OPTIMAL FACTORIAL DESIGNS WITH ROBUST PROPERTIES

by

Ryan Lekivetz

B.Sc., University of Regina, 2004

M.Sc., Simon Fraser University, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the Department
of
Statistics and Actuarial Science

SIMON FRASER UNIVERSITY
Summer 2011

APPROVAL

Name: Ryan Lekivetz

Degree: Doctor of Philosophy

Title of Thesis: Optimal Factorial Designs with Robust Properties

Examining Committee: Dr. Steve Thompson
Chair

____

Dr. Boxin Tang, Senior Supervisor

____

Dr. Derek Bingham, Supervisor

____

Dr. Tom Loughin, Supervisor

____

Dr. Tim Swartz, SFU Examiner

____

Dr. Julie Zhou, External Examiner,
University of Victoria

Date Approved: ____

Abstract

Fractional factorial designs are used in a wide variety of disciplines as a means of studying how changes in the settings of a set of factors influence a response variable. Two important considerations in choosing a fractional factorial design are identifying which effects can be jointly estimated and how the effects not estimated influence the estimation.

Orthogonal arrays with clear two-factor interactions provide a class of designs robust to nonnegligible effects. In the first part of this thesis, we introduce the concept of partially clear interactions which leads to a richer class of robust designs when specific interactions are known to be negligible a priori. We develop several methods to construct designs that allow for additional factors to be studied in comparison to designs with clear two-factor interactions. When used in conjunction with non-regular designs, the results become even more powerful as they provide additional flexibility and retain the robust properties.

In some situations, the experimenter would like to study factors at more than two levels, such as when curvature has the potential to occur within the experimental region. The second part of this thesis focuses on the estimation of main effects and specified interactions for designs with more than two levels. As designs with more than two levels have additional complications, results are provided that aid in the search for efficient designs that also have robust properties.

For two-level designs, the criteria of G and G_2-aberration are based on J-character-
istics and they provide measures of the projection properties of a design. For multi-level designs, extension to G_2 was previously done without the use of J-characteristics.

The J-characteristics for multi-level designs are introduced in the last part of this thesis as an intuitively appealing means to measure lower-dimensional properties, which leads to more natural definitions of G and G_2-aberration. We show how the properties of a design can be gleaned by using an analysis of variance as taught in introductory statistics courses.

Acknowledgments

I want to start by thanking my examining committee: Derek Bingham, Tom Loughin, Tim Swartz, and Julie Zhou. I am especially indebted to my supervisor, Boxin Tang, as without his patience and guidance this thesis would not be possible.

To my family: thank you for your support during all these years. I left Regina as a boy, and have now become... a boy with a PhD.

I will be forever grateful for my time spent at SFU. This is all due to the people I have met during my time here (although I will have fond memories of the daily trek up Cardiac Hill). From fellow graduate students, professors, and other acquaintances, there are far too many people who have left an impression on me during both my MSc and PhD to list, and out of fear of missing anyone's name, I will just say "thank you all." If you are reading this acknowledgement page, then chances are your name is probably on that list (even if your name happens to be Jervyn).

And of course CB, my best friend, thank you for all of your support in helping me make it through this.

Contents

List of Tables

Chapter 1

Introduction

1.1 Factorial Designs

Experimenters are often concerned with how changes of certain factors impact a process. If the experimenter has some control over the settings of these factors, then experimentation through changing these settings can aid in understanding the process. By using settings of the factors at a fixed number of values, referred to as levels, a factorial design is typically used as a plan to conduct such an experiment. Instead of focusing on one factor at a time, a factorial experiment varies the levels of these factors simultaneously. Depending on the different combinations of factors used in the experiment, not only can the experimenter study how the factors impact the response individually, but also their interaction. If enough attention is not placed in the design phase, the experimenter may not be able to meet the objectives of the experiment. A well-designed experiment can be analyzed in a valid and objective way, making it an important aspect of the experimental process.

Throughout this thesis, we assume that the experimental data can be adequately modeled by the general linear model. For a factorial design with n runs and m factors, the general linear model is written as

$$Y = X_0\beta_0 + X_1\beta_1 + X_2\beta_2 + \cdots + X_m\beta_m + \epsilon \tag{1.1}$$

where Y is the vector of n observations for the response, X_0 is the vector of 1's, β_0 the intercept, X_i is the matrix of covariate values for the i-factor interactions with β_i the corresponding effects, and ϵ the vector of independent random errors. For convenience, when discussing the use of (1.1), we assume the use of orthogonal contrasts and denote the effects as factorial effects throughout. This allows independent tests on the effects and estimates can still be related back to original values. We refer the reader to Chapter 1 of Wu and Hamada (2000) or an introductory text on design and analysis of experiments for more details.

Running all possible level combinations of the factors, called a full factorial design, allows for the estimation of all factorial effects. While this makes performing a full factorial design desirable, running all possible combinations of the factors may not be feasible for a variety of reasons including economic limitations, ethical concerns for certain combinations of factors, and situations where certain combinations do not make practical sense or are not possible to be run together. In such situations, a fractional factorial design (FFD) that uses a fraction of the runs of the full factorial design is frequently used. However, using a smaller subset of runs comes with a price; factorial effects become *aliased*. In *regular* designs, those determined by a defining relation, factorial effects are either orthogonal or fully aliased. If a design does not have the aforementioned property we refer to the design as *non-regular*. The question then becomes how to determine the desirability of a design based on the nature of this aliasing.

1.2 Choosing Fractional Factorial Designs

While the appropriateness of a design is ultimately based on the experimental objectives, there are some working assumptions that are generally accounted for in choosing designs. These are based on empirical evidence and play an important role in ranking designs and analyzing results. Three fundamental principles for factorial effects in fractional factorial designs (Wu and Hamada, 2000) are:

1. **Hierarchical Ordering Principle**: (i) Lower order effects are more likely to be important than higher order effects, and;

(ii) effects of the same order are equally likely to be important.

2. **Effect Sparsity Principle**: The number of relatively important effects in a factorial experiment is small (Box and Meyer, 1986).

3. **Effect Heredity Principle**: In order for an interaction to be active, at least one of its parent factors should be active (Hamada and Wu, 1992).

Using these principles, FFDs can be constructed and ranked. Following the hierarchical ordering principle, the main effects are the most important factorial effects, followed by two-factor interactions, three-factor interactions, etc. In the ANOVA model (1.1), these apply to β_1, β_2, $\beta_3, \ldots$. With this ranking of effects in mind, Box and Hunter (1961) introduced the criterion of maximum resolution for regular designs to measure the aliasing between lower order effects. As a means of further distinguishing between designs having the same resolution, Fries and Hunter (1980) proposed the minimum aberration criterion. These ideas were generalized to non-regular two-level designs by Deng and Tang (1999) and Tang and Deng (1999) through G_2 and G-aberration. Generalized minimum aberration was derived for multi-level designs by Xu and Wu (2001). The notion of G_2 and G-aberration for multi-level designs will be returned to later in this thesis. We now discuss other considerations that may arise in choosing an appropriate design for an experiment.

1.2.1 Requirement Sets

By the hierarchical ordering principle, effects of the same order are equally likely to be important. However, there are situations where a certain subset of effects are of interest to the experimenter. For example, the experimenter's interest may be in the estimation of the main effects and a specified set of two-factor interactions. Considering a specific subset of interactions is important from a practical standpoint, but breaks the hierarchical ordering principle, meaning maximum resolution and minimum aberration designs will no longer be the best choice for these cases. The goal is then to find designs that allow estimation of the effects of interest, if such designs exist. This idea goes back to Addelman (1962). Franklin and Bailey (1977) presented an algorithm for generating designs with a requirement set for two levels, and was extended by Franklin (1985) additional levels. In this thesis we call this set of effects a *requirement set*, as done by Greenfield (1976). If we are able to find more than one design that allows for estimation of the requirement set, we can differentiate between these designs using another criterion. Popular examples of these include D-optimality, which minimizes the volume of the confidence ellipsoid for the set of factorial effects, and A-optimality, which minimizes the sum of the variances of the regression coefficients.

1.2.2 Robustness to Nonnegligible Effects

If we can find a design that can estimate the requirement set, the estimates may be aliased with effects that are not estimated. If the effects not estimated are nonnegligible, this aliasing should be accounted for. For example, if the experimenter is interested in estimating the main effects of the factors, they may fit the main effects model given by

$$Y = X_0\beta_0 + X_1\beta_1 + \epsilon. \tag{1.2}$$

Using model (1.2), the least squares estimate of β_1 is calculated as $\hat{\beta}_1 = (X_1^T X_1)^{-1} X_1^T Y$. This estimate is unbiased for model (1.2), but if model (1.1) is the true model, we have

$$E(\hat{\beta}_1) = \beta_1 + C_2\beta_2 + \cdots + C_m\beta_m, \tag{1.3}$$

where the $C_i = (X_1^T X_1)^{-1} X_1^T X_i$ are called the aliasing matrices for $i = 2, \ldots, m$. If elements of β_i are nonnegligible, they can introduce bias on the estimation of β_1 as seen in (1.3). We will see later that, for a requirement set with some two-factor interactions, replacing X_1 in (1.2) by the model matrix for the requirement set can be easily adapted to (1.2) and (1.3).

This examination of the bias dates back to Box and Draper (1959) for response surface designs. If we are only interested in the main effects, the ranking of designs can be done with G and G_2-aberration. When certain interactions are to be estimated, a different tactic needs to be taken in regards to bias. One approach is to treat all those effects not estimated as negligible, but this may not be a reasonable assumption.

Another typical approach is to treat higher-order interactions as negligible. This can still leave nonnegligible effects that are not estimated. If the effects that are estimated are aliased with these effects, we get biased estimates. To prevent this, we want a design that is robust to nonnegligible effects. Wu and Chen (1992) defined *clear* two-factor interactions as those which are orthogonal to all main effects and other two-factor interactions. Clear two-factor interactions are valuable because their estimates are robust to nonnegligible effects if interactions involving three or more factors are negligible.

One of the themes of this thesis is combining the requirement set problem with robustness to nonnegligible interactions. In one case, robustness will be achieved through prior information about some of the effects to ensure robustness to nonnegligible effects. In another, we look to efficient estimation of the requirement set firstly, and then try to establish robustness properties. Related work for two-level designs is seen in Ke and Tang (2003) and Wu (2009), and by Jones and Nachstheim (2011) for three-level designs.

1.2.3 Screening Experiments and Response Surface Exploration

Much of the discussion to this point has been concerned with a specific set of effects to be estimated. Without a requirement set, FFDs can be used for factor *screening*, with the intention of identifying a subset of the factors that have a large influence on the response. More detailed experimentation can then be performed on this smaller subset of factors. By the effect sparsity principle, this subset is expected to be small in size. If there is the possibility that one or more of the lower order interactions are significant, ideally we should be able to entertain as many of the smaller subsets that include some interactions as possible. For regular designs, the nature of the aliasing is well-defined, but leaves some effects indistinguishable from each other. Non-regular designs have a complex aliasing pattern, but have the potential to allow for the estimation of a greater number of models involving smaller subsets of factors. Some metrics for choosing designs for this purpose include estimation capacity (Sun, 1993 and Cheng, Steinberg and Sun, 1999), projection estimation capacity (Loeppky, Sitter and Tang, 2007), and average D-efficiency (Cheng, Deng and Tang, 2002).

In non-regular designs, complex aliasing can make it difficult to discriminate between sets of models. Some proposed solutions to identifying subsets of factors influencing the response include Hamada and Wu (1992), Chipman, Hamada, and Wu (1997) and Wolters and Bingham (2011). If the purpose of the experiment is to maximize or minimize a response, one strategy is to use response surface methodology on this smaller subset of factors. Response surface methodology is a two-phase process which involves using first order models to find an experimental region near the optimum and fit a second-order model that allows for curvature within that region. In Chapter 3 we will consider an alternative to standard response surface methodology, when estimating curvature is desirable but not follow-up experimentation.

1.3 Orthogonal Arrays

The fractional factorial designs considered in this thesis are orthogonal arrays. Orthogonal arrays date back to the 1940's to Rao (1947) and Plackett and Burman (1946) and form the basis for most factorial designs. An orthogonal array $OA(n, s_1, \ldots, s_m, t)$ is an $n \times m$ matrix in which column i has s_i levels for $i = 1, \ldots, m$, such that for any subset of t columns all possible level combinations occur equally often. In the experimental sense, the n rows can be thought of as the experimental runs and each column represents the settings for a factor. We refer to t as the *strength* of the orthogonal array. The strength of an orthogonal array is related to the estimability of the interaction terms in the linear regression model. The higher the strength of an orthogonal array, the more interaction terms can be estimated independently of each other. In terms of clear effects, for an orthogonal array of strength 2, the main effects and clear two-factor interactions can be estimated with the clear two-factor interactions robust to nonnegligible two-factor interactions, under the assumption that interactions involving three or more factors are negligible. Under the same assumption with interactions involving three or more factors being negligible, both the main effects and clear two-factor interactions are estimable and robust to nonnegligible two-factor interactions in an orthogonal array of strength 3. Unfortunately, the higher the strength of an orthogonal array for a fixed set of factors, the larger the run size needs to be. Similarly, for a fixed run size, the higher strength implies fewer factors can be considered.

Regular designs presented in introductory design of experiments textbooks are a special case of orthogonal arrays. With the inclusion of non-regular designs, orthogonal arrays provide a much richer class of designs to consider in terms of run size economy and flexibility, while ensuring the main effect estimates are mutually orthogonal. Throughout this thesis, we will make use of catalogs such as those in Chen, Sun and Wu (1993) and Evangelaras, Koukouvinos and Lappas (2011) in searching for designs for different experimental objectives. An extensive look at orthogonal arrays is done in Hedayat, Sloane, and Stufken (1999). More recent work on designs using orthogonal arrays appears throughout this thesis.

1.4 Outline

We now give an outline for the remainder of the thesis with a brief description of each chapter and motivation behind it.

Orthogonal arrays with clear two-factor interactions provide a class of designs that are robust to nonnegligible effects. If certain prior knowledge is available, then robust designs may allow additional factors to be studied. This is done through partially clear two-factor interactions. In Chapter 2 we investigate the existence and construction of such robust designs and present an upper bound on the maximum number of clear two-factor interactions.

While two-level designs allow the estimation of linear effects, the experimenter may require the use of more than two levels per factor. The ideas of a requirement set and influence of those effects not estimated become more complicated when factors are studied at more than two levels. Chapter 3 studies the problem of requirement sets for factors with more than two levels and robustness to those effects not estimated.

Chapter 4 considers finding designs with desirable projection properties. In an orthogonal array of strength t, in any subset of t columns, all level permutations occur equally often. If strength t is not attainable, ideally the lower-dimensional projections of a design should resemble this property for smaller values of t. With this idea, we introduce the J-characteristics for multi-level designs which are based on the frequency of design points and can be used to examine lower dimensions. We will use these J-characteristics to rank designs.

The thesis is concluded in Chapter 5 with a summary of Chapters 2, 3 and 4 and a general discussion on possible future work.

Chapter 2

Robust Designs Through Partially Clear Two-Factor Interactions

2.1 Introduction

In this chapter, we consider factorial designs with two levels, although the main ideas are generally applicable. Two-level factorial designs are widely used for screening experiments in industrial applications. Two-level orthogonal arrays enjoy run size economy and flexibility beyond those of regular factorial designs.

Orthogonal arrays with *clear* two-factor interactions are robust to nonnegligible two-factor interactions. In an orthogonal array, a two-factor interaction (2fi) is said to be clear if it is orthogonal to all main effects and all other two-factor interactions (2fi's). Under the assumption that interactions containing three or more factors are negligible, an orthogonal array of strength 2 allows estimation of clear 2fi's even if the other 2fi's are nonnegligible, although main effects may be aliased with some 2fi's. Under the same assumption regarding interactions of three or more factors, if an orthogonal array is of strength 3, main effects and clear 2fi's can all be estimated regardless of the nonnegligible 2fi's.

Clear 2fi's were introduced by Wu and Chen (1992) and their existence was examined by Chen and Hedayat (1998). Cheng, Steinberg and Sun (1999) and Wu and Wu (2002) explored the relationship between the criterion of minimum aberration and that of the maximum number of clear 2fi's. Other work on clear 2fi's includes Tang, Ma, Ingram and Wang (2002), Ke, Tang and Wu (2005), Chen, Li, Liu, and Zhang (2006), and Lisonek (2006). The concept of clear 2fi's was generalized to orthogonal arrays by Tang (2006).

In this chapter, instead of treating all 2fi's that are not estimated as either negligible or nonnegligible, we consider the situation in which some 2fi's are assumed to be negligible while others are not. Assuming all 2fi's that are not estimated are negligible such as in Addelman (1962), Greenfield (1976) and Sun (1993) may be too strong whereas assuming all are nonnegligible may be too restrictive. The assumption that some 2fi's are negligible leads to a class of robust designs that allow more factors to be studied as compared to designs with clear 2fi's. They are obtained using the concept of *partially clear* 2fi's. Partially clear 2fi's are orthogonal to nonnegligible 2fi's but allowed to be aliased with negligible 2fi's.

Section 2.2 introduces basic concepts and presents some preliminary results. Section 2.3 establishes several theoretical results for the construction of robust designs and Section 2.4 provides tables of robust designs for 32 and 64 runs. We conclude this chapter with a result on the maximum number of clear 2fi's in an orthogonal array in Section 2.5.

2.2 Concepts and Preliminary Results

A two-level fractional factorial design D of n runs for m factors is represented by an $n \times m$ matrix of ± 1. Design D is a two-level orthogonal array of strength t if, in every $n \times t$ submatrix of D, each of 2^t level combinations appears with the same frequency. We denote such an orthogonal array by $OA(n, 2^m, t)$. The run size n must be a multiple of 4 for $t = 2$, and a multiple of 8 for $t = 3$. Orthogonal arrays of strength 2 allow orthogonal estimation of all main effects when all interactions are negligible. For orthogonal arrays of strength 3, the main effects can be orthogonally estimated under the weaker assumption that interactions involving three or more factors are negligible. In this chapter, we assume that interactions involving three or more factors are negligible. Under this assumption, orthogonal arrays of strength 3 allow unbiased estimation of all main effects regardless of two-factor interactions. Clear 2fi's carry this idea further. Orthogonal arrays of strength 3 with clear 2fi's allow unbiased estimation of all main effects and all clear 2fi's even if the other 2fi's are nonnegligible.

A 2fi in an orthogonal array of strength 3 is clear if it is orthogonal to all other 2fi's. A clear 2fi can be estimated regardless of other 2fi's. In the situation where prior knowledge suggests that certain 2fi's are negligible, this property of robust estimation for a 2fi remains intact so long as it is orthogonal to all nonnegligible 2fi's. Such a 2fi is called partially clear. A partially clear 2fi is allowed to be aliased with negligible 2fi's but this does not affect its robust estimation. The idea for partially clear 2fi's is now formally developed.

Let S_1 denote the set of 2fi's that are to be estimated, S_2 the set of nonnegligible 2fi's and S_3 the set of negligible 2fi's. Thus, the entire set of all 2fi's is partitioned into three mutually exclusive and exhaustive sets S_1, S_2 and S_3. The problem of interest here is to find designs that allow estimation of all main effects and the 2fi's in S_1 in the presence of the nonnegligible 2fi's in S_2. Designs considered for this problem are orthogonal arrays of strength 3, which guarantee that main effects are orthogonal to 2fi's. The 2fi's in S_1 may or may not be orthogonal under a given orthogonal array. For most of this chapter, we consider arrays such that the 2fi's in S_1 are orthogonal. In this case, we have orthogonal designs for estimating main effects and the 2fi's in S_1. According to Greenfield (1976), the set of main effects together with S_1 is called a requirement set. We call S_1 a requirement set in this chapter. In Section 2.3.2, we will allow the requirement set S_1 to be non-orthogonal.

Definition 2.1. *For a given orthogonal array of strength 3, a 2fi in the requirement set S_1 is said to be* partially clear *if it is orthogonal to all 2fi's in the nonnegligible set S_2.*

When the negligible set S_3 is empty, a partially clear 2fi becomes clear. Definition 2.1 allows prior knowledge as summarized by S_3 to be utilized when finding designs for a given requirement set. If an orthogonal array of strength 3 is such that all 2fi's in the requirement set S_1 are partially clear, then this array provides a robust design for the requirement set S_1. This chapter studies the existence and construction of such robust designs.

We now introduce some structures for S_1, S_2 and S_3 which will allow us to gain insights into and obtain theoretical results on robust designs. The factors to be studied are divided into two groups of factors, G_1 and G_2. Let $G_1 \times G_1$ denote the set of 2fi's within the factors in G_1, $G_2 \times G_2$ the set of 2fi's within the factors in G_2, and $G_1 \times G_2$ the set of 2fi's between the factors in G_1 and those in G_2. We consider the following three cases:

Case 1: $S_1 = G_1 \times G_1$, $S_2 = G_1 \times G_2$ and $S_3 = G_2 \times G_2$;
Case 2: $S_1 = G_1 \times G_1$, $S_2 = G_2 \times G_2$ and $S_3 = G_1 \times G_2$;
Case 3: $S_1 = G_1 \times G_2$, $S_2 = G_1 \times G_1$ and $S_3 = G_2 \times G_2$.

For convenience, robust designs for Cases 1, 2 and 3 are referred to as robust designs of types 1, 2 and 3, respectively.

2.2.1 Preliminary Results

Simple characterization of robust designs can be obtained using J-characteristics. For an arbitrary number of vectors $x_j = (x_{1j}, \ldots, x_{nj})^{\mathrm{T}}$, where $j = 1, \ldots, k$, their J-characteristic is defined as

$$J(x_1, \ldots, x_k) = \sum_{i=1}^{n} x_{i1} \cdots x_{ik}.$$

Let $D = (d_1, \ldots, d_m)$ be a two-level factorial design of n runs for m factors, where $d_j = (d_{1j}, \ldots, d_{nj})^{\mathrm{T}}$ is the jth column of D. Then design D is an orthogonal array of strength t if and only if $J(d_{j_1}, \ldots, d_{j_k}) = 0$ for all $j_1, \ldots, j_k$ such that $1 \le j_1 < \cdots < j_k \le m$ and $k \le t$.

Let $a_1, \ldots, a_{m_1}$ be the columns of design D corresponding to the factors in G_1, and $b_1, \ldots, b_{m_2}$ be the columns of design D corresponding to the factors in G_2. Thus $D = (a_1, \ldots, a_{m_1}, b_1, \ldots, b_{m_2})$. We also write $G_1 = (a_1, \ldots, a_{m_1})$ and $G_2 = (b_1, \ldots, b_{m_2})$. Define

$$
\begin{aligned}
A_{40} &= \sum_{1\le j_1<j_2<j_3<j_4\le m_1} \left[J(a_{j_1}, a_{j_2}, a_{j_3}, a_{j_4})\right]^2, \\
A_{31} &= \sum_{1\le j_1<j_2<j_3\le m_1} \sum_{j=1}^{m_2} \left[J(a_{j_1}, a_{j_2}, a_{j_3}, b_j)\right]^2 \text{ and} \\
A_{22} &= \sum_{1\le j_1<j_2\le m_1} \sum_{1\le i_1<i_2\le m_2} \left[J(a_{j_1}, a_{j_2}, b_{i_1}, b_{i_2})\right]^2.
\end{aligned}
$$

The following result is immediate.

Lemma 2.1. *Let $D = (G_1, G_2)$ be an orthogonal array of strength 3. We have that*
(i) design D is a robust design of type 1 if and only if $A_{40} = A_{31} = 0$;
(ii) design D is a robust design of type 2 if and only if $A_{40} = A_{22} = 0$;
(iii) design D is a robust design of type 3 if and only if $A_{31} = A_{22} = 0$.

In Lemma 2.1(i), condition $A_{40} = 0$ ensures an orthogonal requirement set $S_1 = G_1 \times G_1$. Condition $A_{31} = 0$ makes the design robust. More precisely, that the 2fi's in $S_1 = G_1 \times G_1$ are partially clear is captured by $A_{31} = 0$. As seen from parts (ii) and (iii) of Lemma 2.1, robust designs of types 2 and 3 both require that $A_{22} = 0$. This is quite a strong condition to impose on a design, as shown by the following lemma.

Lemma 2.2. *Let $D = (G_1, G_2)$ be an orthogonal array of strength 3 such that $2 \le m_1 \le m_2 \le m - 2$.*

1. *If $A_{22} = 0$, then it must be true that $m \le n/4 + 1$ for every m_1 except for $m_1 = 3$.*

2. *When $m_1 = 3$, we have that $m \le n/4 + 2$. If D is regular, we have $m \le n/4 + 1$.*

Proof. For the case of $m_1 = 2$, condition $A_{22} = 0$ implies that a_1a_2 is a clear 2fi. By Proposition 2 of Tang (2006), it must be true that $m \le n/4 + 1$.

Let $m_1 \ge 3$. Since $D = (G_1, G_2)$ is an orthogonal array of strength 3, all 2fi's are orthogonal to main effects. Because $A_{22} = 0$, the 2fi's a_ib_j for $i = 1, \ldots, m_1$

and $j = 1, \ldots, m_2$ are mutually orthogonal. Thus, the following vectors $a_1, \ldots, a_{m_1}$, $b_1, \ldots, b_{m_2}$, and all $a_i b_j$ for $i = 1, \ldots, m_1$ and $j = 1, \ldots, m_2$ are mutually orthogonal, and all are orthogonal to the column of all plus ones. This establishes that

$$1 + m + m_1(m - m_1) \leq n. \tag{2.1}$$

Solving for m gives

$$m \leq n/(m_1 + 1) + m_1 - 1. \tag{2.2}$$

Taking $m_1 = 3$, we obtain $m \leq n/4 + 2$.

Now consider $m_1 \geq 4$, in which case we must have $n > 16$. (It would be impossible to have $A_{22} = 0$ if $n \leq 16$ and $m_2 \geq m_1 \geq 4$.) We prove Lemma 2.2 for this case by contradication. Assume that $m \geq n/4 + 2$. Then from (2.2), we obtain $n/4 + 2 \leq n/(m_1 + 1) + m_1 - 1$. Solving for n gives $n \leq 4(m_1 + 1)$, implying that $m_1 \geq n/4 - 1$. Since $m_2 = m - m_1 \geq m_1 \geq n/4 - 1$, by (2.1) we obtain $1 + 2(n/4 - 1) + (n/4 - 1)^2 \leq n$, which leads to $n \leq 16$, a contradiction. The proof of Lemma 2.2 is completed.

We next prove that $m \leq n/4 + 1$ for $m_1 = 3$ if D is a regular design. Now assume that $A_{22} = 0$ holds for a strength 3 array $D = (a_1, a_2, a_3, b_1, \ldots, b_{m_2})$ where $m = n/4 + 2$ and $m_2 = n/4 - 1$. Consider design $D^* = (a_1, a_2, b_1, \ldots, b_{m_2})$. Since $A_{22} = 0$, the 2fi $a_1 a_2$ is clear in design D^*. This implies that $a_1, a_2, b_1, \ldots, b_{m_2}$, $a_1 b_1, \ldots, a_1 b_{m_2}$, $a_2 b_1, \ldots, a_2 b_{m_2}$, $a_1 a_2, a_1 a_2 b_1, \ldots, a_1 a_2 b_{m_2}$ together form a saturated regular design, as $m_2 = n/4 - 1$. Noting that $a_1, a_2, b_1, \ldots, b_{m_2}$, $a_1 b_1, \ldots, a_1 b_{m_2}$, $a_2 b_1, \ldots, a_2 b_{m_2}$, $a_3, a_3 b_1, \ldots, a_3 b_{m_2}$ are all mutually orthogonal, we conclude that the set of columns $a_1 a_2, a_1 a_2 b_1, \ldots, a_1 a_2 b_{m_2}$ is identical to the set of columns $a_3, a_3 b_1, \ldots, a_3 b_{m_2}$. Thus a_3 must equal to $a_1 a_2 b_i$ for some i. For simplity, we take $a_3 = a_1 a_2 b_1$. Now consider $a_3 b_2$. Then there must be b_j such that $a_3 b_2 = a_1 a_2 b_j$. This shows that $b_1 = b_2 b_j$, contradicting that D is of strength 3.

□

Using the catalog of regular designs of 64 runs in Chen, Sun and Wu (1993), we conduct a complete search for robust designs. Table 2.1 gives the maximum value of m for given m_1 for each of the three types of robust designs of 64 runs. Ke, Tang and Wu (2005) studied clear compromise plans. In Table 2.1, the maximum number of factors given by clear compromise plans with the same requirement set is provided under max(m_c).

Table 2.1: Robust designs of 64 runs and corresponding clear compromise plans.

	type 1		type 2		type 3	
m_1	max(m)	max(m_c)	max(m)	max(m_c)	max(m)	max(m_c)
7	17	8	10	8	12	8
6	22	9	10	9	12	12
5	22	10	11	10	11	11
4	28	11	11	11	11	11
3	31	17	17	17	17	11

For an orthogonal array of strength 3 to have clear 2fi's, it is necessary that $m \leq n/4+1$; see Chen and Hedayat (1998) and Tang (2006). Lemma 2.2 reveals that robust designs of types 2 and 3 are unable to break this barrier and the gain from considering such designs is unlikely to be substantial. This is all confirmed by Table 2.1. However, much gain can be achieved by robust designs of type 1, as shown in Table 2.1. In the next section, we present several general methods for constructing this type of robust design.

