NEW DEVELOPMENTS IN DESIGNS FOR COMPUTER EXPERIMENTS AND PHYSICAL EXPERIMENTS

by

Chunfang Devon Lin

B.Sc., University of Science and Technology of China, 2002

M.Sc., Simon Fraser University, 2004

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 2008

APPROVAL

Name:	Chunfang Devon Lin
Degree:	Doctor of Philosophy
Title of thesis:	New Developments in Designs for Computer Experiments and Physical Experiments

Examining Committee: Dr. Brad McNeney
Chair

____

Dr. Boxin Tang, Senior Supervisor

____

Dr. Derek Bingham, Supervisor

____

Dr. Carl Schwarz, Supervisor

____

Dr. Tom Loughin, SFU Examiner

____

Dr. Ching-Shui Cheng, External Examiner,
University of California, Berkeley

Date Approved: June 20, 2008

presented. The second problem is to provide a catalogue of "good" two-level folded-over non-orthogonal designs. Such designs are useful in screening experiments. To assess the goodness of designs, we introduce the MDS-resolution and MDS-aberration, based on the notion of minimal dependent sets (MDS). With both criteria, it is possible to systematically compare the statistical properties of designs. Obtaining a catalogue, however, remains challenging because it involves determining whether or not two designs are isomorphic. A fast isomorphism check is developed for this purpose. A catalogue of minimum MDS-aberration designs is obtained for many useful run sizes. An algorithm for obtaining "good" larger designs is discussed.

Dedication

To Randy R. Sitter

Acknowledgments

I owe an enormous debt of gratitude to my advisors, Randy Sitter, Boxin Tang and Derek Bingham. I appreciate their instruction, steadfast encouragement, and guidance throughout. I was exceptionally fortunate to have Randy's faith in me. His contagious enthusiasm has pulled me up and helped me believe in what I am doing. I am especially grateful to Boxin for his generous guidance through the most difficult time in my life. I have also been lucky enough to benefit from his comprehensive knowledge. I thank Derek for providing an efficient and fun working environment, many interesting and stimulating conversations, and new research problems.

I also need to express my gratitude to my supervisory committee. I thank Professors Ching-Shui Cheng, Tom Loughin and Carl Schwarz for their time and suggestions to make my thesis better. I am thankful to enjoy the friendship of my officemates, Wilson, Jason, Crystal, Pritam, Matt and Ryan.

At SFU, I would like to thank the faculty and staff in the department of statistics and actuarial science for making this the great place to learn that it is. I thank my friends: Jason Nielson, Farouk Nathoo, Simon Bonner, Celes Ying, Xin Feng, Na Lei, Lihui Zhao, Saman Muthukumarana, Elizabeth Juarez-Colunga, Carolyn Huston, Dongming Sun and Ada Zhang. I must separately thank Jean's company for the past six years no matter if it was a good time or a bad time.

Finally, my heart-felt thanks go to my family - my parents, brother, and grandmother - for their patience, understanding and love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Computer experiments

Deterministic computer experiments are becoming more commonly used in science and engineering. This is primarily because the underlying physical processes are too time-consuming, expensive, or even impossible to observe. Rapid growth in computer power has made it possible to perform deterministic experiments on simulators. The first computer experiment appeared to be conducted by Enrico Fermi and colleagues in Los Alamos in 1953. Since then, scientists in diverse areas such as engineering, cosmology, particle physics and aircraft design have turned to computer experiments as a powerful tool to understand their respective processes. For instance, in the design of a vehicle, computer experiments are used to study the effect of a collision of the vehicle with a barrier before manufacturing the prototype of the vehicle. See Bayarri et al. (2002) for details.

Similar to physical experiments, computer experiments can be planned and implemented in the following steps:

1. *State the objectives.* Computer experiments are performed with a variety of goals in mind. For example, objectives include factor screening, building an

emulator of the simulator, optimization, and model calibration.

2. *Choose a response.* It can be univariate, multivariate, temporal or functional.
3. *Choose input variables.* They can be qualitative or quantitative or both. They can also be categorized into control variables, environmental variables, and model variables.
4. *Represent and implement the underlying physical process using a computer code.*
5. *Choose an experimental plan.*
6. *Perform the experiments on the simulators.*
7. *Analyze the data.* This includes identifying the active factors and model fitting.
8. *Interpret the model and draw conclusions.*

A detailed discussion of each step can be found in Santner, Williams and Notz (2003) and Fang, Li and Sudjianto (2006). This thesis will investigate the indispensable step 5 - choosing an experimental plan. It is a crucial step because often the computer code is expensive in that it may take hours or days to produce one single output. We will briefly review several types of designs for computer experiments in Section 1.1.2. To help understand the selection of designs, we give an overview of modeling techniques in Section 1.1.1. Before doing so, we introduce some necessary notation and outline the framework.

In many scientific investigations, complicated physical phenomena are represented by a mathematical model

$$\mathbf{Y} = f(\mathbf{X}), \quad \mathbf{X} \in [0,1]^m, \tag{1.1}$$

where $\mathbf{X}$ consists of m input variables, f is the computer code, and $\mathbf{Y}$ represents the response. Model (1.1) is usually a solution to a set of equations, which can be linear, nonlinear, ordinary or partial differential. Because the solution to the equations is often impossible to obtain analytically, scientists study the complex relationship between the inputs and outputs by varying the inputs to the computer code and observing how their process outputs are affected. Such studies are called computer experiments. A key feature of computer experiments is that the computer code is deterministic. That is, the response is unchanged if an input setting is replicated. The lack of random errors presents challenges, which necessitate new approaches to the design and analysis of experiments (see, e.g., Sacks, Welch, Mitchell and Wynn, 1989).

1.1.1 Model

One important objective of computer experiments is to find a model that describes the empirical relationship between the inputs and outputs. That is, we wish to build a statistical model to approximate the true model (1.1). We refer to the approximate model as an emulator. Obtaining an accurate, informative yet simple emulator plays a crucial role in the analysis of computer experiments in that the emulator will replace the true model to make predictions at unsampled points and perform other analyses such as uncertainty analysis and sensitivity analysis.

The true model (1.1) can be viewed as a nonparametric model without a random error component. Therefore, building an emulator can be treated as a nonparametric regression problem with no random error. To deal with the absence of random errors, researchers have developed diverse models for users. Fang, Li and Sudjianto (2006) provided a comprehensive review on modeling techniques to build an emulator. The commonly used modeling techniques include polynomial regression, spline regression, Gaussian process stochastic model, and local polynomial regression. Comparisons of different modeling techniques have been made in the literature (see, e.g., Simpson , Peplinski, Koch and Allen, 2001; Ben-Ari and Steinberg, 2007).

1.1.2 Design

In the previous subsection, we have mentioned that there are a variety of modeling techniques for building an emulator. There is no correct statistical model for computer experiments. In addition, little knowledge is available about which model would fit the data well before they are collected. Thus, designs for computer experiments should facilitate diverse modeling methods. Space-filling designs are a class of designs that serve this purpose. They meet the basic requirement of designs for computer experiments - designs should not have repeated runs due to the deterministic nature of computer models. Furthermore, when making prediction at unsampled points is the primary goal, space-filling designs are more likely to provide better prediction accuracy. A design that is not space-filling leaves most of the design space unexplored and clearly yields a poor predictor.

Most commonly used space-filling designs in computer experiments are Latin hypercube designs, maximin distance designs and uniform designs. Latin hypercube designs have the one-dimensional space-filling property in that when projected onto each dimension, each portion of the design range has a design point. They were proposed by McKay, Beckman and Conover (1979), which is commonly recognized as the first paper on the designs for deterministic computer experiments. This class of designs is easy to generate. Maximin distance designs were first introduced by Johnson, Moore and Ylvisaker (1990) in the context of computer experiments. The basic idea behind this class of designs is quantifying how spread out the design points using distance criteria. A maximin distance design maximizes the smallest distance between any two design points so that no two design points are too close. Johnson, Moore and Ylvisaker (1990) showed that maximin distance designs are asymptotically D-optimal under some regularity conditions. Uniform designs were proposed by Fang (1980) and Wang and Fang (1981). They were chosen based on the discrepancy between the empirical cumulative distribution function of a design and that of the uniform distribution in the design region. The discrepancy is a measure of uniformity; lower discrepancy implies better uniformity. A more detailed account of the above three types of designs can be found in Santner, Williams and Notz (2003), Fang, Li and Sudjianto (2006) and the references therein. An alternative approach to space-filling designs is to use some model-dependent criteria such as the integrated mean square error and maximum mean square error to select designs for computer experiments (Santner, Williams and Notz, 2003).

The curse of dimensionality comes into serious play in the construction of space-filling designs for computer experiments. When the dimensionality of the input space is high, providing a good coverage of the entire input space as suggested by the original idea of space-filling designs with limited design points is a hopeless undertaking. A realistic and fruitful approach is to construct designs that are space-filling in the low dimensional projections. Randomized orthogonal arrays (Owen, 1992) and orthogonal array-based Latin hypercubes (Tang, 1993) enjoy this property of low dimensional space-filling. Research on the use of orthogonal designs for computer experiments has been gaining momentum recently. As argued in Bingham, Sitter and Tang (2008), orthogonality is directly useful when polynomial models are considered, and it can also be viewed as stepping stones to designs that are space-filling in low dimensional projections. Chapters 2 and 3 of the thesis are devoted to the construction of orthogonal and nearly orthogonal Latin hypercubes.

Another consequence of the high dimensionality is that design points are very far apart in a space-filling design. As the spatial correlation (see, e.g., Santner, Williams and Notz, 2003) decreases with the distance dramatically relative to the spacing, there are no points close enough to give reliable estimates of the correlation parameters. To enhance the estimation of the correlation parameters, Handcock (1991) recommended what he terms a cascading Latin hypercube in which small Latin hypercube designs with closely clustered points are dispersed through the space as clusters while the cluster centers also form a Latin hypercube. By doing so, the space-filling property is maintained, and some points that are close together are also ensured. Chapters 2 and 3 of this thesis will provide a method for constructing a rich class of designs with a cascading structure.

1.2 Factorial designs

Factorial designs play a fundamental role in the theory and practice of physical experiments. They have been used in a wide range of fields including engineering, social science, agriculture and biology. They allow experimenters to study simultaneously the effects of multiple input variables on the response. In physical experiments, the input variables are called factors. Each factor must have at least two settings so that the effect of change in factor settings on the response can be studied. These settings are called levels of the factor. A combination of the level settings of factors is referred to as a treatment or a run. Physical experiments differ from computer experiments introduced in the previous section in that the former has random errors in the response. The experimental designs that deal with the arrangement of treatments are called factorial designs. In this thesis, we consider factorial designs with factors at two levels represented by ± 1. Specifically, two-level fractional factorial (FF) designs and two-level folded over non-orthogonal designs are the subjects of Chapters 4 and 5, respectively.

1.2.1 Fractional factorial designs

A full factorial design consists of all possible treatments. That is, if a factorial experiment involves m factors at two levels, a full factorial design requires 2^m runs. This run size grows rapidly as the number m of factors increases. For example, this run size grows from 32 to 512 as the number of factors increases from 5 to 9. Therefore, running a full factorial design becomes impractical even for an moderately large value of m. Instead, fractional factorial (FF) designs are commonly used in practice as they only use a fraction or a subset of the full factorial design. FF designs can be classified into regular designs and nonregular designs. Regular designs are specified through defining relations. In a regular design, any two factorial effects are either orthogonal or fully aliased. Designs that do not have this property are called nonregular designs.

Many important problems regarding FF designs have been studied by researchers and practitioners. We here discuss three major ones. The first and probably most important problem is the choice of FF designs. The first criterion for selecting optimal regular fractions is the maximum resolution proposed by Box and Hunter (1961a, b). Because many designs with the same resolution exist, Fries and Hunter (1980) proposed a more discriminating criterion, known as the minimum aberration. However, these criteria are only applicable to regular designs. Deng and Tang (1999) extended the notions of resolution and minimum aberration to nonregular designs and proposed generalized resolution and minimum aberration. Subsequently, criteria such as minimum G_2-aberration (Tang and Deng, 1999) and minimum moment aberration (Xu, 2003) were introduced. These criteria reduce to their counterparts for regular designs. Meanwhile, other criteria for selecting FF designs have arisen from different statistical points of view. These include the criteria of maximum number of clear two-factor interactions (Wu and Chen, 1992), estimation capacity (Sun, 1993; Cheng and Mukerjee, 1998; Cheng, Steinberg and Sun, 1999), projection estimation capacity (Cheng, 1995; Loeppky, Sitter and Tang, 2007), and average D efficiency (Cheng, Deng and Tang, 2002). The second problem is that, for a given number of factors, we want to find the minimum run size for a design with certain desirable properties to exist. It is equivalent to seeking the maximum number of factors for a given run size and optimality criteria. The problem is practically important for obvious economic reasons. Third, a catalogue of non-isomorphic designs can be very helpful for identifying the design patterns or searching for optimal designs. Two factorial designs are said to be isomorphic if one can be obtained from the other by relabeling the factors having the same number of levels, reordering the treatment combinations and/or relabeling the levels of one or more factors. Otherwise, the two designs are non-isomorphic. When such a catalogue is computationally infeasible to obtain, a catalogue of good designs based on major criteria would be still beneficial for searching designs based on other criteria. In Chapter 4, we aim to provide a collection of good two-level FF designs based on the criteria of minimum G and G_2-aberration.

1.2.2 Folded over non-orthogonal designs

Fold-over (Box and Wilson, 1951) is a clever technique in factorial experiments because it is able to de-alias main effects and two-factor interactions. In other words, in the folded over design, main effects and all two-factor interactions can be estimated independently. For any run in an initial factorial design, its fold-over is the run with the levels of all the factors sign-switched. Thus the fold-over of (1, -1, -1) is (-1, 1, 1), where 1 and -1 designate the high and low levels of a factor. The fold-over of a design is simply the union of the initial design and the fold-overs of the runs in the design.

The fold-over technique has been used primarily to create orthogonal resolution IV designs from orthogonal resolution III designs with notable exceptions of the early literature by John (1962, 1964), Banerjee and Federer (1967), Webb (1968) and Margolin (1969). Folding over a non-orthogonal resolution III design produces a resolution IV design, in which the main effects and two-factor interactions are orthogonal, implying that all the main effects are estimable, ignoring three and more factor interactions. Folded over non-orthogonal designs are recommended because such designs have fewer runs than the competing orthogonal resolution IV designs and only a small efficiency loss in estimating main effects. Miller and Sitter (2005) explored the use of such designs for screening experiments. In screening experiments, the primary goal is to identify the important main effects and the secondary goal is to identify the important two-factor interactions. Folded-over non-orthogonal designs sacrifice some orthogonality of main effects to achieve the complete separation of main effects from two-factor interactions. As argued in Miller and Sitter (2005), this may be a prudent trade-off; that is, these non-orthogonal resolution IV designs can outperform the more commonly recommended orthogonal resolution III designs. Chapter 5 of this thesis is devoted to the selection of folded over non-orthogonal designs and aims to provide a catalogue of good designs based on the proposed design criteria.

1.3 Outline

An outline of the remainder of this thesis is as follows. Chapters 2 and 3 will be devoted to developing new methods for constructing designs for computer experiments. In Chapter 2, we will present methods for constructing many orthogonal Latin hypercubes that are not available in the literature. Construction of nearly-orthogonal and cascading Latin hypercubes are also considered here. In addition, we prove a theorem regarding the existence of orthogonal Latin hypercubes and propose an algorithm for finding orthogonal and nearly orthogonal Latin hypercubes of small runs. In Chapter 3, two generalizations of the basic method proposed in Chapter 2 will be introduced and studied. In Chapters 4 and 5, we turn to two-level FF designs. Specifically, Chapter 4 is concerned with two-level FF designs and provides a collection of good designs based on two criteria of minimum G and G_2 aberration. In Chapter 5, we introduce MDS-resolution and MDS-aberration as criteria for comparing folded over non-orthogonal designs. These criteria and a proposed fast isomorphism check together are used to obtain a catalogue of top two-level folded over non-orthogonal designs. Finally, we will conclude the thesis with a discussion of future research directions in Chapter 6.

Chapter 2

Orthogonal and Cascading Latin Hypercubes

Since the introduction of Latin hypercube sampling by McKay, Beckman and Conover (1979), Latin hypercube designs have become increasingly popular in the area of computer experiments. Except for achieving uniformity in one-dimensions, a Latin hypercube design is merely a combinatorial structure and not directly associated with any criteria such as space-filling or orthogonality. One natural way to find "good" designs within the whole class of Latin hypercube designs is to restrict the attention to a certain class of Latin hypercubes. Such classes include orthogonal array - based Latin hypercubes, orthogonal Latin hypercubes and cascading Latin hypercubes. Here we consider the latter two.

The rest of the chapter is organized as follows. In Section 2.1, we will provide a brief review of Latin hypercubes and each of the two classes of Latin hypercubes. In Section 2.2, we will present methods for constructing many orthogonal Latin hypercubes that are not available in the literature, and the construction of nearly orthogonal Latin hypercubes and cascading Latin hypercubes are also considered here. In Section 2.3, a theorem regarding the existence of orthogonal Latin hypercubes will be proved. In Section 2.4, an adapted algorithm is used to find orthogonal and nearly orthogonal Latin hypercubes of small runs. Section 2.5 provides further methods for constructing orthogonal Latin hypercubes. Section 2.6 concludes the chapter with results and discussions.

