,i_m} \tag{4.7}$$

subject to the constraints

$$A_{0,0,\ldots,0} = 1, \tag{4.8}$$

$$A_{i_1,i_2,\ldots,i_m} \ge 0, i_j = 0, 1, 2, \ldots, \kappa_j, 1 \le j \le m, \tag{4.9}$$

$$\sum_{j_1=0}^{\kappa_1} \sum_{j_2=0}^{\kappa_2} \cdots \sum_{j_k=0}^{\kappa_m} A_{j_1,j_2,\ldots,j_m} P_{i_1,i_2,\ldots,i_m}(j_1, j_2, \ldots, j_m) \ge 0, i_j = 0, 1, \ldots, \kappa_j, 1 \le j \le m, \tag{4.10}$$

$$\sum_{j_1=0}^{\kappa_1} \sum_{j_2=0}^{\kappa_2} \cdots \sum_{j_k=0}^{\kappa_m} A_{j_1,j_2,\ldots,j_m} P_{i_1,i_2,\ldots,i_m}(j_1, j_2, \ldots, j_m) = 0, (i_1, i_2, \ldots i_m) \in \mathcal{T}. \tag{4.11}$$

Then the size N of any $POA(N, (\kappa_1, \kappa_2, \ldots, \kappa_m), 2, (t_1, t_2, \ldots, t_m))$ satisfies

$$N \ge N_{LP}(\kappa_1, \kappa_2, \ldots, \kappa_m; t_1, t_2, \ldots, t_m). \tag{4.12}$$

■

Note that LP problems in Theorem 4.1 has at most $\kappa_1 \times \kappa_2 \times \cdots \times \kappa_m$ variables and constraints, whereas LP problems in Theorem 3.4 has 2^k variables or constraints. Therefore, LP problems in Theorem 4.1 can be solved easier than those in Theorem 3.4.

4.3 An Application to Unequal Error-Protection Codes

Next, we apply the result in Section 4.2 to UEP codes and propose LP bounds for UEP codes.

UEP codes were proposed by Masnick et al. [12] and have been studied by many researchers [5, 6, 2]. UEP codes are divided to two types. The one is bit-wise UEP codes [12, 2], and the other is message-wise UEP codes [5, 6]. In this thesis, we focus on bit-wise UEP codes.

The definition of (bit-wise) UEP codes is as follows.

Definition 4.2 Let $m, k, \kappa_1, \kappa_2, \ldots, \kappa_m, d_1, d_2, \ldots, d_m$ be positive integers, where $k = \kappa_1 + \kappa_2 + \cdots + \kappa_m$. If a code C is a (k, N, D) extended code, where

$$\begin{aligned} D = \{\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m) \in \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2} \times \cdots \times \{0,1\}^{\kappa_m} | \\ (\boldsymbol{z}_1 \ne \boldsymbol{0}, w(\boldsymbol{z}) \le d_1 - 1) \text{ or } (\boldsymbol{z}_2 \ne \boldsymbol{0}, w(\boldsymbol{z}) \le d_2 - 1) \text{ or } \ldots \\ \text{or } (\boldsymbol{z}_m \ne \boldsymbol{0}, w(\boldsymbol{z}) \le d_m - 1)\}, \end{aligned} \tag{4.13}$$

then C is called a $((\kappa_1, \kappa_2, \ldots, \kappa_m), N, (d_1, d_2, \ldots, d_m))_2$ unequal error protection (UEP) code.

■

Next, we propose LP bounds for UEP codes. In what follows, the notation $\mathcal{D}$ is defined by

$$\begin{aligned}\mathcal{D} := \{(i_1, i_2, \ldots, i_m) \in \{0, 1, \ldots, \kappa_1\} \times \{0, 1, \ldots, \kappa_2\} \times \ldots \{0, 1, \ldots, \kappa_m\} | \\ (i_1 \neq 0, \sum_{j=1}^{m} i_j \leq d_1 - 1) \text{ or } (i_2 \neq 0, \sum_{j=1}^{m} i_j \leq d_2 - 1) \text{ or} \\ \ldots \text{ or } (i_m \neq 0, \sum_{j=1}^{m} i_j \leq d_m - 1)\}. \end{aligned} \tag{4.14}$$

UEP codes are corresponding to OAs with different strengths in each column. Therefore, we can obtain LP bounds for UEP codes in the same way as Theorem 4.1.

Theorem 4.2 *Let $M_{LP}(\kappa_1, \kappa_2, \ldots, \kappa_m; d_1, d_2, \ldots, d_m)$ be the solution to the following linear programming problem: choose real numbers $A_{i_1,i_2,\ldots,i_m}$, $i_j = 0, 1, \ldots, \kappa_j$, $1 \leq j \leq m$, so as to*

$$maximize \quad \sum_{i_1=0}^{\kappa_1} \sum_{i_2=0}^{\kappa_2} \cdots \sum_{i_m=0}^{\kappa_m} A_{i_1,i_2,\ldots,i_m} \tag{4.15}$$

subject to the constraints

$$A_{0,0,\ldots,0} = 1, \tag{4.16}$$

$$A_{i_1,i_2,\ldots,i_m} = 0, (i_1, i_2, \ldots, i_m) \in \mathcal{D}, \tag{4.17}$$

$$A_{i_1,i_2,\ldots,i_m} \geq 0, (i_1, i_2, \ldots, i_m) \notin \mathcal{D}, \tag{4.18}$$

$$\sum_{j_1=0}^{\kappa_1} \sum_{j_2=0}^{\kappa_2} \cdots \sum_{j_k=0}^{\kappa_m} A_{j_1,j_2,\ldots,j_m} P_{i_1,i_2,\ldots,i_m}(j_1, j_2, \ldots, j_m) \geq 0, i_j = 0, 1, \ldots, \kappa_j, 1 \leq j \leq m. \tag{4.19}$$

Then the size N of any $((\kappa_1, \kappa_2, \ldots, \kappa_m), N, (d_1, d_2, \ldots, d_m))_2$ UEP code satisfies

$$N \leq M_{LP}(\kappa_1, \kappa_2, \ldots, \kappa_m; d_1, d_2, \ldots, d_m). \tag{4.20}$$

■

4.4 Verification of Linear Programming Bounds for Unequal Error Protection Codes

In this section, we verify the effectiveness of the LP bounds for UEP codes as shown in Theorem 4.2. For this verification, we compare the LP bounds for UEP codes with the modified Hamming bounds and provide some numerical examples of the LP bounds for UEP codes.