2.3 Some Theoretical Results

This section focuses on robust designs of type 1. For simplicity, when we speak of robust designs in this section, we actually mean robust designs of type 1. We present five methods for constructing robust designs in this section. The first and fifth methods are applicable to regular designs.

Regular fractional factorials are linear orthogonal arrays and treated in almost all design textbooks. See Dey and Mukerjee (1999), Hedayat, Sloane and Stufken (1999), and Wu and Hamada (2000). Regular fractional factorials are specified by their defining relations. The resolution of a regular factorial provides a simple characterization of the aliasing properties of the design. A regular fractional factorial of resolution R is an orthogonal array of strength $R-1$; in particular regular designs of resolution IV are special orthogonal arrays of strength 3. The saturated resolution IV design has $n = 2^k$ runs and $m = 2^{k-1}$ factors where k is a positive integer. Our first method of constructing robust designs makes use of this satuarated resolution IV design.

While the first method and fifth methods only apply to regular designs, our other three methods can be used to construct robust orthogonal arrays, regular or non-regular. The scope of application is in fact even greater as these three methods also allow robust designs to be constructed for non-orthogonal requirement sets. This will be discussed in Secion 2.3.2.

Some of the construction methods are best presented using Kronecker products, which we now introduce. Let $x = (x_1, \ldots, x_{n_1})^{\mathrm{T}}$ and $y = (y_1, \ldots, y_{n_2})^{\mathrm{T}}$. The Kronecker product of two vectors x and y is defined as

$$x \otimes y = (x_1 y_1, \ldots, x_1 y_{n_2}, \ldots, x_{n_1} y_1, \ldots, x_{n_1} y_{n_2})^{\mathrm{T}}.$$

Tang (2006) provided a simple way for calculating the J-characteristic of Kronecker products.

Lemma 2.3. *We have that*

$$J(a_1 \otimes b_1, \ldots, a_k \otimes b_k) = J(a_1, \ldots, a_k)\ J(b_1, \ldots, b_k),$$

where $a_j = (a_{1j}, \ldots, a_{n_1 j})^{\mathrm{T}}$ *and* $b_j = (b_{1j}, \ldots, b_{n_2 j})^{\mathrm{T}}$ *for* $j = 1, \ldots, k$.

2.3.1 Construction of Robust Designs

Method 1. Let X be a saturated resolution IV design with $n = 2^k$ runs and $m = 2^{k-1}$ factors. Then for any three distinct columns x, y, z of X, their Hadamard product xyz must also belong to X. The Hadamard product of two vectors $x = (x_1, \ldots, x_n)^T$ and $y = (y_1, \ldots, y_n)^T$ is defined as $xy = (x_1 y_1, \ldots, x_n y_n)^T$. We obtain the following result.

Proposition 2.1. *Design $D = (G_1, G_2)$ is a robust design, where*
(i) G_1 is a subset of the columns of X and G_1 has resolution VI or higher;
(ii) G_2 is obtained by removing columns xyz from $X \setminus G_1$ for all three distinct x, y, z in G_1.

Proposition 2.1 is immediate from part (i) of Lemma 2.1, as it is obvious that $A_{40} = A_{31} = 0$ for design $D = (G_1, G_2)$.

As before, let m_1 and m_2 denote the numbers of columns in G_1 and G_2, respectively. When $m_1 \leq 5$, the number of columns that need to be removed from $X \setminus G_1$ is $\binom{m_1}{3}$. Thus we have that $m_2 = 2^{k-1} - m_1 - \binom{m_1}{3}$. For $m_1 \geq 6$, it is advantageous to choose G_1 to have resolution VI as the number of columns that have to be removed to obtain G_2 will be less than $\binom{m_1}{3}$ due to the existence of the relation $x_1y_1z_1 = x_2y_2z_2$ when G_1 is of resolution VI.

We next look at the application of the above construction method to designs of 32 and 64 runs. For designs of 32 runs, we obtain $m_2 = 12$ for $m_1 = 3$, $m_2 = 8$ for $m_1 = 4$, and $m_2 = 1$ for $m_1 = 5$. Our computer search of robust designs from among all resolution IV designs shows that the maximum number of columns in G_2 for given m_1 is $m_2 = 12$ for $m_1 = 3$, $m_2 = 8$ for $m_1 = 4$, and $m_2 = 5$ for $m_1 = 5$. This reveals that our method of construction provides the best results for $m_1 = 3, 4$. However, the method is not very effective for $m_1 = 5$. We will return to this example when we present other construction methods.

For designs of 64 runs, our construction method gives $m_2 = 28$ for $m_1 = 3$, $m_2 = 24$ for $m_1 = 4$, and $m_2 = 17$ for $m_1 = 5$. For $m_1 = 6$, choosing G_1 to be of resolution VI gives $m_2 = 16$. In all theses cases, the method maximizes m_2 for given m_1, as is clear from Table 2.1. This method does not produce a useful design for $m_1 = 7$ as it gives $m_2 = 0$. This case will be further examined later.

Method 2. Let D^* and D_0 be two orthogonal arrays of n runs such that D^* has strength 4 and D_0 has strength 2 and that $D^* \subset D_0$. We use I to denote a column of all plus ones of length n. Further let $a_1 = (-1, -1, 1, 1)^T$, $a_2 = (-1, 1, -1, 1)^T$ and $a_1a_2 = (1, -1, -1, 1)^T$.

Proposition 2.2. *Design $D = (G_1, G_2)$ is a robust design, where*

$$G_1 = a_1 \otimes (I, D^*) \text{ and } G_2 = a_1 a_2 \otimes (I, D_0).$$

The proof is straightforward. To prove Proposition 2.2, we need to show that $D = (G_1, G_2)$ is of strength 3 and that $A_{40} = A_{31} = 0$. All these can easily be verified using Lemma 2.3.

When D_0 is regular, the construction in Proposition 2.2 can be thought as a special case of that in Proposition 2.1. This can be seen by noting that $X = (a_1, a_1a_2) \otimes (I, D_0)$ is a saturated resolution IV design of $4n$ runs when D_0 is a saturated resolution III design of n runs, and that G_1, a subset of X, has resolution VI or higher when D^* has resolution V.

The usefulness of Proposition 2.2 lies in the fact that it is applicable to non-regular designs. More importantly, this method provides robust designs for non-orthogonal requirement sets, a topic we will discuss in Section 2.3.2.

Method 3. This method is a modification of Method 2 but allows one extra column in G_1. Let D_0, D^*, a_1, a_2 be defined as in Method 2. Then Method 3 chooses

$$G_1 = (a_2 \otimes I, a_1 \otimes I, a_1 \otimes D^*) \text{ and } G_2 = a_1 a_2 \otimes (D_0 \setminus D^*). \quad (2.3)$$

Proposition 2.3. *Let G_1 and G_2 be defined as in (2.3). Then $D = (G_1, G_2)$ is a robust design.*

Similar to Method 2, Method 3 is applicable to both regular and non-regular designs and also allows non-orthogonal requirement sets. When applied to regular designs, Method 3 is not a special case of Method 1 as design $D = (G_1, G_2)$ given by Method 3 is not an even design. Even designs are those that can be obtained by selecting columns from a saturated resolution IV design. For designs of 32 runs, Method 3 gives a robust design with $m_1 = 5$ and $m_2 = 4$. For designs of 64 runs, Method 3 gives a robust design with $m_1 = 7$ and $m_2 = 10$, the same result as that obtained by computer search given in Table 2.1. Note that Method 1 produces $m_2 = 0$ when $m_1 = 7$ for designs of 64 runs.

Method 4. This method can be viewed as a generalization of doubling (Chen and Cheng 2006; Cheng, Mee and Yee 2008). Let D^* be an orthogonal array of strength 4 and D_0 be an orthogonal array of strength 3 such that $D^* \subset D_0$. Let

$$G_1 = \begin{pmatrix} D^* \\ D^* \end{pmatrix} \text{ and } G_2 = \begin{pmatrix} D_0 \\ -D_0 \end{pmatrix}. \tag{2.4}$$

Proposition 2.4. *Let G_1 and G_2 be defined as in (2.4). Then $D = (G_1, G_2)$ is a robust design.*

If D_0 is chosen to be a saturated resolution IV design, then Method 4 is quite similar to Method 1. The generality of Method 4 is that D_0 can be chosen to be any other maximal design (Chen and Cheng 2006) than the saturated resolution IV design. Let D^* and D_0 be the same maximal design with 16 runs and 5 factors. Then Method 4 gives $m_1 = m_2 = 5$ for designs of 32 runs. Finally, Method 4 applies to non-regular designs and non-orthogonal requirement sets as well.

We present Method 5 through an example. Method 5 is similar to Method 3, but is only valid for regular designs. The purpose of Method 5 is to increase m_1 over that obtained with Method 3.

Example 2.1. Let $n = 128$ and denote the columns for the independent factors from the full factorial design as $1, \ldots, 7$. Let D^* be a resolution V design for 8 factors with generators from factors $2, \ldots, 7$, such as $8 = 2345$ and $9 = 4567$. We form G_1 with columns 1, 2, and all the columns of D^*, where if the column of D^* is not generated by column 2, we take the Hadamard product of column 1 with that column. In this example, this gives $G_1 = \{1, 2, 13, 14, 15, 16, 17, 2345, 14567\}$. Let D_0 be a saturated resolution III design based on factors $3, \ldots, 7$ including the column of 1's, and $D_{0(12)}$ the Hadamard product of columns 1 and 2 with the columns of D_0. To form G_2, we will use a subset of the columns of $D_{0(12)}$. That $A_{40} = 0$ and $A_{31} = 0$ is immediate from the construction. To make $D = (G_1, G_2)$ strength 3, we remove the columns $x_i x_j$ from $D_{0(12)}$, where x_i, x_j are columns of G_1 such that x_i is generated by factor 1 and x_j generated by factor 2. This corresponds to the removal of $7 \times 2 = 14$ columns. Then Method 5 gives a design with $m_1 = 9$ and $m_2 = 18$. In comparison, Method 1 gives $m_1 = 9$ and $m_2 = 1$. For $n = 128$, the maximum m_1 for Method 3 gives a design with $m_1 = 8$ and $m_2 = 25$.

Method 5 follows the approach in Example 2.1, and is applicable when the resolution V design D^* can be formed by more than one generator, as this provides the additional columns for G_1. In general, column 2 should be chosen as a factor that appears in the fewest number of generators in D^*. This minimizes the number of factors to be removed from $D_{0(12)}$ through the number of $x_i x_j$ terms.

2.3.2 Non-orthogonal Requirement Sets

A robust design, as constructed in Methods 1-5, is an orthogonal array $D = (G_1, G_2)$ of strength 3 such that $A_{40} = A_{31} = 0$. The condition $A_{31} = 0$ ensures robustness in that the 2fi's in the requirement set $G_1 \times G_1$ are orthogonal to the set $G_1 \times G_2$ of nonnegligible 2fi's. The condition $A_{40} = 0$ says that G_1 is an orthogonal array of strength 4, which guarantees orthogonal estimation of the requirement set $G_1 \times G_1$. Instead of requiring G_1 to be an orthogonal array of strength 4, we now introduce a weaker condition on G_1. A design is said to have *property* A if it allows estimation of all main effects and all 2fi's. Property A is weaker than strength 4. An orthogonal array of strength 4 has property A as it allows *orthogonal* estimation of all main effects and all 2fi's.

If $D = (G_1, G_2)$ is an orthogonal array of strength 3 that satisfies $A_{31} = 0$ and G_1 has property A, then $D = (G_1, G_2)$ provides a design for the requirement set $G_1 \times G_1$ that is robust to the nonnegligible 2fi's in $G_1 \times G_2$. In Methods 2, 3 and 4, if we choose D^* to have property A, then it can be easily checked that G_1 also has property A. We summarize this result in the following theorem.

Theorem 2.1. *In Methods 2, 3 and 4, if* D^* *has property* A*, then design* $D = (G_1, G_2)$

(i) allows estimation of the requirement set $G_1 \times G_1$*, and*

(ii) is robust to the set $G_1 \times G_2$ *of nonnegligible 2fi's.*

Note that the full requirement set is $G_1 \times G_1$ plus all main effects. Design $D = (G_1, G_2)$ in Theorem 2.1 supports this full requirement set follows from the fact D is an orthogonal array of strength 3. Theorem 2.1 is a powerful result. We give two examples to illustrate.

Example 2.2. Let D_0 be an orthogonal array of 12 runs for 11 factors and let D^* consist of any four columns of D_0. According to Cheng (1995), design D^* has property A. Using D^* and D_0, Method 2 gives a robust design $D = (G_1, G_2)$ of 48 runs with $m_1 = 5$ and $m_2 = 12$ and Method 3 constructs a robust design $D = (G_1, G_2)$ of 48 runs with $m_1 = 6$ and $m_2 = 7$.

Example 2.3. Let D_0 be an orthogonal array of 20 runs for 19 factors and D^* be a subarray of D_0 with five columns that has property A. The existence of such D^* follows from the results in Loeppky, Sitter and Tang (2007). Applying Methods 2 and 3, we obtain a robust design $D = (G_1, G_2)$ of 80 runs with $m_1 = 6$ and $m_2 = 20$, and a robust design $D = (G_1, G_2)$ of 80 runs with $m_1 = 7$ and $m_2 = 14$, respectively.

By introducing non-orthogonality, we gain in terms of estimation and robustness. The trade-off here is the loss of estimation efficiency. One common measure of the D-efficiency is given by $(X^T X/n)^{1/p}$, where X denotes the full model matrix and p is the number of parameters. For the designs in Examples 2.2 and 2.3, we have found that their D-efficiencies are all well above 90%. So at least for these examples, the efficiency loss is quite minor.

2.4 Catalog of Designs for 32 and 64 Runs

Chen, Sun, and Wu (1993) gave a complete catalog for non-isomorphic designs of resolution IV having 32 and 64 runs. We use these designs to search for robust designs of types 1, 2, and 3. While the results in the previous section provide a method for constructing robust designs of type 1, we present here which designs from the catalog of Chen, Sun, and Wu (1993) these robust designs can be obtained from. In Tables 2.2 to 2.5, "parent design" refers to the design from Chen, Sun, and Wu (1993) which

Table 2.2: 32-Run Partially Clear Designs of Type 1

$\|G_1\|$	$max(m)$	$max(m_c)$	parent design	Columns in G_1	Columns in G_2
6	6	6	6-1.1, 6-1.2		
5	10	6	10-5.1	(1, 4, 8, 16, 29)	(2, 7, 11, 19, 30)
4	12	7	12-7.1	(16, 19, 21, 25)	(1, 2, 4, 7, 8, 11, 13, 14)
3	15	9	15-10.1	(7, 8, 16)	(1, 2, 4, 11, 13, 14, 19, 21, 22, 25, 26, 28)

allows for the construction of the robust design and $max(m)$ the maximum number of columns in a robust design for given m_1. For robust designs of types 2 and 3, the results come from multiple parent designs, which can then be chosen according to some secondary criterion. Tables 2.2 - 2.5 can be used to construct robust designs, as they specify a set of columns to form both G_1 and G_2. For 32 runs, only robust designs of type 1 provide useful results, as given in Table 2.2, which also lists the maximum number ($max(m_c)$) of columns allowable for a clear design.

For a design of type 1, additional type 1 designs can be created by moving an element from G_1 to G_2. If we have a design in which two-factor interactions in $G_1 \times G_2$ are clear, moving an element from G_2 to G_1 results in a partially clear design of type 3. For the designs listed, further partially clear designs of the same type can be created by removing factors from either G_1 or G_2.

For $n = 64$ and $m_1 = 6$, the partially clear design of type 1 is unique in that for all 15 non-isomorphic designs and all possible choices for columns in G_1, there is only one design that is partially clear of type 1. In terms of minimum aberration of these designs, it is the 13th out of 15. Also of note for $n = 64$ and $m_1 = 5$, the only choice for G_1 that results in a partially clear design of type 1 is achieved by moving an element from G_1 to G_2 from the partially clear design for $m_1 = 6$.

Table 2.3: 64-Run Partially Clear Designs of Type 1

$\|G_1\|$	$max(m)$	parent design	Columns in G_1	Columns in G_2
7	17	17-11.2	(16, 19, 29, 32, 37, 41, 47)	(1, 2, 4, 7, 8, 11, 49, 55, 59, 62)
6	22	22-16.13	(32, 35, 37, 41, 49, 62)	(1, 2, 4, 7, 8, 11, 13, 14, 16, 19, 21, 22, 25, 26, 28, 31)
5	22	22-16.13	(32, 35, 37, 41, 49)	(1, 2, 4, 7, 8, 11, 13, 14, 16, 19, 21, 22, 25, 26, 28, 31, 62)
4	28	28-22.1	(49, 50, 52, 56)	(1, 2, 4, 7, 8, 11, 13, 14, 16, 19, 21, 22, 25, 26, 28, 31, 32, 35, 37, 38, 41, 42, 44, 47)
3	31	31-25.1	(14, 16, 32)	(1, 2, 4, 7, 8, 11, 13, 19, 21, 22, 25, 26, 28, 31, 35, 37, 38, 41, 42, 44, 47, 49, 50, 52, 55, 56, 59, 61)

Table 2.4: 64-Run Partially Clear Designs of Type 2

$\|G_1\|$	$max(m)$	parent design	Columns in G_1	Columns in G_2
7	10	10-4.1	(1, 2, 4, 8, 16, 27, 32)	(7, 43, 53)
		10-4.2	(2, 4, 7, 8, 16, 25, 32)	(1, 42, 53)
6	10	10-4.1	(1, 2, 4, 16, 27, 32)	(7, 8, 43, 53)
		10-4.2	(1, 2, 4, 8, 16, 32)	(7, 25, 42, 53)
		10-4.7	(1, 2, 4, 8, 16, 32)	(7, 25, 42, 52)
5	11	11-5.1	(1, 16, 29, 32, 51)	(2, 4, 7, 8, 11, 45)
		11-5.3	(2, 16, 29, 32, 49)	(1, 4, 7, 8, 11, 46)
		11-5.4	(11, 16, 32, 46, 56)	(1, 2, 4, 7, 8, 21)
		11-5.11	(2, 16, 30, 32, 49)	(1, 4, 7, 8, 11, 13)
4	11	11-5.1	(1, 16, 29, 32)	(2, 4, 7, 8, 11, 45, 51)
		11-5.3	(1, 16, 29, 32)	(2, 4, 7, 8, 11, 46, 49)
		11-5.4	(8, 16, 32, 46)	(1, 2, 4, 7, 11, 21, 56)
		11-5.6	(1, 29, 32, 62)	(2, 4, 7, 8, 11, 16, 19)
		11-5.7	(8, 16, 32, 57)	(1, 2, 4, 7, 11, 21, 38)
		11-5.9	(1, 29, 32, 45)	(2, 4, 7, 8, 11, 16, 19)
		11-5.11	(1, 16, 30, 32)	(2, 4, 7, 8, 11, 13, 49)
		11-5.13	(1, 16, 30, 32)	(2, 4, 7, 8, 11, 13, 46)
		11-5.20	(13, 16, 32, 53)	(1, 2, 4, 7, 8, 11, 19)
		11-5.22	(13, 16, 32, 46)	(1, 2, 4, 7, 8, 11, 19)
		11-5.24	(13, 19, 32, 61)	(1, 2, 4, 7, 8, 11, 16)
		11-5.30	(1, 16, 32, 51)	(2, 4, 7, 8, 11, 13, 14)
		11-5.31	(2, 16, 32, 49)	(1, 4, 7, 8, 11, 13, 14)
3	17	17-11.6	(1, 32, 63)	(2, 4, 7, 8, 11, 13, 14, 16, 19, 21, 22, 25, 26, 28)

Table 2.5: 64-Run Partially Clear Designs of Type 3

$\lvert G_1 \rvert$	$max(m)$	parent design	Columns in G_1	Columns in G_2
7	12	12-6.1	(1, 2, 4, 7, 8, 11, 16)	(29, 32, 45, 51, 62)
		12-6.8	(1, 2, 4, 7, 8, 11, 13)	(16, 30, 32, 46, 49)
6	12	12-6.1	(1, 2, 4, 7, 8, 11)	(16, 29, 32, 45, 51, 62)
		12-6.2	(8, 16, 32, 46, 54, 56)	(1, 2, 4, 7, 11, 21)
		12-6.8	(1, 16, 30, 32, 46, 49)	(2, 4, 7, 8, 11, 13)
5	11	11-5.1	(1, 16, 29, 32, 45)	(2, 4, 7, 8, 11, 51)
		11-5.2	(1, 2, 4, 7, 63)	(8, 16, 25, 32, 42, 52)
		11-5.4	(8, 16, 32, 46, 56)	(1, 2, 4, 7, 11, 21)
		11-5.11	(1, 16, 30, 32, 49)	(2, 4, 7, 8, 11, 13)
		11-5.13	(1, 16, 30, 32, 46)	(2, 4, 7, 8, 11, 13)
4	11	11-5.1	(16, 29, 32, 45)	(1, 2, 4, 7, 8, 11, 51)
		11-5.2	(1, 2, 4, 7)	(8, 16, 25, 32, 42, 52, 63)
		11-5.3	(1, 16, 32, 49)	(2, 4, 7, 8, 11, 29, 46)
		11-5.4	(11, 21, 32, 46)	(1, 2, 4, 7, 8, 16, 56)
		11-5.6	(1, 29, 32, 62)	(2, 4, 7, 8, 11, 16, 19)
		11-5.7	(8, 16, 32, 57)	(1, 2, 4, 7, 11, 21, 38)
		11-5.9	(16, 29, 32, 45)	(1, 2, 4, 7, 8, 11, 19)
		11-5.11	(1, 16, 32, 49)	(2, 4, 7, 8, 11, 13, 30)
		11-5.13	(16, 30, 32, 46)	(1, 2, 4, 7, 8, 11, 13)
		11-5.20	(13, 16, 32, 53)	(1, 2, 4, 7, 8, 11, 19)
3	17	17-11.6	(1, 32, 63)	(2, 4, 7, 8, 11, 13, 14, 16, 19, 21, 22, 25, 26, 28)

2.5 Discussion and Further Results

The following two problems of theoretical importance arise from robust designs of type 1. One is to construct such robust designs that maximize m_2 for given m_1, and the other is to construct robust designs with m_1 maximized. For designs of 32 and 64 runs, we have shown in Section 2.3 that our methods allow such designs to be found. Although completely solving these problems is likely to be quite nontrivial, it would be possible to obtain some useful general results. We leave the problems to future research.

We conclude this chapter with a result on the maximum number of clear 2fi's in an orthogonal array of strength 3.

Theorem 2.2. *We have that*

$$\beta(n,m) \leq m(n-2m)/(m-2),$$

where $\beta(n,m)$ denotes the number of clear 2fi's in an orthogonal array of strength 3 with n runs for m factors.

Proof. Let $D = (a_1, \ldots, a_m)$ be an orthogonal array of strength 3 with n runs for m factors. Suppose that it has β clear 2fi's and we use $c_1, \ldots, c_\beta$ to denote the column vectors corresponding to these clear 2fi's. Further let R_0 denote the $n-1$ dimensional vector space that collects all vectors with n real entries that are orthogonal to the vector of all plus ones. Then $a_1, \ldots, a_m$, $c_1, \ldots, c_\beta$ are mutually orthogonal and all belong to R_0. Let R_1 denote the linear subspace of R_0 that consists of all vectors that are orthogonal to $a_1, \ldots, a_m$, $c_1, \ldots, c_\beta$. Then R_1 is a p dimentional subspace with $p = n-1-m-\beta$. Let $d_1, \ldots, d_p$ be an orthonormal basis of R_1. That is, $d_1, \ldots, d_p$ are orthogonal vectors with length unity in R_1. Since $a_1a_2, \ldots, a_1a_m$ are mutually orthogonal, for any given $k = 1, \ldots, p$, we have

$$| < a_1a_2/\sqrt{n}, d_k > |^2 + \cdots + | < a_1a_m/\sqrt{n}, d_k > |^2 \leq 1,$$

where $< x, y >$ denotes the inner product of two vectors x and y. Equivalently, we have

$$| < a_1 a_2, d_k > |^2 + \cdots + | < a_1 a_m, d_k > |^2 \leq n.$$

Noting that $< a_1 a_1, d_k >= 0$, we obtain $\sum_{j=1}^{m} | < a_1 a_j, d_k > |^2 \leq n$. In general, we have $\sum_{j=1}^{m} | < a_i a_j, d_k > |^2 \leq n$, for every $i = 1, \ldots, m$. Upon combining, we obtain

$$\sum_{1 \leq i < j \leq m} | < a_i a_j, d_k > |^2 \leq nm/2.$$

Summing over k and re-arranging, we obtain

$$\sum_{1 \leq i < j \leq m} \sum_{k=1}^{p} | < a_i a_j, d_k > |^2 \leq pnm/2. \tag{2.5}$$

Now consider $a_i a_j$ for fixed i, j. This 2fi is orthogonal to R_1 if it is clear and belongs to R_1 otherwise. Therefore, we have that $\sum_{k=1}^{p} | < a_i a_j, d_k > |^2$ is equal to 0 if $a_i a_j$ is clear and is equal to n otherwise. We thus obtain $\binom{m}{2} - \beta \leq pm/2$. Substituting $p = n - 1 - m - \beta$ in and solving for β, we obtain $\beta \leq m(n - 2m)/(m - 2)$. $\square$

The same result for regular designs was obtained earlier in Tang, Ma, Ingram and Wang (2002). However, the proof of Theorem 2.2 does not follow from the idea of proving the result for regular designs.

In the case of regular designs, the bound in Theorem 2.2 can be slightly improved; see Wu and Wu (2002) and Yang and Butler (2008).

Chapter 3

Multi-Level Orthogonal Arrays for Estimating Main Effects and Specified Interactions

3.1 Introduction

In this chapter, we examine the requirement set problem for orthogonal arrays with more than two levels. There are two key differences when handling more than two levels. Firstly, main effects and interaction terms can be broken up into orthogonal components, of which only some may be of interest. Secondly, level permutations of the factors become an issue when using these orthogonal components. This chapter investigates how these differences impact searching for designs. In Chapter 2, we enforced orthogonality between the nonnegligible effects and the effects in the requirement set. This chapter relaxes the requirement of orthogonality, but we will still attempt to find designs with robust properties.

Given a factor with q levels, the main effect for this factor has $q-1$ degrees of freedom and can be broken up into $q-1$ orthogonal contrasts. In this chapter, we adopt the approach from Xu and Wu (2001) which uses normalized main effect contrasts so that all factorial effects have the same variance if a full factorial design were used. For example, for a three-level factor, we have linear and quadratic main effects coded as $(-1, 0, 1) \times \sqrt{3}/\sqrt{2}$ and $(1, -2, 1)/\sqrt{2}$, respectively. For convenience, our discussion refers to the linear and quadratic contrasts for the three-level coding, but the results presented hold for any set of orthogonal contrasts.

A two-factor interaction has $(q-1)^2$ orthogonal components, each corresponding to one degree of freedom. In a two-level design, this implies that a two-factor interaction corresponds to one component and, as such, one degree of freedom. For a three-level design, a two-factor interaction has four orthogonal components corresponding to linear-by-linear, linear-by-quadratic, quadratic-by-linear, and quadratic-by-quadratic effects.

The experimenter may be interested in only a subset of the two-factor interaction components. In this chapter we consider experiments in which the experimenter is interested in a model with the grand mean, all main effects, and certain two-factor interaction components, the set of which we call the *requirement set*. For a requirement set S, define the *core set*, $C(S)$, as the subset of S which includes all two-factor interaction components in S and all main effects for which a main effect component occurs in one of the two-factor interaction components in S. For example, consider a special case of the second-order model as in Cheng and Wu (2001) and Xu, Cheng and Wu (2004) in which some or all of the linear-by-linear two-factor interaction components are of interest:

$$y = \beta_0 + \sum_{i=1}^{r+k} \beta_i x_{il} + \sum_{i=1}^{r+k} \beta_{ii} x_{iq} + \sum_{1 \le i < j \le r} \beta_{ij} x_{il} x_{jl} + \epsilon, \tag{3.1}$$

where y is the response, ϵ the error term, x_{il} and x_{iq} the linear and quadratic contrast coefficients for factor i and β_i, β_{ii} their corresponding effects, β_{ij} the linear-by-linear two-factor interaction for factors i and j, and β_0 is the grand mean. Cheng and Wu (2001) and Xu, Cheng and Wu (2004) consider the case where $k = 0$. If $r = 3$ and $k = 2$, the requirement set is $\{\beta_0, \beta_1, \beta_{11}, \beta_2, \beta_{22}, \beta_3, \beta_{33}, \beta_4, \beta_{44}, \beta_5, \beta_{55}, \beta_{12}, \beta_{13}, \beta_{23}\}$, while the core set is $\{\beta_0, \beta_1, \beta_{11}, \beta_2, \beta_{22}, \beta_3, \beta_{33}, \beta_{12}, \beta_{13}, \beta_{23}\}$. We will consider Model (3.1) again in Section 3.6. A design is said to *support* a requirement set if all effects in the requirement set are estimable. If we have multiple non-isomorphic designs that support a given requirement set, we often distinguish between these designs using one or more optimality criteria, some of which will be discussed in the next section.

If the experimenter believes that curvature may exist in the experimental region and only wants to run one experiment, this can be explored with the designs discussed in this chapter. One common approach for testing the presence of curvature is the addition of center points to a two-level design. The addition of center points can only detect if there is curvature. Here we want to estimate the effects with the design. Other designs for this purpose include Box-Behnken designs (Box and Behnken, 1960) and central composite designs (Box and Wilson, 1951).