2.1 Review

2.1.1 Latin hypercubes

A Latin hypercube design (LHD) is an $n \times m$ matrix, each column of which is a permutation of $\{1, 2, \ldots, n\}$. Each of the $1 \times m$ row vectors is taken as a design point in an m-dimensional design space. The main feature of an LHD is that it achieves uniformity in each of the m univariate margins. An example of a 6×2 LHD has design matrix

$$\begin{pmatrix} 1 & 4 \\ 2 & 3 \\ 3 & 2 \\ 4 & 6 \\ 5 & 1 \\ 6 & 5 \end{pmatrix},$$

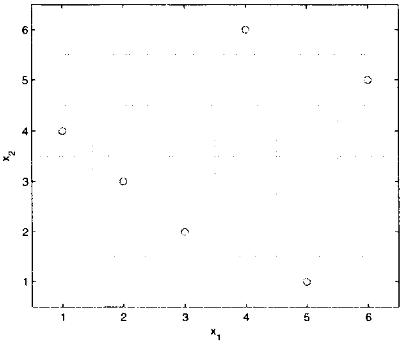
which can be represented graphically by Figure 2.1.

For ease of presentation, hereafter we use a slightly different definition for an LHD. The n entries in each column are taken to be centered at zero and equally-spaced. Thus each column is a permutation of $\{-(n-1)/2, \ldots, 0, \ldots, (n-1)/2\}$ and $\{-(n-1)/2, \ldots, -1/2, 1/2, \ldots, (n-1)/2\}$ when n is odd and even, respectively.

2.1.2 Orthogonal and nearly orthogonal Latin hypercubes

Let $u = [u_1, \ldots, u_n]$ and $v = [v_1, \ldots, v_n]$ be two vectors. The correlation between u and v is then defined as $\sum_i (u_i - \bar{u})(v_i - \bar{v}) / [\sum_i (u_i - \bar{u})^2 \sum_i (v_i - \bar{v})^2]^{1/2}$, where

Figure 2.1: A 6 × 2 Latin hypercube design



$\bar{u} = \sum_i u_i/n$ and $\bar{v} = \sum_i v_i/n$.

Definition 2.1. *A Latin hypercube is said to be orthogonal if all pairs of its columns have zero correlation.*

It is easy to verify that the design in Example 2.1 is an orthogonal LHD.

Example 2.1. An orthogonal LHD with $n = 9$ and $m = 5$ is given by

$$\begin{pmatrix} -4 & -1 & -4 & -2 & -3 \\ -3 & -3 & -1 & 3 & 3 \\ -2 & 2 & 3 & -3 & 1 \\ -1 & 4 & 2 & 0 & -1 \\ 0 & -2 & 4 & 4 & -2 \\ 1 & 1 & 0 & -1 & 0 \\ 2 & 3 & -3 & 2 & 4 \\ 3 & -4 & 1 & -4 & 2 \\ 4 & 0 & -2 & 1 & -4 \end{pmatrix}.$$

The construction of orthogonal LHDs have been considered by Ye (1998), Steinberg and Lin (2006) and Cioppa and Lucas (2007). However, the problem is far from completely solved. In the orthogonal LHDs constructed by Ye (1998), the run size n must have form $n = 2^k$ or 2^k+1 and the corresponding number of factors is $m = 2k-2$

where $k \geq 2$. This means that a very large number of runs is needed to entertain a moderately large number of factors. The orthogonal LHDs constructed by Steinberg and Lin (2006) have a more severe restriction on the run size n, which must be of form $n = 2^{2^k}$. This implies that they in fact only provide two practical run sizes, i.e. $n = 16$ and $n = 256$. Recently, Cioppa and Lucas (2007) extended Ye's approach and thus the constraint on the run size remains. Consequently, there is only a handful of orthogonal LHDs available in the literature. This motivates our work in Section 2.2, where we provide a general method for constructing orthogonal LHDs with much more flexible run sizes.

By slightly sacrificing the orthogonality requirement, we can obtain nearly orthogonal LHDs with even more factors. The definition of what is meant by "nearly" is not unique. In Section 2.2, we find nearly orthogonal LHDs by minimizing the maximum correlation and the average correlation.

The rationale for using orthogonal and nearly orthogonal LHDs has been discussed by various researchers (See, e.g., Iman and Conover 1982; Owen , 1994; Tang, 1998). In particular, the following arguments are related to the use in computer experiments. First, when a lower order polynomial model is employed to fit the data from computer experiments, orthogonal LHDs ensure uncorrelated estimates of linear effects of each input variable; see Ye (1998). Second, Bingham, Sitter and Tang (2008) argued that orthogonal or nearly orthogonal designs can be viewed as useful stepping stones to space-filling designs.

2.1.3 Cascading Latin hypercubes

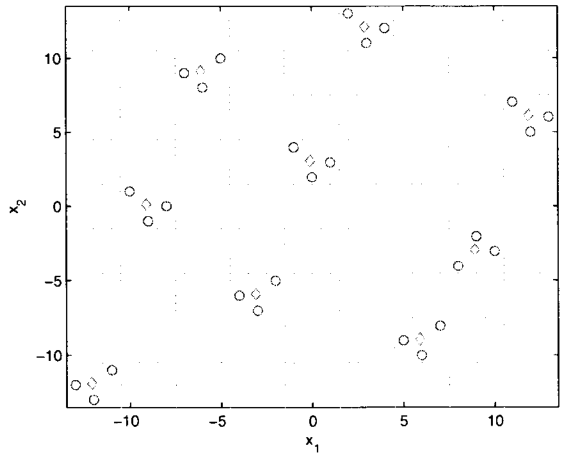
Cascading LHDs were introduced by Handcock (1991). A formal definition is given as follows.

Definition 2.2. *A cascading Latin hypercube of* $n = \prod_{k=1}^{p} n_k$ *points with levels*

$(n_1, \dots, n_p)$ is an n_p-point Latin hypercube about each point in the $(n_1, \dots, n_{p-1})$ cascading Latin hypercube.

Clearly, the usual LHD is the special case with a single level ($p = 1$). Handcock's (1991) experience is that two or three levels ($p = 2$ or 3) are adequate. We illustrate the definition with Example 2.2 below.

Figure 2.2: A cascading Latin hypercube of 27 points with levels $(9, 3)$



Example 2.2. Consider Figure 2.2. The 27 circles together form an LHD. In addition, the 9 diamonds constitute an LHD. Moreover, each of these diamonds is surrounded by a 3-point LHD. Thus, the 27 circles represent a cascading LHD of 27 points with levels $(9, 3)$. □

By Definition 2.2, it is easy to verify a cascading LHD by looking at the geometric distribution of the design points. Next, we give a new definition of a cascading LHD based on its design matrix L. This definition is employed to show a design constructed by the proposed method is a cascading LHD in Section 2.2. Let $V^* = (v^*_{ij})$ be $\lceil V \rceil$ where v^*_{ij} is the nearest integer greater than or equal to v_{ij}.

Definition 2.3. *A Latin hypercube L is termed a cascading Latin hypercube of n points with levels $(n_1, \ldots, n_p)$ if the matrix $U = \lceil L / \prod_{k=q+1}^{p} n_k \rceil$ has $\prod_{k=1}^{q} n_k$ distinct rows, each of which has $\prod_{k=q+1}^{p} n_k$ replicates, for all $q = 1, \ldots, p-1$.*

Cascading LHDs enjoy global space-filling properties as well as having local points. Here global space-filling properties represent the spread of the clustered LHDs. Local points are expected to provide reliable estimates of the scale and smoothness parameters in an additive stochastic model, .as reported by Handcock (1991).

An obvious way to obtain cascading LHDs is replacement. For example, suppose that we wish to construct a cascading LHD of n points with levels (n_1, n_2) for m variables. We first select an $n_1 \times m$ LHD, D_0, as a base design and then replace each design point in D_0 by an $n_2 \times m$ LHD. Note that the dimensionality of the resulting cascading LHD is the same as that of the base design. In contrast, the cascading LHDs constructed in Section 2.2 will have the dimensionality up to n_2 times dimensionality of the base design D_0.

2.2 A flexible construction method

In this section, we will first introduce a construction method. We then show how this method can be used to construct LHDs, orthogonal LHDs, nearly orthogonal LHDs and cascading LHDs.

Consider designs with n runs and m factors, each factor at s levels, where $2 \leq s \leq n$. We denote such designs as $D(n, s^m)$, represented by an $n \times m$ matrix, $D = (d_{ij})$, with entries from a set of s levels. Without loss of generality, the s levels are taken to be centered at zero and equally-spaced. Thus the levels are $\{-(s-1)/2, \ldots, 0, \ldots, (s-1)/2\}$ and $\{-(s-1)/2, \ldots, -1/2, 1/2, \ldots, (s-1)/2\}$ when s is odd and even, respectively. In particular, LHDs are such designs with $s = n$.

Let $A = (a_{ij})$ be an $n_1 \times m_1$ matrix with entries $a_{ij} = \pm 1$, $B = (b_{ij})$ be a $D(n_2, s_2^{m_2})$, $C = (c_{ij})$ be a $D(n_1, s_1^{m_1})$, and $D = (d_{ij})$ be an $n_2 \times m_2$ matrix with entries $d_{ij} = \pm 1$. Let γ be any real number. Consider the construction:

$$L = A \otimes B + \gamma C \otimes D, \tag{2.1}$$

where Kronecker product $A \otimes B$ is the $n_1 n_2 \times m_1 m_2$ block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \ldots & a_{1m_1}B \\ a_{21}B & a_{22}B & \ldots & a_{2m_1}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_1 1}B & a_{n_1 2}B & \ldots & a_{n_1 m_1}B \end{bmatrix}$$

with $a_{ij}B$ itself being an $n_2 \times m_2$ matrix. Designs B and C are called base designs. While A and D are technically two-level designs, they play a different role from designs B and C and are used to provide replicates of designs B and C respectively. The resulting design L in (2.1) has $n = n_1 n_2$ runs and $m = m_1 m_2$ factors.

The above construction has some interesting features. To explain, consider a simple case in which $A = (1,1)^T$ and $C = (1/2, -1/2)^T$. Design L in (2.1) has a column

$$\begin{pmatrix} b + \frac{\gamma}{2}d \\ b - \frac{\gamma}{2}d \end{pmatrix}, \tag{2.2}$$

where b is a column of B and d is a column of D. Further let $b = (b_1, \ldots, b_{n_2})^T$ and $d = (d_1, \ldots, d_{n_2})^T$. The entries in the column (2.2) are $b_i + \gamma d_i/2$ and $b_i - \gamma d_i/2$ with $i = 1, \ldots, n_2$. Because $d_i = \pm 1$, the column (2.2) has entries $b_i + \gamma/2$ and $b_i - \gamma/2$. Consequently, the column (2.2) can be viewed as simultaneously shifting each level in b to the left and the right by the same length $\gamma/2$. If we view b as a block, this is equivalent to shifting two identical blocks b, one to the left and the other to the right. We will show that with the appropriate choices of A, B, C, D and γ (in Proposition 2.1 in the next section), the levels in each column of L in (2.1) are equally-spaced and unreplicated, resulting in a Latin hypercube. Now consider all m columns of L under this simple case. Each one-dimensional block b becomes an m-dimensional stratum B. Suppose D is a matrix of plus ones, then the design points in $B + \gamma D/2$ can be obtained by shifting the entire stratum B to the right by the length $\gamma/2$. Similarly, the design points in $B - \gamma D/2$ can be obtained by shifting the entire stratum B to the left by the length $\gamma/2$. In this case, closely clustered points in each stratum are expected in design L in (2.1). This feature will be utilized to construct cascading Latin hypercubes in the next section.

The orthogonality or near orthogonality of L in (2.1) is determined by the orthogonality or near orthogonality of A, B, C and D, the correlations between the columns in A and those in C, and the correlations between the columns in B and those in D. As a result, the method allows orthogonal and nearly orthogonal Latin hypercubes to be easily constructed.

Vartak (1955) appears to be the first to use Kronecker product to construct statistical experimental designs. In a recent work, Bingham, Sitter and Tang (2008) introduced a method for constructing a rich class of designs that are suitable for use in computer experiments. Their method is a special case of our proposed method in (2.1) with $\gamma = 0$. Unlike our method, theirs does not produce LHDs.

2.2.1 Constructing Latin hypercubes

The following proposition tells us how to obtain a large LHD based on small LHDs using the method (2.1).

Proposition 2.1. *A design L, formed as in (2.1), is a Latin hypercube if*

(i) $s_1 = n_1$, $s_2 = n_2$;

(ii) $\gamma = n_2$;

(iii) *there do not exist i and j, where $i = 1, \ldots, m_1$ and $j = 1, \ldots, m_2$, such that $a_{pi} = -a_{p'i}$ and $d_{qj} = -d_{q'j}$ simultaneously hold, where p and p' are such that $c_{pi} = -c_{p'i}$ and q and q' are such that $b_{qj} = -b_{q'j}$.*

Proof. Let $n = n_1 n_2$. Conditions (i), (ii) and (iii) ensure that each column of L is $\{-(n-1)/2, \ldots, 0, \ldots, (n-1)/2\}$ if both n_1 and n_2 are odd and $\{-(n-1)/2, \ldots, -1/2, 1/2, \ldots, (n-1)/2\}$ otherwise. □

Proposition 2.1 gives the conditions under which designs constructed in (2.1) result in Latin hypercubes. Condition (i) implies that both B and C are required to be Latin hypercubes. Recall that in the previous section, we view the column produced by the term $A \otimes B$ as n_1 blocks. Here a Latin hypercube B is used to ensure that the levels in each block are equally-spaced and unreplicated. Condition (iii) is needed to prevent replicated levels from occurring in each column of L. Furthermore, a Latin hypercube C in combination with condition (ii) guarantees that among the n_1 blocks after shifting, any two consecutive blocks have spacing equal to 1.

We now discuss condition (iii). First, it implies that if there exists (p, p', i) such that $c_{pi} = -c_{p'i}$ and $a_{pi} = -a_{p'i}$, then D must satisfy $d_{qj} = d_{q'j}$ where q and q' are row indices such that $b_{qj} = -b_{q'j}$ for all $j = 1, \ldots, m_2$. Second, the following are three cases in which condition (iii) is met: (a) either A or D or both are identity matrices; (b) if C is a symmetric LHD in the sense that $C = (C_0^T, -C_0^T)^T$, then using $A = (A_0^T, A_0^T)^T$ will satisfy condition (iii); (c) B is a symmetric LHD and D has form $D = (D_0^T, D_0^T)^T$.

Example 2.3. Suppose one wishes to construct 32×32 Latin hypercubes. There are various choices of n_1, n_2, m_1 and m_2 such that $n = n_1 n_2 = 32$ and $m = m_1 m_2 = 32$. One such choice is $n_1 = m_1 = 2$, $n_2 = m_2 = 16$. To meet condition (iii) in Proposition 2.1, we can choose either A or D to be a matrix of plus ones. Suppose we let A be a matrix of plus ones. By Proposition 2.1, we now let $\gamma = n_2 = 16$, choose any 16×16

matrix $D = (d_{ij})$ with $d_{ij} = \pm 1$ and any Latin hypercubes B and C. For example, we can let $C = \{(1/2, -1/2)^T, (-1/2, 1/2)^T\}^T$ and

$$B = \frac{1}{2}\begin{pmatrix}
-15 & 5 & 9 & -3 & 7 & 11 & -11 & 7 & -9 & 3 & -15 & 5 & 11 & -11 & 7 & -7 \\
-13 & 1 & 1 & 13 & -7 & -11 & 11 & -7 & -1 & -13 & -13 & 1 & 13 & 5 & 5 & -3 \\
-11 & 7 & -7 & -11 & 13 & -1 & -1 & -13 & 9 & -3 & 15 & -5 & -5 & 11 & -7 & 7 \\
-9 & 3 & -15 & 5 & -13 & 1 & 1 & 13 & 1 & 13 & 13 & -1 & -13 & -5 & -5 & 3 \\
-7 & -11 & 11 & -7 & 11 & -7 & 7 & 11 & 5 & 15 & -3 & -9 & -9 & 3 & 9 & 11 \\
-5 & -15 & 3 & 9 & -11 & 7 & -7 & -11 & 13 & -1 & -1 & -13 & -1 & 9 & 11 & 15 \\
-3 & -9 & -5 & -15 & 1 & 13 & 13 & -1 & -5 & -15 & 3 & 9 & 1 & 7 & -11 & -11 \\
-1 & -13 & -13 & 1 & -1 & -13 & -13 & 1 & -13 & 1 & 1 & 13 & 9 & -9 & -9 & -15 \\
1 & 13 & 13 & -1 & -9 & 3 & -15 & 5 & 11 & -7 & 7 & 11 & -7 & -7 & -15 & -9 \\
3 & 9 & 5 & 15 & 9 & -3 & 15 & -5 & 3 & 9 & 5 & 15 & -15 & -13 & -13 & -13 \\
5 & 15 & -3 & -9 & -3 & -9 & -5 & -15 & -11 & 7 & -7 & -11 & 15 & -3 & 15 & 9 \\
7 & 11 & -11 & 7 & 3 & 9 & 5 & 15 & -3 & -9 & -5 & -15 & 7 & 15 & 13 & 13 \\
9 & -3 & 15 & -5 & -5 & -15 & 3 & 9 & -7 & -11 & 11 & -7 & 5 & 13 & -3 & 5 \\
11 & -7 & 7 & 11 & 5 & 15 & -3 & -9 & -15 & 5 & 9 & -3 & 3 & -1 & -1 & 1 \\
13 & -1 & -1 & -13 & -15 & 5 & 9 & -3 & 7 & 11 & -11 & 7 & -11 & -15 & 3 & -5 \\
15 & -5 & -9 & 3 & 15 & -5 & -9 & 3 & 15 & -5 & -9 & 3 & -3 & 1 & 1 & -1
\end{pmatrix}.$$

The design formed as in (2.1) is then a 32×32 Latin hypercube.

Note that an LHD given by $B \otimes A + \gamma D \otimes C$ is equivalent to that in (2.1) up to row permutations and column permutations, and we therefore only consider the construction (2.1) in the later development. It is worthwhile to mention that in the method (2.1), we can obtain a rich class of new LHDs by applying different row permutations, column permutations, and/or sign-switching columns of A, B, C and D.