4.4.1 Comparison with Modified Hamming Bounds

Next, we compare the LP bounds for UEP codes as shown in Theorem 4.2 with the modified Hamming bounds as proposed by Masnick et al. [12]. The modified Hamming bounds can be described as follows.

Theorem 4.3 *Let*

$$E = \{\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m) \in \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2} \times \cdots \times \{0,1\}^{\kappa_m} | (\boldsymbol{e} = \boldsymbol{0}) \text{ or } (\boldsymbol{e}_1 \neq \boldsymbol{0}, w(\boldsymbol{e}) \leq \lfloor \frac{d_1 - 1}{2} \rfloor) \text{ or } (\boldsymbol{e}_2 \neq \boldsymbol{0}, w(\boldsymbol{e}) \leq \lfloor \frac{d_2 - 1}{2} \rfloor) \text{ or } \ldots \text{ or } (\boldsymbol{e}_m \neq \boldsymbol{0}, w(\boldsymbol{e}) \leq \lfloor \frac{d_m - 1}{2} \rfloor)\}. \tag{4.21}$$

Then, any $((\kappa_1, \kappa_2, \ldots, \kappa_m), N, (d_1, d_2, \ldots, d_m))_2$ *UEP code satisfy*

$$N \leq \frac{2^{\kappa_1 + \kappa_2 + \cdots + \kappa_m}}{|E|}. \tag{4.22}$$

■

Further, the LP bounds for UEP codes and the modified Hamming bounds satisfy the following relation.

Theorem 4.4 *Let* $M_{LP}(\kappa_1, \kappa_2, \ldots, \kappa_m; d_1, d_2, \ldots, d_m)$ *be the solution to the LP problem in Theorem 4.2 and let* E *be defied by (4.21). Then*

$$M_{LP}(\kappa_1, \kappa_2, \ldots, \kappa_m; d_1, d_2, \ldots, d_m) \leq \frac{2^{\kappa_1 + \kappa_2 + \cdots + \kappa_m}}{|E|}. \tag{4.23}$$

■

Theorem 4.4 shows that the LP bounds for UEP codes are tighter than the modified Hamming bounds. We prove Theorem 4.4 in the next section 4.4.2.

4.4.2 Proof of Theorem 4.4

For simplicity, we prove Theorem 4.4 in the case $m = 2$. The generalization to m, $1 \leq m \leq k$ can be done easily.

In order to prove Theorem 4.4, we provide some lemmas. The next Lemma 4.1 and 4.2 provide properties of the Krawtchouk polynomial.

Lemma 4.1 [11, Ch.5 Theorem 16] *For any* $a, b \in \{0, 1, \ldots, k\}$,

$$\sum_{c=0}^{k} \begin{pmatrix} k \\ c \end{pmatrix} P_a(c) P_b(c) = 2^k \begin{pmatrix} k \\ a \end{pmatrix} \delta_{a,b}, \tag{4.24}$$

where $\delta_{a,b} = 1$, *if* $a = b$, $\delta_{a,b} = 0$, *if* $a \neq b$ *is the Kronecker symbol.* ■

Lemma 4.2 [11, Ch.5 Theorem 17] *For any* $a, b \in \{0, 1, \ldots, n\}$,

$$\begin{pmatrix} n \\ a \end{pmatrix} P_b(a) = \begin{pmatrix} n \\ b \end{pmatrix} P_a(b). \tag{4.25}$$

■

Moreover, Lemma 4.1 and 4.2 can be extended to Lemma 4.3 and 4.4.

Lemma 4.3 *For any* $a_1, b_1 \in \{0, 1, \ldots, \kappa_1\}$, $a_2, b_2 \in \{0, 1, \ldots, \kappa_2\}$,

$$\sum_{c_1=0}^{\kappa_1} \sum_{c_2=0}^{\kappa_2} \left|W_{c_1,c_2}\right| \; P_{a_1,a_2}(c_1, c_2) \; P_{b_1,b_2}(c_1, c_2) = 2^{\kappa_1+\kappa_2} \left|W_{a_1,a_2}\right| \; \delta_{a_1,b_1}\delta_{a_2,b_2}, \tag{4.26}$$

where δ_{a_1,b_1}, δ_{a_2,b_2} *are the Kronecker symbols.*

Proof: For any $a_1, b_1 \in \{0, 1, \ldots, \kappa_1\}$, $a_2, b_2 \in \{0, 1, \ldots, \kappa_2\}$,

$$\begin{aligned}
&\sum_{c_1=0}^{\kappa_1} \sum_{c_2=0}^{\kappa_2} \left|W_{c_1,c_2}\right| \; P_{a_1,a_2}(c_1, c_2) P_{b_1,b_2}(c_1, c_2) \\
&= \sum_{c_1=0}^{\kappa_1} \sum_{c_2=0}^{\kappa_2} \begin{pmatrix} \kappa_1 \\ c_1 \end{pmatrix} \begin{pmatrix} \kappa_2 \\ c_2 \end{pmatrix} P_{a_1}(c_1; \kappa_1) P_{a_2}(c_2; \kappa_2) && (4.27) \\
&= 2^{\kappa_1} \begin{pmatrix} \kappa_1 \\ a_1 \end{pmatrix} \delta_{a_1,b_1} \times 2^{\kappa_2} \begin{pmatrix} \kappa_2 \\ a_2 \end{pmatrix} \delta_{a_2,b_2} && (4.28) \\
&= 2^{\kappa} \left|W_{a_1,a_2}\right| \; \delta_{a_1,b_1}\delta_{a_2,b_2}, && (4.29)
\end{aligned}$$

where (4.28) is from Lemma 4.1. ■

Lemma 4.4 *For any* $a_1, b_1 \in \{0, 1, \ldots, \kappa_1\}$, $a_2, b_2 \in \{0, 1, \ldots, \kappa_2\}$,

$$\left|W_{b_1,b_2}\right| \; P_{a_1,a_2}(b_1, b_2) = \left|W_{a_1,a_2}\right| \; P_{b_1,b_2}(a_1, a_2). \tag{4.30}$$

Proof: For any $a_1, b_1 \in \{0, 1, \ldots, \kappa_1\}$, $a_2, b_2 \in \{0, 1, \ldots, \kappa_2\}$,

$$\begin{aligned}
&\left|W_{b_1,b_2}\right| \; P_{a_1,a_2}(b_1, b_2) \\
&= \begin{pmatrix} \kappa_1 \\ b_1 \end{pmatrix} \begin{pmatrix} \kappa_2 \\ b_2 \end{pmatrix} P_{a_1}(b_1; \kappa_1) P_{a_2}(b_2; \kappa_2) && (4.31) \\
&= \begin{pmatrix} \kappa_1 \\ a_1 \end{pmatrix} \begin{pmatrix} \kappa_2 \\ a_2 \end{pmatrix} P_{b_1}(a_1; \kappa_1) P_{b_2}(a_2; \kappa_2) && (4.32) \\
&= \left|W_{a_1,a_2}\right| \; P_{b_1,b_2}(a_1, a_2), && (4.33)
\end{aligned}$$

where (4.32) is from Lemma 4.2. ■

Furthermore, the Krawtchouk polynomial has the following properties.