In Section 3.2 we introduce various design criteria and Section 3.3 examines how these are affected by level permutations. Section 3.4 investigates the existence of designs for a requirement set when a saturated orthogonal array exists. Section 3.5 uses the results to propose a method of searching for designs from a saturated orthogonal array, which is then applied to 27-run designs under a second-order model in Section 3.6. We conclude the chapter with a brief discussion in Section 3.7.

3.2 Optimality Criteria

In this section we look at some optimality criteria which can be used to differentiate between designs that support a given requirement set S. Consider a model that includes all effects in the requirement set:

$$Y = X_M \beta_M + \epsilon \tag{3.2}$$

where Y is the vector of n observations, β_M is the vector of effects in our requirement set, X_M is the corresponding matrix of contrast coefficients, and ϵ is the vector of n independent random errors. The matrix X_M is also referred to as the model matrix. Define

$$M = {X_M}^T X_M / n = [m_{ij}] \tag{3.3}$$

as the moment matrix. A D-optimal design maximizes $|M|$, the determinant of M, and minimizes the volume of the joint confidence region on the vector of regression coefficients. For a design that supports S, its D-efficiency is defined as

$$D_{eff} = |X_M{}^T X_M / n|^{1/p},$$

where p is the number of parameters in the model (ie. the size of the requirement set). If $D_{eff} = 1$, the columns of X_M are orthogonal.

There are other optimality criteria that use the moment matrix. These include the $E(s^2)$ criterion, where

$$E(s^2)(M) = \sum_{1 \le i < j \le p} m_{ij}^2 / \binom{p}{2},$$

and E-optimality, which maximizes the minimum eigenvalue of the moment matrix and minimizes the maximum possible variance of a normalized linear function of β_M. Another common criterion is the A-optimality which minimizes $tr(M^{-1})$, the trace of the inverse of the moment matrix, and minimizes the average variance of the estimates of the regression coefficients.

In this chapter, our first means of differentiating between designs is to look at the D-efficiency. After D-efficiency, we further distinguish designs by measuring their robustness to nonnegligible effects outside of the requirement set.

To get an estimate for β_M in Model (3.2), we use $\hat{\beta}_M = (X_M{}^T X_M)^{-1} X_M{}^T Y$. This estimate is unbiased for β_M if Model (3.2) is true. While the model we fit is (3.2), assume that the true model is actually

$$Y = X_0\beta_0 + X_1\beta_1 + X_2\beta_2 + \boldsymbol{\epsilon}, \tag{3.4}$$

where X_1 and X_2 correspond to the matrix of contrast coefficients for the main effects, β_1, and two-factor interactions, β_2, X_0 the vector of ones with β_0 being the grand mean, and ϵ the error term. Note that X_M contains the columns of X_0, X_1, and a subset of columns from X_2. An alternative way to write (3.4) is then

$$Y = X_M\beta_M + X_{2o}\beta_{2o} + \epsilon, \tag{3.5}$$

where β_{2o} refers to the two-factor interaction components from β_2 outside of the requirement set, and X_{2o} the corresponding matrix of contrast coefficients.

In order to differentiate between designs having the same D-efficiency, we can minimize the contamination of nonnegligible two-factor interactions outside of the model. Due to restrictions on the run size, Model (3.5) may not be estimable. If we fit Model (3.2) when (3.5) is the truth,

$$\begin{aligned} E(\hat{\beta_M}) &= E(({X_M}^T X_M)^{-1} {X_M}^T Y) \\ &= ({X_M}^T X_M)^{-1} {X_M}^T E(Y) \\ &= ({X_M}^T X_M)^{-1} {X_M}^T (X_M \beta_M + X_{2o}\beta_{2o}) \\ &= \beta_M + ({X_M}^T X_M)^{-1} {X_M}^T X_{2o}\beta_{2o} \\ &= \beta_M + C_{2o}\beta_{2o}. \end{aligned}$$

In this chapter we call $C_{2o} = [c_{ij}] = ({X_M}^T X_M)^{-1} {X_M}^T X_{2o}$ the alias matrix. Because of β_{2o}, estimation of β_M is contaminated by these nonnegligible two-factor interactions outside of the model, so ideally C_{2o} should be small. One measure of this contamination is through the minimum contamination criterion which minimizes $\|C_{2o}\|^2 = \sum c_{ij}^2$ (Tang and Deng, 1999, Xu and Wu, 2001, and Steinberg and Bursztyn, 2001).

In Equation (3.5), X_{2o} is the matrix of contrast coefficients corresponding to all two-factor interactions outside of the requirement set. In practice, linear-by-linear two-factor interactions are more often active than other interactions (Xu, Cheng and Wu, 2004), so one could consider the contamination of linear-by-linear two-factor interactions not in the requirement set. In this case, we would minimize $\|C_{2l}\|^2$, with

$$\|C_{2l}\|^2 = \|({X_M}^T X_M)^{-1} {X_M}^T X_{2l}\|^2, \tag{3.6}$$

where X_{2l} is the subset of columns from X_{2o} that correspond to the linear-by-linear components outside of the model. If we let X_{2q} be the matrix consisting of the remaining columns from X_{2o} whose components contain a quadratic effect (ie. $X_{2q} = X_{2o} \setminus X_{2l}$), we have

$$\|C_{2q}\|^2 = \|({X_M}^T X_M)^{-1} {X_M}^T X_{2q}\|^2. \tag{3.7}$$

To minimize the contamination from the two-factor interactions containing a quadratic main effect, we would minimize $\|C_{2q}\|^2$. Combining (3.6) and (3.7), we have

$$\|C_{2o}\|^2 = \|C_{2l}\|^2 + \|C_{2q}\|^2. \tag{3.8}$$

In our discussion of contamination of nonnegligible effects, our consideration has only been placed on two-factor interactions. In general, we can consider i-factor interactions through $C_i = ({X_M}^T X_M)^{-1} {X_M}^T X_i$, where X_i is the matrix of contrast coefficients for the i-factor interactions and measure the contamination of these effects with $\|C_i\|^2$. If we use the hierarchical ordering principle, on the effects outside of the requirement set, we can sequentially minimize $\|C_{2o}\|^2, \|C_3\|^2, \ldots$ as a way to distinguish between designs.

3.3 Level Permutations

In this section we look at the effect of permuting the levels of a factor on the properties of a design. For multi-level designs, because of the multiple components for main effects and interactions, level permutations of factors can impact a design to the point that one set of level permutations supports a requirement set while another does not for the same design. Remark 2.3.1 from Dey and Mukerjee (1999) states that their main results do not depend on the choice of level permutations. While this holds in situations where we are interested in all components of interactions, this invariance to level permutations may no longer hold if our requirement set contains only certain interaction components. In this section, we examine the influence of level permutations on the optimality criteria of the previous section. The results focus on level permutations on factors outside of the core set, as it will be shown that such level permutations retain many design properties. Throughout this chapter we assume that our design is an orthogonal array, implying main effects are orthogonal and the lack of orthogonality with effects in the requirement set comes through the two-factor interaction components in the requirement set.

Throughout this chapter, we will make use of the fact that this level permutation can be performed via an orthonormal matrix Q_1, where $Q_1^T Q_1 = I$. Let A denote the $n \times (q-1)$ matrix containing the $q-1$ vectors of contrast coefficients for the main effects of a factor. Let A^* consist of the vectors of contrast coefficients for the same factor after a level permutation. The relationship between A^* and A can be expressed with the $(q-1) \times (q-1)$ orthonormal matrix Q_1 as

$$A^* = AQ_1. \tag{3.9}$$

For example, consider the case of three levels and six runs in which we have linear and quadratic main effects for a factor such that A is

$$A = \begin{pmatrix} -\sqrt{3}/\sqrt{2} & 1/\sqrt{2} \\ -\sqrt{3}/\sqrt{2} & 1/\sqrt{2} \\ 0 & -2/\sqrt{2} \\ 0 & -2/\sqrt{2} \\ \sqrt{3}/\sqrt{2} & 1/\sqrt{2} \\ \sqrt{3}/\sqrt{2} & 1/\sqrt{2} \end{pmatrix},$$

and we consider a level permutation to A^* given by

$$A^* = \begin{pmatrix} 0 & -2/\sqrt{2} \\ 0 & -2/\sqrt{2} \\ \sqrt{3}/\sqrt{2} & 1/\sqrt{2} \\ \sqrt{3}/\sqrt{2} & 1/\sqrt{2} \\ -\sqrt{3}/\sqrt{2} & 1/\sqrt{2} \\ -\sqrt{3}/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}.$$

We can then obtain Q_1 as

$$Q_1 = A^T A^*/n = \begin{pmatrix} -1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{pmatrix}.$$

From equation (3.9), we have a matrix Q_1 to perform a level permutation on a factor in the model. Consider a level permutation on a factor outside of the core set. Denote X_M as the model matrix before the level permutation, and X_M^* as the model matrix after the level permutation. If we let

$$Q = \begin{pmatrix} Q_1 & 0 \\ 0 & I_{p-(q-1)} \end{pmatrix},$$

we can relate X_M and X_M^* through

$$X_M^* = X_M Q, \tag{3.10}$$

where, without loss of generality, the first $q-1$ columns of X_M correspond to the main effect components of the level permuted factor. The moment matrix after level permutations, M^*, is then

$$M^* = (X_M Q)^T (X_M Q)/n = Q^T M Q, \tag{3.11}$$

where M is the moment matrix before the level permutation.

At this point, it is important to make the distinction between level permutations of factors inside the core set versus factors outside of the core set. A level permutation of a factor inside the core set causes permutations of the two-factor interaction components in the requirement set, which may not allow us to write X_M^* in the form given in (3.10). With this formulation of level permutations in mind, we can now look at the effect on the optimality criteria from the previous section.

Theorem 3.1. *When the levels of a factor outside of the core set are permuted, the D, A, E, and $E(s^2)$ criteria remain the same.*

Proof. In order to establish that the level permutations outside of the core set preserve the criteria, we will start by showing that the eigenvalues of M^* and M are the same. Let Q be the orthonormal matrix that performs the level permutation as in (3.10). We have $M^* = Q^T M Q$ from (3.11). Recall that the characteristic equation of a square matrix W is: $0 = |W - \lambda I|$. The roots of the characteristic equation are the eigenvalues of W. The characteristic equation of M^* is $0 = |M^* - \lambda I| = |(Q^T M Q) - \lambda I| = |Q^T (M - \lambda I) Q| = |Q^T||M - \lambda I||Q| = |M - \lambda I|$. So M^* and M have the same characteristic equation, and as such the same eigenvalues.

Recalling that the determinant of a matrix is the product of its eigenvalues, and the trace equal to the sum of its eigenvalues, we have that the D, A, and E criteria are the same after level permutation.

For $E(s^2)$, note that $E(s^2)(M) = tr(M^T M)/2 - \sum m_{ii}^2$. Since we assume normalized length for contrast coefficients, $m_{ii}^2 = n$ for all i, which remains true after a level permutation. Examining M^TM, we have $tr(M^{*T}M^*) = tr((Q^TMQ)^T(Q^TMQ)) = tr(M^TM)$, implying $E(s^2)(M) = E(s^2)(M^*)$. □

Theorem 3.1 tells us that we can make level permutations to factors outside of the core set without impacting many criteria. What Theorem 3.1 does not address is if level permutations affect the contamination of the two-factor interaction components outside of the requirement set. This connection is not immediately obvious, since measurement of this contamination is based on X_M and X_{2o}, both of which are changed after level permutations. We address this relationship in the next theorem.

Theorem 3.2. *When the levels of a factor outside of the core set are permuted, the contamination of the two-factor interaction components outside of the model, measured by $\|C_{2o}\|^2$, remains the same.*

Proof. After the level permutation, $\|C_{2o}\|^2 = \|({X_M}^TX_M)^{-1}{X_M}^TX_{2o}\|^2$ becomes $\|C^*_{2o}\|^2 = \|(X^{*T}_MX^*_M)^{-1}X^{*T}_MX^*_{2o}\|^2$. In order to show $\|C_{2o}\|^2 = \|C^*_{2o}\|^2$, we will look at the relationship between X^*_M and X_M and X^*_{2o} and X_{2o}. We will make use of the fact that

$$\|C\|^2 = tr(C^TC).$$

Let Q be the orthonormal matrix from (3.10) such that $X^*_M = X_MQ$. Since $Q^T = Q^{-1}$, we can simplify the terms of $\|C^*_{2o}\|^2$ involving X^*_M to

$$(X^{*T}_MX^*_M)^{-1}X^{*T}_M = Q^T(X^T_MX_M)^{-1}X^T_M.$$

As the matrix X_{2o} corresponds to the contrast coefficients for all two-factor interaction components outside of the model, it can be partitioned into sets of two-factor interaction components corresponding to pairs of the m factors in the model. We denote this partition as X_{2op} for $p \in (1, \ldots, \binom{m}{2})$, where p is an index for a pair of factors from the design. For p indexing a pair of factors in the core set, X_{2op} may not contain the full set of $(q-1)^2$ two-factor interaction components, as some of these components may be in the requirement set. From this partition of X_{2o}, we have that

$$\|C_{2o}\|^2 = \sum_p \|(X^T_MX_M)^{-1}X^T_MX_{2op}\|^2. \tag{3.12}$$

Consider i for which X_{2oi} corresponds to a set of two-factor interaction components for a pair of factors that have not had levels permuted. After the level permutation, we have $X^*_{2oi} = X_{2oi}$. Looking at the effect of this subset of X_{2o} on $\|C^*_{2o}\|^2$ after level permutation,

$$\|Q^T(X_M^TX_M)^{-1}X_M^TX_{2oi}\|^2 = \|(X_M^TX_M)^{-1}X_M^TX_{2oi}\|^2, \tag{3.13}$$

showing that the contamination is unchanged for the pair of factors indexed by i.

The remaining columns of X_{2o} correspond to two-factor interaction components a pairs of columns in which one involves a level permutation. Let X_{2oj} correspond to the $n \times (q-1)^2$ subset for such a pair (all of the interaction components are outside of the requirement set, otherwise the factor with a level permutation would be in the core set). Let X^*_{2oj} be the same matrix after level permutations, and in a similar fashion to (3.9), there is a $(q-1)^2 \times (q-1)^2$ orthonormal matrix Q_2 such that $X^*_{2oj} = X_{2oj}Q_{2j}$. The influence on $\|C^*_{o2}\|^2$ is then

$$\|Q^T(X_M^TX_M)^{-1}X_M^TX_{2oj}Q_{2j}\|^2 = \|(X_M^TX_M)^{-1}X_M^TX_{2oj}\|^2, \tag{3.14}$$

which is also unchanged from the level permutation. Combining (3.13) and (3.14) into (3.12), we have the required result. □

From the proof of Theorem 3.2, we have not only that the overall contamination from two-factor interaction components remains the same, but also the component-wise contamination from each pair of factors. As a result, if we were to sequentially minimize these pairwise contaminations starting from the largest, in a manner analogous to G-aberration, the ranking would not change after level permutations outside of the core set. Likewise, if contamination is only considered for interactions involving one or more factors from the core set, nothing is changed. One situation Theorem 3.2 does not apply to is when contamination is to be considered for certain two-factor interaction components, such as all linear-by-linear components outside of the requirement set.

Our discussion in this section assumes that we have a design to support a requirement set. The next section discusses the existence of such designs from orthogonal arrays.

3.4 Existence of Designs for a Given Requirement Set through Orthogonal Arrays

Tang and Zhou (2009) showed that for two-level designs, the existence of an orthogonal array that supports a requirement set is equivalent to that of an orthogonal array that supports its core set. For designs with greater than two levels, a similar result holds, but the fact that main effects and interactions have multiple components presents some restrictions. While only some components of interactions may be in the requirement set, all main effect components are in the requirement set. In this section, we consider removing columns representing main effect components from a saturated orthogonal array that occupy the same space as the interaction components in the core set. If any one of the main effect components of a factor occupies the space of the interaction components, we remove all remaining main effect components for that factor. The results are presented based on three-level designs, as they are most powerful in this case, but the arguments are applicable for greater than three levels.

Theorem 3.3. *For a requirement set S with m factors and e two-factor interaction components, if an orthogonal array that supports the core set $C(S)$ with m_1 factors exists, then an orthogonal array that supports S exists, provided $m \leq (n-1)/2 - e$.*

Proof. Consider a matrix D constructed from a saturated three-level orthogonal array where each column is replaced by normalized main effect contrasts, such as the coding for linear and quadratic main effects, and a column of all 1's added to the array. We will show that so long as there is a subset D_1 of D that supports $C(S)$, there exists a partition $D = (D_1, D_2, D_3)$, where D_2 is the set of columns in S that are not in $C(S)$ and D_3 are the columns to be removed from D that occupy the same space as the vectors corresponding to interaction components. We will denote the number of factors in D_1, D_2, and D_3 as m_1, m_2, and m_3, respectively, and $m = m_1 + m_2$ is the total number of factors in the requirement set.

Let $X_{1c} = (\mathbf{1}, D_1)$, where $\mathbf{1}$ is the column of all 1s corresponding to the grand mean and D_1 a subset of D that supports $C(S)$. The model matrix for $C(S)$ is then (X_{1c}, X_{2c}), where X_{2c} corresponds to the interaction components in the core set. Since

D is a saturated orthogonal array, the partition $D = (X_{1c}, D_2, D_3)$ forms the column space of D. If we let $D^* = (D_2, D_3)$, by the properties of block matrices,

$$\begin{aligned}
& det[(X_{1c}, X_{2c})^T(X_{1c}, X_{2c})] \\
= \; & det(X_{1c}^T X_{1c}) det(X_{2c}^T X_{2c} - X_{2c}^T X_{1c}(X_{1c}^T X_{1c})^{-1} X_{1c}^T X_{2c}) \\
= \; & det(X_{1c}^T X_{1c}) det(X_{2c}^T D^* D^{*T} X_{2c}) \\
= \; & n^{2m_1+1-e} det(X_{2c}^T D^* D^{*T} X_{2c}). \qquad (3.15)
\end{aligned}$$

Since D_1 supports $C(S)$, $det(X_{2c}^T D^* D^{*T} X_{2c}) > 0$, so $X_{2c}^T D^* D^{*T} X_{2c}$ must be of rank e. Recalling that $rank(AA^T) = rank(A)$, this also implies that $X_{2c}^T D^*$ has rank e.

The dimensions of $X_{2c}^T D^*$ is $e \times (2m_2 + 2m_3)$. In order for X_{2c} to be estimated, it must be that $(2m_2 + 2m_3) \geq 2m_3 \geq e$. Then $X_{2c}^T D^*$ must contain a subset of $2m_3$ columns that has rank e, corresponding to m_3 factors (each with two main effect components) from D^*. As the columns in D_3 are pairs of main effect components for factors, the preceding bound is $m_3 \geq \lceil e/2 \rceil$. It is possible that there exists a subset of $m_3^* < m_3$ columns such that $X_{2c}^T D^*$ has a subset of $2m_3^*$ columns with rank e.

Let D_3^* be a subset of $2m_3$ columns from D^*, so that $rank(X_{2c}^T D_3^*) = e$. This also gives us $rank(X_{2c}^T D_3^* D_3^{*T} X_{2c}) = e$ and

$$det(X_{2c}^T D_3^* D_3^{*T} X_{2c}) > 0. \qquad (3.16)$$

With $D^* = (D_2^*, D_3^*)$, let $X = (X_1, D_2^*)$ be be the model matrix for the requirement set S. Similar to (3.15), we have

$$det[(X, X_{2c})^T(X, X_{2c})] = n^{2m+1-e} det(X_{2c}^T D_3^* D_3^{*T} X_{2c}), \qquad (3.17)$$

which is greater than 0 by (3.16), and implies that (D_1, D_2^*) supports S. If there exists m_3^* as described above, S can accommodate additional factors.

□

As an alternative to the proof from Theorem 3.1, (3.17) shows that level permutations on factors outside of the core set, as represented by D_2, do not effect the D-efficiency of the design. The connection to D-efficiency and D_3 is also seen through the following theorem.

Theorem 3.4. *The D-efficiency is maximized when* $det(X_{2c}^T D_3^* D_3^{*T} X_{2c})$ *is maximized.*

Theorem 3.3 provides a lower bound on the maximum number of factors outside of the core set that can be supported. In the formation of D_3^* in the proof of Theorem 3.3, the worst case scenario is that the e two-factor interaction components take up an e-dimensional space that corresponds to e different factors occupying a $2e$-dimensional space. However, it is possible that the e two-factor interaction components occupy the $2e^*$-dimensional space from $e^* < e$ different factors, a consideration we return to in Section 3.6 when searching for designs. Theorems 3.3 and 3.4 provide insight which will be useful in the next section to aid in searching for designs.

It is clear that level permutations for factors outside of the core set or model, such as in D_2 and D_3, have no effect on the results of Theorems 3.3 and 3.4. This is not true for factors in the core set, as level permutations change X_{2c}. Even if the core set is supported by a D_1 after a level permutation of a factor within it, the corresponding D_2 and D_3 will not be the same. In addition, it is possible that level permutations on the core set allow additional factors to be considered outside of the core set or an improvement to the D-efficiency. This implies that when searching for designs, different level permutations of factors in the core set should be considered, as will be done in the next section.

3.5 Searching for D-efficient Designs with Robust Properties from Orthogonal Arrays

In this section, the previous results are applied to search for designs that support a requirement set, S, if a saturated orthogonal array of the required run size is available. As Theorems 3.3 and 3.4 are applicable when a design that supports the core set, $C(S)$, is specified, the algorithm will consider different combinations of factors from the orthogonal array to form the core set, and find the best choices among the remaining columns for the selection of the factors outside of the core set. Level permutations are considered, but only those on the factors within the core set need to be considered, based on the results in Section 3.3.

Let r be the number of factors in the core set, and k the number of factors outside of the core set. For an orthogonal array with n runs and m columns having q levels each, and replace each column by a set of $q-1$ orthogonal contrasts to create the matrix D. When we refer to choosing a factor from $\{1,\ldots,m\}$, we mean the set of orthogonal contrasts of that factor. The search proceeds as follows:

1. Let $c=(c_1,\ldots,c_r)$, a subset of size r from $\{1,\ldots,m\}$, denote the factors in the core set.

2. Let $p_c=(p_{c_1},\ldots,p_{c_r})$ denote a set of level permutations on the factors in the core set.

3. Let D_1 be the set of columns from D formed by the factors given by c after performing the level permutations upon the factors as given by p_c. With the main effect components in the core set specified, X_{2c} is the set of contrast coefficients for the interaction components in the requirement set.

4. If $C(S)$ is supported by D_1,

   (a) Take a subset of size $m-r-k$ from $\{1,\ldots,m\}\backslash c$ to be the factors not to be used in the design, and use these to form D_3^* from D. The design under consideration is (D_1,D_2^*), where D_2^* is comprised of the remaining columns from D not in D_1 or D_3^*.

   (b) Calculate $det(X_{2c}^TD_3^*D_3^{*T}X_{2c})$.

   (c) Repeat Steps 4a and 4b for all $\binom{m-r}{m-r-k}$ possible subsets from the remaining columns.

5. Repeat Steps 2 to 4 for all possible level permutations p_c.

6. Repeat Steps 1 to 5 for all $c\subset\{1,\ldots,m\}$ such that $|c|=r$.

By Theorem 3.4, the D-optimal design among those from the saturated orthogonal array is the design for which $det(X_{2c}^TD_3^*D_3^{*T}X_{2c})$ in Step 4b is maximized. Recall that Theorem 3.3 gives a lower bound on the number of factors that can be supported outside of the core set. Provided there are sufficient degrees of freedom, the algorithm presented allows us to search for designs in which k is greater than the bound in Theorem 3.3.

The algorithm as presented is applicable to search for D-optimal designs. If we want to minimize the contamination from the two-factor interaction components outside of the requirement set, $\|C_{2o}\|^2$ can be calculated in Step 4b, or calculated for those designs with the largest D-efficiencies after the search is complete. Theorems 3.1 and 3.2 imply that we do not need to consider any level permutations for the factors outside of the core set, as this will have no influence on the D-efficiency or contamination from the two-factor interaction components outside of the model.

This method for searching for designs is applied to designs of 27 runs in the next section.

3.6 Efficient Designs of 27 Runs Robust to Two-factor Interactions

In this section, we consider a special case of the second order model with requirement sets that contain two-factor interaction components for a subset of the factors in the requirement set. The algorithm from the previous section will be used to search for D-efficient designs that minimize the contamination from the two-factor interaction components outside of the model.

The model of interest in this section is the second order model (3.1) as introduced in Section 3.1:

$$y = \beta_0 + \sum_{i=1}^{r+k} \beta_i x_{il} + \sum_{i=1}^{r+k} \beta_{ii} x_{iq} + \sum_{1\leq i<j\leq r} \beta_{ij} x_{il} x_{jl} + \epsilon, \tag{3.18}$$

where we are interested only in the linear-by-linear two-factor interaction effects from a subset of r factors out of $m = r + k$ factors.

The results presented concentrate on 27-run three-level designs. We use the catalog of Evangelaras, Koukouvinos and Lappas (2011) in which they identified 129 non-isomorphic 27-run saturated orthogonal arrays with three levels. We refer to a design from this catalog as Di where i refers to the i'th design as ordered in the catalog from Evangelaras, Koukouvinos and Lappas (2011). We only consider cases of r and k in which there are sufficient degrees of freedom to estimate all effects in (3.18). A saturated design from which we take a subset to form a design is referred to as a parent design.

To gauge the impact of the results in searching for designs, consider the case $r = 3$ and $k = 8$. If we were to consider all possible column choices and level permutations, we would need $\binom{13}{3} \times \binom{10}{8} \times 3^{11} = 2279881890$ different combinations. Theorems 3.1 and 3.2 allow us to restrict level permutations to the core set, giving $\binom{13}{3} \times \binom{10}{8} \times 3^3 =$ 347490 different combinations. Only three level permutations per factor in the core set are considered, as reflection of a factor around the center level retains the projection properties of the design, including D-efficiency, as discussed in Xu, Cheng and Wu (2004). With $\mapsto$ indicating the levels before and after level permutation, we refer to permutation 0 as $(012) \mapsto (012)$, permutation 1 as $(012) \mapsto (120)$ and permutation 2 as $(012) \mapsto (201)$.

Even with the reduction of possible searches, in some situations an exhaustive search is still computationally intensive. We will consider a means of reducing the search in Section 3.6.2.

3.6.1 Example of a Search for a Requirement Set from a Saturated Orthogonal Array

We now examine a complete search of two saturated orthogonal arrays to look for efficient designs. For parent designs $D24$ and $D52$, we use the algorithm to search for designs with $r = 3$ and $k = 8$. That is, we use the 24th and 52nd designs from the catalog of Evangelaras, Koukouvinos and Lappas (2011) to search for designs in which the requirement set contains the linear and quadratic main effect components for 11 factors, and a subset of size 3 of these factors in which the linear-by-linear two-factor interaction components are in the requirement set.

Tables 3.1 and 3.2 show the top 20 D-efficient designs found for each parent design. We rank the designs firstly by D-efficiency, followed by contamination of the two-factor interaction components outside of the model. We also include the contamination from only the linear-by-linear two-factor interaction components outside of the model. The **design** column refers to the parent design Di, **core** refers to the set of columns from Di that form the r factors in the core set, and **outside** the columns used for the k remaining factors. The set of level permutations for the factors in the core set is denoted by **perm**, the D-efficiency by **D-eff**, and the contamination from the two-factor interaction components outside of the model $\|C_{2o}\|^2$, with the linear-by-linear contribution being $\|C_{2l}\|^2$.

Comparing Tables 3.1 and 3.2, we see that $D24$ and $D52$ provide different D-efficient designs for the second-order model (3.18). The highest D-efficiency comes from $D52$, and the highest D-efficiency for $D24$ has less contamination from the two-factor interaction components measured by $\|C_{2o}\|^2$. Also, the number of possible designs that give the same D-efficiency differs between the parent designs. For $D24$, there tends to be a large number of level permutations and factor assignments that give the same D-efficiency in comparison to $D52$. If we were to restrict ourselves to a random selection of core sets rather than all possibilities, we are more likely to miss D-efficient designs among $D52$ versus $D24$.

Xu, Cheng and Wu (2004) suggested using G_2 and G-aberration in choosing designs for second-order models for factor screening and interaction detection. These will be discussed in more detail in the next chapter, but we mention them here to make the distinction between the approaches. The criteria of G and G_2-aberration do not take a specific requirement set into consideration. While minimum G and G_2-aberration designs may perform well when we consider finding designs that are applicable under a variety of models with two-factor interaction components involving fewer factors, in this situation there are specific two-factor interaction components to be estimated. The use of G and G_2-aberration is to give designs that remain efficient over many subsets of factors. However, when we have a particular model in mind, they may not necessarily lead to a good design.

3.6.2 Searching among all Saturated Orthogonal Arrays

Even with the preceding results, a complete search of all possible designs among all 129 saturated designs is still a time-consuming undertaking. For one, we have to consider all $\binom{13}{r}$ possibilities for the core set and level permutations for factors in the core set. For each of these core set choices and level permutations, there is a large amount of time spent in step 4 when searching for the best choice of factors to be removed from the design, particularly if the number of these subsets is large. In order to reduce the time spent on searching, one possibility is to consider only a subset of the core set and level permutation combinations in steps 5 and 6 of the algorithm. While this random sample of the different combinations may be used, ideally we would like a means of choosing factors and level permutations for the core sets that look promising and investigate these further.