2.2.2 Constructing orthogonal Latin hypercubes

In this section, the proposed method is adapted to construct orthogonal LHDs of size $n = 8k$, where k is any positive integer, which provide much more flexible run sizes than those given by the methods of Ye (1998), Steinberg and Lin (2006) and Cioppa and Lucas (2007).

Consider a design or matrix $D = (d_1, \ldots, d_m)$, where d_j is the jth column of D. A design or matrix D is called column-orthogonal if any two columns of D are orthogonal, i.e., $d_i^T d_j = 0$ for any $i \neq j$. Column-orthogonality is weaker than orthogonality because it does not require each column of D to be balanced. In the proposition below, column-orthogonal matrices with entries ±1 are used. Hadamard matrices and two-level orthogonal arrays with levels ±1 are such column-orthogonal matrices. The following proposition provides sufficient conditions for a design L in (2.1) to be column-orthogonal.

Proposition 2.2. *Let A and D be column-orthogonal. A design L, formed as in (2.1), is column-orthogonal if designs B and C are both orthogonal, and at least one of the two, $A^TC = 0$ and $B^TD = 0$, holds.*

Proof. Let $L(i,j,p,q)$ be the entry produced by a_{ip}, b_{jq}, c_{ip} and d_{jq}. The validity of this proposition can be easily established by noting that

$$\begin{aligned}
&\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} L(i,j,p,q)L(i,j,p',q') \\
= &\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}(a_{ip}b_{jq}+\gamma c_{ip}d_{jq})(a_{ip'}b_{jq'}+\gamma c_{ip'}d_{jq'}) \\
= &\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}(a_{ip}b_{jq}a_{ip'}b_{jq'}+\gamma a_{ip}b_{jq}c_{ip'}d_{jq'}+\gamma c_{ip}d_{jq}a_{ip'}b_{jq'}+\gamma^2 c_{ip}d_{jq}c_{ip'}d_{jq'}) \\
= &\sum_{i=1}^{n_1} a_{ip}a_{ip'}\sum_{j=1}^{n_2} b_{jq}b_{jq'}+\gamma\sum_{i=1}^{n_1} a_{ip}c_{ip'}\sum_{j=1}^{n_2} b_{jq}d_{jq'} \\
&\quad +\gamma\sum_{i=1}^{n_1} c_{ip}a_{ip'}\sum_{j=1}^{n_2} d_{jq}b_{jq'}+\gamma^2\sum_{i=1}^{n_1} c_{ip}c_{ip'}\sum_{j=1}^{n_2} d_{jq}d_{jq'}. \qquad (2.3)
\end{aligned}$$

Clearly, the first and last terms on the right hand side are zero by orthogonality of A, B, C and D. The second and third terms vanish due to either of the two additional conditions. □

Theorem 2.1. *A design L, formed as in (2.1), is an orthogonal Latin hypercube if*

(i) $\gamma = n_2$;

(ii) *A and D are column-orthogonal;*

(iii) *B and C are orthogonal Latin hypercubes;*

(iv) *there do not exist i and j, where $i = 1, \ldots, m_1$ and $j = 1, \ldots, m_2$, such that $a_{pi} = -a_{p'i}$ and $d_{qj} = -d_{q'j}$ simultaneously hold, where p and p' are such that $c_{pi} = -c_{p'i}$ and q and q' are such that $b_{qj} = -b_{q'j}$;*

(v) *at least one of the two, $A^T C = 0$ and $B^T D = 0$, holds.*

Theorem 2.1 is a direct consequence of Propositions 2.1 and 2.2. Conditions (i) and (iv), and LHDs B and C are sufficient to obtain an LHD. The orthogonality of LHDs B and C, conditions (ii) and (v) are needed for the orthogonality of the LHD L. Condition (v) implies that either the correlations between the columns of A and those of C are zero, or the correlations between the columns of B and those of D are zero. In addition, the run sizes n_1 and n_2 must be either 2 or a multiple of 4 because of the column-orthogonality of A and D. Thus, the run size n must be of form $n = 8k$ ($k = 1, 2, \ldots$). Note that, technically, orthogonal designs must have at least two factors, but if a design B or C has only one factor, it is orthogonal by our definition. Next, we revisit Example 2.3 and give an example of the use of Theorem 2.1.

Example 2.4. Recall Example 2.3. The first 12 columns of B form an orthogonal LHD of 16 runs, due to Steinberg and Lin (2006). Theorem 2.1 tells us that if D is column-orthogonal, the first 12 columns of L, in Example 2.3, constitute a 32 × 12 orthogonal LHD (See Appendix A), which offers more orthogonal factors than the existing designs in the literature; a 32 × 11 orthogonal LHD was provided by Cioppa and Lucas (2007). □

We are now in a position to present a result that goes beyond Theorem 2.1 by doubling the dimensionality of certain orthogonal LHDs constructed by the method (2.1).

Theorem 2.2. *Suppose that in (2.1), $n_1 = n_2 = n_0$ and A, B, C, D, and γ satisfy the conditions in Theorem 2.1. Let $U = -n_0 A \otimes B + C \otimes D$. Then design [L, U] is an $n_0^2 \times 2m$ orthogonal Latin hypercube, where $m = m_1 m_2$.*

Proof. Design U is obviously an $n_0^2 \times m$ orthogonal LHD as L is. Thus, it remains to show every column from L and every column from U have zero correlation. Consider a column $L(i, j, p, q)$ and a column $U(i, j, p', q')$, we then have

$$\begin{aligned}
& \sum_{i=1}^{n_0}\sum_{j=1}^{n_0} L(i,j,p,q)U(i,j,p',q') \\
= & \sum_{i=1}^{n_0}\sum_{j=1}^{n_0} (a_{ip}b_{jq} + n_0 c_{ip}d_{jq})(-n_0 a_{ip'}b_{jq'} + c_{ip'}d_{jq'}) \\
= & \sum_{i=1}^{n_0}\sum_{j=1}^{n_0} (-n_0 a_{ip}b_{jq}a_{ip'}b_{jq'} + a_{ip}b_{jq}c_{ip'}d_{jq'} - n_0^2 c_{ip}d_{jq}a_{ip'}b_{jq'} + n_0 c_{ip}d_{jq}c_{ip'}d_{jq'}) \\
= & -n_0 \sum_{i=1}^{n_0} a_{ip}a_{ip'} \sum_{j=1}^{n_0} b_{jq}b_{jq'} + \sum_{i=1}^{n_0} a_{ip}c_{ip'} \sum_{j=1}^{n_0} b_{jq}d_{jq'} \\
& \qquad -n_0^2 \sum_{i=1}^{n_0} c_{ip}a_{ip'} \sum_{j=1}^{n_0} d_{jq}b_{jq'} + n_0 \sum_{i=1}^{n_0} c_{ip}c_{ip'} \sum_{j=1}^{n_0} d_{jq}d_{jq'} \\
= & \ 0.
\end{aligned}$$

The second and third terms on the right hand side equal zero because of either of the two additional conditions for L to be an orthogonal LHD. For the first and last terms, both of them equal zero in the cases $(p, q) \neq (p', q')$. In the case of $(p, q) = (p', q')$, the first and last terms are canceled out. Hence, we conclude that the new design $[L, U]$ is an $n_0^2 \times 2m$ orthogonal LHD. □

Theorem 2.2 is applicable to the case in which the run size n must be of form $n = n_0^2$. Note that n_0 must be a multiple of 4 because of the column-orthogonality of A and D. The orthogonal LHD $[L, U]$ in Theorem 2.2 generally possesses more columns than any one obtained by directly using (2.1). This point can be illustrated using the following example.

Example 2.5. Suppose that we wish to construct orthogonal LHDs of 64 runs. The first approach is to set

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix} \text{ and } C = \frac{1}{2}\begin{pmatrix} 1 & -3 \\ 3 & 1 \\ -1 & 3 \\ -3 & -1 \end{pmatrix},$$

take the first 12 columns of B in Example 2.3 to be a new B. Choosing any column-orthogonal 16×12 matrix $D = (d_{ij})$ with entries $d_{ij} = \pm 1$ and applying the method (2.1) with $\gamma = 16$, we obtain a 64×24 orthogonal LHD.

Alternatively, we can choose

$$A = D = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \text{ and } B = C = \frac{1}{2}\begin{pmatrix} 1 & -3 & 7 & 5 \\ 3 & 1 & 5 & -7 \\ 5 & -7 & -3 & -1 \\ 7 & 5 & -1 & 3 \\ -1 & 3 & -7 & -5 \\ -3 & -1 & -5 & 7 \\ -5 & 7 & 3 & 1 \\ -7 & -5 & 1 & -3 \end{pmatrix}.$$

Let L and U be $A \otimes B + 8C \otimes D$ and $-8A \otimes B + C \otimes D$, respectively. By Theorem 2.2, design $[L, U]$ is a 64×32 orthogonal LHD, which has more columns than the one obtained by the first approach. □

Theorem 2.1 and Theorem 2.2 are powerful results for constructing orthogonal LHDs. Theorem 2.1 is useful for providing orthogonal LHDs of size $8k$, thereby filling some of the vast gaps between the available run sizes. Theorem 2.1 and Theorem 2.2 will be used to construct many orthogonal LHDs not available in the literature in Section 2.6.

2.2.3 Constructing nearly orthogonal Latin hypercubes

In this section, we show how the method (2.1) can be adapted for constructing nearly orthogonal LHDs. The basic result is that if the base designs B and C are nearly orthogonal, the method (2.1) produces a nearly orthogonal LHD under some mild conditions.

To assess the near orthogonality, we use two measures defined in Bingham, Sitter and Tang (2008). For a design $D = (d_1, \ldots, d_m)$, where d_j is the jth column of D, they define $\rho_{ij}(D)$ to be $J(d_i, d_j)/[J(d_i, d_i)J(d_j, d_j)]^{1/2}$, where $J(d_i, d_j) = d_i^T d_j$. If the mean of the levels in d_j for all $j = 1, \ldots, m$ is zero, then $\rho_{ij}(D)$ is simply the correlation coefficient between columns d_i and d_j. This is the case for any LHD with levels as described in this chapter. Bingham, Sitter and Tang (2008) then defined two measures of near orthogonality, namely, $\rho_M(D) = \max_{i<j}|\rho_{ij}(D)|$ and $\rho^2(D) = \sum_{i<j} \rho_{ij}^2(D)/[(m(m-1)/2]$. Smaller values of $\rho_M(D)$ and $\rho^2(D)$ imply the near orthogonality. Obviously, if $\rho_M(D)$ or $\rho^2(D)$ is equal to zero, an orthogonal LHD is obtained. The following theorem relaxes the conditions in Theorem 2.1 by allowing LHDs B and C to be nearly-orthogonal.

Theorem 2.3. *Suppose that A, B, C, D and γ in (2.1) satisfy the conditions in Proposition 2.1. Furthermore, let A and D be column-orthogonal, and either $A^T C = 0$, or $B^T D = 0$. We then have that*

(i) $\rho^2(L) = w_1\rho^2(B) + w_2\rho^2(C)$, and

(ii) $\rho_M(L) = Max\{w_3\rho_M(B), w_4\rho_M(C)\}$,

where w_1, w_2, w_3 and w_4 are given by $w_1 = (m_2-1)(n_2^2-1)^2/[(m_1m_2-1)(n^2-1)^2]$, $w_2 = n_2^4(m_1-1)(n_1^2-1)^2/[(m_1m_2-1)(n^2-1)^2]$, $w_3 = (n_2^2-1)/(n^2-1)$ and $w_4 = n_2^2(n_1^2-1)/(n^2-1)$.

Proof. Let $L(i, j, p, q)$ be the entry produced by a_{ip}, b_{jq}, c_{ip} and d_{jq}. Let $L(p, q)$ be the column to which the entry $L(i,j,p,q)$ belongs. Provided that either $A^TC=0$, or $B^TD=0$, the equation (2.3) reduces to

$$\begin{aligned}
&\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} L(i,j,p,q)L(i,j,p',q') \\
=\; &\sum_{i=1}^{n_1} a_{ip}a_{ip'}\sum_{j=1}^{n_2} b_{jq}b_{jq'}+\gamma^2\sum_{i=1}^{n_1} c_{ip}c_{ip'}\sum_{j=1}^{n_2} d_{jq}d_{jq'} \\
=\; &\sum_{i=1}^{n_1} a_{ip}a_{ip'}\sum_{j=1}^{n_2} b_{jq}b_{jq'}+n_2^2\sum_{i=1}^{n_1} c_{ip}c_{ip'}\sum_{j=1}^{n_2} d_{jq}d_{jq'}. \qquad (2.4)
\end{aligned}$$

The last step follows as $\gamma=n_2$, a condition such that L is an LHD. In addition, the value setting of (p,q,p',q') consists of three cases (a) $p=p', q\neq q'$; (b) $p\neq p', q=q'$; (c) $p\neq p', q\neq q'$.

To derive the quantities $\rho^2(L)$ and $\rho_M(L)$, we first consider $\rho[L(p,q),L(p',q')]$, which has form

$$\rho[L(p,q),L(p',q')]=J[L(p,q),L(p',q')]/\{J[L(p,q),L(p,q)]J[L(p',q'),L(p',q')]\}^{1/2},$$

where $J[L(p,q),L(p',q')]=[L(p,q)]^T[L(p',q')]$. Note that matrices A and D are required to be column-orthogonal, thereby implying that both n_1 and n_2 are even. Thus, $J[L(p,q),L(p,q)]=n(n^2-1)/12\triangleq N$. We then have

$$\begin{aligned}
\rho[L(p,q),L(p',q')] &= J[L(p,q),L(p',q')]/N \\
&= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} L(i,j,p,q)L(i,j,p',q')/N \\
&= \{n_1\rho_{pp'}(A)\rho_{qq'}(B)n_2(n_2^2-1)/12 \\
&\quad +n_2^2[n_1(n_1^2-1)/12]\rho_{pp'}(C)\rho_{qq'}(D)n_2\}/N. \qquad (2.5)
\end{aligned}$$

The last step follows by the equation (2.4).

We now consider the right hand side of (2.5) under the three cases of the value settings of (p, q, p', q'). First, consider the case (c). Matrices A and D are assumed to be column-orthogonal. Thus $\rho_{pp'}(A)$ and $\rho_{qq'}(D)$ equal zero, thereby giving us $\rho[L(p,q), L(p',q')] = 0$.

For the cases (a) and (b), we have $\rho_{pp'}(A) = \rho_{pp'}(C) = 1$ and $\rho_{qq'}(B) = \rho_{qq'}(D) = 1$, respectively. Consequently, we obtain

$$\begin{aligned}
\rho^2(L) &= \sum_{p=1}^{m_1}\sum_{q=1}^{m_2}\sum_{p'=1}^{m_1}\sum_{q'=1_{(p,q)\neq(p',q')}}^{m_2} \rho^2[L(p,q), L(p',q')]/M \\
&= [m_1 \sum_{q\neq q'} (\frac{n_2^2-1}{n^2-1})^2 \rho^2_{qq'}(B) + m_2 \sum_{p\neq p'} n_2^4 (\frac{n_1^2-1}{n^2-1})^2 \rho^2_{pp'}(C)]/M \\
&= [(\frac{n_2^2-1}{n^2-1})^2 m_1 m_2 (m_2-1)\rho^2(B) + n_2^4 (\frac{n_1^2-1}{n^2-1})^2 m_1 m_2 (m_1-1)\rho^2(C)]/M \\
&= \frac{m_2-1}{m_1m_2-1}(\frac{n_2^2-1}{n^2-1})^2 \rho^2(B) + n_2^4 \frac{m_1-1}{m_1m_2-1}(\frac{n_1^2-1}{n^2-1})^2 \rho^2(C),
\end{aligned}$$

where $M = m_1 m_2 (m_1 m_2 - 1)$.

Let w_1 be $(m_2-1)(n_2^2-1)^2/[(m_1m_2-1)(n^2-1)^2]$ and w_2 be $n_2^4(m_1-1)(n_1^2-1)^2/[(m_1m_2-1)(n^2-1)^2]$. Note that

$$\begin{aligned}
w_1 + w_2 &< (n_2^2-1)^2/(n^2-1)^2 + n_2^4(n_1^2-1)^2/(n^2-1)^2 \\
&= \{n^4 - 2n_2^4[n_1^2 - 1 + 1/n_2^2 - 1/(2n_2^4)]\}/\{n^4 - 2n_2^4(n_1^2/n_2^2 - 1/(2n_2^4))]\} \\
&< 1.
\end{aligned}$$

We then finish the proof for part (i) in Theorem 2.3. Part (ii) is obvious by the definition of $\rho_M(L)$. □

Theorem 2.3 is a generalization of Theorem 2.1. That is, if we relax the conditions in Theorem 2.1 by allowing LHDs B and C to be nearly orthogonal, the LHD L constructed is then also nearly orthogonal, in terms of both measures of near orthogonality. An example, illustrating the use of this result, is considered below.