Lemma 4.5 *If* $h + i < j$, *then*

$$\sum_{l=0}^{k} \begin{pmatrix} k \\ l \end{pmatrix} P_h(l) P_i(l) P_j(l) = 0. \tag{4.34}$$

Proof: The left-hand side of (4.34) is the coefficient of $x^h y^i z^j$ in

$$\sum_{l=0}^{n} \left(\begin{array}{c} k \\ l \end{array} \right) (1+x)^{k-l}(1-x)^l(1+y)^{k-l}(1-y)^l(1+z)^{k-l}(1-z)^l \tag{4.35}$$

$$= \{(1+x)(1+y)(1+z) + (1-x)(1-y)(1-z)\}^k \tag{4.36}$$

$$= 2^k(1+xy+yz+zx)^k, \tag{4.37}$$

where (4.36) is from binomial theorem. If $h+i<j$, the coefficient of $x^h y^i z^j$ in (4.37) is 0. Thus, we can obtain Eq. (4.34). ■

Next, we define the dual problem of the LP problem in Theorem 4.2, which is important to prove Theorem 4.4.

Problem 4.1 *Choose real numbers* α_{i_1,i_2}, $i_1 = 0,1,\ldots,\kappa_1$, $i_2 = 0,1,\ldots,\kappa_2$, *so as to*

$$minimize \quad \sum_{i_1=0}^{\kappa_1}\sum_{i_2=0}^{\kappa_2} \left(\begin{array}{c} \kappa_1 \\ i_1 \end{array} \right) \left(\begin{array}{c} \kappa_2 \\ i_2 \end{array} \right) \alpha_{i_1,i_2} \tag{4.38}$$

subject to the constraints

$$\alpha_{0,0} = 1, \tag{4.39}$$

$$\alpha_{i_1,i_2} \geq 0, i_1 = 0,1,\ldots,\kappa_1, i_2 = 0,1,\ldots,\kappa_2, \tag{4.40}$$

$$\sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_2} \alpha_{i_1,i_2} P_{i_1,i_2}(j_1,j_2) \leq 0, \forall (j_1,j_2) \in \{0,1,\ldots,\kappa_1\} \times \{0,1,\ldots,\kappa_2\} \setminus \mathcal{D}. \tag{4.41}$$

■

From the duality theorem of LP [11, Ch.17 Theorem 15, 16], it follows that any feasible solution to the dual problem gives an upper bound on the optimal solution to the primal problem. Therefore, we can prove Theorem 4.4 as follows.

Proof of Theorem 4.4: In what follows, let

$$\mathcal{E} := \{(i_1,i_2) \in \{0,1,\ldots,\kappa_1\} \times \{0,1,\ldots,\kappa_2\} | (i_1 \neq 0, i_1+i_2 \leq \lfloor \frac{d_1-1}{2} \rfloor) \text{ or } (i_1+i_2 \leq \lfloor \frac{d_2-1}{2} \rfloor)\}, \tag{4.42}$$

In Problem 4.1, let

$$\alpha_{i_1,i_2} = \left\{ \frac{\sum_{(a_1,a_2)\in\mathcal{E}} P_{a_1,a_2}(i_1,i_2)}{|E|} \right\}^2. \tag{4.43}$$

Then $\alpha_{0,0} = 1$, and $\alpha_{i_1,i_2} \geq 0$, $i_1 = 0,1,\ldots,\kappa_1$, $i_2 = 0,1,\ldots,\kappa_2$. Further, for any $(j_1,j_2) \in$

$\{0,1,\ldots,\kappa_1\}\times\{0,1,\ldots,\kappa_2\}\setminus\mathcal{D}$,

$$\sum_{i_1=0}^{\kappa_1}\sum_{i_2=0}^{\kappa_2}\left\{\frac{\sum_{(a_1,a_2)\in\mathcal{E}}P_{a_1,a_2}(i_1,i_2)}{|E|}\right\}^2 P_{i_1,i_2}(j_1,j_2)$$

$$=\frac{1}{|E|^2}\sum_{i_1=0}^{\kappa_1}\sum_{i_2=0}^{\kappa_2}\sum_{(a_1,a_2)\in\mathcal{E}}\sum_{(b_1,b_2)\in\mathcal{E}}P_{a_1,a_2}(i_1,i_2)P_{b_1,b_2}(i_1,i_2)P_{i_1,i_2}(j_1,j_2) \tag{4.44}$$

$$=\frac{1}{|E|^2\cdot|W_{j_1,j_2}|}\sum_{(a_1,a_2)\in\mathcal{E}}\sum_{(b_1,b_2)\in\mathcal{E}}\sum_{i_1=0}^{\kappa_1}\sum_{i_2=0}^{\kappa_2}\left|W_{i_1,i_2}\right|P_{a_1,a_2}(i_1,i_2)P_{b_1,b_2}(i_1,i_2)P_{j_1,j_2}(i_1,i_2) \tag{4.45}$$

$$=\frac{1}{|E|^2\cdot|W_{j_1,j_2}|}\sum_{(a_1,a_2)\in\mathcal{E}}\sum_{(b_1,b_2)\in\mathcal{E}}\left\{\sum_{i_1=0}^{\kappa_1}\left|W_{i_1}^{(\kappa_1)}\right|P_{a_1}(i_1;\kappa_1)P_{b_1}(i_1;\kappa_1)P_{j_1}(i_1;\kappa_1)\right\}$$
$$\times\left\{\sum_{i_2=0}^{\kappa_2}\left|W_{i_2}^{(\kappa_2)}\right|P_{a_2}(i_2;\kappa_2)P_{b_2}(i_2;\kappa_2)P_{j_2}(i_2;\kappa_2)\right\} \tag{4.46}$$

$$=0. \tag{4.47}$$

where (4.45) is from Lemma 4.4 and (4.47) is from (4.14), (4.42), and Lemma 4.5. Thus, we can see that (4.39)-(4.41) hold.