Wu (2009) showed that for two-level designs, consideration of D-efficient designs for the core set lead to D-efficient designs for the entire requirement set. In essence, this is due to the two-factor interactions occupying the same space as a subset of columns from a Hadamard matrix that are outside of the core set. In the three-level situation, this approach becomes more complicated due to the multiple components attached to a factor - instead of removing individual columns, we remove sets of columns corresponding to the main effect components for a factor. Choosing a D-efficient design for the core set does not take into consideration the multiple componenents for a factor nor which factors should be removed from the remaining.

Letting $W = D_3^{*T} X_{2c}$, where D_3^{*T} and X_{2c} are defined as in Section 3.4, by Theorem 3.4 we want to maximize $det(W^T W)$. This requires evaluation of all factor assignments to D_3^*. We now consider an alternative approach to the D-efficiency of the core set. The $(M.S)$-criterion can be used as a surrogate for D-efficiency and is less computationally intensive to calculate. The $(M.S)$-criterion firstly maximizes $tr(W^T W)$, followed by minimizing $tr((W^T W)^2)$. For a given core set and level permutations, let D_o be the columns from the saturated orthogonal array not in the core set that correspond to $m_o = m-r$ factors. We want a $W = D_3^{*T} X_{2c}$ for some $D_3^* \subset D_o$ for $m_3 = m - r - k$ factors ($2m_3$ columns) that maximizes $det(W^T W)$. Following the

$(M.S)$-criterion, an alternative is to find $D_3^+ \subset D_o$ by firstly maximizing $tr(W^TW)$, followed by minimizing $tr((W^TW)^2)$, where $W = D_3^{+T}X_{2c}$. Looking at the matrix $W_o = D_o^TX_{2c}$, the rows correspond to main effect components for all factors not in the core set and the columns the two-factor interaction components in the core set. We want to pick the best $2m_3$ rows from W_o which would correspond to the m_3 factors needed for D_3^+. Since $tr((W^TW))$ is the sum of the squared components of W, taking W_o^2 as the squared components of W_o, we can collect the row sums of W_o^2 corresponding to the m_o factors. The top m_3 of these correspond to D_3^+ and their sum gives $tr(W^TW)$ with $W = D_3^{+T}X_{2c}$. We can then calculate $tr((W^TW)^2)$ as the sum of the squared column sums of W_o^2, over the rows corresponding the the factors in D_3^+ just chosen.

If a complete search of all combinations of level permutations and factor assignments is infeasible, using the $(M.S)$-approach is appealing for several reasons. Firstly, W_o only needs to be calculated once for each core set and level permutation, and can calculate the quantities $tr(W^TW)$ and $tr((W^TW)^2)$ to rank those core sets that warrant further investigation while taking into account the nature of the factors to be removed. Perhaps more importantly, the $(M.S)$-approach also identifies a potential set of factors to form D_3. Instead of searching through all possible factor assignments to D_3 for a given core set, we can restrict the search to a subset of those identified as best during the maximization of $tr(W^TW)$. This is particularly appealing if the number of combinations among the remaining columns to be removed is large. We will come back to this point in Section 3.6.6.

Use of this $(M.S)$-approach for $D24$ and $D52$ for $r = 3$ and $k = 8$ provides the same D-efficient designs as the complete search in Tables 3.1 and 3.2. Tables 3.3 and 3.4 show the top designs from a complete search for $r = 4$ and $k = 6$ for $D24$ and $D52$. Using the $(M.S)$-approach, Tables 3.5 and 3.6 show the best D-efficient designs using the top 500 core sets for each parent design. While we do not get the top D-efficient designs, the D-efficiencies are close to those found through a complete search. It is worth noting, that if we use the top 1000 designs using the $(M.S)$-criterion, we do obtain the designs as given in Tables 3.3 and 3.4.

3.6.3 Searching for efficient designs with fewer factors outside of the core set

In Sections 3.6.1 and 3.6.2, we examined a complete search for the cases of $r = 3$ and $k = 8$ and $r = 4$ and $k = 6$. In both situations, the number of factors outside of the core set is the maximum allowable for the given r. With the search algorithm, designs with fewer factors outside of the core set than the maximum allowable tend to take longer to search for. The reason for this is that the number of combinations to choose the k factors outside of the core set increases, in addition to extra dimensions in the matrix D_3^* for evaluating the determinant. As k decreases, the advantage of using D_3 over the model matrix decreases, as step 4 of the algorithm can be done by calculating $det(X_M^T X_M)$. The algorithm in Section 3.5 can also be used with the $(M.S)$-criterion as discussed in Section 3.6.2.

In Section 3.6.4 we provide tables with D-efficient designs with the maximum number of allowable factors outside of the core set for different values of r using the $(M.S)$-criterion. We can use these designs and remove factors outside of the core set in an attempt to find D-efficient designs.

Table 3.7 presents the results from a complete search for $r = 3$ and $k = 7$. We get the same results using the top 500 designs with the $(M.S)$-criterion, or by using a search of the top 400 D-efficient designs from $r = 3$ and $k = 8$ and checking all subsets that remove one of the factors outside of the core set. We see that at least for this case, we do not lose any efficiency taking a simpler approach. While this approach may not always find the D-optimal design, we do have

$$det(X_{Mnew}^T X_{Mnew}) \geq det(X_M^T X_M),$$

where X_{Mnew} is the model matrix after removing factors from outside the core set. This suggests that so long as the D-efficiency is high for larger k, it should remain so when removing additional factors.

3.6.4 Tables of Efficient Designs

Using the $(M.S)$-approach, we now search for D-efficient designs that minimize contamination of the two-factor interaction components outside of the model. Table 3.8 presents the top 20 designs in terms of D-efficiency followed by contamination from the two-factor interaction components. Table 3.9 gives the same results where parent designs are presented only once for a unique D-efficiency. The top designs have D-efficiencies better than was found with the complete search of $D24$ and $D52$. For the top designs, the core set and level permutations from the parent design that lead to the best D-efficiencies are few. In addition, the designs coming from the same parent design are not typically found from the same subset of factors. Table 3.10 gives the top 20 designs for $r = 4$ and $k = 6$. Many of the observations from the previous case hold here as well. Also of note, is that the top D-efficiencies occur in less frequency than in comparison to $r = 3$; there appear to be fewer parent design and core set choices that lead to D-efficient designs.

3.6.5 Minimizing Contamination from Two-factor Interaction Components Outside of the Requirement Set

As discussed previously, if all two-factor interaction components outside of the requirement set are treated equally, Theorem 3.2 tells us we no longer need to consider level permutations for the factors outside of the core set. However, if we differentiate among two-factor interaction components by considering linear-by-linear components as more likely to be nonnegligible, we can first find a D-optimal design and then consider level permutations of the factors outside of the core set to minimize the contamination from the linear-by-linear components. By Theorem 3.1, the D-efficiency will be unchanged.

Tables 3.11 and 3.12 show level permutations of factors outside of the core set for two D-efficient designs for $r = 4$ and $k = 2$. Table 3.11 is from $D46$ and 3.12 from $D110$. In this case, the core set has level permutations 0 for all factors in the core set, and the $\mathbf{perm_{out}}$ refers to the level permutations for the factors outside of the core set. For each design, we report the D-efficiency, the contamination of the two-factor interaction components given by $\|C_{2l}\|^2$, $\|C_{2q}\|^2$, and $\|C_{2o}\|^2$ as defined by equations (3.6), (3.7), and (3.8), respectively. Note that within each table, $\|C_{2o}\|^2$ is the same, as expected via Theorem 3.2.

In general, level permutations of factors outside of the core set influence the impact of the linear-by-linear two-factor interaction components outside of the requirement set. In this example, the D-efficiency of the designs in Table 3.11 are better than those in Table 3.12, as is the minimum of $\|C_{2l}\|^2$. We do have $\|C_{2o}\|^2$ smaller in Table 3.12. Since $\|C_{2o}\|^2$ is invariant to level permutations outside of the core set, it would be ideal to use this value before considering level permutations if choosing among designs with the same D-efficiency. As we have seen in this example, looking at $\|C_{2o}\|^2$ before considering level permutations outside of the core set has no guarantee of finding the minimum $\|C_{2l}\|^2$. The number of all possible level permutations of the factors outside of the core set can still take a significant amount of time to search. However, this minimization of $\|C_{2l}\|^2$ is essentially the last step in the search, completed after finding D-efficient designs. Using this search for only a few top D-efficient designs, the experimenter might trade off a slight reduction in D-efficiency for less contamination from the linear-by-linear two-factor interaction components.

3.6.6 Searching for an 81-Run Design

We conclude this section with an example where a complete search is not feasible. Taking the 81-run, 40-factor design from Neil Sloane's website (Sloane, 2011), consider the case where $r = 5$ and $k = 25$. According to Theorem 3.3, if we have a subset of 5 factors that supports the core set, then a design that supports the requirement set exists. However, for such a core set, there are $\binom{35}{25} = 183579396$ different combinations for the factors outside of the core set. This is for a single choice of columns that supports a core set - aiming for D-efficient designs becomes even more cumbersome. The $(M.S)$-approach not only identifies core sets to be investigated further, but can also be used to select possible choices for factors outside of the core set. While full enumeration of all core sets and level permutations is not possible, we are still able to consider a subset of these and restrict our search for the factors outside of the core set.

For a small sample of core sets and level permutations, we used the $(M.S)$-approach to select the top 12 factors and search among those $\binom{12}{10}$ to be the columns removed from the design. The $(M.S)$-approach found designs with D-efficiencies of about 90%, whereas a random sample of several thousands out of the $\binom{35}{25}$ possibilities did not find a design that supports the requirement set.

3.7 Discussion

While much of the attention in this chapter has been placed on three-level designs, all of the results hold generally for designs with more than three levels. Depending on the nature of the interaction components in the core set, it is possible that the search can be simplified even further if an orthonormal matrix can be used on the interactions, such as when we consider all interaction components for a pair of factors. The results also apply to mixed level designs. This is particularly useful if the effects of interest vary among the factors. For example, in the second order model studied in this chapter, if we are not interested in curvature for the factors outside of the core set, these factors can be set at two levels. Our searches were based on a saturated orthogonal array. If we have an orthogonal array available that is not saturated, the results of this chapter still apply. In this case, the columns in the null space of the design are automatically in D_3.

Table 3.1: Full Search for D24 with $r = 3$, $k = 8$.

Rank	**design**	**core**			**outside**								**perm**			**D-eff**	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	24	2	3	7	4	5	6	8	9	10	11	13	1	1	0	0.90751	210.3599	126.0924
2	24	2	11	12	1	3	4	5	7	9	10	13	1	1	0	0.90751	211.7624	121.6943
3	24	3	5	12	1	2	4	6	7	9	11	13	0	2	1	0.90751	211.9441	124.1502
4	24	2	5	8	1	3	6	9	10	11	12	13	1	2	2	0.90751	211.9999	127.5223
5	24	7	8	13	1	2	3	4	6	9	11	12	2	0	1	0.90751	212.1622	121.0393
6	24	1	11	12	2	3	4	5	7	8	9	10	1	2	0	0.90751	212.3426	128.7172
7	24	4	6	11	1	2	3	8	9	10	12	13	2	0	2	0.90751	212.6203	116.1502
8	24	3	12	13	1	4	5	6	7	8	9	11	0	2	0	0.90751	212.7159	128.2826
9	24	5	7	10	1	3	4	6	8	11	12	13	0	1	1	0.90751	212.7527	124.0225
10	24	3	8	11	1	2	4	5	6	9	10	12	1	1	0	0.90751	212.7751	123.4190
11	24	4	6	13	1	2	5	8	9	10	11	12	1	0	2	0.90751	212.7822	122.4536
12	24	1	6	10	2	3	4	7	9	11	12	13	0	1	0	0.90751	212.8084	121.4064
13	24	2	10	13	3	4	5	7	8	9	11	12	2	1	2	0.90751	212.8944	117.4062
14	24	1	4	9	2	3	5	6	7	8	10	13	2	0	2	0.90751	212.9416	121.6931
15	24	1	3	6	2	4	5	7	9	10	12	13	0	2	2	0.90751	212.9611	122.0260
16	24	5	10	11	1	3	4	6	7	8	9	13	0	0	2	0.90751	212.9723	124.3190
17	24	2	9	13	1	3	5	7	8	10	11	12	2	2	0	0.90751	213.0855	130.5637
18	24	8	10	12	1	2	3	4	5	6	7	13	1	2	2	0.90751	213.1078	127.5364
19	24	1	2	12	3	4	5	6	7	9	10	11	2	0	0	0.90751	213.1779	113.1100
20	24	2	6	8	1	3	4	5	10	11	12	13	0	1	2	0.90751	213.1784	122.3386

Table 3.2: Full Search for D52 with $r = 3$, $k = 8$.

Rank	**design**	**core**			**outside**								**perm**			**D-eff**	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	52	1	3	11	4	5	6	7	8	9	10	13	2	0	2	0.91642	212.5225	119.0060
2	52	1	3	11	2	4	5	6	7	8	10	13	2	0	2	0.91642	212.6413	122.6324
3	52	3	7	9	1	2	4	6	8	10	12	13	0	2	2	0.91642	212.6723	127.5303
4	52	3	7	9	1	2	4	6	8	11	12	13	0	2	2	0.91642	212.6723	130.0322
5	52	2	3	10	4	5	7	8	9	11	12	13	2	0	2	0.91642	212.8388	121.9882
6	52	2	3	10	1	4	5	8	9	11	12	13	2	0	2	0.91642	212.9387	126.4317
7	52	1	4	7	3	5	8	9	10	11	12	13	0	1	0	0.91018	211.0688	114.7868
8	52	1	4	7	2	3	5	8	9	11	12	13	0	1	0	0.91018	211.0787	123.9678
9	52	4	10	11	1	2	3	6	8	9	12	13	1	1	1	0.91018	212.2993	130.6163
10	52	4	10	11	1	2	3	6	7	8	12	13	1	1	1	0.91018	212.3142	133.0460
11	52	2	4	9	3	5	6	7	8	10	11	13	1	1	0	0.91018	214.3700	125.4501
12	52	2	4	9	1	3	5	6	7	8	10	13	1	1	0	0.91018	214.5439	127.2921
13	52	1	2	13	3	4	6	8	9	10	11	12	2	2	1	0.90474	211.5843	105.1457
14	52	1	2	13	3	4	6	7	8	10	11	12	2	2	1	0.90474	211.5843	107.5666
15	52	7	10	13	2	3	4	5	6	8	9	11	0	1	1	0.90474	212.1420	124.6016
16	52	7	10	13	1	2	3	4	5	6	8	9	0	1	1	0.90474	212.1420	125.6868
17	52	9	11	13	1	3	4	5	7	8	10	12	0	1	1	0.90474	212.9785	125.3592
18	52	9	11	13	1	2	3	4	5	7	8	12	0	1	1	0.90474	212.9785	126.7352
19	52	1	2	13	3	4	6	8	9	10	11	12	0	1	1	0.90474	213.8150	121.8106
20	52	1	2	13	3	4	6	7	8	10	11	12	0	1	1	0.90474	213.8150	122.5939

Table 3.3: Full Search for D24 with $r = 4$, $k = 6$.

Rank	**design**	**core**				**outside**						**perm**				**D-eff**	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	24	1	4	5	9	3	6	7	10	11	12	0	0	0	1	0.81496	162.3934	89.64550
2	24	3	5	12	13	1	2	7	8	10	11	0	0	0	1	0.81496	162.3934	93.77505
3	24	5	7	10	11	2	3	4	8	9	12	0	0	2	0	0.81496	162.6169	94.57458
4	24	1	3	6	10	4	5	8	11	12	13	0	0	0	2	0.81496	162.6169	95.74347
5	24	4	8	10	12	1	2	5	6	9	11	0	2	2	0	0.81496	162.8417	99.40190
6	24	1	2	11	12	3	6	8	9	10	13	0	2	0	2	0.81496	163.6982	81.27873
7	24	1	7	8	13	2	3	4	5	6	10	0	0	2	1	0.81496	163.9439	97.01499
8	24	6	7	9	12	1	2	3	4	11	13	0	0	1	0	0.81496	164.0664	94.07995
9	24	4	6	11	13	1	3	5	7	8	9	0	0	0	1	0.81496	164.0664	94.51895
10	24	5	7	10	11	2	3	4	8	9	12	2	0	2	0	0.81496	164.2559	88.50985
11	24	1	2	11	12	3	6	8	9	10	13	0	2	0	0	0.81496	164.2900	76.67385
12	24	2	3	4	7	1	5	9	10	12	13	2	0	0	0	0.81496	164.2900	86.31036
13	24	2	5	6	8	4	7	9	10	11	13	2	0	0	2	0.81496	164.5147	78.08652
14	24	3	5	12	13	1	2	7	8	10	11	1	0	0	1	0.81496	165.0664	97.94988
15	24	2	9	10	13	1	4	6	7	8	12	2	1	2	1	0.81496	165.1687	101.91022
16	24	3	8	9	11	2	5	6	7	12	13	0	2	1	0	0.81496	165.6169	91.77722
17	24	1	7	8	13	2	3	4	5	6	10	0	0	2	2	0.81496	165.8136	92.62801
18	24	2	5	6	8	4	7	9	10	11	13	2	0	0	0	0.81496	165.9630	81.70841
19	24	2	3	4	7	1	5	9	10	12	13	2	0	0	1	0.81496	165.9975	81.86348
20	24	3	8	9	11	2	5	6	7	12	13	0	2	1	1	0.81496	166.2218	96.82280

Table 3.4: Full Search for D52 with $r = 4$, $k = 6$.

Rank	design	core				outside						perm				D-eff	$\|\|C_{2o}\|\|^2$	$\|\|C_{2l}\|\|^2$
1	52	2	5	11	13	3	4	6	8	9	12	0	1	2	0	0.83792	162.4883	97.00187
2	52	6	7	11	13	3	4	5	8	10	12	0	2	0	0	0.83792	163.0312	88.83047
3	52	9	10	12	13	3	4	5	6	8	11	2	0	0	0	0.83792	164.0350	88.85841
4	52	1	5	10	13	3	4	6	7	8	12	1	0	2	0	0.83792	165.1516	95.71076
5	52	9	10	12	13	3	4	5	6	8	11	0	0	1	2	0.83792	165.1856	91.56477
6	52	1	6	9	13	2	3	4	5	8	12	2	1	1	0	0.83792	165.4647	93.95153
7	52	2	7	12	13	1	3	4	5	6	8	2	1	1	0	0.83792	166.0109	87.22564
8	52	6	7	11	13	3	4	5	8	10	12	1	0	0	2	0.83792	166.1894	98.86032
9	52	1	6	9	13	2	3	4	5	8	12	0	0	1	2	0.83792	167.0529	90.68273
10	52	2	5	11	13	3	4	6	8	9	12	0	0	1	2	0.83792	167.3160	93.69656
11	52	2	7	12	13	1	3	4	5	6	8	1	1	0	2	0.83792	167.3691	89.23705
12	52	1	5	10	13	3	4	6	7	8	12	1	1	1	2	0.83792	169.3557	96.54786
13	52	2	3	6	7	4	5	8	9	12	13	0	1	1	2	0.82637	164.8869	98.48818
14	52	1	3	9	12	4	5	6	7	8	13	1	1	2	1	0.82637	164.9460	95.64872
15	52	2	3	11	12	4	5	6	8	10	13	2	1	0	0	0.82637	166.1974	87.74581
16	52	1	3	6	10	4	5	8	11	12	13	2	1	0	0	0.82637	166.5262	86.34181
17	52	3	5	7	11	1	4	6	8	12	13	1	0	1	2	0.82637	167.4242	91.53866
18	52	3	5	9	10	2	4	6	8	12	13	1	1	1	2	0.82637	169.0701	95.40400
19	52	1	4	5	10	3	6	7	8	12	13	2	0	0	1	0.82474	164.6268	86.66257
20	52	2	4	5	11	3	6	8	9	12	13	2	0	1	1	0.82474	164.8185	89.98030

Table 3.5: Using Search Method for D24 with $r = 4$, $k = 6$.

Rank	**design**	**core**				**outside**						**perm**				**D-eff**	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	24	3	8	9	11	1	2	4	7	10	13	2	1	0	2	0.80346	159.1687	87.52976
2	24	4	6	11	13	2	3	5	7	10	12	1	1	2	0	0.80346	160.7192	91.35896
3	24	6	7	9	12	1	3	5	8	10	13	1	2	0	1	0.80346	160.7192	96.42893
4	24	2	9	10	13	1	3	4	5	6	11	1	0	1	0	0.80346	161.6181	95.31477
5	24	2	3	4	7	1	6	8	10	11	12	1	2	1	2	0.80346	162.1687	92.97066
6	24	1	2	11	12	4	5	6	7	8	13	2	1	2	1	0.80346	162.1687	94.72860
7	24	1	7	8	13	4	5	9	10	11	12	2	2	1	0	0.80346	162.5147	90.03270
8	24	3	5	12	13	2	4	6	8	9	10	2	1	1	0	0.80346	164.0652	89.76182
9	24	1	4	5	9	2	3	8	11	12	13	2	1	1	0	0.80346	164.0652	93.12114
10	24	5	7	10	11	1	2	6	9	12	13	1	2	1	2	0.80346	165.5147	87.52230
11	24	1	3	6	10	2	5	7	8	9	11	2	2	1	1	0.80346	165.5147	94.30790
12	24	2	5	6	8	1	3	4	7	9	12	1	1	1	1	0.80346	165.6181	101.34181
13	24	4	8	10	12	3	6	7	9	11	13	1	1	1	1	0.80346	168.9642	102.52963
14	24	1	2	11	12	5	6	8	9	10	13	0	0	0	2	0.78800	156.9516	88.78948
15	24	5	7	10	11	1	2	3	8	9	12	2	0	2	1	0.78800	158.2443	84.49759
16	24	5	7	10	11	2	3	4	6	9	12	2	1	2	0	0.78800	160.9944	88.21398
17	24	1	2	11	12	3	6	7	8	10	13	1	2	0	2	0.78800	161.3952	99.95896
18	24	1	2	11	12	3	4	6	8	9	13	0	2	1	2	0.78800	162.1376	89.17005
19	24	3	8	9	11	2	4	5	7	12	13	1	2	1	1	0.78800	162.1542	96.29928
20	24	3	5	12	13	2	6	7	8	10	11	1	2	0	1	0.78800	163.2546	91.92878

Table 3.6: Using Search Method for D52 with $r = 4$, $k = 6$.

Rank	design	core				outside						perm				D-eff	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	52	1	5	8	11	2	3	4	6	10	13	1	2	1	0	0.81291	164.1246	93.66659
2	52	7	8	9	12	2	3	4	6	10	13	1	1	1	2	0.81291	165.7855	94.66795
3	52	6	7	8	9	1	3	4	11	12	13	2	1	1	1	0.81291	166.0298	88.08316
4	52	2	8	10	12	3	4	5	7	9	13	0	1	0	2	0.81291	166.5332	81.33443
5	52	1	6	8	11	3	4	5	7	9	13	1	2	1	0	0.81291	168.1113	81.56717
6	52	2	5	8	10	1	3	4	11	12	13	0	2	1	0	0.81291	168.2393	85.19275
7	52	1	2	10	11	3	5	6	7	9	12	1	2	1	1	0.80571	163.2840	96.50605
8	52	1	2	10	11	3	5	6	7	9	12	0	1	0	2	0.80571	163.2840	97.72632
9	52	1	7	9	11	2	3	5	6	10	12	2	1	0	1	0.80571	164.3994	91.60579
10	52	1	7	9	11	2	3	5	6	10	12	0	0	2	0	0.80571	164.3994	92.71080
11	52	2	7	9	10	1	3	5	6	11	12	2	0	1	1	0.80571	165.9570	89.16467
12	52	2	7	9	10	1	3	5	6	11	12	1	2	0	0	0.80571	165.9570	93.73994
13	52	1	2	10	11	3	5	6	7	9	12	0	1	2	0	0.80571	166.0724	90.65135
14	52	1	2	10	11	3	5	6	7	9	12	2	0	1	1	0.80571	166.0724	92.51222
15	52	2	7	9	10	1	3	5	6	11	12	1	1	0	2	0.80571	167.0724	87.89259
16	52	2	7	9	10	1	3	5	6	11	12	0	0	2	1	0.80571	167.0724	91.03420
17	52	1	7	9	11	2	3	5	6	10	12	1	2	0	1	0.80571	168.3031	86.96101
18	52	1	7	9	11	2	3	5	6	10	12	0	0	1	2	0.80571	168.3031	90.47473
19	52	1	5	7	11	2	3	6	9	10	12	2	0	1	1	0.79871	164.7085	89.94885
20	52	1	6	10	11	2	3	5	7	9	12	0	0	0	2	0.79871	164.8753	94.37449

Table 3.7: Results for D24 with $r = 3$, $k = 7$.

Rank	**design**	**core**			**outside**							**perm**			**D-eff**	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	24	4	5	11	1	2	3	7	9	10	13	2	1	0	0.93056	170.7317	96.33509
2	24	2	4	10	1	3	8	9	11	12	13	0	0	1	0.93056	171.4650	100.31228
3	24	6	7	13	1	3	4	8	10	11	12	2	0	0	0.93056	171.4923	103.73614
4	24	3	7	12	1	2	4	5	6	8	13	2	1	0	0.93056	171.9793	96.92269
5	24	5	9	11	1	3	7	8	10	12	13	2	1	2	0.93056	172.1675	102.59238
6	24	3	6	11	2	4	5	8	9	10	13	0	2	2	0.93056	172.3346	107.04504
7	24	2	6	9	3	4	5	7	10	12	13	0	0	0	0.93056	172.4373	96.41721
8	24	4	5	6	1	2	7	9	11	12	13	1	0	2	0.93056	172.4417	103.91636
9	24	4	7	13	1	2	3	5	8	11	12	0	2	2	0.93056	172.5734	100.79833
10	24	2	8	9	1	3	5	6	7	10	13	1	0	1	0.93056	172.6229	95.31995
11	24	1	6	12	2	3	5	7	10	11	13	2	0	2	0.93056	172.7086	97.76794
12	24	1	3	4	2	5	6	8	9	10	12	2	0	2	0.93056	172.7302	96.89505
13	24	1	3	5	2	6	9	10	11	12	13	1	2	0	0.93056	172.7638	101.14507
14	24	6	8	10	2	3	4	5	7	11	12	0	1	0	0.93056	172.7808	97.31855
15	24	1	9	12	2	3	5	6	7	8	11	0	2	1	0.93056	172.8053	100.21616
16	24	8	11	12	1	2	3	4	6	7	9	2	2	2	0.93056	172.8703	102.99662
17	24	3	11	13	4	5	7	8	9	10	12	2	1	1	0.93056	172.8861	104.21742
18	24	2	8	11	1	3	4	5	6	9	13	2	1	1	0.93056	173.0412	105.75809
19	24	4	6	8	2	3	5	9	11	12	13	0	1	0	0.93056	173.0958	99.07562
20	24	2	3	12	1	4	5	6	10	11	13	2	1	1	0.93056	173.4056	98.98429

Table 3.8: Using Search Method for $r = 3$, $k = 8$.

Rank	design	core			outside								perm			D-eff	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	129	2	10	11	1	4	5	6	7	9	12	13	2	1	1	0.93953	210.8255	117.1609
2	127	2	9	11	1	4	6	7	8	10	12	13	2	1	1	0.93953	210.8255	118.2195
3	125	4	9	11	1	2	6	7	8	10	12	13	2	1	1	0.93953	210.8255	119.9672
4	126	5	10	13	1	3	4	7	8	9	11	12	0	0	2	0.93953	210.8255	122.2694
5	95	5	9	11	1	3	4	7	8	10	12	13	0	0	2	0.93953	210.8255	122.2694
6	129	7	9	11	1	3	4	5	6	8	10	13	1	0	0	0.93953	212.0503	118.9779
7	97	4	5	9	1	3	6	7	8	10	11	12	2	2	0	0.93953	212.0503	119.7267
8	124	3	7	11	1	2	5	6	9	10	12	13	2	1	2	0.93953	212.0503	121.6325
9	114	2	10	11	1	4	5	7	8	9	12	13	0	0	0	0.93953	212.0503	122.7779
10	128	10	11	12	2	3	4	5	7	8	9	13	1	0	0	0.93953	212.0503	124.2831
11	113	5	7	9	1	3	4	8	10	11	12	13	1	1	2	0.93953	212.2744	123.5556
12	128	7	12	13	1	3	4	5	6	8	9	11	0	2	2	0.93953	212.3291	123.8918
13	95	8	11	12	1	3	4	5	6	7	9	10	0	2	0	0.93953	212.4985	126.1110
14	126	9	11	13	1	3	4	5	6	7	8	10	0	0	2	0.93953	212.4985	126.1110
15	128	5	11	13	1	2	3	6	7	8	9	12	0	2	2	0.93953	212.8868	122.7850
16	97	5	11	12	2	3	4	6	7	8	9	10	2	1	1	0.93953	213.0562	124.5567
17	115	5	11	12	2	3	4	6	7	8	9	10	2	1	1	0.93953	213.0562	124.6173
18	127	2	7	10	1	4	5	6	8	9	11	13	2	2	2	0.93953	213.2750	128.2857
19	125	4	7	10	1	2	5	6	8	9	11	13	2	2	2	0.93953	213.2750	130.1711
20	96	4	7	11	1	2	6	8	9	10	12	13	0	1	1	0.93953	213.7233	118.9792

Table 3.9: Using Search Method for $r = 3$, $k = 8$ - unique D-efficiencies.