Example 2.6. Let A be $(1,1)^T$ and C be $(1/2,-1/2)^T$. Set γ to be 16 and choose a nearly orthogonal LHD

$$B=\frac{1}{2}\begin{pmatrix}
-15 & 15 & -13 & 13 & -5 & -13 & 5 & 3 & -1 & 5 & -7 & 5 & -9 & -9 & 5\\
-13 & -15 & -3 & 3 & 7 & 3 & 15 & -11 & 13 & -5 & 7 & -13 & -7 & -3 & -3\\
-11 & -9 & -5 & -11 & -15 & 13 & -5 & 11 & -9 & 9 & 9 & 3 & -5 & -1 & -11\\
-9 & -1 & 9 & -15 & -11 & 1 & -1 & -13 & 5 & -1 & -15 & 7 & 1 & 3 & 15\\
-7 & 1 & -7 & 7 & 15 & 15 & -13 & 9 & -5 & -13 & -3 & -1 & -1 & 7 & 13\\
-5 & 13 & 11 & -5 & 9 & -7 & -3 & -9 & -13 & 11 & 13 & -9 & -3 & 13 & 1\\
-3 & -5 & 13 & 15 & -9 & -9 & -11 & 1 & 7 & -9\ . & 15 & 11 & 9 & 1 & -1\\
-1 & -11 & 3 & -7 & 11 & -15 & 13 & 15 & -7 & -3 & -9 & 9 & 7 & 9 & -5\\
1 & 3 & -9 & -3 & -1 & -5 & -15 & -1 & 11 & 3 & -11 & -15 & 15 & 5 & -15\\
3 & -3 & 15 & 11 & 3 & 9 & 1 & -7 & -15 & 1 & -13 & -3 & 3 & -15 & -9\\
5 & 9 & 7 & -1 & 5 & 11 & 9 & 13 & 15 & 15 & 5 & 1 & 11 & -7 & 9\\
7 & 7 & -1 & -13 & 13 & -1 & -7 & -5 & 9 & -7 & 3 & 15 & -13 & -11 & -13\\
9 & 5 & -11 & -9 & -7 & -3 & 7 & -3 & -11 & -15 & 11 & -7 & 13 & -13 & 7\\
11 & 11 & 5 & 5 & -13 & 7 & 11 & 5 & 3 & -11 & -5 & -5 & -11 & 15 & -7\\
13 & -7 & -15 & 9 & 1 & 5 & 3 & -15 & -3 & 13 & 1 & 13 & 5 & 11 & 3\\
15 & -13 & 1 & 1 & -3 & -11 & -9 & 7 & 1 & 7 & -1 & -11 & -15 & -5 & 11
\end{pmatrix},$$

with $\rho^2(B)=0.0003$ and $\rho_M(B)=0.0765$. Taking any 15 columns of a Hadamard matrix of order 16 to be D and applying the approach (2.1), we obtain an LHD L of 32 run and 15 factors. As $\rho^2(C)=\rho_M(C)=0$, we have $\rho^2(L)=(n_2^2-1)^2\rho^2(B)/(n^2-1)^2=0.0621\rho^2(B)=0.00002$ and $\rho_M(L)=(n_2^2-1)\rho_M(B)/(n^2-1)=0.2493\rho_M(B)=0.0191$. □

In the use of Theorem 2.3, we notice that both A and D are required to be column-orthogonal. This means that both n_1 and n_2 must be 2 or a multiple of 4. In other words, the run size n must be a multiple of 8. To cope with other run sizes, we can instead let B and C be orthogonal LHDs and a result similar to Theorem 2.3 can also be obtained. That is, if A and D are nearly column-orthogonal matrices, the method (2.1) gives a nearly orthogonal LHD when B and C are orthogonal LHDs and either $A^TC=0$ or $B^TD=0$ is true.

2.2.4 Constructing cascading Latin hypercubes

We begin by presenting a theorem based on which we construct two-level cascading LHDs using the method (2.1). We then mention how k-level cascading LHDs ($k = 3, 4, \ldots$) can be further constructed.

Theorem 2.4. *Let D be an $n_2 \times m_2$ matrix of plus ones. A design L, formed as in (2.1), is a two-level cascading Latin hypercube of $n = n_1 n_2$ points with level (n_1, n_2) if (i) $s_1 = n_1$ and $s_2 = n_2$; (ii) $\gamma = n_2$.*

Proof. Provided that D is a matrix of plus ones, the resulting design L is an LHD, following from Proposition 2.1. To establish the cascading property, we employ Definition 2.3. Let $\lceil r \rceil$ be the nearest integer greater than or equal to r. Let $L(i, j, p, q)$ be the entry produced by a_{ip}, b_{jq}, c_{ip} and d_{jq}. We have

$$\begin{aligned} U(i,j,p,q) &= \lceil L(i,j,p,q)/n_2 \rceil = \lceil a_{ip}b_{jq}/n_2 + c_{ip}d_{jq} \rceil \\ &= \lceil a_{ip}b_{jq}/n_2 + c_{ip} \rceil = \lceil c_{ip} \rceil, \end{aligned} \tag{2.6}$$

which follows from the fact that $d_{jq} = 1$, $|a_{ip}b_{jq}/n_2| < 1/2$, and c_{ip} is either a multiple of $1/2$ (when n_1 is even) or an integer (when n_1 is odd). The equation (2.6) informs us that each row of matrix U is completely determined by a row of C. More specifically, matrix U has n_1 distinct rows of n_2 replicates, yielding Theorem 2.4. □

We discuss a couple of issues regarding Theorem 2.4. First, the proof of Theorem 2.4 reveals that the global space-filling properties of cascading LHDs constructed in the theorem are controlled by the space-filling properties of the LHD C. Locally, $a_{ip}B$ determines the layout of the design points in each clustered LHD. Therefore, in the use of Theorem 2.4, we may use good space-filling designs such as maximin LHDs (see, e.g., Morris and Mitchell, 1995) for B and C in order to achieve good space-filling properties, both globally and locally.

Second, the consequence of using a matrix D of plus ones in the method (2.1) to construct cascading LHDs is that the bivariate projection of the columns $\{(i-1)m_2+1,\ldots,im_2, i=1,\ldots,m_1\}$ has undesirable diagonal patterns. This disadvantage partly motivates the work in the next chapter .

We now briefly describe how we can construct k-level ($k>2$) cascading LHDs. Suppose that we have obtained a two-level cascading LHD, we then can create a k-level cascading LHD simply by taking a $(k-1)$-level cascading LHD to be C and using Theorem 2.4.

2.3 An existence result

In this section, we present a theorem on the existence of orthogonal LHDs. An LHD with one column is orthogonal in the previous sections, but an orthogonal LHD must have two or more columns in Theorem 2.5 below.

Theorem 2.5. *There exists an orthogonal Latin hypercube if and only if the run size n is not equal to 3 and does not have form $4k+2$, where $k=0,1,2,\ldots$.*

Proof. We will prove Theorem 2.5 by showing that (i) there exists an orthogonal LHD of odd size ($n \neq 3$) ; (ii) there exists an orthogonal LHD of size n having the form $n=4k$; (iii) there does not exist an orthogonal LHD of size 3 and $4k+2$, where $k \geq 0$.

To show (i) and (ii), we will make use of the following orthogonal LHDs of 4, 5 and 7 runs with two factors

$$\begin{pmatrix} 3/2 & 1/2 & -1/2 & -3/2 \\ -1/2 & 3/2 & -3/2 & 1/2 \end{pmatrix}^T, \begin{pmatrix} 2 & 1 & 0 & -1 & -2 \\ -1 & 2 & 0 & -2 & 1 \end{pmatrix}^T \text{ and }$$
$$\begin{pmatrix} 3 & 2 & 1 & 0 & -1 & -2 & -3 \\ -2 & -1 & 3 & 0 & 2 & 1 & -3 \end{pmatrix}^T.$$

Let

$$O_2 = \begin{pmatrix} x_1 & -x_1 & x_2 & -x_2 \\ x_2 & -x_2 & -x_1 & x_1 \end{pmatrix}^T .$$

If we stack an 4×2 orthogonal LHD on O_2 with $x_1 = 5/2, x_2 = 7/2$, the resulting design is an 8×2 orthogonal LHD. Similarly, if we stack an 5×2 orthogonal LHD on O_2 with $x_1 = 3, x_2 = 4$, we obtain a 9×2 orthogonal LHD. In the same fashion, we can obtain 11×2 orthogonal LHDs. In general, suppose we have an $n \times 2$ orthogonal LHD, where the run size n has form $4k$, $4k+1$ or $4k+3$, we can obtain an $(n+4) \times 2$ orthogonal LHD by stacking an $n \times 2$ orthogonal LHD on O_2 with $x_1 = (n-3)/2, x_2 = (n-1)/2$.

It is easy to verify that there does not exist an orthogonal LHD of size 3. Thus, to show (iii), it remains to show that there does not exist an orthogonal LHD of size $4k+2$ ($k = 0, 1, \ldots$). Equivalently, our target is to show that there are no two orthogonal columns in an LHD of size $4k + 2$. Let $a = (a_1, \ldots, a_n)^T$ and $b = (b_1, \ldots, b_n)^T$ be the first and second column of such an LHD and both a and b are permutations of $\{1/2, 3/2, \ldots, (n-1)/2, -1/2, -3/2, \ldots, -(n-1)/2\}$. Note that $\sum_{i=1}^n a_i = 0$, $\sum_{i=1}^n b_i = 0$. Without loss of generality, we assume that a has form $(1/2, 3/2, \ldots, (n-1)/2, -1/2, -3/2, \ldots, -(n-1)/2)^T$. In other words, $a_i = -a_{i+n/2} = (2i-1)/2$. We will prove the result by contradiction. Suppose columns a and b are orthogonal, that is, $\sum_{i=1}^n a_i b_i = 0$, which can be rewritten as

$$2^{-1} \sum_{i=1}^{n/2} [(2b_i)i - (2b_{i+n/2})(i-1)] = 0. \tag{2.7}$$

Note that both $2b_i$ and $2b_{i+n/2}$ are odd, $i = 1, \ldots, n/2$. The quantity $(2b_i)i - (2b_{i+n/2})(i-1)$ must be odd as $(2b_i)i$ and $(2b_{i+n/2})(i-1)$ cannot be both even or both odd. In addition, $n/2$ must be odd. It is obvious that the addition or substraction among odd numbers of odd integers gives an odd integer. This leads to a contradiction and we therefore conclude that there does not exist an orthogonal LHD of size $n = 4k + 2$ where $k \geq 0$. □

2.4 An algorithm for constructing designs of small runs

In Section 2.2, we have presented a method that allows us to construct LHDs of large runs based on LHDs of small runs. More importantly, the method can build large-run (nearly) orthogonal LHDs based on small-run (nearly) orthogonal LHDs. Hence, to obtain a rich class of large-run (nearly) orthogonal LHDs, it is important to have a catalogue of small-run (nearly) orthogonal LHDs.

To the best of our knowledge, the problem of obtaining a catalogue of small-run (nearly) orthogonal LHDs has not been considered in the literature. It is a challenging problem because of the astronomical number of possible LHDs and the computational complexity in determining the isomorphism of any two LHDs. Two LHDs are called isomorphic if one can be obtained from the other by reordering the runs, relabeling the factors and/or sign-switching one or more factors. To determine if two such designs of n runs with m factors are isomorphic, a complete search compares $2^m n! m!$ designs. Although Clark and Dean (2001) proposed an efficient method based on Hamming distances for checking the isomorphism between any two factorial designs, it is not applicable here since the Hamming distance between any two rows of an LHD is a constant, which is equal to the number m of factors. For a fixed small run size, instead, we aim to find some orthogonal LHDs and the best nearly orthogonal LHD according to some optimality criteria. To do so, we adapt Xu's algorithm (2002), which we will briefly describe first. We then discuss our optimality criteria and present the designs found by the adapted algorithm.

2.4.1 Xu's algorithm

Xu (2002) presented a simple and effective algorithm for constructing orthogonal arrays and nearly orthogonal arrays with mixed levels and small runs. The key idea of his algorithm is to add columns sequentially to an existing design. To add a column, two operations, pairwise switch and exchange, are used. A pairwise switch switches a pair of distinct symbols in a column. For a candidate column, the algorithm searches for all possible pairwise switches and makes the pairwise switch that achieves the best improvement of the optimality criteria. This search and pairwise switch procedure is repeated until a bound is reached or there is no further improvement. An exchange replaces the candidate column by a randomly generated column in which all levels appear equally often. The exchange step is repeated at most T_1 (user-specified) times if no bound of optimality criteria is achieved. The procedure relies on the initial random columns, therefore the entire procedure is repeated T_2 times. Apart from the sequential idea, the efficiency of the algorithm benefits from its fast updates of the optimality criteria. An update is needed when a pairwise switch is applied. The update is fast because the calculation of the value change of the criteria does not involve multiplications.

2.4.2 Optimality criteria

In Section 2.2, we have adopted two measures, $\rho^2(D)$ and $\rho_M(D)$, to evaluate exact and near orthogonality of a design D. Here only $\rho^2(D)$ is used in the algorithm. We will explain the reason of not using $\rho_M(D)$ shortly.

To apply Xu's algorithm, we need to calculate the update of the $\rho^2(D)$ value when the pairwise switch and exchange are carried out. Recall that for a design $D_k = (d_1, \dots, d_k)$, where d_j is the jth column of D_k, $\rho^2(D_k) = \sum_{i<j} \rho^2(d_i, d_j)/[k(k-1)/2]$ where $\rho(d_i, d_j) = J(d_i, d_j)/[J(d_i, d_i)J(d_j, d_j)]^{1/2}$ and $J(d_i, d_j) = d_i^T d_j$. Suppose that the existing design D_{k-1} has $k-1$ columns and d_k is added to D_{k-1}. The new design is denoted by $D_k = (D_{k-1}, d_k)$. We then have

$$\eta_k \rho^2(D_k) = \eta_{k-1}\rho^2(D_{k-1}) + \sum_{i=1}^{k-1} \rho^2(d_i, d_k),$$

where $\eta_k = k(k-1)/2$.

Now suppose that two symbols in rows j and l in the added column d_k are switched. Then for $i = 1, \ldots, k-1$, we get $J_{jl}(d_i, d_k) = J(d_i, d_k) - (d_{ji} - d_{li})(d_{jk} - d_{lk})$ where $J_{jl}(d_i, d_k)$ represents the value of $J(d_i, d_k)$ after the pairwise switch and $d_i = (d_{1i}, \ldots, d_{ni})^T$. Let $d_k^{j,l}$ and $D_k^{j,l}$ denote the added column and the design after the switch of the row pair (j, l), respectively. A straightforward calculation leads to

$$\rho^2(d_i, d_k^{j,l}) = \rho^2(d_i, d_k) - (2J(d_i, d_k)\delta_{jl} - \delta_{jl}^2)/[J(d_i, d_i)J(d_k, d_k)]$$

and

$$\begin{aligned}
\eta_k \rho^2(D_k^{j,l}) &= \eta_{k-1}\rho^2(D_{k-1}) + \sum_{i=1}^{k-1} \rho^2(d_i, d_k^{j,l}) \\
&= \eta_{k-1}\rho^2(D_{k-1}) + \sum_{i=1}^{k-1} \rho^2(d_i, d_k) \\
&\quad - \sum_{i=1}^{k-1} \Delta_{ik}(j,l)/[J(d_i, d_i)J(d_k, d_k)] \\
&= \eta_{k-1}\rho^2(D_k) - \sum_{i=1}^{k-1} \Delta_{ik}(j,l)/[J(d_i, d_i)J(d_k, d_k)], \qquad (2.8)
\end{aligned}$$

where $\Delta_{ik}(j,l) = 2J(d_i, d_k)\delta_{jl} - \delta_{jl}^2$ and $\delta_{jl} = (d_{ji} - d_{li})(d_{jk} - d_{lk})$.

In this section, only LHDs are considered so $J(d_i, d_i) = n(n^2-1)/12$ in our notation. Thus on the right hand side of (2.8), only $\Delta(j,l) = \sum_{i=1}^{k-1} \Delta_{ik}(j,l)$ varies due to the pairwise switch. The equation (2.8) implies that we perform the pairwise switch which produces a positive and biggest value of $\Delta(j,l)$.

An important aspect of an optimality criterion is a lower bound. In our situation, $\rho^2(D) = 0$ is clearly a lower bound for an orthogonal LHD. When there does not exist an orthogonal LHD, i.e. $n = 4k+2$ $(k = 0, 1, \ldots)$, the following corollary provides a useful lower bound of ρ^2 due to Theorem 2.5.

Corollary 2.1. *For an $n \times m$ Latin hypercube L, each column of which is a permutation of $\{-(n-1)/2, \ldots, -1/2, 1/2, \ldots, (n-1)/2\}$, where n is of form $n = 4k+2$*

$(k = 0, 1, \ldots)$, *we have that*

$$\rho^2(L) \geq b(n) = 36/[n^2(n^2-1)^2]. \tag{2.9}$$

Proof. Consider any two column vectors a and b from the LHD L, we have

$$\sum_{i=1}^{n} a_i b_i = 2^{-1} \sum_{i=1}^{n/2} [(2b_i)i - (2b_{i+n/2})(i-1)],$$

given by the equation (2.7) in the proof of Theorem 2.5. Because $|\sum_{i=1}^{n/2}[(2b_i)i - (2b_{i+n/2})(i-1)]| \geq 1$, we have $|\rho(a,b)| = |\sum_i a_i b_i|/(\sum_i a_i^2 \sum_i b_i^2)^{1/2} \geq 2^{-1}/[n(n^2-1)/12] = 6/[n(n^2-1)]$. Therefore, $\rho^2(L) = \sum_{i \leq j} \rho^2(l_i, l_j)/[m(m-1)/2] \geq 36/[n^2(n^2-1)^2]$. □

We have tried to use $\rho_M(D_k)$ as an optimality criterion for several cases in the adapted algorithm. However, the results are not so good as those from using $\rho^2(D)$. An intuitive explanation is as follows. If we choose $\rho_M(D_k)$ as an optimality criterion, obviously, $\rho_M(D_k) = \text{Max}\{\rho_M(D_{k-1}), |\rho(d_1, d_k)|, \ldots, |\rho(d_{k-1}, d_k)|\}$. This means if $|\rho(d_i, d_k)| < \rho_M(D_{k-1})$ for all $i = 1, \ldots, k-1$, $\rho_M(D_k) = \rho_M(D_{k-1})$ and the added column d_k has no contribution to the selection of optimal designs. Therefore, $\rho_M(D_k)$ as an optimality criterion does not give results as fruitful as $\rho^2(D)$.