Therefore, the following is a feasible solution to Problem 4.1.

$$\sum_{i_1=0}^{\kappa_1}\sum_{i_2=0}^{\kappa_2}\binom{\kappa_1}{i_1}\binom{\kappa_2}{i_2}\left\{\frac{\sum_{(a_1,a_2)\in\mathcal{E}}P_{a_1,a_2}(i_1,i_2)}{|E|}\right\}^2$$

$$=\frac{1}{|E|^2}\sum_{i_1=0}^{\kappa_1}\sum_{i_2=0}^{\kappa_2}\binom{\kappa_1}{i_1}\binom{\kappa_2}{i_2}\sum_{(a_1,a_2)\in\mathcal{E}}\sum_{(b_1,b_2)\in\mathcal{E}}P_{a_1,a_2}(i_1,i_2)P_{b_1,b_2}(i_1,i_2) \tag{4.48}$$

$$=\frac{1}{|E|^2}\sum_{(a_1,a_2)\in\mathcal{E}}\sum_{(b_1,b_2)\in\mathcal{E}}2^{\kappa_1+\kappa_2}\binom{n_1}{a_1}\binom{n_2}{a_2}\delta_{a_1,b_1}\delta_{a_2,b_2} \tag{4.49}$$

$$=\frac{2^{\kappa_1+\kappa_2}}{|E|^2}\sum_{(a_1,a_2)\in\mathcal{E}}|W_{a_1,a_2}| \tag{4.50}$$

$$=\frac{2^{\kappa_1+\kappa_2}}{|E|}, \tag{4.51}$$

where (4.49) is from Lemma 4.3. We can obtain (4.23) because any feasible solution to the dual problem is upper bound on the optimal solution to the primal problem. ■

4.4.3 Numerical Examples

Next, we provide two examples to illustrate Theorem 4.2.

Example 4.1 We consider the case that $m=2$, $\kappa_1=6$, $\kappa_2=3$, $d_1=3$ and $d_2=5$. In this case, we can obtain the optimal solution $M_{LP}(6,3;3,5)=16$ using a computer. Moreover, this optimal solution has $A_{0,0}=A_{6,3}=1$, $A_{3,0}=A_{3,3}=4$, $A_{4,0}=A_{2,3}=3$ and the other $A_{i_1,i_2}=0$.

Table 4.1: $((\kappa_1, \kappa_2), M, (5, 3))$ UEP codes $(0 \leq \kappa_1, \kappa_2 \leq 15)$

κ_1	κ_2	LP bounds	Construction
0	15	2048.00	2048 (Hamming)
1	14	1024.00	
2	13	585.14	
3	12	585.14	
4	11	390.10	
5	10	390.10	
6	9	336.84	
7	8	318.58	
8	7	290.59	
9	6	280.70	
10	5	273.07	
11	4	263.49	
12	3	262.76	
13	2	260.06	
14	1	260.06	
15	0	260.06	256 (Preparata)

In fact, the UEP code with the parity check matrix

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \tag{4.52}$$

is a $((6, 3), 16, (3, 5))_2$ UEP code and has these 2 split distance distribution. ■

Example 4.2 We consider $((\kappa_1, \kappa_2), M, (5, 3))$ UEP codes, where $\kappa_1 + \kappa_2 = 15$ and $0 \leq \kappa_1 \leq 15$. Note that a $((0, 15), M, (5, 3))$ UEP code is a $(15, M, 3)$ code and a $((15, 0), M, (5, 3))$ UEP code is a $(15, M, 5)$ code. In this case, the optimal solutions $M_{LP}((\kappa_1, \kappa_2), (5, 3))$ obtained using a computer are in the column "LP bounds" of Table 4.1. In Table 4.1, the column "Construction" means the size of the best code with these parameters [11, p.675]. From Table 4.1, we can say that there can be useful UEP codes in the parameter $\kappa_1 = 1, 2, \ldots, 14$. ■

4.5 Concluding Remarks

In this Chapter, we defined OAs with different strengths in each column and proposed its LP bounds. Moreover, we proposed LP bounds for UEP codes and showed the effectiveness of the

LP bounds for UEP codes.

We first defined OAs with different strengths in each column as a subclass of POAs and proposed its LP bounds. Further, we showed that the LP problems corresponding to this subclass had few variables and constraints. Next, we proposed LP bounds for UEP codes. Lastly, we compared the LP bounds for UEP codes with the modified Hamming bounds and provided some numerical examples of the LP bounds for UEP codes.

Chapter 5

Construction of Orthogonal Arrays with Partial Strength from Unequal Error Protection Codes

5.1 Introduction

As stated in Chapter 3.5, main problems in the study of POAs are as follows.

- Find the POA with a minimum number of rows N, given the number of columns k and the partial strength $T(\subseteq \{0,1\}^k)$.
- Find the lower bound for a number of rows N, given the number of columns k and the partial strength $T(\subseteq \{0,1\}^k)$.

In previous works, many construction methods for POAs were proposed by researchers of experimental designs [1, 3, 13, 22]. Most of these methods are algorithmic and it is hard to construct POAs with a large number of columns and partial strength T whose size $|T|$ is large.

On the other hand, in the study of OAs, there are many construction methods for OAs [7]. Some of these methods use the relation with error-correcting codes in Theorem 2.3. Further, these methods can construct OAs with the large number columns and large strength easily.

In this Chapter 5, we propose construction methods for OAs with different strengths in each column, which was defined as a subclass of POAs in Chapter 4. We first propose two construction methods , construction method 1 and construction method 2, for linear OAs with different strengths in each column. These two methods use the relation between POAs and extended codes as shown in Theorem 3.3. Moreover, construction method 1 and construction method 2 use the construction methods for UEP codes as proposed by Masnick et al. [12] and Boyarinov et al. [2], respectively. Next, we propose construction method 3 for nonlinear OAs with different strengths in each column. The construction method 3 is an extension of the construction method 1. Lastly, we provide some numerical examples of these construction methods.

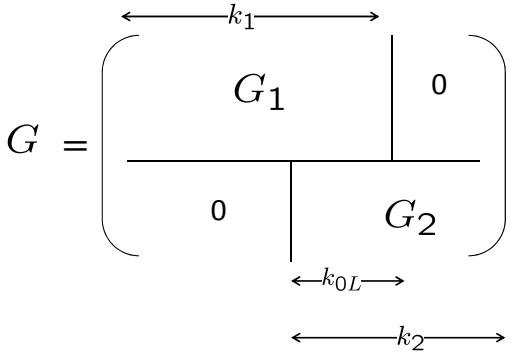

Figure 5.1: The construction method of linear OA

This chapter is organized as follows. In Section 5.2, we propose construction methods for linear and nonlinear OAs with different strengths in each column. In Section 5.3, we provide some numerical example of these proposed methods.