Rank	**design**	**core**			**outside**								**perm**			**D-eff**	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	129	2	10	11	1	4	5	6	7	9	12	13	2	1	1	0.93953	210.8255	117.1609
2	127	2	9	11	1	4	6	7	8	10	12	13	2	1	1	0.93953	210.8255	118.2195
3	125	4	9	11	1	2	6	7	8	10	12	13	2	1	1	0.93953	210.8255	119.9672
4	126	5	10	13	1	3	4	7	8	9	11	12	0	0	2	0.93953	210.8255	122.2694
5	95	5	9	11	1	3	4	7	8	10	12	13	0	0	2	0.93953	210.8255	122.2694
6	97	4	5	9	1	3	6	7	8	10	11	12	2	2	0	0.93953	212.0503	119.7267
7	124	3	7	11	1	2	5	6	9	10	12	13	2	1	2	0.93953	212.0503	121.6325
8	114	2	10	11	1	4	5	7	8	9	12	13	0	0	0	0.93953	212.0503	122.7779
9	128	10	11	12	2	3	4	5	7	8	9	13	1	0	0	0.93953	212.0503	124.2831
10	113	5	7	9	1	3	4	8	10	11	12	13	1	1	2	0.93953	212.2744	123.5556
11	115	5	11	12	2	3	4	6	7	8	9	10	2	1	1	0.93953	213.0562	124.6173
12	96	4	7	11	1	2	6	8	9	10	12	13	0	1	1	0.93953	213.7233	118.9792
13	46	2	5	8	1	4	6	9	10	11	12	13	2	1	1	0.92919	209.3596	115.4164
14	49	2	7	10	1	4	6	8	9	11	12	13	2	1	1	0.92919	209.3596	115.4164
15	50	2	5	8	1	4	6	9	10	11	12	13	2	1	1	0.92919	209.3596	115.4164
16	13	2	7	9	1	4	6	8	10	11	12	13	2	1	1	0.92919	209.3596	115.4164
17	47	5	9	12	1	2	6	7	8	10	11	13	0	1	1	0.92919	210.5843	124.8164
18	14	7	9	11	1	2	5	6	8	10	12	13	0	1	1	0.92919	210.5843	124.8164
19	48	3	5	6	1	4	7	8	9	10	11	12	1	0	0	0.92919	211.1420	123.5673
20	127	5	10	11	1	2	6	7	8	9	12	13	0	2	0	0.92767	210.1307	119.4287

Table 3.10: Using Search Method for $r = 4$, $k = 6$.

Rank	**design**	**core**				**outside**						**perm**				**D-eff**	$\|C_{2o}\|^2$	$\|C_{2l}\|^2$
1	2	5	7	11	12	2	4	6	8	9	13	1	1	1	1	0.88509	161.1687	98.15536
2	76	7	10	11	13	1	2	5	6	8	9	0	2	0	0	0.86761	163.2900	93.78200
3	76	9	10	11	13	2	4	6	7	8	12	1	2	1	0	0.86761	163.5130	93.93308
4	76	7	9	10	13	1	4	5	6	11	12	2	2	2	0	0.86761	163.5984	92.42728
5	75	1	2	9	13	4	5	6	8	10	11	2	2	2	1	0.86761	164.1561	89.39627
6	75	1	4	9	13	2	5	6	7	11	12	1	0	2	1	0.86761	164.6894	96.30732
7	75	2	4	9	13	1	6	7	8	10	12	0	1	2	1	0.86761	169.2443	92.07690
8	66	9	10	11	13	1	2	5	6	7	12	1	0	1	1	0.85940	162.2075	92.83496
9	66	7	9	10	13	2	4	5	6	8	11	2	2	0	1	0.85940	165.3944	92.85750
10	66	7	10	11	13	1	4	6	8	9	12	0	0	0	1	0.85940	166.8548	89.62653
11	68	6	8	12	13	2	4	5	7	10	11	1	0	2	2	0.85302	164.7261	90.92066
12	68	6	7	12	13	1	2	8	9	10	11	1	0	2	1	0.85302	166.6398	94.51921
13	67	2	4	6	13	1	5	7	10	11	12	1	1	1	2	0.85302	167.1338	96.88900
14	67	1	2	6	13	4	7	8	9	10	12	0	2	1	2	0.85302	167.6260	86.36760
15	68	6	7	8	12	1	4	5	9	10	13	1	2	1	2	0.85302	167.8491	91.30406
16	67	1	4	6	13	2	5	8	9	10	11	1	2	1	2	0.85302	168.4627	88.96247
17	69	1	2	9	13	3	4	5	7	11	12	1	2	2	1	0.85045	163.6013	95.86586
18	70	7	8	10	12	2	3	4	9	11	13	2	1	2	0	0.85045	165.6422	93.10338
19	69	1	4	9	13	2	3	7	8	10	12	2	0	2	1	0.85045	166.2363	92.74255
20	70	7	10	12	13	1	2	3	5	8	9	1	2	0	2	0.85045	166.8771	90.22760

Table 3.11: Level Permutations of Factors Outside of the Core Set from Design $D46$ for $r = 4$ and k=2.

main				**outside**		**perm$_{\mathbf{out}}$**		**Dopt**	$\|C_{2l}\|^2$	$\|C_{2q}\|^2$	$\|C_{2o}\|^2$
2	4	5	12	9	10	0	0	0.960041	5.3802	31.6758	37.0559
2	4	5	12	9	10	0	1	0.960041	6.3898	30.6661	37.0559
2	4	5	12	9	10	1	0	0.960041	6.3898	30.6661	37.0559
2	4	5	12	9	10	2	0	0.960041	6.3898	30.6661	37.0559
2	4	5	12	9	10	0	2	0.960041	6.3898	30.6661	37.0559
2	4	5	12	9	10	2	1	0.960041	7.0795	29.9764	37.0559
2	4	5	12	9	10	1	1	0.960041	7.0795	29.9764	37.0559
2	4	5	12	9	10	1	2	0.960041	7.0795	29.9764	37.0559
2	4	5	12	9	10	2	2	0.960041	7.0795	29.9764	37.0559

Table 3.12: Level Permutations of Factors Outside of the Core Set from Design $D110$ for $r = 4$ and k=2.

main				**outside**		**perm$_{\mathbf{out}}$**		**Dopt**	$\|C_{2l}\|^2$	$\|C_{2q}\|^2$	$\|C_{2o}\|^2$
3	9	11	13	6	10	2	2	0.959229	6.6922	30.1542	36.8464
3	9	11	13	6	10	0	2	0.959229	6.8698	29.9766	36.8464
3	9	11	13	6	10	2	0	0.959229	6.8698	29.9766	36.8464
3	9	11	13	6	10	2	1	0.959229	6.9092	29.9372	36.8464
3	9	11	13	6	10	1	2	0.959229	6.9092	29.9372	36.8464
3	9	11	13	6	10	1	0	0.959229	7.1888	29.6575	36.8464
3	9	11	13	6	10	0	1	0.959229	7.1888	29.6575	36.8464
3	9	11	13	6	10	0	0	0.959229	7.2514	29.5950	36.8464
3	9	11	13	6	10	1	1	0.959229	7.3303	29.5161	36.8464

Chapter 4

J-Characteristics for Multi-Level Factorial Designs

4.1 Introduction

In Chapters 2 and 3, designs were studied to estimate a subset of the two-factor interactions based on our knowledge about certain effects. This chapter has no requirement sets, nor prior knowledge about effects. In the absence of a requirement set, the hierarchical ordering principle would suggest that we want an orthogonal array of higher strength. If we think of strength in terms of design points, the higher strength of an orthogonal array implies an even spread of the design points in lower dimensions. In this chapter, our goal is to measure the properties of a design when projected onto lower dimensions. We will do so by measuring the spread of the design points in lower dimensions, which will lead to robust properties for the design. This approach provides designs that are useful for screening and also appealing as 'nearly-orthogonal' arrays either on their own or as building blocks for other designs.

For two-level designs, the criterion of minimum G-aberration (Deng and Tang, 1999) provides a measure of the projection properties of a design, and a relaxed variant, minimum G_2-aberration, also has a projection justification (Tang, 2001). Xu and Wu (2001) studied minimum G_2-aberration for multi-level designs, but did so without the use of J-characteristics. In this chapter we extend the concept of J-characteristics to multi-level designs and derive G and G_2-aberration based on these J-characteristics. The J-characteristics are appealing because they characterize the distributions of design points in various dimensions and do not require the specification of orthonormal contrasts for factors as is done by other approaches.

Section 4.2 introduces the J-characteristics for multi-level designs and Section 4.3 uses them to define G and G_2-aberration, and a new criterion that places consideration on individual factors. Section 4.4 provides an illustration of using the J-characteristics to evaluate designs using the standard analysis of variance (ANOVA) provided in statistical packages. We give some designs ranked according to G and G_2-aberration in Section 4.5.

4.2 J-Characteristics and Their Properties

4.2.1 J-Characteristics for Multi-Level Designs

In this section, we define the J-characteristics for multi-level designs and give an explicit example, followed by some of the properties of J-characteristics in Section 4.2.2. The definition of the J-characteristics is based on decomposing the frequency of the design points analogous to the ANOVA decomposition taught in introductory statistics courses. The key idea is to treat the frequency of design points as the response variable.

Let D be a design with n runs and m factors, where factor j has s_j levels, given by $0, 1, \ldots, s_j - 1$, for $j = 1, \ldots, m$. Denote $N(x_1, \ldots, x_m)$ as the number of design points at level combination $(x_1, \ldots, x_m)$ where $x_j = 0, \ldots, s_j - 1$. Consider the ANOVA decomposition of $N(x_1, \ldots, x_m)$ given by

$$N(x_1, \ldots, x_m) = \sum_{u \subseteq Z_m} N_u(x_1, \ldots, x_m), \tag{4.1}$$

where the summation is over all subsets of $Z_m = \{1, \ldots, m\}$,

$$N_\emptyset = (s_1 \cdots s_m)^{-1} \sum_{x_1, \ldots, x_m} N(x_1, \ldots, x_m) \tag{4.2}$$

is the grand mean, and

$$N_u(x_1,\ldots,x_m) = (\prod_{j\notin u} s_j)^{-1} \sum_{x_j, j\notin u} N(x_1,\ldots,x_m) - \sum_{v\subset u} N_v(x_1,\ldots,x_m) \tag{4.3}$$

is the interaction involving the factors in u.

The J-characteristics are defined as

$$J_u(x_1,\ldots,x_m) = s_1\cdots s_m N_u(x_1,\ldots,x_m). \tag{4.4}$$

The J-characteristics can then be related back to the design points through

$$s_1\cdots s_m N(x_1,\ldots,x_m) = \sum_{u\subseteq Z_m} J_u(x_1,\ldots,x_m). \tag{4.5}$$

Ai and Zhang (2004) defined J-characteristics using orthonormal contrasts following the approach of Xu and Wu (2001). In their formulation, the J-characteristics require a set of orthonormal contrasts for each factor and are calculated based on these contrasts. In our formulation, the J-characteristics are based only on the design points and do not require the specification of contrasts.

The value of $J_u(x_1,\ldots,x_m)$ depends only on x_j's for $j\in u$. For the remainder of this chapter, unless otherwise specified, by referring to the J-characteristics we mean those as defined in equation (4.4). The ANOVA decomposition allows for standard statistical software to evaluate the properties of a design, a point which we return to in Section 4.4.

The multi-level J-characteristics are consistent with those used for two-level designs. For a two-level design, $D = (x_{ij})$ with $x_{ij} = \pm 1$, Tang and Deng (1999) and Deng and Tang (1999) defined the J-characteristics as

$$J_u = \sum_{i=1}^{n} \prod_{j\in u} x_{ij}. \tag{4.6}$$

for any $u \subseteq Z_m = \{1,\ldots,m\}$. Tang (2001) related the distribution of design points as a vector, N, to the vector of J-characteristics of a design through

$$J_u = H^T N, \tag{4.7}$$

or equivalently $N = 2^{-m}HJ$, where H is a Hadamard matrix that contains all possible Hadamard products of the column vectors of the full factorial design. For two-level designs, J_u is characterized by a single number and the calculation is based on ± 1. In the two-level case, for any of the 2^m combinations of $x_1, \ldots, x_m$, $|J_u(x_1, \ldots, x_m)|$ $= |J_u|$. More precisely we have $J_u(x_1, \ldots, x_m) = (\prod_{i \in u} x_i) J_u$. For multi-level designs, the situation becomes more complicated, as $J_u(x_1, \ldots, x_m)$ can take on numerous values. We demonstrate this in the following example.

Example 4.1. Consider the 9-run, three-level design

$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 1 & 0 \\ 2 & 2 & 1 \end{pmatrix}.$$

For the above design D, we have $N(x_1, x_2, x_3) = 1$ if (x_1, x_2, x_3) is one of the nine rows and $N(x_1, x_2, x_3) = 0$ otherwise. We use the decomposition in (4.1) to obtain the N_u terms which will allow us to calculate the J-characteristics by Equation (4.4). Starting with the grand mean, we have $N_\emptyset = (1/27) * 9 = 1/3$. For the subsets u with $|u| = 1$, $N_1(x_1, x_2, x_3) = N_2(x_1, x_2, x_3) = N_3(x_1, x_2, x_3) = (1/9) * 3 - (1/3) = 0$, for any $x_1, x_2, x_3 \in \{0, 1, 2\}$. To clarify, examining the calculation of $N_1(0, 0, 0)$ in greater detail,

$N_1(0, 0, 0) = (1/(s_2 * s_3)) \sum_{x_2, x_3} N(0, x_2, x_3) - N_\emptyset = (1/(3 * 3)) * 3 - (1/3) = 0,$

where $\sum_{x_2, x_3} N(0, x_2, x_3) = 3$ is the number of times 0 occurs in column 1. For $u = \{1, 2\}$ and $u = \{1, 3\}$,

$N_{1,2}(x_1, x_2, x_3) = N_{1,3}(x_1, x_2, x_3) = (1/3) * 1 - (1/3) = 0,$

for all $x_1, x_2, x_3 \in \{0, 1, 2\}$, while for $u = \{2, 3\}$,

$N_{2,3}(x_1, 1, 0) = N_{2,3}(x_1, 1, 1) = N_{2,3}(x_1, 1, 2) =$

$N_{2,3}(x_1, 0, 2) = N_{2,3}(x_1, 2, 2) = 1/3 - 1/3 = 0$, for all $x_1 \in \{0, 1, 2\}$,

and $N_{2,3}(x_1, 0, 0) = N_{2,3}(x_1, 2, 1) = (2/3 - 1/3) = 1/3$,
$N_{2,3}(x_1, 0, 1) = N_{2,3}(x_1, 2, 0) = (0 - 1/3) = -1/3$ for all $x_1 \in \{0, 1, 2\}$.
Table 4.1 gives the set of values for $N_{1,2,3}$ for all (x_1, x_2, x_3). The entries correspond to the design points giving those values. For instance,

$$\begin{aligned} N_{1,2,3}(0,0,0) &= N(0,0,0) - N_{1,2}(0,0,0) - N_{1,3}(0,0,0) - N_{2,3}(0,0,0) \\ &\quad - N_1(0,0,0) - N_2(0,0,0) - N_3(0,0,0) - N_\emptyset \\ &= 1 - 1/3 - 1/3 = 1/3. \end{aligned}$$

The remaining values in Table 4.1 are calculated similarly.

Table 4.1: $N_{1,2,3}$ values over all combinations of (x_1, x_2, x_3).

$N_{1,2,3}$				
-2/3	-1/3	0	1/3	2/3
(1,2,1)	(0,0,2)	(0,0,1)	(0,0,0)	(0,1,2)
(2,0,0)	(0,1,0)	(0,2,0)	(0,2,1)	(1,1,1)
	(0,1,1)	(1,0,1)	(1,0,0)	(1,2,2)
	(0,2,2)	(1,2,0)	(2,2,1)	(2,0,2)
	(1,0,2)	(2,0,1)		(2,1,0)
	(1,1,0)	(2,2,0)		
	(1,1,2)			
	(2,1,1)			
	(2,1,2)			
	(2,2,2)			

If we return to Equation (4.1), we have the decomposition of $N(0,0,0)$ as

$$\begin{aligned} N(0,0,0) &= N_\emptyset + N_1(0,0,0) + N_2(0,0,0) + N_3(0,0,0) + N_{1,2}(0,0,0) + \\ &\quad N_{1,3}(0,0,0) + N_{2,3}(0,0,0) + N_{1,2,3}(0,0,0) \\ &= 1/3 + 0 + 0 + 0 + 0 + 0 + 1/3 + 1/3 \\ &= 1. \end{aligned}$$

The J-characteristics are calculated by Equation (4.4) as $J_u(x_1, x_2, x_3) = 27N_u(x_1, x_2, x_3)$ for all $u \subseteq Z_m$ and $x_1, x_2, x_3 \in \{0, 1, 2\}$. The J-characteristics for the individual factors, $J_1(x_1,x_2,x_3), J_2(x_1,x_2,x_3)$ and $J_3(x_1,x_2,x_3)$, are 0, which reflects the fact we have a balanced design. For the two-factor interactions, $J_{1,2}(x_1,x_2,x_3)$ and $J_{1,3}(x_1,x_2,x_3)$ are 0 for all x_1,x_2 and x_3, whereas $J_{2,3}(x_1,x_2,x_3)$ is non-zero for some x_1,x_2 and x_3. This indicates column 1 is orthogonal to columns 2 and 3, but columns 2 and 3 are not orthogonal. In this example, $J_{1,2,3}(x_1,x_2,x_3)$ takes on five different values.

In the next subsection, we take a closer look at some of the properties of J-characteristics.

4.2.2 Properties of J-Characteristics

The results in the following Lemma provide some insight into the usefulness of the J-characteristics.

Lemma 4.1. *(i) A design is uniquely determined by its J-characteristics, and vice versa.*

(ii) A design is a full factorial or several replicates of it if and only if $J_u = 0$ for all nonempty subsets u of Z_m.

(iii) A design is an orthogonal array of strength t if and only if $J_u = 0$ for all nonempty subsets u such that $|u| \leq t$.

(iv) when $s_1 = \cdots = s_m = s$ and the design is an orthogonal array of strength t, then J_u must be a multiple of s^t.

If our interest lies in the lower dimensions of the factors, Lemma 4.1 tells us that for smaller values of $|u|$, a good design should have J_u values as close to zero as possible. As was seen in Example 4.1, we need to be mindful that unlike the two-level case, J_u can take on a number of different values for a given u. The construction of the J-characteristics gives a convenient way to summarize the size of J_u. By using the ANOVA decomposition for the frequency of the design points, we have

$$\sum_{x_1=0}^{s_1-1} \cdots \sum_{x_m=0}^{s_m-1} [s_1 \cdots s_m N(x_1,\ldots,x_m)]^2 = \sum_{u \subseteq Z_m} \sum_{x_1=0}^{s_1-1} \cdots \sum_{x_m=0}^{s_m-1} [J_u(x_1,\ldots,x_m)]^2, \quad (4.8)$$

as $\sum_{x_1=0}^{s_1-1} \cdots \sum_{x_m=0}^{s_m-1} J_u(x_1, \ldots, x_m) J_v(x_1, \ldots, x_m) = 0$ for $u \neq v$ because the ANOVA decomposition is an orthogonal decomposition.

If we let

$$S_u^2 = \sum_{x_1=0}^{s_1-1} \cdots \sum_{x_m=0}^{s_m-1} [J_u(x_1, \ldots, x_m)]^2 \tag{4.9}$$

represent the size of J_u, then the decomposition becomes

$$\sum_{x_1=0}^{s_1-1} \cdots \sum_{x_m=0}^{s_m-1} [s_1 \cdots s_m N(x_1, \ldots, x_m)]^2 = \sum_{u \subseteq Z_m} S_u^2. \tag{4.10}$$

The values of S_u^2 in (4.9) will be used throughout the remainder of the chapter as a means of evaluating the properties of a design, and will be justified in the next section.

Example 4.1 (continued). Calculating the left-hand side of Equation (4.10),

$$\sum_{x_1=0}^{2} \sum_{x_2=0}^{2} \sum_{x_3=0}^{2} [3^3 N(x_1, x_2, x_3)]^2 = 6561.$$

Using Equation (4.9), $S_\emptyset^2 = 2187$, $S_1^2 = S_2^2 = S_3^2 = 0$, $S_{1,2}^2 = S_{1,2}^2 = 0$, $S_{2,3}^2 = 972$ and $S_{1,2,3}^2 = 3402$. Taking the right-hand side of (4.10), we have $\sum_{u \subseteq Z_3} S_u^2 = 6561$.

The quantities of S_u^2 are the partitioned sums of squares for the factorial effects in an ANOVA. The m factors are a full factorial design corresponding to all level combinations of $x_1, \ldots, x_m$ and the response variable is $s_1 \cdots s_m N(x_1, \ldots, x_m)$. The sum of squares component S_u^2 corresponds to the factors in u. In the standard usage of ANOVA for a general factorial design, we typically want lower order effects with a large sum of squares. For our situation, the lower order effects should be as small as possible, ideally zero, as these represent the projections onto factors in u. An example of analyzing a design using this ANOVA approach on the frequency of design points will be given in Section 4.4. We now explore the justification for the use of S_u^2.

4.3 Criteria of Aberration and their Justification

In this section, we introduce a means of ranking designs that makes use of the J-characteristics through S_u^2. This will be done through the variability of design points and contamination from nonnegligible interactions.

4.3.1 G_2-Aberration

If we consider Equation (4.10), moving the grand mean to the left-hand side, we have the total corrected sum of squares in an ANOVA. This quantity represents the total variability of the design points. The S_u^2 terms give the partitioned sums of squares corresponding to the different effects in the design. For any subset v of Z_m, if we want to look at the variability of the projection onto the factors in v we can look at the S_u^2 for $u \subseteq v$. To ensure good projections onto lower dimensions, we want the S_u^2 to be small when $|u|$ is small. Based on J-characteristics, we now define the criterion of minimum G_2-aberration. Let

$$B_j = \sum_{|u|=j} S_u^2. \tag{4.11}$$

Then minimum G_2-aberration is to sequentially minimize $B_1, \ldots, B_m$. Similar to Tang (2001), Ai and Zhang (2004) gave a projection justification for generalized minimum aberration, which we discuss shortly, based on their version of the J-characteristics. Their justification used the same idea of the variability of design points. Our measure of S_u^2 is equivalent to the sum of the squared J-characteristics used by Ai and Zhang (2004), but in our derivation, S_u^2 is directly linked to the distribution of design points.

The J-characteristics of Ai and Zhang (2004) are based on the orthonormal contrasts as used in Xu and Wu (2001) in defining generalized minimum aberration (GMA). Generalized minimum aberration sequentially minimizes $A_3, A_4, \ldots$ for

$$A_j = n^{-2} \sum_{k=1}^{m_j} \left| \sum_{i=1}^{n} x_{ik}^{(j)} \right|^2, \tag{4.12}$$

where $x_{ik}^{(j)}$ is the ith component of the kth factor effect contrast, of which there are m_j. We have

$$A_j = B_j/(s_1 \cdots s_m n^2), \tag{4.13}$$

for $j = 1, \ldots, m$. The value of A_j measures the overall aliasing between the j-factor effects and the grand mean.

In regards to robustness to nonnegligible interactions, consider the ANOVA model written as

$$Y = X_0\beta_0 + X_1\beta_1 + X_2\beta_2 + \cdots + X_m\beta_m + \epsilon \tag{4.14}$$

where Y is the vector of n observations, X_i is the matrix of contrast coefficients for the i-factor interactions with β_i the corresponding effects, β_0 the grand mean and X_0 the vector of 1's, and ϵ the vector of independent random errors. The main effects model for (4.14) is given by

$$Y = X_0\beta_0 + X_1\beta_1 + \epsilon. \tag{4.15}$$

Using model (4.15), the least squares estimate of β_1 is given by $\hat{\beta}_1 = (X_1^T X_1)^{-1} X_1^T Y$. This estimate is unbiased for model (4.15), but if model (4.14) is the true model, we have

$$E(\hat{\beta}_1) = \beta_1 + C_2\beta_2 + \cdots + C_m\beta_m,$$

where $C_i = (X_1^T X_1)^{-1} X_1^T X_i$ are the aliasing matrices for $i = 2, \ldots, m$. In the situation our design D is an orthogonal array, C_i simplifies to $C_i = (1/n)X_1^T X_i$. By the hierarchical ordering principle, lower order effects are more likely to be important than higher order effects, and effects of the same order are equally likely to be important. If we want to minimize the contamination from nonnegligible interactions, we would like to sequentially minimize $C_2, C_3, \ldots$. To sequentially minimize $\|C_2\|^2, \|C_3\|^2, \ldots$ gives the minimum contamination criterion such as in Tang and Deng (1999), Xu and Wu (2001), and Steinberg and Bursztyn (2001). Xu, Cheng and Wu (2004) suggested using this approach for screening with three-level designs. In Xu and Wu (2001), they showed that sequentially minimizing $A_3, A_4, \ldots$ is equivalent to sequentially minimizing $\|C_2\|^2, \|C_3\|^2, \ldots$. This means that for orthogonal arrays ($B_1 = B_2 = 0$), G_2-aberration sequentially minimizes $\|C_2\|^2, \|C_3\|^2, \ldots$.

The entries of C_i are dependent on the choice of orthonormal contrasts, as are the components $x_{ik}^{(j)}$ in (4.12). Our definition of B_j in (4.11) does not depend on orthonormal contrasts. As noted in Xu and Wu (2001), $\|C_i\|^2$ is independent of the choice of orthonormal contrasts. If we look at the projection design onto the factors referenced by v, note that the projected A_j values and S_u^2 remain the same regardless of the choice of orthonormal contrasts for all $u \subseteq v$.

In related work, Liu, Fang, and Hickernell (2006) studied the connection between different criteria for asymmetrical fractional factorial designs showing that generalized minimum aberration is equivalent to the minimum χ^2-criterion (an extension of

Yamada and Matsui, 2002) and minimum projection uniformity (Hickernell and Liu, 2002).

4.3.2 G-Aberration

By the effect sparsity principle, few of the effects are expected to be significant. A more conservative approach than G_2-aberration is to minimize the worst case scenario. Assuming interactions involving three or more factors are negligible, if $C_2 = [c_{ij}]$, in order to best protect estimation of the main effects from nonnegligible two-factor interactions, we can try to sequentially minimize the largest of the c_{ij}^2's. However, the $c'_{ij}s$ are based on a set of orthonormal contrasts. Without prior knowledge or choice of contrasts, a good surrogate is to sequentially minimize the frequency of S_u^2 starting from the largest value among with smaller values of $|u|$, effectively collecting the $\sum c_{ij}^2$ for i, j corresponding to factors in u. That is, we first sequentially minimize the largest S_u^2 values for the smallest nonzero $|u|$.

Sequentially minimizing the frequencies of S_u^2 can also be thought of from the projection standpoint. From this view, minimum G-aberration sequentially minimizes the worst of the lower-dimensional projections as measured by the variability of the design points. Whereas B_j measures the overall projection properties, the projected S_u^2 values provide a more conservative approach. If we expect that the response is active in a smaller subset of the factors, G-aberration aims to minimize poor projections in smaller subsets. Our concentration for G-aberration is for $|u_0|$ with the first nonzero $S_{u_0}^2$ from $|u_0| = 1, 2, \ldots$. The reason is that if we consider a $|u^*| > |u_0|$, the projection properties are based on S_u^2 for all subsets $u \subseteq u^*$. This causes no problems for G_2-aberration.

Another possibility to define G-aberration is to sequentially minimize the frequency of $|J_u(x_1, \ldots x_m)|$'s. In the two-level case, defining G-aberration with either $|J_u(x_1, \ldots x_m)|$ or S_u^2 are equivalent, but not so when there are more than two levels due to $|J_u(x_1, \ldots x_m)|$ taking on different values. This approach can also be used to further differentiate designs with the same G-aberration as defined using S_u^2. Based on the direct relationship to the variability of the design points, for the remainder of this chapter we use S_u^2 in relation to G-aberration.

Our definition of minimum G-aberration is equivalent to the definition for two-level designs. For multi-level designs, using the frequency of S_u^2 is more general than the projection aberration criterion in Xu, Cheng and Wu (2004), in which they sequentially minimized the projected A_3 values starting from the largest as a means of combining factor screening and interaction detection for three-level designs.

4.3.3 $G_{2(i)}$-Aberration

While minimum G_2-aberration considers an overall measure of lower dimensions and minimum G-aberration takes a conservative approach, the impact on individual factors is not taken into account from both the contamination and projection perspectives. Ideally, given any factor, we should have good projection properties for all those projections involving that factor. Equivalently, we want the contamination to the estimation of each factor minimized so that no individual main effect has a substantial amount of contamination from nonnegligible interactions. Minimum $G_{2(i)}$-aberration aims to address this by sequentially minimizing the largest values of

$$S_{i:k}^2 = \sum_{\{u:i\in u,|u|=k\}} S_u^2 \tag{4.16}$$

for $i = 1, \ldots, m$, beginning from $k = 1, 2, \ldots$. In general, we expect that a design with smaller G_2-aberration should also have smaller $G_{2(i)}$-aberration as G_2-aberration is based on the sum of the $S_{i,k}^2$'s.