2.4.3 The adapted algorithm and results

Xu's algorithm is adapted to obtain orthogonal and nearly orthogonal LHDs. Suppose we aim to construct a nearly orthogonal LHD of n runs with m factors. Let $\mathcal{S}$ be $\{-(n-1)/2, \ldots, 0, \ldots, (n-1)/2\}$ and $\{-(n-1)/2, \ldots, (n-1)/2\}$ when n is odd and even, respectively. The lower bound of ρ^2 is defined as

$$\ell(n) = \begin{cases} b(n) = 36/[n^2(n^2-1)^2], & n = 4k+2 \ (k = 1, 2, \ldots); \\ 0, & \text{otherwise.} \end{cases}$$

The adapted algorithm works as follows.

Step 1: Randomly select a permutation of $\mathcal{S}$. Set $m_0 = 0$.

Step 2: For $k = 2, \ldots, m$, do the following:

- **(a)** Generate a random permutation of $\mathcal{S}$ for the kth candidate column, d_k. Let D_{k-1} be the present design and D_k denote the design obtained by adding d_k to D_{k-1}. Compute $\rho^2(D_k) = [(k-1)(k-2)\rho^2(D_{k-1})/2 + \sum_{i=1}^{k-1} \rho^2(d_i, d_k)]/[k(k-1)/2]$. If $\rho^2(D_k) = 0$, set $m_0 = m_0 + 1$. If $\rho^2(D_k) = \ell(n)$, go to (d).
- **(b)** For every pair of rows, j and l, compute $\Delta(j, l)$ as defined in Section 2.4.2. Choose the pair with the positive and largest $\Delta(j, l)$ and switch the symbols in rows j and l of d_k. Reduce $\rho^2(D_k)$ by $\Delta(j, l)/[\eta_k n(n^2 - 1)/12]$. If $\rho^2(D_k) = 0$, then set $m_0 = m_0 + 1$. If $\rho^2(D_k) = \ell(n)$, go to (d); otherwise, repeat (b) until no further improvement is possible.
- **(c)** Repeat (a) and (b) T_1 times and choose a column d_k that produces the smallest value of $\rho^2(D_k)$.
- **(d)** Add the column d_k to D_{k-1} and update the value of $\rho^2(D_k)$.

Step 3: Repeat Steps 1 and 2 T_2 times. Keep the design D_m with the smallest value of $\rho^2(D_m)$ and the orthogonal LHD D_{m_0} with the largest m_0.

We now tabulate the orthogonal LHDs obtained by the adapted algorithm with $T_1 = 3000$ and $T_2 = 3000$ in Tables 2.1 and 2.2. At step 3 of the algorithm, T_2 designs are ranked according to ρ_M and the one with the smallest value of ρ_M is also kept. Tables 2.3 and 2.4 summarize the smallest values of ρ and ρ_M obtained by the algorithm for various n and m. Note that the design that has the smallest value of ρ may not have the smallest value of ρ_M and vice versa. When the run size n is not of form $4k + 2$, $\rho = \rho_M = 0$ corresponds to an orthogonal LHD. In the case of $n = 4k + 2$, the lower bound in Corollary 2.1 is attained for some values of m.

For instance, in Table 2.4, LHDs of 22 runs with m factors achieve the lower bound $\rho = \rho_M = 6/[n(n^2 - 1)] = 0.00056$ for $m \leq 7$.

Table 2.1: Orthogonal Latin hypercubes of n runs, $4 \leq n \leq 13$

n										
4		5		7			8			
−1.5	−0.5	−2	1	−3	−3	0	−3.5	1.5	−0.5	−2.5
−0.5	1.5	−1	0	−2	2	−3	−2.5	−2.5	0.5	3.5
0.5	−1.5	0	−1	−1	1	3	−1.5	3.5	−1.5	2.5
1.5	0.5	1	−2	0	3	1	−0.5	−0.5	1.5	−3.5
		2	2	1	−2	−1	0.5	−3.5	−2.5	−1.5
				2	−1	2	1.5	−1.5	3.5	1.5
				3	0	−2	2.5	2.5	2.5	−0.5
							3.5	0.5	−3.5	0.5

n											
9					11						
−4	−1	−4	−2	−3	−5	−4	−5	−5	−3	0	0
−3	−3	−1	3	3	−4	2	−1	3	4	5	4
−2	2	3	−3	1	−3	−2	4	5	−4	−2	−1
−1	4	2	0	−1	−2	3	−3	4	1	−4	−2
0	−2	4	4	−2	−1	4	2	−4	3	2	−4
1	1	0	−1	0	0	−5	5	−2	5	−3	2
2	3	−3	2	4	1	5	3	−3	−5	−1	5
3	−4	1	−4	2	2	−1	1	1	−2	3	−5
4	0	−2	1	−4	3	0	0	−1	0	1	−3
					4	1	−4	0	2	−5	1
					5	−3	−2	2	−1	4	3

n											
12						13					
−5.5	−5.5	−1.5	−5.5	−3.5	−3.5	−6	−6	−6	0	−5	−1
−4.5	−2.5	−2.5	5.5	4.5	0.5	−5	1	4	−1	6	5
−3.5	4.5	5.5	−4.5	−0.5	1.5	−4	6	−4	5	5	−2
−2.5	0.5	0.5	0.5	0.5	5.5	−3	2	6	−4	−6	2
−1.5	2.5	−0.5	1.5	5.5	−4.5	−2	−2	2	2	−2	−4
−0.5	5.5	2.5	3.5	−2.5	−1.5	−1	3	1	1	−3	3
0.5	1.5	−5.5	2.5	−5.5	−2.5	0	4	−2	−6	1	−5
1.5	−1.5	1.5	−1.5	1.5	2.5	1	−4	−5	−2	3	4
2.5	−4.5	3.5	4.5	−4.5	3.5	2	−5	5	6	2	1
3.5	−0.5	−4.5	−3.5	3.5	4.5	3	−3	3	−5	4	−6
4.5	3.5	−3.5	−2.5	−1.5	−0.5	4	−1	0	4	−1	−3
5.5	−3.5	4.5	−0.5	2.5	−5.5	5	5	−1	3	−4	0
						6	0	−3	−3	0	6

Table 2.2: Orthogonal Latin hypercubes of n runs, $15 \leq n \leq 21$

n																	
15						17						19					
−7	4	−7	−6	−2	−5	−8	−8	−8	3	1	−7	−9	−9	−9	−8	2	7
−6	3	5	3	5	2	−7	0	5	−1	−6	8	−8	5	5	4	−8	9
−5	−6	−2	5	6	−3	−6	−2	3	−2	−2	−1	−7	−6	−5	5	0	−5
−4	1	4	4	−5	−2	−5	7	−6	2	3	5	−6	−3	0	−9	−4	−2
−3	0	−4	−7	0	3	−4	3	−5	−7	0	3	−5	9	6	1	7	4
−2	−2	3	0	−4	−7	−3	2	8	0	7	6	−4	7	9	−5	−2	−6
−1	−7	6	−2	−3	5	−2	−3	−1	−4	6	−4	−3	−2	3	3	−9	−7
0	7	−5	7	3	7	−1	5	4	6	4	−8	−2	6	−8	6	6	−3
1	−4	1	−5	4	6	0	4	1	7	−8	−5	−1	−4	4	9	5	2
2	−1	0	6	−6	1	1	−5	−3	5	2	2	0	4	−6	−2	9	−8
3	6	2	−3	−7	4	2	−6	7	−6	−1	0	1	0	−2	−4	−5	−1
4	−3	−3	−1	2	−1	3	8	2	1	−3	−2	2	8	−3	−1	3	1
5	5	7	−4	7	−4	4	−4	6	−3	−4	−6	3	−8	8	0	4	3
6	−5	−6	1	−1	0	5	1	−4	4	−7	4	4	−7	2	8	−1	−9
7	2	−1	2	1	−6	6	−1	−7	−8	−5	1	5	−5	7	−3	8	5
						7	6	−2	−5	8	−3	6	1	−4	7	−7	6
						8	−7	0	8	5	7	7	2	−7	2	−3	8
												8	3	−1	−6	−6	−4
												9	−1	1	−7	1	0

n											
20						21					
−9.5	6.5	−9.5	−8.5	−9.5	3.5	−10	−2	−9	4	−9	−9
−8.5	−6.5	4.5	8.5	−5.5	−1.5	−9	−5	8	9	10	−7
−7.5	−4.5	5.5	2.5	5.5	4.5	−8	0	−3	−8	−4	−6
−6.5	4.5	−1.5	−4.5	7.5	1.5	−7	10	4	7	6	6
−5.5	−7.5	0.5	−7.5	−6.5	−7.5	−6	9	3	−5	4	−5
−4.5	9.5	−0.5	5.5	6.5	−4.5	−5	5	−1	−1	−7	3
−3.5	7.5	−4.5	4.5	−3.5	2.5	−4	1	−2	−9	−5	9
−2.5	−1.5	3.5	−1.5	0.5	8.5	−3	−4	7	2	−1	10
−1.5	−9.5	−3.5	3.5	3.5	7.5	−2	6	−5	−4	8	8
−0.5	0.5	7.5	−5.5	−1.5	−6.5	−1	−7	9	6	−10	−1
0.5	−0.5	8.5	−0.5	−0.5	−2.5	0	−3	0	−6	3	0
1.5	−8.5	−5.5	6.5	4.5	−8.5	1	−1	−6	10	−3	5
2.5	2.5	−2.5	−6.5	1.5	−3.5	2	−8	−4	−2	2	7
3.5	5.5	1.5	1.5	9.5	−0.5	3	−10	5	−3	5	−8
4.5	1.5	−7.5	0.5	8.5	0.5	4	−6	2	−10	7	1
5.5	8.5	9.5	7.5	−7.5	−5.5	5	−9	−7	0	−2	2
6.5	−2.5	−6.5	9.5	−8.5	6.5	6	2	−10	5	9	−4
7.5	−5.5	6.5	−9.5	2.5	5.5	7	8	1	−7	−6	−10
8.5	−3.5	−8.5	−3.5	−2.5	−9.5	8	4	−8	8	1	−3
9.5	3.5	2.5	−2.5	−4.5	9.5	9	3	6	3	−8	4
						10	7	10	1	0	−2

Table 2.3: The best values of ρ and ρ_M (in bracket) for $4 \leq n \leq 15$

m	n					
	4	5	6	7	8	9
2	0.000(0.000)	0.000(0.000)	0.029(0.029)	0.000(0.000)	0.000(0.000)	0.000(0.000)
3	0.258(0.400)	0.082(0.100)	0.055(0.086)	0.000(0.000)	0.000(0.000)	0.000(0.000)
4		0.135(0.200)	0.072(0.086)	0.015(0.036)	0.000(0.000)	0.000(0.000)
5			0.093(0.143)	0.037(0.071)	0.011(0.024)	0.000(0.000)
6				0.053(0.107)	0.015(0.024)	0.009(0.017)
7					0.036(0.071)	0.016(0.033)
8						0.028(0.067)

m	n					
	10	11	12	13	14	15
2	0.006(0.006)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.002(0.002)	0.000(0.000)
3	0.006(0.006)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.002(0.002)	0.000(0.000)
4	0.006(0.006)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.002(0.002)	0.000(0.000)
5	0.006(0.006)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.002(0.002)	0.000(0.000)
6	0.006(0.006)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.002(0.002)	0.000(0.000)
7	0.012(0.030)	0.000(0.000)	0.003(0.007)	0.002(0.005)	0.002(0.002)	0.001(0.004)
8	0.019(0.042)	0.010(0.028)	0.006(0.014)	0.004(0.011)	0.003(0.007)	0.002(0.004)
9	0.030(0.079)	0.017(0.036)	0.011(0.028)	0.006(0.016)	0.005(0.011)	0.003(0.007)
10		0.025(0.064)	0.016(0.035)	0.009(0.022)	0.006(0.015)	0.005(0.011)
11			0.021(0.056)	0.015(0.038)	0.010(0.024)	0.007(0.018)
12				0.021(0.049)	0.014(0.033)	0.010(0.029)
13					0.020(0.059)	0.012(0.039)
14						0.017(0.050)

Table 2.4: The best values of ρ and ρ_M (in bracket) for $16 \leq n \leq 22$

m	n						
	16	17	18	19	20	21	22
2	0.000(0.000)	0.000(0.000)	0.001(0.001)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.001(0.001)
3	0.000(0.000)	0.000(0.000)	0.001(0.001)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.001(0.001)
4	0.000(0.000)	0.000(0.000)	0.001(0.001)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.001(0.001)
5	0.000(0.000)	0.000(0.000)	0.001(0.001)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.001(0.001)
6	0.000(0.000)	0.000(0.000)	0.001(0.001)	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.001(0.001)
7	0.000(0.000)	0.001(0.002)	0.001(0.001)	0.001(0.002)	0.001(0.002)	0.001(0.001)	0.001(0.001)
8	0.000(0.000)	0.001(0.002)	0.001(0.003)	0.001(0.002)	0.001(0.002)	0.001(0.002)	0.001(0.002)
9	0.000(0.000)	0.002(0.005)	0.002(0.003)	0.001(0.004)	0.001(0.003)	0.001(0.003)	0.001(0.003)
10	0.000(0.000)	0.003(0.007)	0.002(0.005)	0.002(0.005)	0.002(0.005)	0.002(0.004)	0.002(0.004)
11	0.000(0.000)	0.004(0.010)	0.003(0.007)	0.003(0.007)	0.002(0.006)	0.002(0.005)	0.002(0.004)
12	0.000(0.000)	0.005(0.015)	0.004(0.009)	0.003(0.009)	0.003(0.006)	0.002(0.006)	0.002(0.005)
13	0.009(0.026)	0.006(0.020)	0.005(0.013)	0.004(0.009)	0.004(0.009)	0.003(0.008)	0.003(0.006)
14	0.011(0.035)	0.008(0.025)	0.007(0.020)	0.005(0.014)	0.004(0.011)	0.003(0.009)	0.003(0.007)
15	0.016(0.047)	0.011(0.034)	0.008(0.024)	0.007(0.018)	0.005(0.014)	0.004(0.010)	0.004(0.010)
16		0.016(0.056)	0.011(0.034)	0.008(0.025)	0.006(0.018)	0.005(0.014)	0.004(0.012)
17			0.013(0.049)	0.010(0.033)	0.008(0.024)	0.006(0.017)	0.005(0.014)
18				0.015(0.058)	0.010(0.036)	0.008(0.026)	0.006(0.019)
19					0.015(0.054)	0.009(0.032)	0.007(0.022)
20						0.013(0.055)	0.009(0.032)
21							0.012(0.049)

2.5 Constructing orthogonal Latin hypercubes of other run sizes

In Section 2.3, we have shown that the run size of an orthogonal LHD must be a multiple of 4 or odd. In Section 2.2, the proposed method enables us to construct orthogonal LHDs of any size n that n is a multiple of 8. In this section, we develop methods for all other run sizes that are odd or multiples of 4. Thus, the problem of constructing orthogonal LHDs is solved completely in terms of run sizes.

Let $\mathcal{S}$ be the n levels of an LHD of size n. Let $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ where $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, and n_1 and n_2 be the numbers of elements in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Suppose that there exist an $n_1 \times m$ orthogonal design D_1 with levels in $\mathcal{S}_1$ and an $n_2 \times m$ orthogonal design D_2 with levels in $\mathcal{S}_2$. Then

$$L = \begin{pmatrix} D_1 \\ D_2 \end{pmatrix} \tag{2.10}$$

is an $n \times m$ orthogonal LHD, where $n = n_1 + n_2$. Note that D_1 and D_2 are not LHDs in general.

In the following two subsections, we discuss two methods for obtaining the designs D_1 and D_2 in (2.10). In the orthogonal LHDs constructed by the first method, the run size n must be $8k+1$ or $8k-1$ ($k = 3, 4, \ldots$). The second method can provide orthogonal LHDs of any run size that does not have form $4k+2$ ($k = 0, 1, \ldots$).

2.5.1 A direct stacking method

This method applies to the situation where n_1 is odd, n_2 is even, $n_1 + n_2 = n$, and $|n_1 - n_2| = 1$. The method works as follows:

(i) Select an $n_1 \times m$ orthogonal LHD to be $\widetilde{D}_1$;

(ii) Select an $n_2 \times m$ orthogonal LHD to be $\widetilde{D}_2$;

(iii) In (2.10), set $D_1 = 2\widetilde{D}_1$ and $D_2 = 2\widetilde{D}_2$.

As an illustration, we consider constructing orthogonal LHDs of 23 runs in the next example.

Example 2.7. Let $n_1 = 11$ and $n_2 = 12$. In Table 2.1, we take the first six columns of the 11×7 orthogonal LHD to be $\widetilde{D}_1$ and the 12×6 orthogonal LHD to be $\widetilde{D}_2$, i.e.,

$$\widetilde{D}_1 = \begin{pmatrix} -5 & -4 & -5 & -5 & -3 & 0 \\ -4 & 2 & -1 & 3 & 4 & 5 \\ -3 & -2 & 4 & 5 & -4 & -2 \\ -2 & 3 & -3 & 4 & 1 & -4 \\ -1 & 4 & 2 & -4 & 3 & 2 \\ 0 & -5 & 5 & -2 & 5 & -3 \\ 1 & 5 & 3 & -3 & -5 & -1 \\ 2 & -1 & 1 & 1 & -2 & 3 \\ 3 & 0 & 0 & -1 & 0 & 1 \\ 4 & 1 & -4 & 0 & 2 & -5 \\ 5 & -3 & -2 & 2 & -1 & 4 \end{pmatrix} \text{ and } \widetilde{D}_2 = \frac{1}{2}\begin{pmatrix} -11 & -11 & -3 & -11 & -7 & -7 \\ -9 & -5 & -5 & 11 & 9 & 1 \\ -7 & 9 & 11 & -9 & -1 & 3 \\ -5 & 1 & 1 & 1 & 1 & 11 \\ -3 & 5 & -1 & 3 & 11 & -9 \\ -1 & 11 & 5 & 7 & -5 & -3 \\ 1 & 3 & -11 & 5 & -11 & -5 \\ 3 & -3 & 3 & -3 & 3 & 5 \\ 5 & -9 & 7 & 9 & -9 & 7 \\ 7 & -1 & -9 & -7 & 7 & 9 \\ 9 & 7 & -7 & -5 & -3 & -1 \\ 11 & -7 & 9 & -1 & 5 & -11 \end{pmatrix}.$$

A 23×6 orthogonal LHD can then be obtained immediately by (2.10) without the effort of computer search. □

2.5.2 Orthogonal designs method

This method proceeds as follows.