5.2 Construction from Unequal Error Protection Codes

5.2.1 Linear Orthogonal Arrays with Partial Strength

From Theorem 3.3, Definition 4.1, and Definition 4.2, we can make a linear $POA((\kappa_1, \kappa_2, \ldots \kappa_m), N, 2, (t_1, t_2, \ldots, t_m))$ from the dual code of a linear $((\kappa_1, \kappa_2, \ldots \kappa_m), N', (t_1+1, t_2+1, \ldots, t_m+1))_2$ UEP code directly. The next Construction Method 1 is from the construction method for UEP codes as proposed by Masnick et al. [12].

Construction Method 1 Let there be two generator matrices of OAs; G_1 is the generator matrix for a linear $OA(N_1, k_1, 2, t_1)$ and G_2 is the one for a linear $OA(N_2, k_2, 2, t_2)$, where $t_2 \leq t_1$. Let G_1 and G_2 be joined as sub-matrices of G where G_1 and G_2 overlap, as shown in Fig.5.1. The array with generator matrix G is an $N_1N_2 \times k_1 + k_2 - k_{0L}$ array. Let $k_{0L} \leq t_2/2$. ■

The array by Construction Method 1 satisfies the following theorem.

Theorem 5.1 *The array by Construction Method 1 is an* $POA(N_1N_2, (k_1-k_{0L}, k_{0L}, k_2-k_{0L}), 2, (t_1, t_1+t_2-k_{0L}, t_2))$. ■

Further, the next Construction Method 2 is from the construction method for UEP codes as proposed by Boyarinov et al. [2].

Construction Method 2 Let α denote a primitive element of the Galois field $GF(2^{2l})$. Then $\beta = \alpha^{2^l+1}$ is a primitive element of the Galois field $GF(2^l)$ that is a subfield of the Galois field $GF(2^{2l})$. Consider an array over $\{0, 1\}$, which have the generator matrix

$$G = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{2^l} & \alpha^{2^l+1} & \alpha^{2^l+2} & \cdots & \alpha^{2^{2l}-2} \\ 1 & 0 & \cdots & 0 & \beta^3 & 0 & \cdots & 0 \end{bmatrix}. \tag{5.1}$$

The array with the generator matrix G is a $2^{3l} \times (2^{2l} - 1)$ array. ■

The array by Construction Method 2 satisfies the following theorem.

Theorem 5.2 *Let l be an odd integer. Then, the array with the generator matrix in (5.1) is an $POA(2^{3m}, (2^m - 1, 2^{2m} - 2^m), 2, (4, 2))$.* ■

The statement of Theorem 5.2 allows for some modifications and generalizations in the same way as UEP codes (cf. [2, Theorem 2]).

5.2.2 Nonlinear Orthogonal Arrays with Partial Strength

Next, we propose a construction method for nonlinear $POA((\kappa_1, \kappa_2, \ldots \kappa_m), N, 2, (t_1, t_2, \ldots, t_m))$. The next Construction Method 3 is extended from Construction Method 1.

Construction Method 3 Let there be two OAs; $\bar{C}_1$ is an $OA(N_1, k_1, 2, t_1)$ and $\bar{C}_2$ is an $OA(N_2, k_2, 2, t_2)$, where $r_2 \leq r_1$. Note that $\bar{C}_1$ and $\bar{C}_2$ are not needed to be linear. Let C_1 be the set of the rows of $\bar{C}_1$ and C_2 be the set of the rows of $\bar{C}_2$. Let

$$C = \{(c_{1,1}, \ldots, c_{1,k_1-k_{0L}}, c_{1,k_1-k_{0L}+1} \oplus c_{2,1}, \ldots, c_{1,k_1} \oplus c_{2,k_{0L}}, c_{2,k_{0L}+1}, \ldots, c_{2,k_2}) \mid \text{for } \forall(c_{1,1}, c_{1,2}, \ldots, c_{1,k_1}) \in C_1, \forall(c_{2,1}, c_{2,2}, \ldots, c_{2,k_2}) \in C_2\}. \quad (5.2)$$

The OA whose rows are formed by the vectors in C is an $(N_1 N_2) \times (k_1 + k_2 - k_{0L})$ array. Let $k_{0L} \leq t_2/2$. ■

The array by Construction Method 3 satisfies the following theorem.

Theorem 5.3 *The array by Construction Method 3 is an $POA(N_1 N_2, (k_1 - k_{0L}, k_{0L}, k_2 - k_{0L}), 2, (t_1, t_1 + t_2 - k_{0L}, t_2))$.*

Proof: Let $N = N_1 N_2$ and $k = k_1 + k_2 - k_{0L}$. Further, let

$$C_1' = \{(a_{1,1}, a_{1,2}, \ldots, a_{1,k_1}, 0, 0, \ldots, 0) \in \{0,1\}^k | \forall(a_{1,1}, a_{1,2}, \ldots, a_{1,k_1}) \in C_1\}, \quad (5.3)$$

$$C_2' = \{(0, 0, \ldots, 0, a_{2,1}, a_{2,2}, \ldots, a_{2,k_2}) \in \{0,1\}^k | \forall(a_{2,1}, a_{2,2}, \ldots, a_{2,k_2}) \in C_2\}. \quad (5.4)$$

From Lemma 3.2, we should prove

$$\sum_{\boldsymbol{v} \in C} (-1)^{\boldsymbol{u} \cdot \boldsymbol{v}} = 0 \quad \text{for } \boldsymbol{u} \in T \setminus \{\boldsymbol{0}\}, \quad (5.5)$$

where

$$T = \{\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3) \in \{0,1\}^{k_1 - k_{0L}} \times \{0,1\}^{k_{0L}} \times \{0,1\}^{k_2 - k_{0L}} | (\boldsymbol{z} = \boldsymbol{0}) \text{ or } (\boldsymbol{z}_1 \neq \boldsymbol{0}, w(\boldsymbol{z}) \leq t_1) \text{ or } (\boldsymbol{z}_2 \neq \boldsymbol{0}, w(\boldsymbol{z}) \leq t_1 + t_2 - k_{0L}) \text{ or } (\boldsymbol{z}_3 \neq \boldsymbol{0}, w(\boldsymbol{z}) \leq t_2)\}. \quad (5.6)$$

Moreover,

$$\sum_{\boldsymbol{v}\in C}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v}}=0 \quad \text{for } \boldsymbol{u}\in T\setminus\{\boldsymbol{0}\} \tag{5.7}$$

$$\iff \quad \sum_{\boldsymbol{v}_1\in C_1'}\sum_{\boldsymbol{v}_2\in C_2'}(-1)^{\boldsymbol{u}\cdot(\boldsymbol{v}_1\oplus\boldsymbol{v}_2)}=0 \quad \text{for } \boldsymbol{u}\in T\setminus\{\boldsymbol{0}\} \tag{5.8}$$

$$\iff \quad \sum_{\boldsymbol{v}_1\in C_1'}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v}_1}\sum_{\boldsymbol{v}_2\in C_2'}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v}_2}=0 \quad \text{for } \boldsymbol{u}\in T\setminus\{\boldsymbol{0}\}, \tag{5.9}$$

so we will prove (5.9).