Using $G_{2(i)}$ can be thought of as somewhere between G and G_2-aberration. Not as conservative as G-aberration, it still gives us the flexibility to distinguish between designs having the same G_2-aberration. When we reach a point that higher strength is not possible, the aim is to have the contamination or projection properties spread out evenly among the factors. The concept of $G_{2(i)}$-aberration can be extended to $G_{(i)}$-aberration whereby we use the frequency vectors of S_u^2 for each factor and rank these according to G-aberration.

4.3.4 Considerations from Previous Chapters

In Chapter 2, we considered a case where certain two-factor interactions were safely assumed to be negligible before conducting the experiment while other two-factor interactions were nonnegligible. If it were known ahead of time that certain factors do not interact, then G, G_2 and $G_{2(i)}$-aberration can all be adapted to account for this information. Instead of using B_j, we can give different weights to S_u^2 to account for this information.

As an example, similar to Chapter 2, consider an orthogonal array with a subset of factors T_2 such that the interactions within $T_2 \times T_2$ are assumed negligible. For $|u| = 1, 2$, all $S_u^2 = 0$. For $|u| = 3$, any S_u^2 where $u \subseteq T_2$ (ie. all three factors are in T_2) can be assigned zero weight. In addition, the S_u^2 where u has two factors from T_2 would be considered less serious than the S_u^2 with only one or zero factors from T_2. There are then a number of ways to rank designs using this information, whether by creating a metric with a weighted sum of the S_u^2, or a partition based on the factors in the subset u. An example of this partitioning was used in Chapter 2 with the A_{40}, A_{31} and A_{22} terms.

One of the appeals of the multi-level J-characteristics and the use of S_u^2 is that they are independent of the choice of orthonormal contrasts. Our concern is with the general structure of the design in lower dimensions. In Chapter 3, we had a particular model in mind with a set of orthonormal contrasts and specified two-factor interaction components. In such a situation, the J-characteristics from Ai and Zhang (2004) may be more appropriate as they are defined based on a set of orthonormal contrasts. The S_u^2 terms can be thought of as summarizing all of the interaction components of the factors in u, when our interest may only be in a subset of these components.

4.4 An ANOVA Example

We now analyze two designs based on their S_u^2 values calculated using a standard ANOVA from the statistical software package R. Consider the $OA(27, 3^6)$'s denoted by D_1 and D_2 in Table 4.2. The design D_1 is a non-regular design, while D_2 is regular. To analyze these designs using an ANOVA, our design matrix is the full factorial design for $m = 6$ factors with 3 levels each, and the response y is the number of occurrences of a particular row multiplied by 3^6 (see the left-hand side of equation (4.5)). If we let $x_1, \ldots, x_6$ represent the six factors in the full factorial, and y be as just described, we can use the R command

```
aov(y∼x1*x2*x3*x4*x5*x6)
```

to calculate S_u^2 for all $u \subseteq Z_m$ and investigate the properties of the design.

Table 4.3 gives the results of the ANOVA for the three-factor interactions for both designs and Table 4.4 for interactions involving four or more factors. The interaction listed refers to the set u. For example, x1:x2:x3 refers to $u = \{1, 2, 3\}$. The degrees of freedom for the interaction, represented by 'df' in the table, give the number of components these interactions have if main effects were broken into orthogonal components, as was done in the previous chapter. We present interactions involving three or more factors as for the main effects and two-factor interactions ($|u| = 1, 2$) $S_u^2 = 0$. Then for both designs, B_j from (4.11) is zero for $j = 1, 2$. In a typical ANOVA, we look for significant effects through larger sums of squares for lower-order effects. In this situation, we want the lower-order terms as close to zero as possible. Design D_2 has less G_2-aberration than D_1 as we have $B_3 = 3188646$ for D_2 versus $B_3 = 4015332$ for D_1. However, D_1 has less G-aberration than D_2 as seen in Table 4.5 with the frequency of S_u^2 values for $|u| = 3$. Table 4.6 gives the $S_{i:3}^2$ as calculated by equation (4.16). By these values, design D_1 has less $G_{2(i)}$-aberration than D_2. From Table 4.6, we see that while the sum of the S_u^2 terms is smaller for D_2, D_1 has more of an even spread of the contribution of each factor to B_3.

Table 4.2: Two $OA(27, 3^6)$, D_1 a non-regular design and D_2 regular.

$$D_1 = \begin{pmatrix} 001001 \\ 011221 \\ 022201 \\ 000022 \\ 010100 \\ 021010 \\ 002212 \\ 012120 \\ 020112 \\ 100200 \\ 112011 \\ 122022 \\ 102110 \\ 111102 \\ 120220 \\ 101111 \\ 111222 \\ 120001 \\ 200121 \\ 212000 \\ 221102 \\ 201020 \\ 210012 \\ 222121 \\ 202202 \\ 210211 \\ 221210 \end{pmatrix} \quad D_2 = \begin{pmatrix} 000000 \\ 001101 \\ 002202 \\ 010120 \\ 011221 \\ 012022 \\ 020210 \\ 021011 \\ 022112 \\ 100111 \\ 101212 \\ 102010 \\ 110201 \\ 111002 \\ 112100 \\ 120021 \\ 121122 \\ 122220 \\ 200222 \\ 201020 \\ 202121 \\ 210012 \\ 211110 \\ 212211 \\ 220102 \\ 221200 \\ 222001 \end{pmatrix}$$

For D_1, the vector of B_j values are $(0, 0, 4015332, 5078214, 3306744, 1417176)$ and for D_2 we have $(0, 0, 3188646, 6377292, 3188646, 1062882)$. It should be noted that $\sum_{j=0}^{6} B_j = 14348907$ for both designs. If we return to Equation (4.9), we see that $\sum_j B_j$ will be the same for two designs with the same factor and level combinations, provided they have no repeated runs. From this perspective, minimizing B_j for smaller values of j is equivalent to maximizing B_j for larger values of j.

Table 4.3: S_u^2 (SS) for 3-factor interactions for D_1 and D_2.

Interaction	df	D_1 S_u^2	D_2 S_u^2
x1:x2:x3	8	157464	0
x1:x2:x4	8	236196	0
x1:x3:x4	8	236196	0
x2:x3:x4	8	157464	0
x1:x2:x5	8	236196	1062882
x1:x3:x5	8	157464	0
x2:x3:x5	8	236196	0
x1:x4:x5	8	157464	0
x2:x4:x5	8	236196	0
x3:x4:x5	8	157464	0
x1:x2:x6	8	157464	0
x1:x3:x6	8	236196	1062882
x2:x3:x6	8	236196	0
x1:x4:x6	8	236196	0
x2:x4:x6	8	236196	1062882
x3:x4:x6	8	236196	0
x1:x5:x6	8	236196	0
x2:x5:x6	8	157464	0
x3:x5:x6	8	157464	0
x4:x5:x6	8	157464	0

4.5 Searching for Designs

In this section we search for designs using S_u^2 as discussed in Section 4.3 to find minimum G and G_2-aberration designs among orthogonal arrays. As was the approach in Chapter 3, we start with an existing orthogonal array and search among smaller subsets for the best designs. We firstly examine orthogonal arrays of 18 runs and then move to 27 runs. Because we are dealing with orthogonal arrays, our attention will be placed on 3-factor projections through S_u^2 with $|u| = 3$. For convenience, throughout this section we report results in terms of projected A_3 values instead of S_u^2 to remain consistent with previous literature. We use projected A_3 and S_u^2 interchangeably in

Table 4.4: S_u^2 (SS) for interactions with greater than 3 factors for D_1 and D_2.

Interaction	df	D_1	D_2
x1:x2:x3:x4	16	275562	1062882
x1:x2:x3:x5	16	393660	0
x1:x2:x4:x5	16	314928	0
x1:x3:x4:x5	16	354294	1062882
x2:x3:x4:x5	16	393660	1062882
x1:x2:x3:x6	16	393660	0
x1:x2:x4:x6	16	314928	0
x1:x3:x4:x6	16	236196	0
x2:x3:x4:x6	16	314928	0
x1:x2:x5:x6	16	275562	0
x1:x3:x5:x6	16	275562	0
x2:x3:x5:x6	16	393660	1062882
x1:x4:x5:x6	16	393660	1062882
x2:x4:x5:x6	16	275562	0
x3:x4:x5:x6	16	472392	1062882
x1:x2:x3:x4:x5	32	551124	0
x1:x2:x3:x4:x6	32	590490	1062882
x1:x2:x3:x5:x6	32	551124	1062882
x1:x2:x4:x5:x6	32	629856	1062882
x1:x3:x4:x5:x6	32	551124	0
x2:x3:x4:x5:x6	32	433026	0
x1:x2:x3:x4:x5:x6	64	1417176	1062882

this section.

Before moving on, we clarify the calculations used in the search for designs in the remainder of this chapter. In Section 4.4, the calculation of S_u^2 was based upon an ANOVA analysis of a full factorial design with the frequency of design points as the response. Since we have only a fraction of the runs from the full factorial, many components of the response vector were zero. In the case of an $OA(27, 3^{13})$, the full factorial would have $3^{13} = 1594323$ runs, but the response vector would have at most 27 nonzero entries. Because of the ANOVA decomposition, the calculation of S_u^2 can be done with simpler formulas through the definition in Section 4.2. The simplified

Table 4.5: Tabulation of S_u^2 for $|u| = 3$ for D_1 and D_2.

	S_u^2 frequency			
design	0	157464	236196	1062882
D_1	0	9	11	0
D_2	17	0	0	3

Table 4.6: Tabulation of $S_{i,3}^2$ for D_1 and D_2.

	$S_{1,3}^2$	$S_{2,3}^2$	$S_{3,3}^2$	$S_{4,3}^2$	$S_{5,3}^2$	$S_{6,3}^2$
D_1	2047032	2047032	1968300	2047032	1889568	2047032
D_2	2125764	2125764	1062882	1062882	1062882	2125764

formulation allows easier calculation of S_u^2 without creating the full factorial and can be thought of as the manual computing formulas for the sum of squares components. Since many of the terms are 0, the computational savings is substantial as the manual computation formulas allow us to ignore any of the terms that are 0 in the calculation.

We examine designs by presenting our results beginning with the maximum number of factors from the larger orthogonal array. With the S_u^2 calculated for all $u \subseteq Z_m$, we can analyze a subset v of $m^* \leq m$ factors by simply looking at the S_u^2 for $u \subseteq v$. If we relate this approach back to linear regression, we are dealing with a best subset selection process. One could then use forward or backwards selection using the error sum of squares (SSE) such that the number of factors in the model is of size m^*. In our situation, the SSE would be formed by taking the sum of all S_u^2 such that u contains a factor outside of the current design. Instead of trying to minimize the SSE, we want to maximize it. By starting with the larger design and calculating the full set of S_u^2 values, it is convenient to start with the larger designs and sequentially remove factors.

4.5.1 Designs with 18 Runs

Table 4.7 provides the three non-isomorphic 18-run designs with 7 three-level factors as identified by Evangelaras, Koukouvinos and Lappas (2007). We refer to these designs as $N18.1$, $N18.2$ and $N18.3$. Schoen (2009) also identified the three non-isomorphic 18-run designs and presented the projected A_3 frequency, which is equivalent to our definition of G-aberration.

For $m = 7$, all three designs are the same by minimum G_2-aberration with an overall $A_3 = 22$. Table 4.8 shows the frequencies for the projected A_3 values. According to minimum G-aberration, we would rank the designs in order from best to worst as $N18.2$, $N18.1$, and $N18.3$. Table 4.9 gives the values for $A_{3(i)}$, where $A_{3(i)}$ corresponds to the sum of the projected A_3 values that involve factor i (the equivalent of $S^2_{i:k}$). The ranking according to $G_{2(i)}$-aberration is the same as G-aberration.

Table 4.10 shows the results for the top designs found by removing one column from the parent designs. For each parent design, the design listed is best in terms of G, G_2, and $G_{2(i)}$-aberration. For $m = 6$, the rank is reversed from the case of $m = 7$. Table 4.11 gives the $A_{3(i)}$ values, where i refers to the column in the resulting design with $m = 6$. By a theoretical result in Xu (2003), any subdesign of $m \leq 6$ from $N18.3$ with column 4 removed has minimum G and G_2-aberration.

In terms of overall A_3, the change in the ranking was expected based on the results for $m = 7$ from Table 4.9. For $N18.3$, we find that the largest value for $A_{3(i)}$ occurs for $i = 4$. If we remove factor 4 from this design, the overall A_3 value for the subdesign will be the overall A_3 from the full design minus all projected A_3 values that include factor 4, which is exactly what is calculated by $A_{3(4)}$. That is, given a design D with k factors, if we remove factor i, the overall A_3 for the subdesign is $A_3(D) - A_{3(i)}$, where $A_3(D)$ is the overall A_3 for D.

The above observations have some practical implications. Intuitively, we might expect that selecting the best larger design as a parent design may lead to better subdesigns. As we have seen in this example, this is not necessarily the case, as all of the best subdesigns come from the parent design with the worst ranking for $m = 7$. This also has implications in constructing designs in a columnwise manner. If our goal is for $m = 7$ factors, design $N18.3$ would be selected as the best design for $m \leq 6$, and the worst for $m = 7$. Computer algorithms that create designs by sequentially adding columns can fall prey to creating less appealing designs in such a situation.

We conclude our studies on 18-run designs with an interesting case. For $N18.2$, with $m = 6$ with factor 1 removed, all projections onto $m = 5$ factors have the same $A_3 = 6$, and projected A_3 frequency of 4 with 0.5 and 6 with 2/3 (recall that $A_{3(i)}$ is the same for all i in Table 4.11), as shown in Table 4.12. At $m = 6$, the $A_{3(i)}$ values make no distinction between any of the remaining factors in terms of G, G_2, or $G_{2(i)}$. However, looking at the $A_{3(i)}$ for $m = 5$ in Table 4.13, $G_{2(i)}$-aberration does make a separation between designs. Table 4.14 shows the projected A_3 frequencies for $m = 4$. For design $N18.2$, if we want a design for $m = 5$, we should remove columns 1 and 2, whereas if we want $m = 4$, we would be removing columns 1 and 3 at $m = 5$. If we were only using G and G_2-aberration and the ultimate goal was $m = 4$, the removal of columns at $m = 6$ to $m = 5$ would find no difference between removing columns 1 and 2 versus columns 1 and 3. In this case, $G_{2(i)}$ provides insight into the lower dimensions, and depending on the desired m, we want to choose designs with worst $G_{2(i)}$. Before we move on to 27 run designs, we state this more precisely.

Proposition 4.1. *Let D_1 and D_2 be two designs with k factors having the same G_2-aberration such that $A_3(D_1) = A_3(D_2)$. If D_1 has less $G_{2(i)}$-aberration than D_2, then the best subdesigns of D_1 and D_2 with $k-1$ factors in terms of A_3, D_1^- and D_2^-, are such that $A_3(D_1^-) \geq A_3(D_2^-)$. That is, D_2^- is at least as good as D_1^- in terms of G_2-aberration.*

Proposition 4.2. *Let D_1 and D_2 be two designs with k factors having the same G-aberration such that they have the same frequency of projected A_3 values. If D_1 has smaller $G_{(i)}$-aberration than D_2, then the best subdesigns of D_1 and D_2 with $k-1$ factors in terms of G-aberration, D_1^- and D_2^-, are such that D_2^- is at least as good as D_1^- in terms of G-aberration.*

Table 4.7: Three nonisomorphic $OA(18, 3^7)$, $N18.1$, $N18.2$, and $N18.3$.

$$N18.1 = \begin{pmatrix} 0000000 \\ 0111100 \\ 0211211 \\ 0022112 \\ 0122021 \\ 0200222 \\ 1010121 \\ 1110012 \\ 1221020 \\ 1021202 \\ 1102210 \\ 1202101 \\ 2012220 \\ 2120201 \\ 2212002 \\ 2001011 \\ 2101122 \\ 2220110 \end{pmatrix} \quad N18.2 = \begin{pmatrix} 0000000 \\ 0111100 \\ 0211211 \\ 0022112 \\ 0122021 \\ 0200222 \\ 1010121 \\ 1110012 \\ 1221020 \\ 1021202 \\ 1102201 \\ 1202110 \\ 2012220 \\ 2120210 \\ 2212002 \\ 2001011 \\ 2101122 \\ 2220101 \end{pmatrix} \quad N18.3 = \begin{pmatrix} 0000000 \\ 0111110 \\ 0211021 \\ 0022122 \\ 0122201 \\ 0200212 \\ 1010101 \\ 1110222 \\ 1221002 \\ 1021210 \\ 1102011 \\ 1202120 \\ 2012012 \\ 2120020 \\ 2212200 \\ 2001221 \\ 2101102 \\ 2220111 \end{pmatrix}$$

Table 4.8: Frequency of projected A_3 values for $N18.1$, $N18.2$ and $N18.3$ and $m = 7$.

	$\frac{1}{2}$	$\frac{2}{3}$	1	2	A_3
$N18.1$	20	12	2	1	22
$N18.2$	16	18	0	1	22
$N18.3$	28	0	6	1	22

Table 4.9: Tabulation of $A_{3(i)}$ for $N18.1$, $N18.2$ and $N18.3$ and $m = 7$.

	$A_{3(1)}$	$A_{3(2)}$	$A_{3(3)}$	$A_{3(4)}$	$A_{3(5)}$	$A_{3(6)}$	$A_{3(7)}$
$N18.1$	9.67	9	10.67	9.67	9	9	9
$N18.2$	10	9	10	10	9	9	9
$N18.3$	9	9	9	12	9	9	9

Table 4.10: Frequency of projected A_3 values for $N18.1$, $N18.2$ and $N18.3$ and $m = 6$.

	removed	$\frac{1}{2}$	$\frac{2}{3}$	1	2	A_3
$N18.1$	3	12	8	0	0	11.33
$N18.2$	1	8	12	0	0	12
$N18.3$	4	20	0	0	0	10

4.5.2 Designs with 27 Runs

In this Section we search through the design catalog of Evangelaras, Koukouvinos and Lappas (2011) of 129 saturated orthogonal arrays with 27 runs to find the minimum G and G_2-aberration designs among these. In this chapter, we refer to the i'th design from the catalog as $D27.i$. Among the saturated designs, we look for the best subdesigns in terms of G and G_2-aberration. As discussed previously, since we are starting with saturated designs and looking for subdesigns from these, our results will be presented starting from $m = 13$ and moving down sequentially. Xu, Cheng and Wu (2004) did a complete search of 27-run orthogonal arrays and found there are eight distinct values that the projected A_3's can take: $(0, 8/27, 4/9, 14/27, 2/3, 20/27, 10/9, 2)$. Our results will be presented with the projected A_3 frequency vector for each design.

Table 4.15 summarizes the saturated orthogonal arrays ordered in terms of G-aberration. All saturated orthogonal arrays have an overall $A_3 = 104$. The worst design in terms of G-aberration is the regular design. Among all these designs, only $D27.1$ and $D27.24$ do not contain a projected A_3 value of 2, corresponding to full aliasing. All of the designs have the same A_3 values of 80 and 60 for $m = 12$ and

Table 4.11: Tabulation of $A_{3(i)}$ for $N18.1$, $N18.2$ and $N18.3$ and $m = 6$.

	removed	$A_{3(1)}$	$A_{3(2)}$	$A_{3(3)}$	$A_{3(4)}$	$A_{3(5)}$	$A_{3(6)}$
$N18.1$	3	5.67	5.67	5.67	5.67	5.67	5.67
$N18.2$	1	6	6	6	6	6	6
$N18.3$	4	5	5	5	5	5	5

Table 4.12: Frequency of projected A_3 values for $N18.1$, $N18.2$ and $N18.3$ and $m = 5$.

	removed	$\frac{1}{2}$	$\frac{2}{3}$	1	2	A_3
$N18.1$	$(1,3)$	6	4	0	0	5.67
$N18.2$	$(1,2)$	4	6	0	0	6
$N18.2$	$(1,3)$	4	6	0	0	6
$N18.3$	$(1,4)$	10	0	0	0	5

$m = 11$, respectively. Tables 4.16 and 4.17 give the top 30 designs in terms of G-aberration. The numbers under **removed** refer to the columns to be removed from the parent design to get a resulting design with the given projected A_3 frequency vector and overall A_3 value as provided. As was the case for $m = 13$, only designs $D27.1$ and $D27.24$ do not have a projected A_3 with a value of 2. The frequency vector of projected A_3 values for each factor is not necessarily the same, as we see some differences in the frequency vectors for $m = 12$ from the same parent design.

At $m = 10$, we begin to see differentiation between designs in terms of G_2-aberration. Table 4.18 provides the top designs in terms of G-aberration, while Table 4.19 in terms of G_2-aberration. For designs having the same overall A_3, our secondary ranking is that of G-aberration. Deng and Tang (2002) showed that for two-level designs with small run sizes, the ranking between G and G_2-aberration is generally consistent. As we see at $m = 10$, this is not the case for 27 run designs. The best G_2-aberration designs have an overall A_3 value of 42, and the best of the G_2-aberration designs in terms of G-aberration does not appear on the list of top G-aberration designs. In particular, all contain projected A_3 values of 2.

We continue to see this gap between G and G_2-aberration designs for $9 \geq m \geq 5$

Table 4.13: Tabulation of $A_{3(i)}$ for $N18.1$, $N18.2$ and $N18.3$ and $m = 5$.

	removed	$A_{3(1)}$	$A_{3(2)}$	$A_{3(3)}$	$A_{3(4)}$	$A_{3(5)}$
$N18.1$	$(1,3)$	3.33	3.67	3.33	3.33	3.33
$N18.2$	$(1,2)$	3.5	3.5	3.67	3.67	3.67
$N18.2$	$(1,3)$	3.5	4	3.5	3.5	3.5
$N18.3$	$(1,4)$	3	3	3	3	3

Table 4.14: Frequency of projected A_3 values for $N18.1$, $N18.2$ and $N18.3$ and $m = 4$.

	removed	$\frac{1}{2}$	$\frac{2}{3}$	1	2	A_3
$N18.1$	$(1,3,4)$	4	0	0	0	2
$N18.2$	$(1,2,5)$	2	2	0	0	2.33
$N18.2$	$(1,3,4)$	4	0	0	0	2
$N18.3$	$(1,2,4)$	4	0	0	0	2

as seen in tables 4.20 - 4.29. At $m = 4$, the best designs for both G and G_2-aberration are the strength three $OA(27, 3^4)$, and not presented here. For $m \geq 6$, we find that the top G_2 designs always have at least one projected A_3 value of 2. The best G-aberration designs tend to come from the same parent designs as we reduce the number of factors. This is also true of the G_2-aberration designs. However, we do not generally see overlap between the top parent designs.

Using an algorithm from Xu (2002), Xu, Cheng and Wu (2004) found designs for $5 \leq m \leq 10$ with less G-aberration than those found as subdesigns from the saturated orthogonal arrays. It is interesting to note however, that they did not find designs having less G_2-aberration than those found through the saturated orthogonal array.

Before concluding this section, we make some comments on G_{2i}-aberration. For $m = 13$ and $m = 12$, all designs are identical by $G_{2(i)}$-aberration. In both cases, the $A_{3(i)}$ values are the same for all i for each design. For $m = 11$ and $m = 10$, the top G-aberration designs are ranked above the G_2-aberration designs. This can also be seen through Proposition 4.1 and looking at the overall A_3 values for $m = 10$ and $m = 9$ factors. For $m \leq 9$, we tend to see a reversal of this ranking. That is, after this point the better G_2-aberration designs rank above the G-aberration ones. This is due in large part to the fact that the overall A_3 values are much smaller.

Based on these results, in designing a 27-run experiment, the choice between G and G_2-aberration becomes much more of an issue as their ranking is not generally consistent. One particular point of interest is that the minimum G_2-aberration designs tend to have at least one projected A_3 value of 2, corresponding to full aliasing. If G-aberration is more of a priority, one compromise is to take the smallest overall A_3 value from a list of the top ranked G-aberration designs. If G_2-aberration is considered more important to the experimenter, one might take the smallest overall A_3 value such that there are no projected A_3 values of 2.

4.6 Discussion

In this chapter, we introduced J-characteristics for multi-level designs that lead to the definition of G and G_2-aberration. Using the ANOVA decomposition has a broad appeal in that it is easily accessible by a wide range of audiences. The algorithmic construction of nearly-orthogonal arrays (Xu, 2002 and Nguyen and Liu, 2008) typically use an equivalent of G_2-aberration and focus on one dimension. With the use of J-characteristics, considerations can be placed on G-aberration and additional lower dimensional properties. By taking the viewpoint of trying to spread out the design points in lower dimensions, we get intuitively good projection properties that also have robustness properties if we return to the linear model. By viewing G and G_2-aberration from a projection standpoint, we get designs that are appealing as building blocks for other designs such as those described in Bingham, Sitter and Tang (2009) and Lin, Bingham, Sitter and Tang (2010).

Table 4.15: All 129 designs ranked by G-aberration.

design	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
1	(78, 0, 156, 0, 52, 0, 0, 0)	104
24	(52, 130, 0, 52, 0, 52, 0, 0)	104
53	(85, 27, 108, 0, 63, 0, 0, 3)	104
73	(92, 30, 80, 24, 50, 0, 7, 3)	104
74	(92, 30, 80, 24, 50, 0, 7, 3)	104
95	(98, 26, 80, 28, 37, 0, 14, 3)	104
96	(98, 26, 80, 28, 37, 0, 14, 3)	104
97	(98, 26, 80, 28, 37, 0, 14, 3)	104
113	(94, 46, 56, 36, 29, 8, 14, 3)	104
114	(94, 46, 56, 36, 29, 8, 14, 3)	104
115	(94, 46, 56, 36, 29, 8, 14, 3)	104
123	(108, 0, 126, 0, 28, 0, 21, 3)	104
124	(98, 34, 94, 12, 16, 8, 21, 3)	104
125	(98, 34, 94, 12, 16, 8, 21, 3)	104
126	(98, 34, 94, 12, 16, 8, 21, 3)	104
127	(94, 56, 68, 18, 10, 16, 21, 3)	104
128	(94, 56, 68, 18, 10, 16, 21, 3)	104
129	(94, 56, 68, 18, 10, 16, 21, 3)	104
2	(91, 73, 45, 28, 0, 25, 21, 3)	104
9	(117, 27, 0, 54, 81, 0, 0, 7)	104
10	(117, 27, 0, 54, 81, 0, 0, 7)	104
3	(108, 27, 45, 36, 45, 18, 0, 7)	104
4	(108, 27, 45, 36, 45, 18, 0, 7)	104
5	(108, 27, 45, 36, 45, 18, 0, 7)	104
6	(108, 27, 45, 36, 45, 18, 0, 7)	104
7	(108, 27, 45, 36, 45, 18, 0, 7)	104
8	(108, 27, 45, 36, 45, 18, 0, 7)	104
11	(117, 27, 12, 54, 63, 0, 6, 7)	104
12	(117, 27, 12, 54, 63, 0, 6, 7)	104
28	(120, 0, 90, 0, 60, 0, 9, 7)	104
29	(120, 0, 90, 0, 60, 0, 9, 7)	104
30	(120, 0, 90, 0, 60, 0, 9, 7)	104
31	(120, 0, 90, 0, 60, 0, 9, 7)	104
32	(120, 0, 90, 0, 60, 0, 9, 7)	104
33	(126, 0, 72, 0, 72, 0, 9, 7)	104
34	(126, 0, 72, 0, 72, 0, 9, 7)	104
15	(108, 18, 84, 24, 24, 12, 9, 7)	104
16	(108, 18, 84, 24, 24, 12, 9, 7)	104
17	(108, 18, 84, 24, 24, 12, 9, 7)	104
18	(108, 18, 84, 24, 24, 12, 9, 7)	104

Table 4.15: All 129 designs ranked by G-aberration.

design	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
19	$(108, 18, 84, 24, 24, 12, 9, 7)$	104
20	$(108, 18, 84, 24, 24, 12, 9, 7)$	104
21	$(108, 18, 84, 24, 24, 12, 9, 7)$	104
22	$(108, 18, 84, 24, 24, 12, 9, 7)$	104
23	$(108, 18, 84, 24, 24, 12, 9, 7)$	104
25	$(111, 18, 75, 24, 30, 12, 9, 7)$	104
26	$(111, 18, 75, 24, 30, 12, 9, 7)$	104
27	$(111, 18, 75, 24, 30, 12, 9, 7)$	104
13	$(120, 30, 36, 12, 60, 12, 9, 7)$	104
14	$(120, 30, 36, 12, 60, 12, 9, 7)$	104
41	$(120, 18, 42, 36, 51, 0, 12, 7)$	104
42	$(120, 18, 42, 36, 51, 0, 12, 7)$	104
43	$(120, 18, 42, 36, 51, 0, 12, 7)$	104
44	$(120, 18, 42, 36, 51, 0, 12, 7)$	104
45	$(120, 18, 42, 36, 51, 0, 12, 7)$	104
37	$(114, 18, 72, 24, 27, 12, 12, 7)$	104
38	$(114, 18, 72, 24, 27, 12, 12, 7)$	104
39	$(114, 18, 72, 24, 27, 12, 12, 7)$	104
40	$(114, 18, 72, 24, 27, 12, 12, 7)$	104
35	$(114, 18, 72, 24, 27, 12, 12, 7)$	104
36	$(114, 18, 72, 24, 27, 12, 12, 7)$	104
58	$(117, 18, 57, 36, 36, 0, 15, 7)$	104
59	$(117, 18, 57, 36, 36, 0, 15, 7)$	104
54	$(114, 18, 78, 24, 18, 12, 15, 7)$	104
55	$(114, 18, 78, 24, 18, 12, 15, 7)$	104
56	$(114, 18, 78, 24, 18, 12, 15, 7)$	104
57	$(114, 18, 78, 24, 18, 12, 15, 7)$	104
51	$(114, 18, 78, 24, 18, 12, 15, 7)$	104
52	$(114, 18, 78, 24, 18, 12, 15, 7)$	104
46	$(120, 30, 48, 12, 42, 12, 15, 7)$	104
47	$(120, 30, 48, 12, 42, 12, 15, 7)$	104
48	$(120, 30, 48, 12, 42, 12, 15, 7)$	104
49	$(120, 30, 48, 12, 42, 12, 15, 7)$	104
50	$(120, 30, 48, 12, 42, 12, 15, 7)$	104
62	$(120, 18, 54, 36, 33, 0, 18, 7)$	104
63	$(120, 18, 54, 36, 33, 0, 18, 7)$	104
60	$(117, 18, 75, 24, 15, 12, 18, 7)$	104
61	$(117, 18, 75, 24, 15, 12, 18, 7)$	104
64	$(126, 0, 96, 0, 36, 0, 21, 7)$	104
65	$(126, 0, 96, 0, 36, 0, 21, 7)$	104