(i) Given n, choose n_1 and n_2 such that $n_1 + n_2 = n$ and n_2 is a multiple of 8;

(ii) Choose an $n_1 \times m$ orthogonal LHD to be D_1;

(iii) Construct an $n_2 \times m$ orthogonal design D_2 with levels $\{-(n_1+n_2-1)/2, \ldots, -(n_1+1)/2, (n_1+1)/2, \ldots, (n_1+n_2-1)/2\}$.

Two approaches for obtaining a design D_2 in (iii) are now considered. To describe the first approach, we consider an $n \times m$ orthogonal matrix V that satisfies the following:

(i) V has entries $\pm x_1, \ldots, \pm x_{n/2}$, where $x_1, \ldots, x_{n/2}$ are real variables;

(ii) Both x_i and $-x_i$ must occur exactly once in each column of of V, $i = 1, \ldots, n/2$;

(iii) Every two columns v_i and v_j of V are orthogonal, i.e., $v_i^T v_j = 0$ for $i \neq j$.

Unfortunately, no general construction method for V is available at present. Nevertheless, we have obtained a few orthogonal matrices V for $n = 2, 4, 8, 16$, as listed in Table 2.5. Substituting each x_i $(i = 1, \ldots, n_2/2)$ in an $n_2 \times m$ orthogonal matrix V by $(n_1 + 2i - 1)/2$ will then yield an $n_2 \times m$ orthogonal design D_2, required by the method in (2.10). For example, 27 × 7, 28 × 6, 29 × 6 and 31 × 6 orthogonal LHDs can easily be constructed using this method together with orthogonal LHDs in Table 2.1. We note that orthogonal matrices V considered above are related to but different from orthogonal designs in the combinatorics literature (Geramita and Seberry, 1979).

Table 2.5: Orthogonal matrices of n runs, $n = 2, 4, 8, 16$

								n						
2	4		8				16							
x_1	x_1	x_2	x_1	$-x_2$	x_4	x_3	x_1	$-x_2$	$-x_4$	$-x_3$	$-x_8$	x_7	x_5	x_6
$-x_1$	$-x_1$	$-x_2$	x_2	x_1	x_3	$-x_4$	x_2	x_1	$-x_3$	x_4	$-x_7$	$-x_8$	$-x_6$	x_5
	x_2	$-x_1$	x_3	$-x_4$	$-x_2$	$-x_1$	x_3	$-x_4$	x_2	x_1	$-x_6$	$-x_5$	x_7	$-x_8$
	$-x_2$	x_1	x_4	x_3	$-x_1$	x_2	x_4	x_3	x_1	$-x_2$	$-x_5$	x_6	$-x_8$	$-x_7$
			$-x_4$	$-x_3$	x_1	$-x_2$	x_5	$-x_6$	$-x_8$	x_7	x_4	x_3	$-x_1$	$-x_2$
			$-x_3$	x_4	x_3	x_1	x_6	x_5	$-x_7$	$-x_8$	x_3	$-x_4$	x_2	$-x_1$
			$-x_2$	$-x_1$	$-x_3$	x_4	x_7	$-x_8$	x_6	$-x_5$	x_2	$-x_1$	$-x_3$	x_1
			$-x_1$	x_2	$-x_4$	$-x_3$	x_8	x_7	x_5	x_6	x_1	x_2	x_4	x_3
							$-x_1$	x_2	x_4	x_3	x_8	$-x_7$	$-x_5$	$-x_6$
							$-x_2$	$-x_1$	x_3	$-x_4$	x_7	x_8	x_6	$-x_5$
							$-x_3$	x_4	$-x_2$	$-x_1$	x_6	x_5	$-x_7$	x_8
							$-x_4$	$-x_3$	$-x_1$	x_2	x_5	$-x_6$	x_8	x_7
							$-x_5$	x_6	x_8	$-x_7$	$-x_4$	$-x_3$	x_1	x_2
							$-x_6$	$-x_5$	x_7	x_8	$-x_3$	x_4	$-x_2$	x_1
							$-x_7$	x_8	$-x_6$	x_5	$-x_2$	x_1	x_3	$-x_4$
							$-x_8$	$-x_7$	$-x_5$	$-x_6$	$-x_1$	$-x_2$	$-x_4$	$-x_3$

We now turn to the second approach thanks to Proposition 2.3 below.

Proposition 2.3. *Let A be an $n_{21} \times m_1$ column-orthogonal matrix, B be an $n_{22} \times m_2$ orthogonal Latin hypercube, $C = (c_{ij})$ be an $n_{21} \times m_1$ orthogonal design with levels*

$\{-[n_1 + kn_{22}]/2, [n_1 + kn_{22}]/2, k = 1, 3, \ldots, n_{21} - 1\}$, D be an $n_{22} \times m_2$ column-orthogonal matrix. Let $n_2 = n_{21}n_{22}$ and $m = m_1m_2$. Suppose that A, B, C and D satisfy conditions (iii) and (iv) in Theorem 2.1. Then choosing $\gamma = 1$ in (2.1) gives an orthogonal design $D_2 = L$ with levels $\{-(n_1 + n_2 - 1)/2, \ldots, -(n_1 + 1)/2, (n_1 + 1)/2, \ldots, (n_1 + n_2 - 1)/2\}$.

Proof. It is straightforward to see the levels of D_2 are $\{-(n_1 + n_2 - 1)/2, \ldots, -(n_1 + 1)/2, (n_1+1)/2, \ldots, (n_1+n_2-1)/2\}$ from the levels in A, B, C and D and the definition of Kronecker product. The orthogonality of D_2 follows directly from Proposition 2.2. □

Proposition 2.3 is particularly useful when $n_1 = 1$. That is because D_1 in (2.10) is a row of zeros in this case and there is no restriction on the number of columns in D_1. To use Proposition 2.3, we choose A, B and D in the same way as in Theorem 2.1. As for C, we can make use of the orthogonal matrices in Table 2.5. For given n_2, we can use different combinations of n_{21} and n_{22}, which yield many choices for D_2.

The next two examples illustrate the use of Proposition 2.3.

Example 2.8. Let $n_{21} = 2$, $n_{22} = n_2/2$, $m_1 = 1$, and $m_2 = m$, and let A, B and D be chosen as in Proposition 2.3. Taking $((n_1 + n_{22})/2, -(n_1 + n_{22})/2)^T$ to be C in the method (2.1), we obtain an orthogonal design D_2 for (2.10). For example, letting $n_1 = 1$ and $n_2 = 24$, a 25 × 6 orthogonal LHD can be obtained immediately.

Example 2.9. Let $n_{21} = 4$, $n_{22} = n_2/4$, $m_1 = 2$, and $m_2 = m/2$. Again, A, B and D are chosen as in Proposition 2.3. Let C be

$$C = \begin{pmatrix} \gamma_1 & \gamma_2 \\ -\gamma_1 & -\gamma_2 \\ \gamma_2 & -\gamma_1 \\ -\gamma_2 & \gamma_1 \end{pmatrix},$$

where $\gamma_1 = (n_1 + n_{22})/2$ and $\gamma_2 = (n_1 + 3n_{22})/2$. For instance, suppose we wish to construct an orthogonal LHD of 65 runs. To do so, we let n_1 be 1 and n_2 be 64. In addition, we choose a 16×12 orthogonal LHD to be B. After choosing A and D appropriately, we can obtain a 65×24 orthogonal LHD.

2.6 Collections of orthogonal Latin hypercubes

Our intention in this section is to provide a comprehensive table of orthogonal LHDs for available run sizes, n. Note that n must not equal 3 and $4k+2$ $(k = 0, 1, \ldots)$.

In the previous sections, six approaches have been introduced to construct orthogonal LHDs of various run sizes. They are (i) the algorithm search in Section 2.4.3; (ii) the orthogonal matrices method in Section 2.5.2; (iii) the direct stacking method in Section 2.5.1; (iv) the construction (2.1) as in Theorem 2.1; (v) the construction as in Theorem 2.2; (vi) the construction (2.10) as in Proposition 2.3. We now summarize orthogonal LHDs provided by each approach. Approach (i) is used to obtain orthogonal LHDs of small run sizes. The corresponding number m of columns in each of these designs is given in Table 2.6. Approach (ii) uses orthogonal matrices to obtain orthogonal designs D_2 in the construction (2.10). Although general construction on orthogonal matrices is not available currently, orthogonal matrices of sizes 8 and 16 are available in Table 2.5. Because n_1 can be any available run size, this approach can provide orthogonal LHDs of any available run size. As an illustration, Table 2.7 provides the number m of columns in those designs of run sizes $n \leq 40$. Note that $n = n_1 + n_2$ and $m = \min\{m_1, m_2\}$. For the larger run sizes, the value of m's can be readily obtained using m's for small run sizes. Approach (iii) applies to the situation where n_1 and n_2 are adjacent. Equivalently, it can offer orthogonal LHDs of run sizes $n = 8k - 1$ and $n = 8k + 1$ where k is any positive integer. For example, Table 2.8 displays the cases $k = 3, 4, 5$. Again, we have $n = n_1 + n_2$ and $m = \min\{m_1, m_2\}$ in this approach. Approach (iv) is applicable to the cases $n = 8k$ $(k = 1, 2, \ldots)$. Table 2.9 gives the value of m, n_1, n_2, m_1 and m_2 in Theorem 2.1 for each of the run size $n \leq 256$. Note that there may exist multiple choices of n_1 and n_2 that gives the same m, but we only report one such choice. Approach (v) requires the run size n to have form $n = 16k^2$. The number m of columns in orthogonal LHDs obtained by this approach is given in Table 2.10. Note that $m = 2m_1m_2$. In approach (vi), n_2 must be a multiple of 8 while n_1 can be any integer that is not equal to $4k + 2$ and 3, where k is any integer. Therefore, this approach can provide orthogonal LHDs of any available run size. It is particularly useful when $n_1 = 1$ and n_2 is a multiple of 16 as indicated in Table 2.11.

Table 2.6: The maximum number m of factors in orthogonal LHDs obtained by approach (i)

n	4	5	7	8	9	11	12	13	15	16	17	19	20
m	2	2	3	4	5	7	6	6	6	6	6	6	6

Table 2.7: The maximum number m of factors in orthogonal LHDs obtained by approach (ii)

n	m	n_1	n_2	m_1	m_2
17	8	1	16	8	8
19	4	11	8	7	4
20	4	12	8	6	4
21	4	13	8	6	4
23	4	15	8	6	4
24	4	16	8	12	4
25	5	9	16	5	8
27	7	11	16	7	8
28	6	12	16	6	8
29	6	13	16	6	8
31	6	15	16	6	8
32	8	16	16	12	8
33	8	17	16	8	8
35	6	19	16	6	8
36	6	20	16	6	8
37	6	21	16	6	8
39	6	23	16	6	8
40	6	24	16	6	8

Given the results above, we now summarize the maximum value of m provided by the aforementioned six approaches and the methods of Ye (1998), Steinberg and Lin (2006) and Cioppa and Lucas (2007) in Tables 2.12, 2.13, 2.14, and 2.15. Therefore, the maximum number m^* of columns obtained by combining all the results is also given in Tables 2.12, 2.13, 2.14, and 2.15. Due to the space consideration, we only list the cases $n \leq 259$.

Several comments are in order. First, Tables 2.12 - 2.15 demonstrate that our approaches outperform others in terms of both flexibility of the run size and the number of orthogonal columns. Second, since large orthogonal LHDs are built based on small ones, we will obtain more columns for large ones if more columns for small ones can be found through algorithms or constructions in the future. Third, suppose we have an $n_1 \times m_1$ orthogonal LHD and an $n_2 \times m_2$ orthogonal LHD. One may expect an $(n_1 n_2) \times (m_1 m_2)$ orthogonal LHD given by the method (2.1). However, this is not always the case. For example, we have a 12 × 6 and 20 × 6 orthogonal LHD that can be used as B and C respectively for constructing orthogonal LHDs of 240 runs. But a 240 × 36 orthogonal LHD cannot be constructed because condition (iv) in Theorem 2.1 is not satisfied. Instead, only 12 columns are available as reported in Table 2.15. Lastly, many small orthogonal LHDs are available and thus a large collection of large ones are obtained via both methods (2.1) and (2.10). In addition, as argued in Section 2.2.1, given A, B, C and D that produce an orthogonal LHD via the method (2.1), we can apply different row permutations (or column permutation or sign-switching or a combination of these operations) between A and C, or between B and D to obtain non-isomorphic orthogonal LHDs.

Table 2.8: The maximum number m of factors in orthogonal LHDs obtained by approach (iii)

n	m	n_1	n_2	m_1	m_2	n	m	n_1	n_2	m_1	m_2
23	6	11	12	7	6	33	8	17	16	8	12
25	6	13	12	6	6	39	6	19	20	6	6
31	6	15	16	6	12	41	6	21	20	6	6

Table 2.9: The maximum number m of factors in orthogonal LHDs obtained by approach (iv)

n	m	n_1	n_2	m_1	m_2
24	6	12	2	6	1
32	12	16	2	12	1
40	6	20	2	6	1
48	12	12	4	6	2
56	6	28	2	6	1
64	24	16	4	12	2
72	6	36	2	6	1
80	12	20	4	6	2
88	6	44	2	6	1
96	24	12	8	6	4
104	6	52	2	6	1
112	12	28	4	6	2
120	6	60	2	6	1
128	48	16	8	12	4
136	6	68	2	6	1
144	12	12	12	6	2
152	6	76	2	6	1
160	24	20	8	6	4
168	6	84	2	6	1
176	12	44	4	6	2
184	6	92	2	6	1
192	48	16	12	8	6
200	6	100	2	6	1
208	12	52	4	6	2
216	6	108	2	6	1
224	24	28	8	6	4
232	6	116	2	6	1
240	12	60	4	6	2
248	6	124	2	6	1
256	96	16	16	12	8

Table 2.10: The maximum number m of factors in orthogonal LHDs obtained by approach (v)

n	m	n_0	m_1	m_2
64	32	8	4	4
144	24	12	6	2
256	192	16	12	8

Table 2.11: The maximum number m of factors in orthogonal LHDs obtained by approach (vi)

n	m	n_1	n_2	n_{21}	n_{22}	m_1	m_2
33	12	1	32	16	2	12	1
41	6	1	40	20	2	6	1
49	12	1	48	12	4	6	2
57	6	1	56	28	2	6	1
65	24	1	64	16	4	12	2
73	6	1	72	36	2	6	1
81	12	1	80	20	4	6	2
89	6	1	88	44	2	6	1
97	24	1	96	12	8	6	4
105	6	1	104	52	2	6	1
113	12	1	112	28	4	6	2
121	6	1	120	60	2	6	1
129	48	1	128	16	8	12	4
137	6	1	136	68	2	6	1
145	12	1	144	12	12	6	2
153	6	1	152	76	2	6	1
161	24	1	160	20	8	6	4
169	6	1	168	84	2	6	1
177	12	1	176	44	4	6	2
185	6	1	184	92	2	6	1
193	48	1	192	12	16	6	8
201	6	1	200	100	2	6	1
209	12	1	208	52	4	6	2
217	6	1	216	108	2	6	1
225	24	1	224	28	8	6	4
233	6	1	232	116	2	6	1
241	12	1	240	60	4	6	2
249	6	1	248	124	2	6	1
257	96	1	256	16	16	12	8

Table 2.12: The maximum number m^* of factors in available orthogonal LHDs of run sizes $n \leq 67$

n	m^*	Ref.$^{(1)}$	New$^{(2)}$	Ye$^{(3)}$	S.L.$^{(4)}$	C.L.$^{(5)}$	n	m^*	Ref.$^{(1)}$	New$^{(2)}$	Ye$^{(3)}$	S.L.$^{(4)}$	C.L.$^{(5)}$
4	2	(i)	2	2	0	2	36	6	(ii)	6	0	0	0
5	2	(i)	2	0	0	0	37	6	(ii)	6	0	0	0
7	3	(i)	3	0	0	0	39	6	(ii),(iii)	6	0	0	0
8	4	(i)	4	2	0	2	40	6	(ii),(iv)	6	0	0	0
9	5	(i)	5	0	0	0	41	6	(ii),(iii)	6	0	0	0
11	7	(i)	7	0	0	0	43	7	(ii)	7	0	0	0
12	6	(i)	6	0	0	0	44	6	(ii)	6	0	0	0
13	6	(i)	6	0	0	0	45	6	(ii)	6	0	0	0
15	6	(i)	6	0	0	0	47	6	(ii),(iii)	6	0	0	0
16	12	S.L.	8	6	12	7	48	12	(iv)	12	0	0	0
17	8	(ii)	8	6	0	7	49	12	(vi)	12	0	0	0
19	6	(i)	6	0	0	0	51	6	(ii)	6	0	0	0
20	6	(i)	6	0	0	0	52	6	(ii)	6	0	0	0
21	6	(i)	6	0	0	0	53	6	(ii)	6	0	0	0
23	6	(iii)	6	0	0	0	55	6	(ii),(iii)	6	0	0	0
24	6	(iv)	6	0	0	0	56	6	(ii),(iv)	6	0	0	0
25	6	(iii)	6	0	0	0	57	6	(ii),(iii)	6	0	0	0
27	7	(ii)	7	0	0	0	59	7	(ii)	7	0	0	0
28	6	(ii)	6	0	0	0	60	6	(ii)	6	0	0	0
29	6	(ii)	6	0	0	0	61	6	(ii)	6	0	0	0
31	6	(ii),(iii)	6	0	0	0	63	6	(ii),(iii)	6	0	0	0
32	12	(iv)	12	8	0	11	64	32	(v)	32	10	0	16
33	12	(vi)	12	8	0	11	65	24	(vi)	24	10	0	16
35	6	(ii)	6	0	0	0	67	6	(ii)	6	0	0	0

Note: (1): The approach that gives m^*; (2): approaches (i) - (vi); (3): Ye (1998); (4): Steinberg and Lin (2006); (5) Cioppa and Lucas (2007).