1. For the case of $\boldsymbol{u}\in\{\boldsymbol{z}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_3)|\boldsymbol{z}_1\neq\boldsymbol{0}, w(\boldsymbol{z})\leq t_1\}(\subseteq T\setminus\{\boldsymbol{0}\})$.

   Then

   $$\sum_{\boldsymbol{v}_1\in C_1'}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v_1}}=0, \tag{5.10}$$

   from (5.3) and Lemma (3.2). Thus (5.9) holds.

2. For the case of $\boldsymbol{u}\in\{\boldsymbol{z}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_3)|\boldsymbol{z}_3\neq\boldsymbol{0}, w(\boldsymbol{z})\leq t_2\}(\subseteq T\setminus\{\boldsymbol{0}\})$.

   In the same way as 1. ,

   $$\sum_{\boldsymbol{v}_1\in C_2'}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v_2}}=0. \tag{5.11}$$

   Thus (5.9) holds.

3. For the case of $\boldsymbol{u}\in\{\boldsymbol{z}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_3)|\boldsymbol{z}_2\neq\boldsymbol{0}, w(\boldsymbol{z})\leq t_1+t_2-k_{0L}\}(\subseteq T\setminus\{\boldsymbol{0}\})$.

   Let $\boldsymbol{u}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\boldsymbol{u}_3)\in\{0,1\}^{k_1-k_{0L}}\times\{0,1\}^{k_{0L}}\times\{0,1\}^{k_2-k_{0L}}$. If $w(\boldsymbol{u}_1,\boldsymbol{u}_2)\leq t_1$, then

   $$\sum_{\boldsymbol{v}_1\in C_1'}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v_1}}=0, \tag{5.12}$$

   from (5.3) and Lemma (3.2). Further, if $w(\boldsymbol{u}_1,\boldsymbol{u}_2)>t_1$, then

   $$w(\boldsymbol{u}_3)\leq t_2-k_{0L}\Rightarrow w(\boldsymbol{u}_2,\boldsymbol{u}_3)\leq t_2, \tag{5.13}$$

   so

   $$\sum_{\boldsymbol{v}_1\in C_2'}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v_2}}=0. \tag{5.14}$$

   from (5.4) and Lemma (3.2). Thus (5.9) holds.

■

5.3 Numerical Examples

5.3.1 Numerical Examples of Construction Method 1 and 3

In this section, we provide some examples of POAs by Construction Method 1 and 3. Further, we compare them with the optimal OAs [7, Table 12.1].

We first compare the following;

- An $OA(2^{14}, 16, 2, 8)$. This is an optimal $M \times 16$ OA with the strength 8, which is in [7, Table 12.1].
- An $POA(2^{13}, (8, 1, 7), 2, (5, 8, 4))$ by Construction Method 1: G_1 in Construction Method 1 is a generator matrix for an $OA(2^7, 9, 2, 5)$. This is an optimal OA with the number of column 9 and strength 5, which is in [7, Table 12.1]. G_2 is a generator matrix for an $OA(2^6, 8, 2, 4)$. This is also an optimal OA with the number of column 8 and strength 4. And $n_{0L} = 1$.

In this case, the number of rows of the POA by Construction Method 1 is fewer than the optimal OA.

Next, we compare the following arrays to discuss the differences between linear and nonlinear OAs with different strengths in each column.

- (OA) $OA(M, k, 2, 4)$, $k = 11, 12, \ldots, 32$. These are optimal OAs with strength 4, which is in [7, Table 12.1].
- (Method 1) $POA(M, (\kappa_1, 1, 2), 2, (3, 4, 2))$, $\kappa_1 = 8, 9, \ldots, 29$ by Construction Method 1: G_1 in Construction Method 1 are generator matrices for the optimal linear OAs with the number of column $\kappa_1 + 1$ and strength 3, which is in [7, Table 12.1], G_2 is the generator matrix for a linear $OA(4, 3, 2, 2)$, and $n_{0L} = 1$.
- (Method 3) $POA(M, (\kappa_1, 1, 2), 2, (3, 4, 2))$, $\kappa_1 = 8, 9, \ldots, 29$ by Construction Method 3: $\bar{C}_1$ in Construction Method 3 are the optimal linear or nonlinear OAs with the number of column $\kappa_1 + 1$ and strength 3, which is in [7, Table 12.1], $\bar{C}_2$ is a $OA(4, 3, 2, 2)$, and $n_{0L} = 1$.

The number of rows of each array is shown in Table 5.1, where $k = \kappa_1 + 1 + 2$ in Method 1 and Method 3.

We first compare the POA by Construction Method 1 with the OAs. We can see that the number of rows of the linear POAs is fewer than that of OAs at many k.

Next, we compare linear and nonlinear POAs by Construction Method 3 with linear POAs by Construction Method 1. The number of rows of POAs by Construction Method 3 is fewer than that of POAs by Construction Method 1. This is because Construction Method 3 is extended from Construction Method 1, so POAs by Construction Method 3 include POAs by Construction Method 1.

Table 5.1: The number of rows of OAs

k	OA	Method 1	Method 3
11	128	128	96
12	128	128	96
13	128	128	96
14	128	128	96
15	128	128	128
16	256	128	128
17	256	128	128
18	256	128	128
19	256	256	160
20	512	256	160
21	512	256	160
22	512	256	160
23	512	256	192
24	1024	256	192
25	1024	256	192
26	1024	256	192
27	1024	256	224
28	1024	256	224
29	1024	256	224
30	1024	256	224
31	1024	256	256
32	1024	256	256

5.3.2 Numerical Examples of the Construction Method 2

Next, we provide an example of an POA by Construction Method 2. Further, we compare it with an OA by using a BCH code [7, 11].

We compare the following arrays;

- The $OA(4096, 63, 2, 4)$, which has the generator matrix

$$G = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{2^l+1} & \cdots & \alpha^{2^{2l}-2} \\ 1 & \alpha^2 & \cdots & \alpha^{2^{l+1}+2} & \cdots & \alpha^{2^{2l+1}-4} \end{bmatrix}.$$

  where $l = 3$.