Table 4.15: All 129 designs ranked by G-aberration.

design	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
66	$(120, 30, 60, 12, 24, 12, 21, 7)$	104
67	$(120, 30, 60, 12, 24, 12, 21, 7)$	104
68	$(120, 30, 60, 12, 24, 12, 21, 7)$	104
71	$(126, 0, 102, 0, 27, 0, 24, 7)$	104
72	$(126, 0, 102, 0, 27, 0, 24, 7)$	104
69	$(129, 0, 93, 0, 33, 0, 24, 7)$	104
70	$(129, 0, 93, 0, 33, 0, 24, 7)$	104
79	$(132, 0, 84, 0, 39, 0, 24, 7)$	104
80	$(132, 0, 84, 0, 39, 0, 24, 7)$	104
81	$(132, 0, 84, 0, 39, 0, 24, 7)$	104
77	$(120, 30, 66, 12, 15, 12, 24, 7)$	104
78	$(120, 30, 66, 12, 15, 12, 24, 7)$	104
75	$(123, 30, 57, 12, 21, 12, 24, 7)$	104
76	$(123, 30, 57, 12, 21, 12, 24, 7)$	104
82	$(126, 0, 108, 0, 18, 0, 27, 7)$	104
83	$(126, 0, 108, 0, 18, 0, 27, 7)$	104
86	$(120, 30, 72, 12, 6, 12, 27, 7)$	104
87	$(120, 30, 72, 12, 6, 12, 27, 7)$	104
84	$(126, 30, 54, 12, 18, 12, 27, 7)$	104
85	$(126, 30, 54, 12, 18, 12, 27, 7)$	104
88	$(123, 30, 69, 12, 3, 12, 30, 7)$	104
89	$(123, 30, 69, 12, 3, 12, 30, 7)$	104
90	$(132, 0, 102, 0, 12, 0, 33, 7)$	104
91	$(132, 0, 102, 0, 12, 0, 33, 7)$	104
92	$(132, 0, 54, 0, 90, 0, 0, 10)$	104
93	$(132, 0, 54, 0, 90, 0, 0, 10)$	104
94	$(132, 18, 30, 36, 45, 0, 15, 10)$	104
98	$(138, 0, 66, 0, 57, 0, 15, 10)$	104
99	$(138, 0, 66, 0, 57, 0, 15, 10)$	104
100	$(138, 0, 96, 0, 12, 0, 30, 10)$	104
101	$(132, 30, 60, 12, 0, 12, 30, 10)$	104
102	$(132, 30, 60, 12, 0, 12, 30, 10)$	104
104	$(126, 0, 108, 0, 36, 0, 0, 16)$	104
105	$(126, 0, 108, 0, 36, 0, 0, 16)$	104
106	$(135, 0, 81, 0, 54, 0, 0, 16)$	104
107	$(135, 0, 81, 0, 54, 0, 0, 16)$	104
103	$(162, 0, 0, 0, 108, 0, 0, 16)$	104
108	$(144, 0, 72, 0, 45, 0, 9, 16)$	104
111	$(144, 0, 90, 0, 18, 0, 18, 16)$	104
109	$(162, 0, 36, 0, 54, 0, 18, 16)$	104

Table 4.15: All 129 designs ranked by G-aberration.

design	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
110	$(162, 0, 36, 0, 54, 0, 18, 16)$	104
112	$(153, 0, 81, 0, 9, 0, 27, 16)$	104
116	$(162, 0, 54, 0, 27, 0, 27, 16)$	104
117	$(162, 0, 54, 0, 27, 0, 27, 16)$	104
118	$(162, 0, 72, 0, 0, 0, 36, 16)$	104
119	$(162, 0, 72, 0, 0, 0, 36, 16)$	104
120	$(180, 0, 0, 0, 81, 0, 0, 25)$	104
121	$(180, 0, 54, 0, 0, 0, 27, 25)$	104
122	$(234, 0, 0, 0, 0, 0, 0, 52)$	104

Table 4.16: Top G-aberration designs for $m = 12$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
1	(1)	(60, 0, 120, 0, 40, 0, 0, 0)	80
24	(1)	(40, 100, 0, 40, 0, 40, 0, 0)	80
53	(1)	(66, 18, 84, 0, 50, 0, 0, 2)	80
73	(3)	(66, 24, 70, 12, 44, 0, 2, 2)	80
74	(3)	(66, 24, 70, 12, 44, 0, 2, 2)	80
73	(2)	(71, 21, 58, 24, 39, 0, 5, 2)	80
74	(8)	(71, 21, 58, 24, 39, 0, 5, 2)	80
73	(1)	(70, 30, 46, 24, 43, 0, 5, 2)	80
74	(13)	(70, 30, 46, 24, 43, 0, 5, 2)	80
73	(8)	(73, 18, 63, 18, 40, 0, 6, 2)	80
74	(1)	(73, 18, 63, 18, 40, 0, 6, 2)	80
95	(2)	(70, 26, 54, 28, 33, 0, 7, 2)	80
96	(3)	(70, 26, 54, 28, 33, 0, 7, 2)	80
97	(13)	(70, 26, 54, 28, 33, 0, 7, 2)	80
73	(7)	(72, 20, 66, 16, 37, 0, 7, 2)	80
74	(5)	(72, 20, 66, 16, 37, 0, 7, 2)	80
113	(2)	(68, 36, 42, 32, 29, 4, 7, 2)	80
114	(3)	(68, 36, 42, 32, 29, 4, 7, 2)	80
115	(13)	(68, 36, 42, 32, 29, 4, 7, 2)	80
95	(3)	(72, 20, 70, 16, 31, 0, 9, 2)	80
96	(13)	(72, 20, 70, 16, 31, 0, 9, 2)	80
97	(3)	(72, 20, 70, 16, 31, 0, 9, 2)	80
113	(3)	(72, 28, 54, 24, 23, 8, 9, 2)	80
114	(7)	(72, 28, 54, 24, 23, 8, 9, 2)	80
115	(2)	(72, 28, 54, 24, 23, 8, 9, 2)	80
95	(1)	(77, 19, 54, 26, 31, 0, 11, 2)	80
96	(6)	(77, 19, 54, 26, 31, 0, 11, 2)	80
97	(8)	(77, 19, 54, 26, 31, 0, 11, 2)	80
113	(1)	(74, 34, 36, 32, 25, 6, 11, 2)	80
114	(8)	(74, 34, 36, 32, 25, 6, 11, 2)	80

Table 4.17: Top G-aberration designs for $m = 11$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
1	(1, 2)	(45, 0, 90, 0, 30, 0, 0, 0)	60
24	(1, 2)	(30, 75, 0, 30, 0, 30, 0, 0)	60
53	(1, 4)	(49, 12, 64, 0, 39, 0, 0, 1)	60
73	(3, 7)	(49, 16, 56, 8, 33, 0, 2, 1)	60
74	(3, 5)	(49, 16, 56, 8, 33, 0, 2, 1)	60
73	(2, 3)	(49, 18, 50, 12, 33, 0, 2, 1)	60
74	(3, 8)	(49, 18, 50, 12, 33, 0, 2, 1)	60
73	(1, 3)	(49, 24, 40, 12, 37, 0, 2, 1)	60
74	(3, 13)	(49, 24, 40, 12, 37, 0, 2, 1)	60
95	(2, 3)	(49, 20, 48, 16, 27, 0, 4, 1)	60
96	(3, 13)	(49, 20, 48, 16, 27, 0, 4, 1)	60
97	(3, 13)	(49, 20, 48, 16, 27, 0, 4, 1)	60
73	(2, 8)	(54, 12, 45, 18, 31, 0, 4, 1)	60
74	(1, 8)	(54, 12, 45, 18, 31, 0, 4, 1)	60
73	(1, 8)	(53, 18, 38, 18, 33, 0, 4, 1)	60
74	(1, 13)	(53, 18, 38, 18, 33, 0, 4, 1)	60
113	(2, 3)	(49, 24, 40, 20, 23, 4, 4, 1)	60
114	(3, 7)	(49, 24, 40, 20, 23, 4, 4, 1)	60
115	(2, 13)	(49, 24, 40, 20, 23, 4, 4, 1)	60
73	(2, 7)	(53, 14, 48, 16, 28, 0, 5, 1)	60
74	(5, 8)	(53, 14, 48, 16, 28, 0, 5, 1)	60
73	(1, 7)	(53, 20, 38, 16, 32, 0, 5, 1)	60
74	(5, 13)	(53, 20, 38, 16, 32, 0, 5, 1)	60
95	(2, 6)	(53, 16, 44, 20, 25, 0, 6, 1)	60
96	(3, 5)	(53, 16, 44, 20, 25, 0, 6, 1)	60
97	(1, 13)	(53, 16, 44, 20, 25, 0, 6, 1)	60
73	(7, 8)	(55, 12, 50, 12, 29, 0, 6, 1)	60
74	(1, 5)	(55, 12, 50, 12, 29, 0, 6, 1)	60
113	(2, 6)	(51, 24, 36, 22, 23, 2, 6, 1)	60
114	(3, 5)	(51, 24, 36, 22, 23, 2, 6, 1)	60

Table 4.18: Top G-aberration designs for $m = 10$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
1	$(1,2,4)$	$(33,0,66,0,21,0,0,0)$	43.3333
1	$(1,2,3)$	$(33,0,65,0,22,0,0,0)$	43.5556
1	$(1,2,11)$	$(32,0,66,0,22,0,0,0)$	44.0000
53	$(1,4,5)$	$(34,8,48,0,30,0,0,0)$	43.7037
24	$(1,2,5)$	$(22,55,0,22,0,21,0,0)$	43.2593
24	$(1,2,3)$	$(22,55,0,21,0,22,0,0)$	43.4815
24	$(1,2,4)$	$(22,54,0,22,0,22,0,0)$	43.7037
24	$(1,2,11)$	$(21,55,0,22,0,22,0,0)$	44.0000
73	$(2,3,7)$	$(34,12,40,8,24,0,2,0)$	43.7037
74	$(3,5,8)$	$(34,12,40,8,24,0,2,0)$	43.7037
73	$(1,3,7)$	$(34,16,32,8,28,0,2,0)$	44.0000
74	$(3,5,13)$	$(34,16,32,8,28,0,2,0)$	44.0000
95	$(2,3,7)$	$(34,12,40,12,18,0,4,0)$	44.0000
96	$(1,3,13)$	$(34,12,40,12,18,0,4,0)$	44.0000
97	$(3,6,13)$	$(34,12,40,12,18,0,4,0)$	44.0000
95	$(2,3,11)$	$(34,16,36,8,22,0,4,0)$	44.0000
96	$(3,4,13)$	$(34,16,36,8,22,0,4,0)$	44.0000
97	$(3,5,13)$	$(34,16,36,8,22,0,4,0)$	44.0000
73	$(2,7,8)$	$(38,8,36,12,22,0,4,0)$	43.7037
74	$(1,5,8)$	$(38,8,36,12,22,0,4,0)$	43.7037
73	$(1,7,8)$	$(38,12,30,12,24,0,4,0)$	43.5556
74	$(1,5,13)$	$(38,12,30,12,24,0,4,0)$	43.5556
113	$(2,3,10)$	$(34,18,30,16,14,4,4,0)$	43.7037
114	$(1,3,7)$	$(34,18,30,16,14,4,4,0)$	43.7037
115	$(2,6,13)$	$(34,18,30,16,14,4,4,0)$	43.7037
113	$(2,3,7)$	$(34,20,28,12,18,4,4,0)$	44.0000
114	$(2,3,7)$	$(34,20,28,12,18,4,4,0)$	44.0000
115	$(2,5,13)$	$(34,20,28,12,18,4,4,0)$	44.0000
95	$(2,6,7)$	$(38,10,36,14,16,0,6,0)$	43.5556
96	$(1,3,5)$	$(38,10,36,14,16,0,6,0)$	43.5556

Table 4.19: Top G_2-aberration designs for $m = 10$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
2	(1, 2, 4)	(49, 20, 22, 8, 0, 8, 11, 2)	42
101	(1, 9, 10)	(49, 20, 22, 8, 0, 8, 11, 2)	42
102	(1, 2, 4)	(49, 20, 22, 8, 0, 8, 11, 2)	42
113	(7, 10, 11)	(49, 20, 22, 8, 0, 8, 11, 2)	42
114	(1, 2, 4)	(49, 20, 22, 8, 0, 8, 11, 2)	42
115	(5, 6, 7)	(49, 20, 22, 8, 0, 8, 11, 2)	42
3	(1, 2, 4)	(54, 9, 0, 18, 36, 0, 0, 3)	42
4	(5, 7, 13)	(54, 9, 0, 18, 36, 0, 0, 3)	42
5	(5, 6, 13)	(54, 9, 0, 18, 36, 0, 0, 3)	42
6	(5, 7, 13)	(54, 9, 0, 18, 36, 0, 0, 3)	42
7	(1, 2, 3)	(54, 9, 0, 18, 36, 0, 0, 3)	42
8	(5, 7, 13)	(54, 9, 0, 18, 36, 0, 0, 3)	42
9	(1, 2, 3)	(54, 9, 0, 18, 36, 0, 0, 3)	42
10	(5, 6, 13)	(54, 9, 0, 18, 36, 0, 0, 3)	42
11	(1, 2, 4)	(54, 9, 0, 18, 36, 0, 0, 3)	42
12	(2, 7, 9)	(54, 9, 0, 18, 36, 0, 0, 3)	42
9	(5, 6, 12)	(60, 9, 0, 18, 27, 0, 0, 6)	42
10	(1, 2, 3)	(60, 9, 0, 18, 27, 0, 0, 6)	42
11	(5, 9, 10)	(60, 9, 0, 18, 27, 0, 0, 6)	42
12	(1, 11, 12)	(60, 9, 0, 18, 27, 0, 0, 6)	42
41	(5, 8, 11)	(60, 9, 0, 18, 27, 0, 0, 6)	42
42	(5, 8, 11)	(60, 9, 0, 18, 27, 0, 0, 6)	42
43	(1, 11, 12)	(60, 9, 0, 18, 27, 0, 0, 6)	42
44	(1, 6, 7)	(60, 9, 0, 18, 27, 0, 0, 6)	42
45	(4, 9, 11)	(60, 9, 0, 18, 27, 0, 0, 6)	42
3	(5, 9, 10)	(54, 9, 24, 12, 9, 6, 0, 6)	42
4	(1, 2, 3)	(54, 9, 24, 12, 9, 6, 0, 6)	42
5	(1, 2, 3)	(54, 9, 24, 12, 9, 6, 0, 6)	42
6	(1, 2, 3)	(54, 9, 24, 12, 9, 6, 0, 6)	42
7	(5, 9, 10)	(54, 9, 24, 12, 9, 6, 0, 6)	42

Table 4.20: Top G-aberration designs for $m = 9$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
1	$(1, 2, 4, 10)$	$(24, 0, 48, 0, 12, 0, 0, 0)$	29.3333
1	$(1, 2, 3, 7)$	$(24, 0, 45, 0, 15, 0, 0, 0)$	30.0000
1	$(1, 2, 3, 11)$	$(23, 0, 46, 0, 15, 0, 0, 0)$	30.4444
1	$(1, 2, 3, 4)$	$(22, 0, 47, 0, 15, 0, 0, 0)$	30.8889
1	$(1, 2, 3, 5)$	$(23, 0, 45, 0, 16, 0, 0, 0)$	30.6667
1	$(1, 2, 5, 12)$	$(22, 0, 46, 0, 16, 0, 0, 0)$	31.1111
1	$(1, 2, 11, 13)$	$(20, 0, 48, 0, 16, 0, 0, 0)$	32.0000
53	$(1, 4, 5, 9)$	$(22, 8, 36, 0, 18, 0, 0, 0)$	30.3704
53	$(1, 2, 4, 5)$	$(25, 4, 32, 0, 23, 0, 0, 0)$	30.7407
24	$(1, 2, 5, 10)$	$(16, 40, 0, 16, 0, 12, 0, 0)$	29.0370
24	$(1, 2, 3, 13)$	$(16, 39, 0, 14, 0, 15, 0, 0)$	29.9259
24	$(1, 2, 3, 5)$	$(16, 38, 0, 15, 0, 15, 0, 0)$	30.1481
24	$(1, 2, 3, 6)$	$(15, 39, 0, 15, 0, 15, 0, 0)$	30.4444
24	$(1, 2, 4, 10)$	$(16, 37, 0, 16, 0, 15, 0, 0)$	30.3704
24	$(1, 2, 4, 9)$	$(15, 38, 0, 16, 0, 15, 0, 0)$	30.6667
24	$(1, 2, 3, 8)$	$(16, 38, 0, 14, 0, 16, 0, 0)$	30.3704
24	$(1, 2, 4, 6)$	$(16, 37, 0, 15, 0, 16, 0, 0)$	30.5926
24	$(1, 2, 3, 4)$	$(15, 38, 0, 15, 0, 16, 0, 0)$	30.8889
24	$(1, 2, 7, 9)$	$(16, 36, 0, 16, 0, 16, 0, 0)$	30.8148
24	$(1, 2, 4, 7)$	$(15, 37, 0, 16, 0, 16, 0, 0)$	31.1111
24	$(1, 2, 11, 12)$	$(12, 40, 0, 16, 0, 16, 0, 0)$	32.0000
73	$(2, 3, 5, 7)$	$(22, 10, 31, 6, 14, 0, 1, 0)$	30.2963
74	$(3, 4, 5, 10)$	$(22, 10, 31, 6, 14, 0, 1, 0)$	30.2963
73	$(2, 3, 7, 10)$	$(22, 10, 29, 6, 16, 0, 1, 0)$	30.7407
74	$(3, 4, 5, 8)$	$(22, 10, 29, 6, 16, 0, 1, 0)$	30.7407
73	$(1, 3, 5, 7)$	$(22, 14, 23, 6, 18, 0, 1, 0)$	30.5926
74	$(3, 4, 5, 13)$	$(22, 14, 23, 6, 18, 0, 1, 0)$	30.5926
73	$(2, 3, 7, 8)$	$(25, 6, 28, 4, 20, 0, 1, 0)$	30.7407
74	$(1, 3, 5, 8)$	$(25, 6, 28, 4, 20, 0, 1, 0)$	30.7407
73	$(1, 3, 7, 8)$	$(25, 8, 24, 4, 22, 0, 1, 0)$	30.8889

Table 4.21: Top G_2-aberration designs for $m = 9$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
3	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
4	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
5	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
6	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
7	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
8	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
9	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
10	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
11	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
12	(2, 7, 9, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
13	(3, 6, 10, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
14	(3, 6, 10, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
28	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
29	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
30	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
31	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
32	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
33	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
34	(5, 6, 7, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
46	(3, 6, 9, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
47	(3, 7, 8, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
48	(1, 11, 12, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
49	(3, 6, 9, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
50	(3, 6, 10, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
79	(3, 6, 9, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
80	(3, 6, 10, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
81	(3, 6, 10, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
92	(1, 2, 3, 4)	(54, 0, 0, 0, 27, 0, 0, 3)	24
93	(3, 8, 10, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24
98	(3, 6, 9, 13)	(54, 0, 0, 0, 27, 0, 0, 3)	24

Table 4.22: Top G-aberration designs for $m = 8$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
95	$(2, 3, 5, 8, 10)$	$(14, 8, 20, 8, 6, 0, 0, 0)$	19.4074
96	$(1, 3, 11, 12, 13)$	$(14, 8, 20, 8, 6, 0, 0, 0)$	19.4074
97	$(3, 4, 6, 11, 13)$	$(14, 8, 20, 8, 6, 0, 0, 0)$	19.4074
1	$(1, 2, 3, 7, 13)$	$(17, 0, 31, 0, 8, 0, 0, 0)$	19.1111
1	$(1, 2, 3, 4, 10)$	$(16, 0, 32, 0, 8, 0, 0, 0)$	19.5556
1	$(1, 2, 4, 6, 10)$	$(15, 0, 33, 0, 8, 0, 0, 0)$	20.0000
73	$(2, 3, 5, 6, 7)$	$(14, 8, 22, 4, 8, 0, 0, 0)$	19.5556
74	$(3, 4, 5, 8, 11)$	$(14, 8, 22, 4, 8, 0, 0, 0)$	19.5556
53	$(1, 4, 5, 9, 10)$	$(14, 8, 24, 0, 10, 0, 0, 0)$	19.7037
1	$(1, 2, 3, 6, 7)$	$(17, 0, 29, 0, 10, 0, 0, 0)$	19.5556
1	$(1, 2, 3, 5, 11)$	$(16, 0, 30, 0, 10, 0, 0, 0)$	20.0000
1	$(1, 2, 3, 4, 5)$	$(15, 0, 31, 0, 10, 0, 0, 0)$	20.4444
1	$(1, 2, 3, 4, 12)$	$(14, 0, 32, 0, 10, 0, 0, 0)$	20.8889
1	$(1, 2, 3, 11, 13)$	$(13, 0, 33, 0, 10, 0, 0, 0)$	21.3333
73	$(1, 3, 5, 7, 12)$	$(14, 12, 16, 4, 10, 0, 0, 0)$	19.4074
74	$(3, 4, 5, 11, 13)$	$(14, 12, 16, 4, 10, 0, 0, 0)$	19.4074
95	$(2, 3, 5, 8, 11)$	$(14, 12, 16, 4, 10, 0, 0, 0)$	19.4074
96	$(3, 4, 7, 8, 13)$	$(14, 12, 16, 4, 10, 0, 0, 0)$	19.4074
97	$(3, 4, 5, 11, 13)$	$(14, 12, 16, 4, 10, 0, 0, 0)$	19.4074
95	$(2, 3, 5, 7, 8)$	$(14, 8, 16, 8, 10, 0, 0, 0)$	20.2963
96	$(1, 3, 7, 8, 13)$	$(14, 8, 16, 8, 10, 0, 0, 0)$	20.2963
97	$(3, 4, 7, 11, 13)$	$(14, 8, 16, 8, 10, 0, 0, 0)$	20.2963
1	$(1, 2, 3, 5, 6)$	$(16, 0, 29, 0, 11, 0, 0, 0)$	20.2222
1	$(1, 2, 3, 4, 8)$	$(15, 0, 30, 0, 11, 0, 0, 0)$	20.6667
1	$(1, 2, 3, 5, 8)$	$(14, 0, 31, 0, 11, 0, 0, 0)$	21.1111
1	$(1, 2, 3, 4, 6)$	$(13, 0, 32, 0, 11, 0, 0, 0)$	21.5556
73	$(2, 3, 5, 7, 13)$	$(15, 4, 22, 4, 11, 0, 0, 0)$	20.3704
74	$(1, 3, 4, 5, 10)$	$(15, 4, 22, 4, 11, 0, 0, 0)$	20.3704
73	$(1, 3, 5, 6, 7)$	$(14, 12, 14, 4, 12, 0, 0, 0)$	19.8519
74	$(3, 4, 5, 12, 13)$	$(14, 12, 14, 4, 12, 0, 0, 0)$	19.8519

Table 4.23: Top G_2-aberration designs for $m = 8$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
3	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
4	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
5	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
6	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
7	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
8	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
9	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
10	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
11	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
12	$(1, 2, 7, 9, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
13	$(1, 3, 6, 10, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
14	$(1, 3, 6, 10, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
28	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
29	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
30	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
31	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
32	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
33	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
34	$(1, 5, 6, 7, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
46	$(1, 3, 6, 9, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
47	$(1, 3, 7, 8, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
48	$(1, 2, 11, 12, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
49	$(1, 3, 6, 9, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
50	$(1, 3, 6, 10, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
79	$(1, 3, 6, 9, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
80	$(1, 3, 6, 10, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
81	$(1, 3, 6, 10, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
92	$(1, 2, 3, 4, 5)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
93	$(1, 3, 8, 10, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16
98	$(1, 3, 6, 9, 13)$	$(36, 0, 0, 0, 18, 0, 0, 2)$	16

Table 4.24: Top G-aberration designs for $m = 7$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
95	(2, 3, 5, 6, 8, 10)	(9, 4, 15, 4, 3, 0, 0, 0)	11.9259
96	(1, 3, 5, 11, 12, 13)	(9, 4, 15, 4, 3, 0, 0, 0)	11.9259
97	(1, 3, 4, 6, 11, 13)	(9, 4, 15, 4, 3, 0, 0, 0)	11.9259
95	(2, 3, 5, 7, 9, 12)	(9, 6, 11, 6, 3, 0, 0, 0)	11.7778
96	(1, 3, 7, 11, 12, 13)	(9, 6, 11, 6, 3, 0, 0, 0)	11.7778
97	(3, 4, 6, 9, 11, 13)	(9, 6, 11, 6, 3, 0, 0, 0)	11.7778
73	(2, 3, 5, 6, 7, 10)	(9, 6, 14, 2, 4, 0, 0, 0)	11.7037
74	(3, 4, 5, 8, 9, 11)	(9, 6, 14, 2, 4, 0, 0, 0)	11.7037
95	(2, 3, 5, 8, 10, 11)	(9, 6, 14, 2, 4, 0, 0, 0)	11.7037
96	(1, 3, 4, 11, 12, 13)	(9, 6, 14, 2, 4, 0, 0, 0)	11.7037
97	(3, 4, 5, 6, 11, 13)	(9, 6, 14, 2, 4, 0, 0, 0)	11.7037
73	(2, 3, 5, 6, 7, 9)	(9, 4, 16, 2, 4, 0, 0, 0)	12.0000
74	(3, 4, 5, 6, 8, 11)	(9, 4, 16, 2, 4, 0, 0, 0)	12.0000
95	(2, 3, 5, 7, 8, 10)	(9, 2, 14, 6, 4, 0, 0, 0)	12.5926
96	(1, 2, 3, 7, 8, 13)	(9, 2, 14, 6, 4, 0, 0, 0)	12.5926
97	(1, 3, 4, 6, 9, 13)	(9, 2, 14, 6, 4, 0, 0, 0)	12.5926
73	(2, 4, 5, 7, 10, 13)	(11, 0, 12, 8, 4, 0, 0, 0)	12.1481
74	(1, 4, 5, 8, 9, 10)	(11, 0, 12, 8, 4, 0, 0, 0)	12.1481
95	(1, 2, 5, 7, 9, 13)	(10, 3, 9, 9, 4, 0, 0, 0)	12.2222
96	(1, 3, 5, 6, 8, 12)	(10, 3, 9, 9, 4, 0, 0, 0)	12.2222
97	(1, 4, 6, 8, 9, 13)	(10, 3, 9, 9, 4, 0, 0, 0)	12.2222
1	(1, 2, 3, 5, 6, 11)	(12, 0, 18, 0, 5, 0, 0, 0)	11.3333
1	(1, 2, 3, 4, 8, 10)	(11, 0, 19, 0, 5, 0, 0, 0)	11.7778
1	(1, 2, 3, 4, 5, 9)	(10, 0, 20, 0, 5, 0, 0, 0)	12.2222
1	(1, 2, 3, 5, 8, 13)	(9, 0, 21, 0, 5, 0, 0, 0)	12.6667
1	(1, 2, 3, 4, 6, 10)	(8, 0, 22, 0, 5, 0, 0, 0)	13.1111
95	(2, 3, 5, 8, 9, 11)	(9, 10, 9, 2, 5, 0, 0, 0)	11.3333
96	(3, 4, 7, 8, 11, 13)	(9, 10, 9, 2, 5, 0, 0, 0)	11.3333
97	(3, 4, 5, 9, 11, 13)	(9, 10, 9, 2, 5, 0, 0, 0)	11.3333
73	(2, 3, 5, 7, 8, 12)	(9, 4, 15, 2, 5, 0, 0, 0)	12.2222