Table 2.13: The maximum number m^* of factors in available orthogonal LHDs of run sizes $68 \leq n \leq 131$

n	m^*	Ref.[(1)]	New[(2)]	Ye[(3)]	S.L.[(4)]	C.L.[(5)]	n	m^*	Ref.[(1)]	New[(2)]	Ye[(3)]	S.L.[(4)]	C.L.[(5)]
68	6	(ii)	6	0	0	0	100	6	(ii)	6	0	0	0
69	6	(ii)	6	0	0	0	101	6	(ii)	6	0	0	0
71	6	(ii),(iii)	6	0	0	0	103	6	(ii),(iii)	6	0	0	0
72	6	(ii),(iv)	6	0	0	0	104	6	(ii),(iv)	6	0	0	0
73	6	(ii),(iii)	6	0	0	0	105	6	(ii),(iii)	6	0	0	0
75	7	(ii)	7	0	0	0	107	7	(ii)	7	0	0	0
76	6	(ii)	6	0	0	0	108	6	(ii)	6	0	0	0
77	6	(ii)	6	0	0	0	109	6	(ii)	6	0	0	0
79	6	(ii),(iii)	6	0	0	0	111	6	(ii),(iii)	6	0	0	0
80	12	(iv)	12	0	0	0	112	12	(iv)	12	0	0	0
81	12	(vi)	6	0	0	0	113	12	(vi)	12	0	0	0
83	6	(ii)	6	0	0	0	115	6	(ii)	6	0	0	0
84	6	(ii)	6	0	0	0	116	6	(ii)	6	0	0	0
85	6	(ii)	6	0	0	0	117	6	(ii)	6	0	0	0
87	6	(ii),(iii)	6	0	0	0	119	6	(ii),(iii)	6	0	0	0
88	6	(ii),(iv)	6	0	0	0	120	6	(ii),(iv)	6	0	0	0
89	6	(ii),(iii)	6	0	0	0	121	6	(ii),(iii)	6	0	0	0
91	7	(ii)	7	0	0	0	123	7	(ii)	7	0	0	0
92	6	(ii)	6	0	0	0	124	6	(ii)	6	0	0	0
93	6	(ii)	6	0	0	0	125	6	(ii)	6	0	0	0
95	6	(ii),(iii)	6	0	0	0	127	6	(ii),(iii)	6	0	0	0
96	24	(iv)	24	0	0	0	128	48	(iv)	48	12	0	22
97	24	(vi)	24	0	0	0	129	48	(vi)	48	12	0	22
99	6	(ii)	6	0	0	0	131	6	(ii)	6	0	0	0

Note: (1): The approach that gives m^*; (2): approaches (i) - (vi); (3): Ye (1998); (4): Steinberg and Lin (2006); (5) Cioppa and Lucas (2007).

Table 2.14: The maximum number m^* of factors in available orthogonal LHDs of run sizes $132 \leq n \leq 195$

n	m^*	Ref.$^{(1)}$	New$^{(2)}$	Ye$^{(3)}$	S.L.$^{(4)}$	C.L.$^{(5)}$	n	m^*	Ref.$^{(1)}$	New$^{(2)}$	Ye$^{(3)}$	S.L.$^{(4)}$	C.L.$^{(5)}$
132	6	(ii)	6	0	0	0	164	6	(ii)	6	0	0	0
133	6	(ii)	6	0	0	0	165	6	(ii)	6	0	0	0
135	6	(ii),(iii)	6	0	0	0	167	6	(ii),(iii)	6	0	0	0
136	6	(ii),(iv)	6	0	0	0	168	6	(ii),(iv)	6	0	0	0
137	6	(ii),(iii)	6	0	0	0	169	6	(ii),(iii)	6	0	0	0
139	7	(ii)	7	0	0	0	171	7	(ii)	7	0	0	0
140	6	(ii)	6	0	0	0	172	6	(ii)	6	0	0	0
141	6	(ii)	6	0	0	0	173	6	(ii)	6	0	0	0
143	6	(ii),(iii)	6	0	0	0	175	6	(ii),(iii)	6	0	0	0
144	24	(v)	24	0	0	0	176	12	(iv)	12	0	0	0
145	12	(vi)	6	0	0	0	177	12	(vi)	12	0	0	0
147	6	(ii)	6	0	0	0	179	6	(ii)	6	0	0	0
148	6	(ii)	6	0	0	0	180	6	(ii)	6	0	0	0
149	6	(ii)	6	0	0	0	181	6	(ii)	6	0	0	0
151	6	(ii),(iii)	6	0	0	0	183	6	(ii),(iii)	6	0	0	0
152	6	(ii),(iv)	6	0	0	0	184	6	(ii),(iv)	6	0	0	0
153	6	(ii),(iii)	6	0	0	0	185	6	(ii),(iii)	6	0	0	0
155	7	(ii)	7	0	0	0	187	7	(ii)	7	0	0	0
156	6	(ii)	6	0	0	0	188	6	(ii)	6	0	0	0
157	6	(ii)	6	0	0	0	189	6	(ii)	6	0	0	0
159	6	(ii),(iii)	6	0	0	0	191	6	(ii),(iii)	6	0	0	0
160	24	(iv)	24	0	0	0	192	48	(iv)	48	0	0	0
161	24	(vi)	24	0	0	0	193	48	(vi)	48	0	0	0
163	6	(ii)	6	0	0	0	195	6	(ii)	6	0	0	0

Note: (1): The approach that gives m^*; (2): approaches (i) - (vi); (3): Ye (1998); (4): Steinberg and Lin (2006); (5) Cioppa and Lucas (2007).

Table 2.15: The maximum number m^* of factors in available orthogonal LHDs of run sizes $196 \leq n \leq 259$

n	m^*	Ref.$^{(1)}$	New$^{(2)}$	Ye$^{(3)}$	S.L.$^{(4)}$	C.L.$^{(5)}$
196	6	(ii)	6	0	0	0
197	6	(ii)	6	0	0	0
199	6	(ii),(iii)	6	0	0	0
200	6	(ii),(iv)	6	0	0	0
201	6	(ii),(iii)	6	0	0	0
203	7	(ii)	7	0	0	0
204	6	(ii)	6	0	0	0
205	6	(ii)	6	0	0	0
207	6	(ii),(iii)	6	0	0	0
208	12	(iv)	12	0	0	0
209	12	(vi)	6	0	0	0
211	6	(ii)	6	0	0	0
212	6	(ii)	6	0	0	0
213	6	(ii)	6	0	0	0
215	6	(ii),(iii)	6	0	0	0
216	6	(ii),(iv)	6	0	0	0
217	6	(ii),(iii)	6	0	0	0
219	7	(ii)	7	0	0	0
220	6	(ii)	6	0	0	0
221	6	(ii)	6	0	0	0
223	6	(ii),(iii)	6	0	0	0
224	24	(iv)	24	0	0	0
225	24	(vi)	24	0	0	0
227	6	(ii)	6	0	0	0
228	6	(ii)	6	0	0	0
229	6	(ii)	6	0	0	0
231	6	(ii),(iii)	6	0	0	0
232	6	(ii),(iv)	6	0	0	0
233	6	(ii),(iii)	6	0	0	0
235	7	(ii)	7	0	0	0
236	6	(ii)	6	0	0	0
237	6	(ii)	6	0	0	0
239	6	(ii),(iii)	6	0	0	0
240	12	(iv)	12	0	0	0
241	12	(vi)	12	0	0	0
243	6	(ii)	6	0	0	0
244	6	(ii)	6	0	0	0
245	6	(ii)	6	0	0	0
247	6	(ii),(iii)	6	0	0	0
248	6	(ii),(iv)	6	0	0	0
249	6	(ii),(iii)	6	0	0	0
251	7	(ii)	7	0	0	0
252	6	(ii)	6	0	0	0
253	6	(ii)	6	0	0	0
255	6	(ii),(iii)	6	0	0	0
256	248	S.L.	192	14	248	29
257	96	(vi)	96	14	0	29
259	6	(ii)	6	0	0	0

Note: (1): The approach that gives m^*; (2): approaches (i) - (vi); (3): Ye (1998); (4): Steinberg and Lin (2006); (5) Cioppa and Lucas (2007).

Chapter 3

Generalizations

Generalizations of the construction in Chapter 2 for better projection properties are the focus of this chapter. The idea is motivated by the generalizations introduced by Bingham, Sitter and Tang (2008). Their generalizations are reviewed and the connection to the present methods is discussed. Our generalizations are then introduced and studied. An example will be provided at the end of the chapter.

3.1 Introduction

Bingham, Sitter and Tang (2008) presented two generalizations, one of which improves the projection properties of their basic method. In this section, we will briefly review this generalization and discuss its connection with our present work.

In Section 2.2, we have used $D(n, s^m)$ to denote a design with n runs and m factors, each factor at s levels, where $2 \leq s \leq n$. Let $A = (a_{ij})$ be an $n_1 \times m_1$ matrix with $a_{ij} = \pm 1$. Let D_0 be a $D(n_2, s^{m_2})$. The basic method proposed by Bingham, Sitter and Tang (2008) provides a design

$$D = A \otimes D_0. \tag{3.1}$$

Let D_j be a $D(n_2, s^{m_2})$, for each $j = 1, \ldots, m_1$. They considered the following generalization

$$D = (a_{ij}D_j) = \begin{bmatrix} a_{11}D_1 & a_{12}D_2 & \ldots & a_{1m_1}D_{m_1} \\ a_{21}D_1 & a_{22}D_2 & \ldots & a_{2m_1}D_{m_1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_11}D_1 & a_{n_12}D_2 & \ldots & a_{n_1m_1}D_{m_1} \end{bmatrix}, \tag{3.2}$$

and went on using a simple case to explain that this generalization offers better projection properties as compared with their basic method (3.1). The idea is as follows. When A in (3.1) has the form $((1,1)^T, (1,-1)^T)^T$, the basic method (3.1) produces two columns of the form

$$\begin{pmatrix} d & d \\ d & -d \end{pmatrix}$$

in the resulting design D, where d is a column of D_0. When the design D is projected onto these two columns, its design points lie on the two diagonal lines $y = x$ and $y = -x$, leaving most of the design space unexplored. The generalization (3.2) uses different D_1 and D_2, thereby producing two columns of the form

$$\begin{pmatrix} d_1 & d_2 \\ d_1 & -d_2 \end{pmatrix}$$

where d_1 is a column of D_1 and d_2 is a column of D_2. Obviously, if the column vectors d_1 and d_2 are different, there will not be the diagonal pattern in any two columns of D in (3.2). When we say d_1 and d_2 are different, we mean that $d_1 \neq \pm d_2$. This implies that D_1 should not be obtained from D_2 just by column-permuting and/or sign-switching if we want to eliminate the diagonal pattern exhibited in D in (3.1).

We now discuss the connection between the basic method (3.1) and the proposed method in the previous chapter. Recall that our proposed method for constructing a

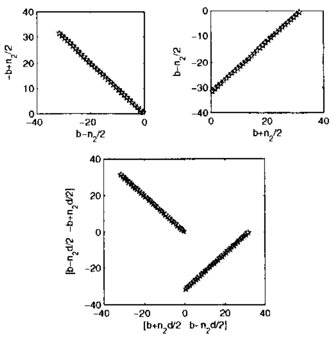


Figure 3.1: d is a column of ones

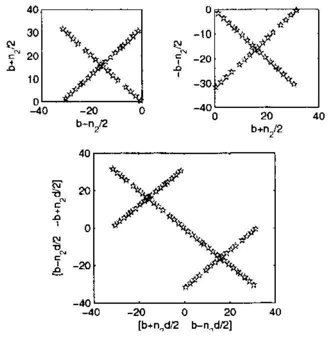


Figure 3.2: d is balanced

Latin hypercube design (LHD) is

$$L = A \otimes B + n_2 C \otimes D, \tag{3.3}$$

where A and D are matrices with entries ± 1 and B and C are LHDs. Now consider a simple case in which

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } C = \tfrac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

The design L, formed as in (3.3), has two columns of the form

$$\begin{pmatrix} b + \frac{n_2}{2} d & b - \frac{n_2}{2} d \\ b - \frac{n_2}{2} d & -b + \frac{n_2}{2} d \end{pmatrix}$$

where b is a column of B and d is a column of D. When the design L is projected onto these two columns, the design points are spread out in two ways based on the column d. If d is a column of plus ones, those design points are plotted in the bottom plot of Figure 3.1. Another case is that the column d is balanced (half 1's and half -1's) and the bottom plot of Figure 3.2 shows the corresponding design points. In both plots, we notice that the design points form two clusters, each having n_2 points. Furthermore, the centers of the two clusters lie on the diagonal line $y = -x$. The top part of Figures 3.1 and 3.2 displays the respective two clusters of design points. The design points in each cluster lie on the diagonal lines $y = x$ or $y = -x$ or both.

In brief, the proposed method in Chapter 2 possesses similar undesirable projection properties as the basic method in Bingham, Sitter and Tang (2008). A natural way to cope with this issue is to adopt their generalization by using different B_j and D_j via

$$L = (a_{ij} \otimes B_j + n_2 c_{ij} \otimes D_j). \tag{3.4}$$

We now use the following example to illustrate the benefit of this generalization. Let

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } C = \frac{1}{2}\begin{pmatrix} -3 & 1 \\ 1 & 3 \\ 3 & -1 \\ -1 & -3 \end{pmatrix}.$$

The design L, formed as in (3.4), becomes

$$L = \begin{pmatrix} B_1 - \frac{3}{2}D_1 & B_2 + \frac{1}{2}D_2 \\ B_1 + \frac{1}{2}D_1 & -B_2 + \frac{3}{2}D_2 \\ B_1 + \frac{3}{2}D_1 & B_2 - \frac{1}{2}D_2 \\ B_1 - \frac{1}{2}D_1 & -B_2 - \frac{3}{2}D_2 \end{pmatrix}.$$

Let $L_1 = (B_1^T - 3D_1^T/2, B_1^T + D_1^T/2, B_1^T + 3D_1^T/2, B_1^T - D_1^T/2)^T$ and $L_2 = (B_2^T + D_2^T/2, -B_2^T + 3D_2^T/2, B_2^T - D_2^T/2, -B_2^T - 3D_2^T/2)^T$. When the design L is projected onto two columns in L_1, the design points are distributed as in Figure 3.3. Similarly, Figure 3.4 displays the design points when the design L is projected onto two columns in L_2. When the design L is projected onto two columns, one from L_1 and the other from L_2, the design points are distributed as in Figure 3.5.

Figures 3.3 and 3.4 indicate that when the design L, formed as in (3.4), is projected onto the two columns from the same B_j and D_j, the design points still roughly lie on the diagonal lines $y = x$ and $y = -x$. This motivates us to consider the generalization,

$$L = (b_{ij} \otimes A_j + n_2 d_{ij} \otimes C_j), \tag{3.5}$$

in which the diagonal pattern of some projected columns will be completely eliminated.

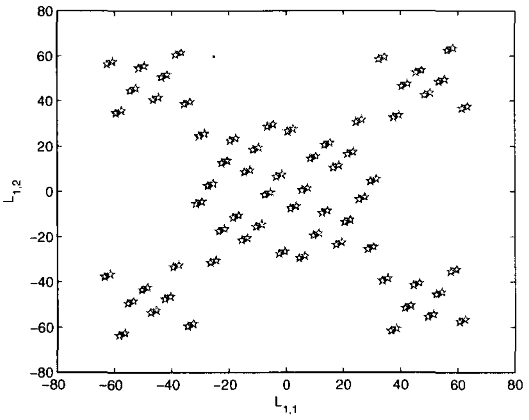


Figure 3.3: Design points of two columns from L_1

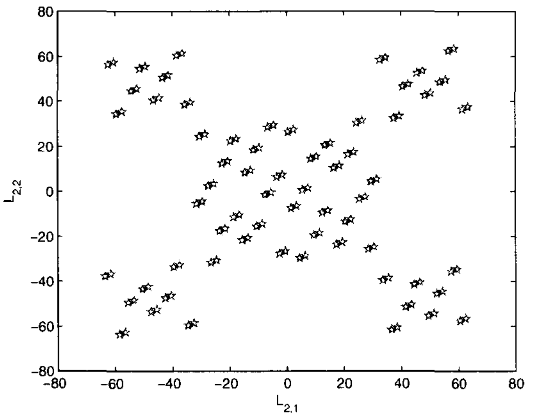
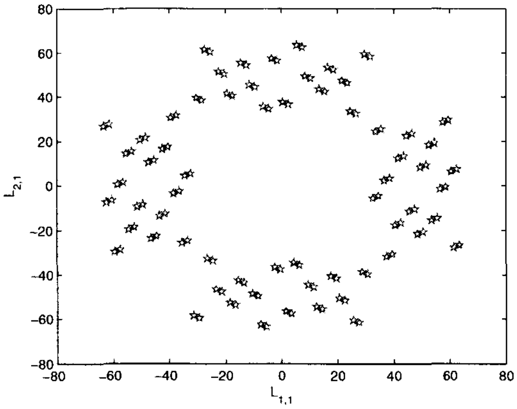


Figure 3.4: Design points of two columns from L_2

Figure 3.5: Design points of two columns, one from L_1 and the other from L_2



3.2 Generalization methods

In this section, we will introduce two generalizations and study the properties of the corresponding designs.