- The $POA(512, (56, 7), 2, (2, 4))$ by Construction Method 2, where let $l = 3$ in Construction Method 2.

In this case, the number of rows of the POA by Construction Method 2 is fewer than that of the OA.

5.4 Concluding Remarks

In this chapter, we proposed three construction methods for OAs with different strengths in each column. Further, we provided numerical examples of these construction methods.

We first proposed construction method 1 and construction method 2 for linear OAs with different strengths in each column. These methods used the relation between POAs and extended codes and used the construction methods for UEP codes. Next, we proposed construction method 3 for nonlinear OAs with different strengths in each column. This method is an extension of the construction method 1. Lastly, we provided some numerical examples of these three construction methods.

Chapter 6

Conclusion

In this thesis, we extended the concept of OAs to POAs and proposed LP bounds for POAs. We also defined OAs with different strengths in each column as a subclass of POAs and proposed its LP bounds. Moreover, we proposed LP bounds for UEP codes by using the results obtained for this subclass. We also proposed construction methods for OAs with different strengths in each column.

In Chapter 2, we discussed previous studies pertaining to OAs. In particular, we provided details about the applications of OAs in experimental designs, the relation between OAs and error-correcting codes, and the LP bounds for OAs.

In Chapter 3, we extended OAs to POAs and showed that POAs were more suitable for experimental design applications than OAs. We also defined extended codes and clarified the relation between POAs and extended codes. Additionally, we derived some properties of POAs from the relation. We further proposed LP bounds for POAs from the properties derived from the relation with the extended codes and presented some numerical examples of the LP bounds for POAs.

In Chapter 4, we defined OAs with different strengths in each column as a subclass of POAs and proposed their LP bounds. Then, we showed that the LP problems corresponding to this subclass can be solved more easily than those corresponding to POAs. Further, we proposed the LP bounds for UEP codes by using the results in this subclass. We also compare the LP bounds for UEP codes with the modified Hamming bounds and provided numerical examples of the LP bounds for UEP codes.

In Chapter 5, we initially proposed two construction methods, construction method 1 and construction method 2, for linear OAs with different strengths in each column. In both these methods, the relation between POAs and extended codes and the construction methods for UEP codes were used. We also proposed construction method 3 for nonlinear OAs with different strengths in each column; this method is an extension of construction method 1. We also provided some numerical examples for these three construction methods.

In the future, we plan to study other subclasses of POAs. In Chapter 4, we stated the importance of considering subclasses of POAs and introduced OAs with different strengths in each column as a subclass of POAs. We expect to find other subclasses of POAs for which the

LP problems corresponding to the subclasses can be solved easily and for which the subclasses are important in applications involving experimental designs or error-correcting codes.

We also plan to study other construction methods for POAs in the future. As stated in Chapter 5, many construction methods were proposed for POAs [1, 3, 13, 22]. Most of these methods can be regarded as search algorithms. Also, the lower bounds for POAs can be useful for narrowing the search ranges of these algorithms. Thus, we believe that we can find a new algorithm by combining the search algorithms and the proposed lower bounds (or properties of POAs to derive these bounds).

References

[1] G.E.P.Box, W.G.Hunter, and J.S.Hunter, Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building, John Wiley & Sons, 1978.

[2] I. M. Boyarinov, and G. L. Katsman, "Linear Unequal Error Protection Codes," *IEEE Trans. Inf. Theory*, Vol.IT-27, No.2, pp.168-175, March, 1981.

[3] R. Fuji-Hara, "On Automatical Construction for Orthogonal Designs of Experiments," *Rep. Stat. Appl. Res., JUES*, Vol.25, No.1, pp.13-25, 1978.

[4] P.Delsarte, "An algebraic approach to the association schemes of coding theory, " *Philips Res. Repts. Suppl.*, No.10, 1973.

[5] L. A. Dunning and W. E. Robbins, "Optimal encodings of linear block codes for unequal error protection, " *Inform. Contr.*, vol. 37, pp.150-177, 1978.

[6] W. J. van Gils, "Two topics on linear unequal error protection codes: bounds on their length and cyclic code classes," *IEEE Trans. Inf. Theory*, vol. IT-29, pp. 866-876, Sept. 1983.

[7] A. S. Hedayat, N. J. A. Sloane and J. Stufken, *Orthogonal Arrays: Theory and Applications*, Springer, New York, 1999.

[8] S. Hirasawa and T. Nishijima, *Introduction to coding theory* (in Japanese), Baifukan, 1999.

[9] W. Cary Huffman and Vera Pless, *Fundamentals of Error-Correcting Codes*, Cambiridge University Press, 2003.

[10] H. Imai, *Coding Theory* (in Japanese), IEICE, 1990.

[11] F. J. MacWilliams and N. J. A. Sloane, *The theory of Error-Correcting Codes*, Amsterdam: North-Holland Publishing Co., 1977.

[12] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Trans. Inform. Theory*, vol. IT-3, pp. 600-607, Oct. 1967.

[13] K. Suda, and H.Miyazaki, "An Algorithm Which Corresponds the Multi-Factors to Orthogonal Arrays in the Design of Orthogonal Experiments (in Japanese)," *Journal of Japan Industrial Management Association*, Vol.37, No.6, pp.345-352, 1987.

[14] T. Saito, T. Matsushima and S. Hirasawa, "A Note on Automatic Construction Algorithms for Orthogonal Designs of Experiments Using Error-Correcting Codes," *Journal of Discrete Mathematical Sciences & Cryptography*(to appear).

[15] T. Saito, Y. Ukita, T. Matsushima and S. Hirasawa, "Linear Programming Bounds of Orthogonal Arrays for Experimental Designs," *Proceedings of IEEE African Winter School on Information Theory and Communications 2010*, p.24, 2010.

[16] T. Saito, Y. Ukita, T. Matsushima and S. Hirasawa, "A Linear Programming Bound for Unequal Error Protection Codes," *Proceedings of the 2010 Australian Communications Theory Workshop*, pp.24-29, 2009.

[17] T. Saito, T. Matsushima and S. Hirasawa, "A Note on Automatic Construction Algorithms for Orthogonal Designs of Experiments Using Error-Correcting Codes," *Proceeding of Pre-ICM International Convention on Mathematical Sciences*, p.112, 2008.

[18] T. Saito, T. Matsushima and S. Hirasawa, "A Note on Construction of Orthogonal Arrays with Unequal Strength from Error-Correcting Codes," *IEICE Trans. Fundamentals*, Vol.E89-A, pp.1307-1315, May 2006.