Table 4.25: Top G_2-aberration designs for $m = 7$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
3	$(1, 2, 3, 4, 5, 6)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
4	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
5	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
6	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
7	$(1, 2, 3, 4, 5, 6)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
8	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
9	$(1, 2, 3, 4, 5, 7)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
10	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
11	$(1, 2, 3, 4, 5, 6)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
12	$(1, 2, 3, 7, 9, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
13	$(1, 3, 5, 6, 10, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
14	$(1, 3, 5, 6, 10, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
28	$(1, 2, 3, 4, 5, 6)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
29	$(1, 2, 3, 4, 5, 6)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
30	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
31	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
32	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
33	$(1, 2, 3, 4, 5, 6)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
34	$(1, 4, 5, 6, 7, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
46	$(1, 3, 5, 6, 9, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
47	$(1, 3, 5, 7, 8, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
48	$(1, 2, 3, 11, 12, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
49	$(1, 3, 5, 6, 9, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
50	$(1, 3, 5, 6, 10, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
53	$(1, 2, 3, 4, 8, 12)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
73	$(1, 2, 3, 4, 8, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
74	$(1, 2, 3, 5, 6, 7)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
79	$(1, 3, 5, 6, 9, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
80	$(1, 3, 5, 6, 10, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10
81	$(1, 3, 5, 6, 10, 13)$	$(22, 0, 0, 0, 12, 0, 0, 1)$	10

Table 4.26: Top G-aberration designs for $m = 6$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
64	$(1,3,5,7,10,11,12)$	$(6,0,14,0,0,0,0,0)$	6.2222
65	$(1,2,3,7,9,10,11)$	$(6,0,14,0,0,0,0,0)$	6.2222
95	$(2,3,5,6,7,9,12)$	$(5,4,9,2,0,0,0,0)$	6.2222
96	$(1,3,5,7,11,12,13)$	$(5,4,9,2,0,0,0,0)$	6.2222
97	$(1,3,4,6,11,12,13)$	$(5,4,9,2,0,0,0,0)$	6.2222
113	$(2,3,5,6,8,9,11)$	$(5,4,8,3,0,0,0,0)$	6.2963
114	$(1,3,5,7,9,10,12)$	$(5,4,8,3,0,0,0,0)$	6.2963
115	$(1,2,6,10,11,12,13)$	$(5,4,8,3,0,0,0,0)$	6.2963
73	$(2,3,5,6,7,9,10)$	$(5,3,10,1,1,0,0,0)$	6.5185
74	$(3,4,5,6,8,9,11)$	$(5,3,10,1,1,0,0,0)$	6.5185
15	$(1,4,5,6,7,8,13)$	$(6,2,9,2,1,0,0,0)$	6.2963
16	$(1,4,7,10,11,12,13)$	$(6,2,9,2,1,0,0,0)$	6.2963
17	$(1,2,5,6,8,10,13)$	$(6,2,9,2,1,0,0,0)$	6.2963
18	$(1,4,6,8,11,12,13)$	$(6,2,9,2,1,0,0,0)$	6.2963
19	$(2,4,5,6,8,9,12)$	$(6,2,9,2,1,0,0,0)$	6.2963
20	$(1,2,4,6,11,12,13)$	$(6,2,9,2,1,0,0,0)$	6.2963
21	$(2,4,5,6,8,9,12)$	$(6,2,9,2,1,0,0,0)$	6.2963
22	$(1,2,4,6,11,12,13)$	$(6,2,9,2,1,0,0,0)$	6.2963
23	$(1,4,6,7,10,11,12)$	$(6,2,9,2,1,0,0,0)$	6.2963
15	$(2,4,5,6,8,9,12)$	$(6,4,5,4,1,0,0,0)$	6.1481
16	$(1,5,7,8,9,11,12)$	$(6,4,5,4,1,0,0,0)$	6.1481
19	$(1,3,5,6,11,12,13)$	$(6,4,5,4,1,0,0,0)$	6.1481
20	$(1,2,5,7,8,9,11)$	$(6,4,5,4,1,0,0,0)$	6.1481
113	$(1,2,5,6,9,11,13)$	$(6,3,6,4,1,0,0,0)$	6.2963
114	$(1,3,5,6,8,9,12)$	$(6,3,6,4,1,0,0,0)$	6.2963
115	$(1,3,4,7,9,10,13)$	$(6,3,6,4,1,0,0,0)$	6.2963
113	$(1,2,5,6,9,10,13)$	$(5,4,6,4,1,0,0,0)$	6.5926
114	$(1,3,5,6,9,12,13)$	$(5,4,6,4,1,0,0,0)$	6.5926
115	$(1,3,4,6,9,10,13)$	$(5,4,6,4,1,0,0,0)$	6.5926
95	$(1,2,5,6,7,9,13)$	$(6,2,7,4,1,0,0,0)$	6.4444

Table 4.27: Top G_2-aberration designs for $m = 6$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
3	$(1, 2, 3, 4, 5, 9, 10)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
4	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
5	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
6	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
7	$(1, 2, 3, 4, 5, 9, 10)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
8	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
9	$(1, 2, 3, 4, 5, 6, 12)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
10	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
11	$(1, 2, 3, 4, 5, 9, 10)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
12	$(1, 2, 7, 9, 11, 12, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
13	$(1, 2, 3, 4, 6, 10, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
14	$(1, 2, 3, 4, 6, 10, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
15	$(1, 2, 3, 4, 5, 8, 11)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
16	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
17	$(1, 2, 3, 4, 5, 9, 10)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
18	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
19	$(1, 2, 3, 4, 5, 8, 11)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
20	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
21	$(1, 2, 3, 4, 5, 8, 11)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
22	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
23	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
25	$(1, 2, 3, 4, 5, 8, 11)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
26	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
27	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
28	$(1, 2, 3, 4, 5, 8, 11)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
29	$(1, 2, 3, 4, 5, 8, 11)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
30	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
31	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
32	$(1, 2, 3, 5, 6, 7, 13)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4
33	$(1, 2, 3, 4, 5, 8, 11)$	$(18, 0, 0, 0, 0, 0, 0, 2)$	4

Table 4.28: Top G-aberration designs for $m = 5$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
95	$(2, 3, 5, 6, 8, 9, 11, 12)$	$(3, 4, 3, 0, 0, 0, 0, 0)$	2.5185
96	$(3, 4, 5, 7, 8, 11, 12, 13)$	$(3, 4, 3, 0, 0, 0, 0, 0)$	2.5185
97	$(1, 3, 4, 5, 9, 11, 12, 13)$	$(3, 4, 3, 0, 0, 0, 0, 0)$	2.5185
113	$(2, 3, 5, 6, 7, 8, 9, 12)$	$(3, 2, 5, 0, 0, 0, 0, 0)$	2.8148
114	$(2, 3, 5, 7, 9, 10, 11, 12)$	$(3, 2, 5, 0, 0, 0, 0, 0)$	2.8148
115	$(1, 2, 5, 9, 10, 11, 12, 13)$	$(3, 2, 5, 0, 0, 0, 0, 0)$	2.8148
95	$(2, 3, 5, 6, 7, 9, 11, 12)$	$(2, 3, 5, 0, 0, 0, 0, 0)$	3.1111
96	$(1, 3, 4, 5, 7, 11, 12, 13)$	$(2, 3, 5, 0, 0, 0, 0, 0)$	3.1111
97	$(1, 3, 4, 5, 6, 11, 12, 13)$	$(2, 3, 5, 0, 0, 0, 0, 0)$	3.1111
13	$(1, 2, 3, 5, 7, 8, 12, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
14	$(1, 2, 3, 4, 5, 8, 10, 11)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
25	$(1, 4, 5, 6, 7, 8, 10, 12)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
26	$(1, 2, 5, 7, 8, 9, 10, 12)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
27	$(1, 2, 3, 4, 5, 7, 8, 9)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
28	$(3, 4, 5, 6, 7, 10, 11, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
29	$(1, 4, 5, 6, 7, 8, 10, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
30	$(1, 2, 3, 4, 6, 7, 8, 10)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
31	$(1, 2, 5, 7, 8, 9, 10, 12)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
32	$(1, 2, 6, 8, 9, 10, 12, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
35	$(2, 3, 5, 7, 8, 9, 10, 12)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
36	$(1, 2, 3, 5, 6, 7, 10, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
37	$(3, 4, 5, 6, 7, 10, 11, 12)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
38	$(1, 2, 3, 5, 6, 10, 11, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
39	$(2, 4, 5, 6, 7, 10, 11, 12)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
40	$(1, 2, 3, 5, 6, 9, 10, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
46	$(1, 2, 3, 5, 7, 9, 10, 11)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
47	$(1, 2, 3, 4, 5, 9, 11, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
48	$(1, 2, 3, 4, 5, 7, 9, 13)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
64	$(1, 2, 3, 5, 7, 9, 10, 11)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667
65	$(1, 2, 3, 4, 5, 8, 10, 11)$	$(4, 0, 6, 0, 0, 0, 0, 0)$	2.6667

Table 4.29: Top G_2-aberration designs for $m = 5$.

design	**removed**	$(0, \frac{8}{27}, \frac{4}{9}, \frac{14}{27}, \frac{2}{3}, \frac{20}{27}, \frac{10}{9}, 2)$	A_3
80	(1, 2, 6, 8, 9, 11, 12, 13)	(6, 0, 3, 0, 1, 0, 0, 0)	2
116	(1, 3, 5, 6, 7, 8, 12, 13)	(6, 0, 3, 0, 1, 0, 0, 0)	2
117	(1, 3, 5, 6, 7, 8, 10, 12)	(6, 0, 3, 0, 1, 0, 0, 0)	2
73	(1, 2, 7, 8, 9, 10, 11, 12)	(7, 0, 0, 0, 3, 0, 0, 0)	2
74	(1, 4, 5, 6, 7, 8, 12, 13)	(7, 0, 0, 0, 3, 0, 0, 0)	2
92	(1, 2, 5, 6, 7, 9, 10, 11)	(7, 0, 0, 0, 3, 0, 0, 0)	2
93	(1, 4, 5, 6, 7, 8, 9, 10)	(7, 0, 0, 0, 3, 0, 0, 0)	2
94	(1, 3, 5, 6, 7, 8, 10, 13)	(7, 0, 0, 0, 3, 0, 0, 0)	2
95	(1, 3, 5, 6, 7, 8, 11, 13)	(7, 0, 0, 0, 3, 0, 0, 0)	2
96	(1, 4, 5, 6, 7, 8, 9, 13)	(7, 0, 0, 0, 3, 0, 0, 0)	2
97	(1, 2, 3, 4, 5, 7, 8, 11)	(7, 0, 0, 0, 3, 0, 0, 0)	2
98	(1, 2, 6, 8, 9, 10, 11, 12)	(7, 0, 0, 0, 3, 0, 0, 0)	2
99	(1, 2, 6, 7, 9, 11, 12, 13)	(7, 0, 0, 0, 3, 0, 0, 0)	2
113	(1, 3, 5, 6, 7, 8, 10, 13)	(7, 0, 0, 0, 3, 0, 0, 0)	2
114	(1, 2, 5, 6, 7, 8, 9, 11)	(7, 0, 0, 0, 3, 0, 0, 0)	2
115	(1, 2, 3, 4, 5, 6, 9, 11)	(7, 0, 0, 0, 3, 0, 0, 0)	2
120	(1, 3, 5, 6, 7, 8, 9, 10)	(7, 0, 0, 0, 3, 0, 0, 0)	2
95	(1, 2, 6, 7, 8, 9, 10, 11)	(7, 0, 2, 0, 0, 0, 1, 0)	2
96	(1, 2, 3, 4, 5, 6, 7, 12)	(7, 0, 2, 0, 0, 0, 1, 0)	2
97	(1, 4, 5, 6, 7, 8, 12, 13)	(7, 0, 2, 0, 0, 0, 1, 0)	2
98	(1, 2, 3, 5, 6, 7, 10, 11)	(7, 0, 2, 0, 0, 0, 1, 0)	2
99	(1, 2, 3, 4, 5, 6, 7, 11)	(7, 0, 2, 0, 0, 0, 1, 0)	2
100	(1, 2, 3, 4, 5, 6, 7, 9)	(7, 0, 2, 0, 0, 0, 1, 0)	2
101	(1, 2, 3, 5, 6, 7, 8, 12)	(7, 0, 2, 0, 0, 0, 1, 0)	2
102	(2, 3, 5, 6, 8, 9, 11, 13)	(7, 0, 2, 0, 0, 0, 1, 0)	2
121	(1, 2, 3, 4, 5, 6, 7, 8)	(7, 0, 2, 0, 0, 0, 1, 0)	2
2	(1, 2, 3, 4, 5, 6, 7, 13)	(9, 0, 0, 0, 0, 0, 0, 1)	2
3	(1, 2, 3, 4, 5, 6, 7, 8)	(9, 0, 0, 0, 0, 0, 0, 1)	2
4	(1, 2, 3, 4, 5, 6, 7, 13)	(9, 0, 0, 0, 0, 0, 0, 1)	2
5	(1, 2, 3, 4, 5, 6, 7, 13)	(9, 0, 0, 0, 0, 0, 0, 1)	2

Chapter 5

Summary and Future Work

We conclude this thesis with a brief summary of the work in each chapter, and some general discussion about potential directions for future work.

5.1 Partially Clear Two-factor Interactions

In Chapter 2 we introduced the concept of robust designs through partially clear two-factor interactions. We allowed effects of interest to be aliased with two-factor interactions that prior knowledge supported as being negligible, while still remaining robust to nonnegligible two-factor interactions. By allowing this aliasing, the goal was to entertain more factors in comparison to designs with clear two-factor interactions. The set of two-factor interactions were separated into three distinct categories: the requirement set, the set of nonnegligible interactions and the set of negligible interactions. By breaking the set of factors into two groups, we looked at different configurations of the sets of two-factor interactions within and between groups. In particular, one of these configurations, which we denoted type 1, showed substantial gain in terms of the number of factors versus designs with clear two-factor interactions. We gave constructions for robust designs of type 1, and showed that these constructions match with the optimal results for designs of 32 and 64 runs. The constructions are even more powerful, as they also apply if we allow nonorthogonality within the requirement set. The Chapter concluded with a bound on the number of clear two-factor interactions in an orthogonal array.

From a practical standpoint, the concept of partially clear two-factor interactions is appealing as it allows prior knowledge to be used to gain more from a design, while still keeping the robustness properties of clear designs. In general, if we have knowledge about some interactions being negligible, we can separate the interactions into the three categories and do a computer search for robust designs. One future direction this research can take is to look at more than two groups of factors which allows more flexibility for the two-factor interactions between groups. With these additional groups, it would be useful to establish bounds on the number of factors that can be entertained depending on the configuration of factors, similar to Lemma 2.2. Another consideration is the D-efficiency for these robust designs when non-orthogonal requirement sets are used. For the robust designs in Chapter 2, we saw good performance in terms of D-efficiency, but it would be worthwhile to see what sacrifice towards the efficiency is made in general by ensuring robustness to the nonnegligible two-factor interactions. In the case of a non-orthogonal requirement set, the approaches in Chapters 2 and 3 take different directions. Here, our primary concern is the robustness to the nonnegligible two-factor interactions with D-efficiency as a secondary criterion, whereas in Chapter 3, D-efficiency was the first priority and then the robustness to nonnegligible two-factor interactions. This consideration depends on what the experimenter deems as more serious. Lemma 2.2 showed that requiring $A_{22} = 0$ places quite a restriction on the factors in a design, and we focused our considerations to robust designs of type 1. For robust designs of type 3, that $A_{22} = 0$ is required for orthogonal estimation of the requirement set and $A_{31} = 0$ gave the robustness to nonnegligible 2fi's. It is possible that introducing nonorthogonality into the requirement set could lead to robust designs of type 3.

We conclude our discussion on partially clear designs by discussing robust parameter design. In robust parameter design, factors are separated into control and noise factors, where noise factors are those that are hard to control. The goal is to find optimal settings for the control factors that minimize the variability of a system by making it less sensitive to noise variation. Wu and Hamada (2000) give the following ordering for the importance of effects:

1. control-by-noise interactions, control main effects and noise main effects;

2. control-by-control interactions and control-by-control-by-noise interactions;

3. noise-by-noise interactions.

The setup is similar by having the two groups of factors and separating the interactions into these subsets. The idea of partially clear designs should be useful for robust parameter design problems but this warrants further investigation.

5.2 Multi-Level Orthogonal Arrays for Estimating Main Effects and Specified Interactions

Chapter 3 investigated the use of multi-level orthogonal arrays for estimating a requirement set. Designs were ranked firstly based on D-efficiency followed by contamination from the two-factor interactions not estimated. Focus was on designs with three-level factors to allow curvature of factors to be studied. These designs are an attractive alternative to other methods that require follow-up experimentation. Factors were grouped into two sets: the core set, which were those factors involved in an interaction in the requirement set, and the set of remaining factors in which only the main effects are of interest. We showed that level permutations of factors outside of the core set maintain many of the optimality measures, including D-efficiency and contamination from the interactions outside of the requirement set. A result giving a lower bound on the number of factors outside of the core set that can be entertained was provided, and a method was proposed to aid in searching for designs. The chapter was concluded with a search for three-level designs with 27 runs.

That the results from Chapter 3 hold for mixed level designs is particularly appealing, as it provides flexibility in how to treat some of the factors, such as only concerning ourselves with two levels for the factors outside of the core set (ie. if there is no interest in the curvature). It would be interesting to see how designs that are not a subset of columns from a saturated orthogonal array perform for the objectives of Chapter 3. The (M.S)-approach shows promise as an efficient means of improving the search for good designs when the complete search is infeasible. This idea of studying the space occupied by the two-factor interactions rather than the model matrix for multi-level designs requires further study to see if anything else can be gleaned from it. Returning to the theme of Chapter 2, we can approach the problem by trying to make the design robust to certain interaction components, such as the linear-by-linear. Dealing with certain interaction components is likely to need consideration of level permutations - but it is worth noting that depending on the level assignments, the linear-by-linear components should contain many zero entries.

5.3 J-Characteristics for Multi-Level Factorial Designs

Whereas Chapters 2 and 3 focused on a specific set of effects, Chapter 4 made no such separation of effects. Instead, the focus was on the distribution of design points, and an ANOVA decomposition to study the uniformity in lower dimensions. This allowed a means of measuring the projection properties of a design through G_2-aberration that is equivalent to generalized minimum aberration as defined by Xu and Wu (2002). We also proposed more conservative approaches using G-aberration and $G_{2(i)}$-aberration. The J-characteristics were used to examine three-level designs of 18 and 27 runs. The ranking between G and G_2-aberration was consistent for 18 runs, but not so for 27 runs.

In terms of measuring near-orthogonality, algorithmic approaches such as Xu (2002), Yamada and Matsui (2002), and Lekivetz, Sitter, Bingham, Hamada, Moore and Wendelberger (2011) are typically driven by G_2-aberration through some equivalent measure to B_2. The approach here has a natural extension to higher dimensions by using $B_3, B_4, \ldots$ if the experimenter wants to consider higher dimensional properties. The designs given in Xu, Cheng and Wu (2004) were found by constructing many orthogonal arrays and choosing the best among them. Using these algorithms to find designs for G-aberration rather than G_2 is currently being investigated. From Propositions 4.1 and 4.2, $G_{2(i)}$ and $G_{(i)}$-aberration may warrant some further research as a tool for searching for designs through their connection to G and G_2-aberration.

One of the most exciting potential uses for the J-characteristics is in measuring the projection properties of designs for computer experiments such as those described in Bingham, Sitter and Tang (2009) and Lin, Bingham, Sitter and Tang (2010). In terms of using designs as building blocks, particularly for nearly-orthogonal arrays, the J-characteristics can help select designs that have good lower-dimensional projection properties in terms of the spread of design points. As another use, we can create a grid over the lower dimensional projections and determine how many points fall within a partition. Ideally, we want to see the points spread out evenly within each of the partitions. We can use the J-characteristics to measure this spread, and still maintain properties such as generalized minimum aberration.

Bibliography

[1] Addelman, S. Symmetrical and asymmetrical fractional factorial plans. *Technometrics 4* (1962), 47–58.

[2] Ai, M.-Y., and Zhang, R.-C. Projection justification of generalized minimum aberration for asymmetrical fractional factorial designs. *Metrika 60* (2004), 279–285.

[3] Bingham, D., Sitter, R. R., and Tang, B. Orthogonal and nearly orthogonal designs for computer experiments. *Biometrika 96* (2009), 51–65.

[4] Box, G. E. P., and Behnken, D. W. Some new three level designs for the study of quantitative variables. *Technometrics 2* (1960), 455–475.

[5] Box, G. E. P., and Draper, N. R. A basis for the selection of a response surface design. *Journal of the American Statistical Association 54* (1959), 622–654.

[6] Box, G. E. P., and Hunter, J. S. The 2^{k-p} fractional factorial designs. *Technometrics 3* (1961), 311–351 and 449–458.

[7] Box, G. E. P., and Meyer, R. D. An analysis for unreplicated fractional factorials. *Technometrics 28* (1986), 11–18.

[8] Box, G. E. P., and Wilson, K. B. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society, Series B 13* (1951), 1–45.

[9] Chen, B.-J., Li, P.-F., Liu, M.-Q., and Zhang, R.-C. Some results on blocked regular 2-level fractional factorial designs with clear effects. *Journal of Statistical Planning and Inference 136* (2006), 4436 – 4449.

[10] Chen, H., and Hedayat, A. S. 2^{n-m} designs with resolution III or IV containing clear two-factor interactions. *Journal of Statistical Planning and Inference 75* (1998), 147 – 158.

[11] CHEN, H. H., AND CHENG, C.-S. Doubling and projection: A method of constructing two-level designs of resolution IV. *The Annals of Statistics 34* (2006), 546–558.

[12] CHEN, J., SUN, D. X., AND WU, C. F. J. A catalogue of two-level and three-level fractional factorial designs with small runs. *International Statistical Review 61* (1993), 131 – 145.

[13] CHENG, C.-S. Some projection properties of orthogonal arrays. *The Annals of Statistics 23* (1995), 1223–1233.

[14] CHENG, C.-S., DENG, L.-Y., AND TANG, B. Generalized minimum aberration and design efficiency for nonregular fractional factorial designs. *Statistica Sinica 12* (2002), 991–1000.

[15] CHENG, C.-S., MEE, R. W., AND YEE, O. Second order saturated orthogonal arrays of strength three. *Statistica Sinica 18* (2008), 105–119.

[16] CHENG, C.-S., STEINBERG, D. M., AND SUN, D. X. Minimum aberration and model robustness for two-level fractional factorial designs. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 61* (1999), 85–93.

[17] CHENG, S.-W., AND WU, C. F. J. Factor screening and response surface exploration. *Statistica Sinica 11* (2001), 553–604. With comments and a rejoinder by the authors.

[18] CHIPMAN, H., HAMADA, M., AND WU, C. F. J. A bayesian variable-selection approach for analyzing designed experiments with complex aliasing. *Technometrics 39* (1997), 372–381.

[19] DENG, L.-Y., AND TANG, B. Generalized resolution and minimum aberration criteria for Plackett-Burman and other nonregular factorial designs. *Statistica Sinica 9* (1999), 1071–1082.

[20] DENG, L.-Y., AND TANG, B. Design selection and classification for Hadamard matrices using generalized minimum aberration criteria. *Technometrics 44* (2002), 173–184.

[21] DEY, A., AND MUKERJEE, R. *Fractional factorial plans*. Wiley series in probability and statistics: Probability and statistics. Wiley, 1999.

[22] EVANGELARAS, H., KOUKOUVINOS, C., AND LAPPAS, E. 18-run nonisomorphic three level orthogonal arrays. *Metrika 66* (2007), 31–37.

[23] Evangelaras, H., Koukouvinos, C., and Lappas, E. 27-run nonisomorphic three level orthogonal arrays: Identification, evaluation and projection properties. *Utilitas Mathematica 84* (2011), 75–87.

[24] Franklin, M. F. Selecting defining contrasts and confounded effects in pn-m factorial experiments. *Technometrics 27* (1985), 165–172.

[25] Franklin, M. F., and Bailey, R. A. Selection of defining contrasts and confounded effects in two-level experiments. *Journal of the Royal Statistical Society. Series C 26* (1977), 321–326.

[26] Fries, A., and Hunter, W. G. Minimum aberration 2^{k-p} designs. *Technometrics 22* (1980), pp. 601–608.

[27] Greenfield, A. A. Selection of defining contrasts in two-level experiments. *Applied Statistics 25* (1976), 64–67.

[28] Hamada, M., and Wu, C. F. J. Analysis of designed experiments with complex aliasing. *Journal of Quality Technology 24* (1992), 130–137.

[29] Hedayat, A., Sloane, N., and Stufken, J. *Orthogonal arrays: theory and applications*. Springer series in statistics. Springer, 1999.

[30] Hickernell, F. J., and Liu, M.-Q. Uniform designs limit aliasing. *Biometrika 89* (2002), 893–904.

[31] Jones, B., and Nachtsheim, C. J. Efficient designs with minimal aliasing. *Technometrics 53* (2011), 62–71.

[32] Ke, W., and Tang, B. Selecting 2^{m-p} designs using a minimum aberration criterion when some two-factor interactions are important. *Technometrics 45* (2003), 352–360.

[33] Ke, W., Tang, B., and Wu, H. Compromise plans with clear two-factor interactions. *Statistica Sinica 15* (2005), 709–715.

[34] Lekivetz, R., Sitter, R. R., Bingham, D., Hamada, M., Moore, L., and Wendelberger, J. A new algorithm for constructing mixed-level orthogonal and nearly-orthogonal arrays. *in preparation*.

[35] Lin, C. D., Bingham, D., Sitter, R. R., and Tang, B. A new and flexible method for constructing designs for computer experiments. *The Annals of Statistics 38* (2010), 1460–1477.

[36] Lisonek, P. Binary caps with many free pairs of points. *Journal of Combinatorial Designs 14* (2006), 490–499.

[37] Liu, M.-Q., Fang, K.-T., and Hickernell, F. J. Connections among different criteria for asymmetrical fractional factorial designs. *Statistica Sinica 16* (2006), 1285–1297.

[38] Loeppky, J. L., Sitter, R. R., and Tang, B. Nonregular designs with desirable projection properties. *Technometrics 49* (2007), 454–467.

[39] Nguyen, N.-K., and Liu, M.-Q. An algorithmic approach to constructing mixed-level orthogonal and near-orthogonal arrays. *Computational Statistics and Data Analysis 52* (2008), 5269 – 5276.

[40] Plackett, R. L., and Burman, J. P. The design of optimum multifactorial experiments. *Biometrika 33* (1946), 305–325.

[41] Rao, C. R. Factorial experiments derivable from combinatorial arrangements of arrays. *Supplement to the Journal of the Royal Statistical Society 9* (1947), 128–139.

[42] Schoen, E. D. All orthogonal arrays with 18 runs. *Quality and Reliability Engineering International 25* (2009), 467–480.

[43] Sloane, N. J. A. A library of orthogonal arrays, accessed online http://www2.research.att.com/ njas/oadir/, July 2011.

[44] Steinberg, D. M., and Bursztyn, D. Discussion of "factor screening and response surface exploration". *Statistica Sinica 11* (2001), 596–599.

[45] Sun, D. X. *Estimation capacity and related topics in experimental design.* PhD thesis, University of Waterloo, 1993.

[46] Tang, B. Theory of J-characteristics for fractional factorial designs and projection justification of minimum G_2-aberration. *Biometrika 88* (2001), 401–407.

[47] Tang, B. Orthogonal arrays robust to nonnegligible two-factor interactions. *Biometrika 93* (2006), 137–146.

[48] Tang, B., and Deng, L.-Y. Minimum G_2-aberration for nonregular fractional factorial designs. *The Annals of Statistics 27* (1999), 1914–1926.

[49] Tang, B., Ma, F., Ingram, D., and Wang, H. Bounds on the maximum numbercomment of clear two-factor interactions for 2^{m-p} designs of resolution III and IV. *Canadian Journal of Statistics 30* (2002), 127–136.

[50] Tang, B., and Zhou, J. Existence and construction of two-level orthogonal arrays for estimating main effects and some specified two-factor interactions. *Statistica Sinica 19* (2009), 1193 – 1201.

[51] Wolters, M., and Bingham, D. Simulated annealing model search for subset selection in screening experiments. *Technometrics* (to appear).

[52] Wu, C., and Hamada, M. *Experiments: planning, analysis, and parameter design optimization.* Wiley series in probability and statistics. J. Wiley, 2000.

[53] Wu, C. F. J., and Chen, Y. A graph-aided method for planning two-level experiments when certain interactions are important. *Technometrics 34* (May 1992), 162–175.

[54] Wu, D. Searching for optimal orthogonal arrays for estimating some main effects and specified two-factor interactions, 2009. M.Sc. thesis, Simon Fraser University.

[55] Wu, H., and Wu, C. F. J. Clear two-factor interactions and minimum aberration. *The Annals of Statistics 30* (2002), 1496–1511.

[56] Xu, H. An algorithm for constructing orthogonal and nearly-orthogonal arrays with mixed levels and small runs. *Technometrics 44* (2002), 356–368.

[57] Xu, H. Minimum moment aberration for nonregular designs and supersaturated designs. *Statistica Sinica 13* (2003), 691–708.

[58] Xu, H., Cheng, S.-W., and Wu, C. Optimal projective three-level designs for factor screening and interaction detection. *Technometrics 46* (2004), 280–292.

[59] Xu, H., and Wu, C. F. J. Generalized minimum aberration for asymmetrical fractional factorial designs. *The Annals of Statistics 29* (2001), 549–560.

[60] Yamada, S., and Matsui, T. Optimality of mixed-level supersaturated designs. *Journal of Statistical Planning and Inference 104* (2002), 459–468.

[61] Yang, G., and Butler, N. A. Grid representations for join designs containing maximum numbercomment of clear two-factor interactions. *Journal of Statistical Planning and Inference 138* (2008), 1472 – 1478.