[19] Y. Song, K. Kurosawa, S. Tsujii and T. Satoh, "Secret Sharing Schemes Based on Combinatorial Designs," (in Japanese), *IEICE Trans. Fundamentals*, Vol.J78-A, no.3, pp.401-406, Mar. 1995.

[20] D.R. Stinson, "Combinatorial Characterizations of Authentication Codes," *Designs, Codes and Cryptography*, 2, 175-187, 1992.

[21] D.R. Stinson, *Cryptography Theory and Practice*, Chapman & Hall/CRC, 2006.

[22] G. Taguchi, *The System of Experimental Design* (in Japanese), maruzen, 1976

[23] I. Takahasi, *Combinatorial Theory and its Application* (in Japanese), Iwanami Syoten, 1979.

[24] Y. Ukita, T. Saito, T. Matsushima and S. Hirasawa, "A Note on a Sampling Theorem for Functions over $GF(q)^n$ Domain," *IEICE Trans. Fundamentals*, vol.E93-A, no.6, pp.1024-1031, 2010-6.

[25] Y. Ukita, T. Saito, T. Matsushima and S. Hirasawa, "A Description of Experimental Design using an Orthonormal System," *Proceeding of 2010 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing*, pp.429-432, 2010.

[26] Y. Ukita, T. Saito, T. Matsushima and S. Hirasawa, "A Note on the Relation between a Sampling Theorem for Functions over a $GF(q)^n$ Domain and Linear Codes," *2009 IEEE International Conference on Systems, Man, and Cybernetics*, pp.2665-2670, 2009.

[27] Y. Ukita, T. Matsushima and S. Hirasawa, "A Note on Learning Boolean Functions Using Orthogonal Designs" (in Japanese), *IEICE Trans. Fundamentals*, Vol.J86-A, no.4, pp.482-490, Apr. 2003.

[28] Y.Washio, *Design and Analysis of Experiments* (in Japanese), Iwanami Syoten, 1988.

Publications

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on Automatic Construction Algorithms for Orthogonal Designs of Experiments Using Error-Correcting Codes," Journal of Discrete Mathematical Sciences & Cryptography(to appear).

Tomohiko Saito, Yoshifumi Ukita, Toshiyasu Matsushima and Shigeichi Hirasawa, "Linear Programming Bounds of Orthogonal Arrays for Experimental Designs," Proceedings of IEEE African Winter School on Information Theory and Communications 2010, p.24, 2010.

Yoshifumi Ukita, Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on a Sampling Theorem for Functions over $GF(q)^n$ Domain," IEICE Trans., vol.E93-A, no.6, pp.1024-1031, 2010-6.

Yoshifumi Ukita, Tomohiko Saito and Toshiyasu Matsushima, "A Description of Experimental Design using an Orthonormal System," Proceeding of 2010 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, pp.429-432, 2010.

Tomohiko Saito, Yoshifumi Ukita, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Linear Programming Bound for Unequal Error Protection Codes," Proceedings of the 2010 Australian Communications Theory Workshop, pp.24-29, 2009.

Yoshifumi Ukita, Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on the Relation between a Sampling Theorem for Functions over a $GF(q)^n$ Domain and Linear Codes," 2009 IEEE International Conference on Systems, Man, and Cybernetics, pp.2665-2670, 2009.

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on Automatic Construction Algorithms for Orthogonal Designs of Experiments Using Error-Correcting Codes," Proceeding of Pre-ICM International Convention on Mathematical Sciences, p.112, 2008.

Nobuhito Mikami, Tomohiko Saito and Toshiyasu Matsushima, "A Note on Construction of Pseudo-Random Number Generator for Stream Cipher, " IEICE Trans. Fundamentals (Japanese Edition), vol.J90-A, no.5, pp.470-476, 2007.

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Notes on Construction of Orthogonal Arrays with Unequal Strength from Error-Correcting Codes," IEICE Trans., vol.E89-A, no.5, pp.1307-1315, 2006-5.

Satoshi Hosobuchi, Tomohiko Saito and Toshiyasu Matsushima, "A Note on the Improvement of Fast Correlation Attack on Stream Ciphers," IEICE Trans. Fundamentals (Japanese Edition), vol.J89-A, no.2, pp.121-128, 2006.

Tomohiko Saito, Yoshifumi Ukita, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Linear Programming Bound for Unequal Error Protection Codes," Proceedings of 32th Symposium on Information Theory and its Applications (SITA2009), pp.359-364, 2009.

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on Automatic Construction Algorithms for Orthogonal Designs of Experiments Using Error-Correcting Codes," Proceedings of 31th Symposium on Information Theory and its Applications (SITA2008), pp.939-944, 2008.

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "On Factorial Effects Corresponding to Orthogonal Arrays with Unequal Strength," Proceedings of 2006 Hawaii, IEICE and SITA Joint Conference on Information Theory (HISC2006), pp.53-58, 2006.

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on Construction of Nonlinear Unequal Orthogonal Arrays from Error-Correcting Codes," Proceedings of 2005 Hawaii, IEICE and SITA Joint Conference on Information Theory (HISC2005), pp.13-18, 2005.

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on the Construction of Orthogonal Designs Using Error Correcting Codes (in Japanese)," Proceedings of 27th Symposium on Information Theory and its Applications (SITA2004), pp.463-466, 2004.

Tomohiko Saito, Toshiyasu Matsushima and Shigeichi Hirasawa, "A Note on the Construction of Orthogonal Designs by using the Construction of Error Correcting Codes (in Japanese)," Proceedings of 25th Symposium on Information Theory and its Applications (SITA2002), pp.663-666, 2002.

Daisuke Shisido, Tomohiko Saito and Toshiyasu Matsushima, "A Note on Robust Ramp Secret Sharing Schemes (in Japanese)," Proceedings of The 2007 Symposium on Cryptography and Information Security (SCIS2007), 2007.

Yuji Unagami, Tomohiko Saito and Toshiyasu Matsushima, "Non-Perfectly Secure Identity

Based Asymmetric Key Distribution Scheme (in Japanese)," Proceedings of The 2006 Symposium on Cryptography and Information Security (SCIS2006), 2006.

Satoshi Hosobuchi, Tomohiko Saito and Toshiyasu Matsushima, "A Note on the Improvement of a Fast Correlation Attack, " Proceedings of 27th Symposium on Information Theory and its Applications (SITA2004), pp.37-40, 2004.

Tomonari Shibuya, Tomohiko Saito and Toshiyasu Matsushima, "On the Security of Collision Free Hash Function Family," Proceedings of 26th Symposium on Information Theory and its Applications (SITA2003), pp.609-612, 2003.