Let $A = (a_{ij})$ be an $n_1 \times m_1$ matrix with $a_{ij} = \pm 1$ and C be a $D(n_1, s_1^{m_1})$. For each $j = 1, \ldots, m_1$, let $B_j = (b_{ik}^j)$ be a $D(n_2, s_2^{m_2})$ and $D_j = (d_{ik}^j)$ be an $n_2 \times m_2$ matrix with $d_{ik}^j = \pm 1$. Further let γ be any real number. Consider the first generalization

$$\begin{aligned} L &= (a_{ij}B_j + \gamma c_{ij}D_j) \\ &= \begin{bmatrix} a_{11}B_1 + \gamma c_{11}D_1 & a_{12}B_2 + \gamma c_{12}D_2 & \cdots & a_{1m_1}B_{m_1} + \gamma c_{1m_1}D_{m_1} \\ a_{21}B_1 + \gamma c_{21}D_1 & a_{22}B_2 + \gamma c_{22}D_2 & \cdots & a_{2m_1}B_{m_1} + \gamma c_{2m_1}D_{m_1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_11}B_1 + \gamma c_{n_11}D_1 & a_{n_12}B_2 + \gamma c_{n_12}D_2 & \cdots & a_{n_1m_1}B_{m_1} + \gamma c_{n_1m_1}D_{m_1} \end{bmatrix}. \end{aligned} \tag{3.6}$$

Let $D = (d_{ij})$ be an $n_2 \times m_2$ matrix with $d_{ij} = \pm 1$ and B be a $D(n_2, s_2^{m_2})$. For each $j = 1, \ldots, m_2$, let $C_j = (c_{ik}^j)$ be a $D(n_1, s_1^{m_1})$ and $A_j = (a_{ik}^j)$ be an $n_1 \times m_1$ matrix with $a_{ik}^j = \pm 1$. Further let γ be any real number. The second generalization provides a design

$$\begin{aligned} L &= (b_{ij}A_j + \gamma d_{ij}C_j) \\ &= \begin{bmatrix} b_{11}A_1 + \gamma d_{11}C_1 & b_{12}A_2 + \gamma d_{12}C_2 & \dots & b_{1m_2}A_{m_2} + \gamma d_{1m_2}C_{m_2} \\ b_{21}A_1 + \gamma d_{21}C_1 & b_{22}A_2 + \gamma d_{22}C_2 & \dots & b_{2m_2}A_{m_2} + \gamma d_{2m_2}C_{m_2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n_21}A_1 + \gamma d_{n_21}C_1 & b_{n_22}A_2 + \gamma d_{n_22}C_2 & \dots & b_{n_2m_2}A_{m_2} + \gamma d_{n_2m_2}C_{m_2} \end{bmatrix}. \end{aligned} \tag{3.7}$$

The first generalization improves the local projection properties while the second one offers better global projection properties. Here the global and local projection properties respectively represent the spread of the clusters and the points in each cluster when the design is projected onto the lower dimensions. Furthermore, the generalizations also permit us to construct LHDs, orthogonal or nearly orthogonal LHDs, and cascading LHDs. We will provide the parallel conditions for the design L constructed by the generalizations to be in these three classes of designs as done in Section 2.2. For simplicity in presentation, we only consider the generalization (3.7) although similar results can readily be obtained for the generalization (3.6).

The proposition below generalizes Proposition 2.1. The proof is analogous to that of Proposition 2.1 and thus omitted here.

Proposition 3.1. *A design L, formed as in (3.7), is a Latin hypercube if*

(i) $s_1 = n_1$, $s_2 = n_2$;

(ii) $\gamma = n_2$;

(iii) *there do not exist j and k, where $j = 1, \dots, m_2$ and $k = 1, \dots, m_1$, such that $a^j_{q_jk} = -a^j_{q'_jk}$ and $d_{pj} = -d_{p'j}$ simultaneously hold, where p and p' are such that $b_{pj} = -b_{p'j}$ and q_j and q'_j are such that $c^j_{q_jk} = -c^j_{q'_jk}$.*

The following result is about the exact and near orthogonality of the design L in the generalization (3.7).

Theorem 3.1. *Suppose that A_j, B, C_j, D and γ are so chosen that a Latin hypercube L is obtained. Furthermore, let A_j and D be column-orthogonal, B be orthogonal, and $B^T D = 0$. We then have that*

(i) $\rho_M(L) = Max\{w_1 \rho_M(C_j), j = 1, \ldots, m_2\}$, where $w_1 = n_2^2(n_1^2-1)/(n_1^2n_2^2-1)$, and

(ii) $\rho^2(L) = w_2 \sum_{j=1}^{m_2} \rho^2(C_j)/m_2$, where $w_2 = (m_1 - 1)w_1^2/(m_1m_2 - 1)]$, and

(iii) L is orthogonal if and only if $C_1, C_2, \ldots, C_{m_2}$ are all orthogonal.

Proof. Let L_{jk} be the column produced by the jth column of B and the kth column A_j. Further let $n = n_1n_2$. Then, parts (i) and (ii) can be easily obtained by noting that

$$\begin{aligned}
\rho(L_{jk}, L_{j'k'}) &= \left[\frac{n(n^2-1)}{12}\right]^{-1} \sum_{i_1=1}^{n_2}\sum_{i_2=1}^{n_1}(b_{i_1j}a^j_{i_2k} + n_2d_{i_1j}c^j_{i_2k})(b_{i_1j'}a^{j'}_{i_2k'} + n_2d_{i_1j'}c^{j'}_{i_2k'}) \\
&= \left[\frac{n(n^2-1)}{12}\right]^{-1} \Big(\sum_{i_1=1}^{n_2} b_{i_1j}b_{i_1j'} \sum_{i_2=1}^{n_1} a^j_{i_2k}a^{j'}_{i_2k'} + n_2 \sum_{i_1=1}^{n_2} d_{i_1j}b_{i_1j'} \sum_{i_2=1}^{n_1} c^j_{i_2k}a^{j'}_{i_2k'} \\
&\quad + n_2 \sum_{i_1=1}^{n_2} b_{i_1j}d_{i_1j'} \sum_{i_2=1}^{n_1} a^j_{i_2k}c^{j'}_{i_2k'} + n_2^2 \sum_{i_1=1}^{n_2} d_{i_1j}d_{i_1j'} \sum_{i_2=1}^{n_1} c^j_{i_2k}c^{j'}_{i_2k'}\Big) \\
&= \left[\frac{n(n^2-1)}{12}\right]^{-1} \Big(\sum_{i_1=1}^{n_2} b_{i_1j}b_{i_1j'} \sum_{i_2=1}^{n_1} a^j_{i_2k}a^{j'}_{i_2k'} + n_2^2 \sum_{i_1=1}^{n_2} d_{i_1j}d_{i_1j'} \sum_{i_2=1}^{n_1} c^j_{i_2k}c^{j'}_{i_2k'}\Big),
\end{aligned}$$

which implies that $\rho(L_{jk}, L_{j'k'}) = 0$ when $j \neq j'$ and $\rho(L_{jk}, L_{j'k'}) = n_2^2(n_1^2 - 1)\rho_{kk'}(C_j)/(n^2 - 1)$ in the case of $j = j'$ and $k \neq k'$. Part (iii) follows directly from parts (i) and (ii). □

Theorem 3.1 says that if every matrix is column-orthogonal and every LHD is orthogonal, the generalization (3.7) results in an orthogonal LHD. If C_j's are relaxed to be nearly orthogonal, the corresponding LHD L is also nearly orthogonal. Note that matrices A_j are required to be column-orthogonal in the theorem. This assumption is not difficult to meet since a column-orthogonal matrix with entries ±1 of n rows can have as many as n columns. As for orthogonal C_j's, they are not rare anymore thanks to the proposed method in Chapter 2. Nevertheless, it is worthwhile to mention that C_j's can be equivalent up to row-permuting, column-permuting and/or sign-switching within one or more columns. It should, however, be noted that column-permuting and sign-switching columns alone do not eliminate the diagonal pattern in the bivariate projections.

We now present the conditions for L in (3.7) to be a two-level cascading LHD in the result below. The proof is omitted as it is similar to that of Theorem 2.4. As discussed in Section 2.2.4, a k-level ($k > 2$) cascading LHD can be easily obtained once we have a two-level cascading LHD.

Theorem 3.2. *Let D be an $n_2 \times m_2$ matrix of plus ones. A design L, formed as in (3.7), is a two-level cascading Latin hypercube of $n = n_1 n_2$ points with level (n_1, n_2) if (i) $s_1 = n_1$ and $s_2 = n_2$; (ii) $\gamma = n_2$.*

To conclude the section, we summarize that the generalizations not only improve the global or local projection properties, but also retain the exact or near orthogonality and the cascading structure.

3.3 A cascading Latin hypercube example

In this section, we use an example to illustrate the benefits gained by using the generalizations as compared with the basic method. In addition, the difference of these two generalizations will be demonstrated.

Example 3.1. Let $n_1 = 9$ and $m_1 = n_2 = m_2 = 3$. We first choose A, B, C and D as follows:

$$B = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}, D = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

$$A = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ -1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & 1 & -1 \end{pmatrix} \text{ and } C = \begin{pmatrix} -4 & -3 & -2 \\ -2 & 0 & 1 \\ -3 & 2 & 3 \\ 0 & -4 & 0 \\ -1 & 1 & 4 \\ 1 & 4 & -4 \\ 3 & -2 & 2 \\ 2 & -1 & -3 \\ 4 & 3 & -1 \end{pmatrix}.$$

Given B, we can row-permute it. We shall denote the resulting LHD after the jth row-permuting by B_j, $j = 1, \ldots, m_1$. Similarly, we use D for all D_j's. For A and C, we row-permute them independently m_2 times instead. The corresponding designs/matrices are denoted by A_k and C_k, $k = 1, \ldots, m_2$.

Table 3.1: Four constructions for cascading LHDs

Designs	Method	Reference
L_1	$A \otimes B + n_2 C \otimes D$	(2.1)
L_2	$B \otimes A + n_2 D \otimes C$	(2.1)
L_3	$a_{ij} \otimes B_j + n_2 c_{ij} \otimes D_j$	(3.6)
L_4	$b_{ik} \otimes A_k + n_2 d_{ik} \otimes C_k$	(3.7)

We now consider four constructions in Table 3.1. They produce four designs L_1, L_2, L_3 and L_4. It is easy to verify that the four designs all are cascading LHDs. Their pairwise plots are shown in Figures 3.6, 3.7, 3.8 and 3.9. There are a few points worth mentioning. First, Figures 3.6 and 3.7 display the identical pattern after row-permuting and column-permuting the pairwise plots because the designs L_1 and L_2 are equivalent up to row-permuting, column-permuting and sign-switching. Second, L_3 constructed by the first generalization provides better local bivariate projection properties, as shown in Figure 3.10. Third, the second generalization improves the global bivariate projection properties of the basic method. For example, Figure 3.11 depicts the sixth and ninth columns of L_2 and L_4. The global diagonal pattern present in the columns of L_2 has vanished in the columns of L_4.

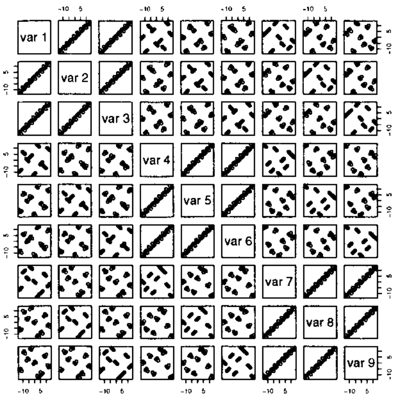


Figure 3.6: Pairwise plot of L_1

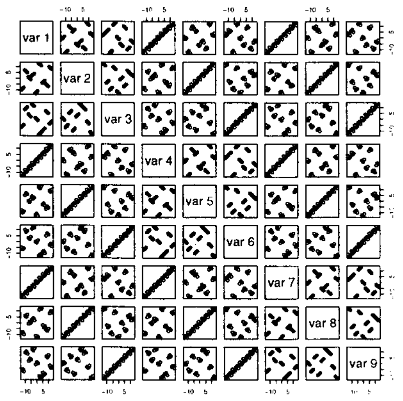


Figure 3.7: Pairwise plot of L_2

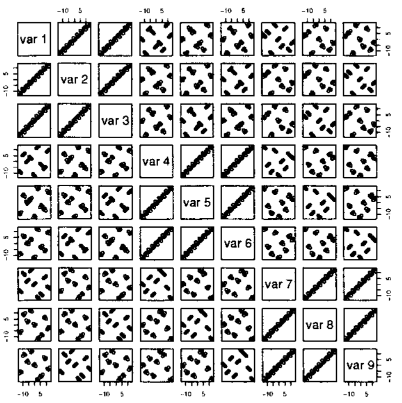


Figure 3.8: Pairwise plot of L_3

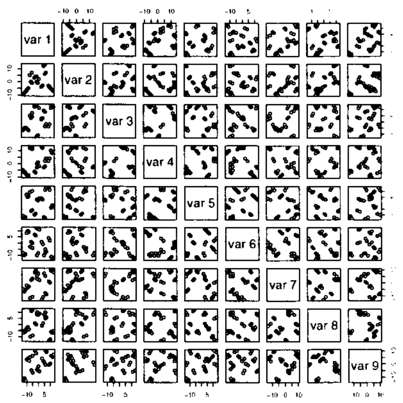
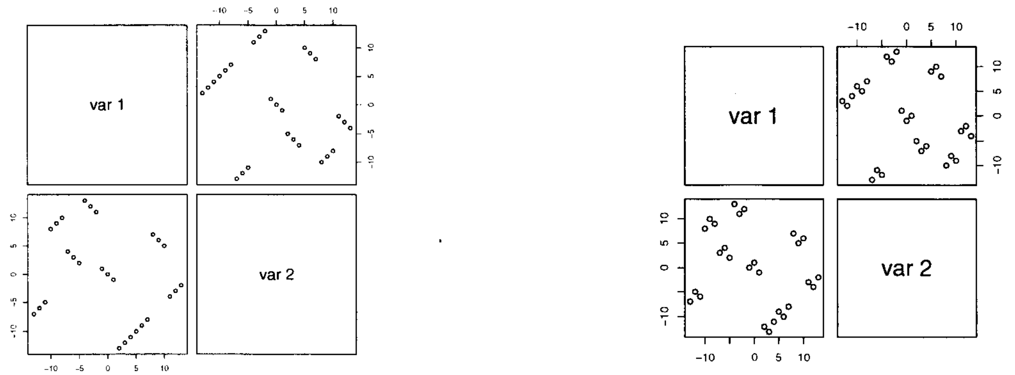
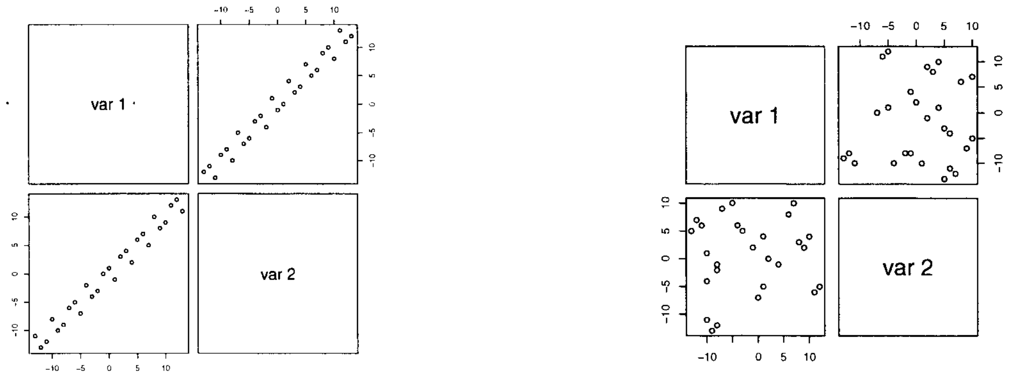


Figure 3.9: Pairwise plot of L_4



(a) (b)

Figure 3.10: Pairwise plot of the second and eighth columns of L_1 and L_3



(a) (b)

Figure 3.11: Pairwise plot of the sixth and ninth columns of L_2 and L_4

Chapter 4

Two-level Fractional Factorial Designs

Fractional factorial designs with factors at two levels are the most widely used in practice. An important question that arises in fractional factorial experimentation is how to judge the "goodness" of designs and select good designs. The minimum G and G_2-aberration are the commonly used criteria for selecting optimal designs. The purpose of this chapter is to provide a collection of good designs based on these two criteria.

A brief outline of this chapter is as follows. In Section 4.1, the problem that we aim to attack is described and relevant work is reviewed. Necessary notation and definitions, as well as the background knowledge, are introduced in Section 4.2. A general method and its implementation are the topics of Section 4.3. The method is then applied to construct designs of 24, 32 and 40 runs in Section 4.4.

4.1 Introduction

We consider factorial experiments with m factors at two levels. A full factorial design requires $n = 2^m$ runs and thus is rarely used in practice unless m is very small. Fractional factorial (FF) designs, which are fractions of full factorial designs, are commonly used instead. Among two-level FF designs, those constructed through the defining relation are called regular designs. The rest are termed nonregular designs. In this chapter, we focus on two-level orthogonal designs in which for every two columns of the design matrix, the four level combinations (1, 1), (1, -1), (-1, 1), (-1, -1) occur equally often.

Minimum aberration (MA) (Fries and Hunter, 1980) is the most popular criterion for choosing a regular design. It has been discussed extensively by many researchers. See Chapter 4 of Wu and Hamada (2000) for a comprehensive review.

In an attempt to evaluate and discriminate general two-level FF designs, Deng and Tang (1999) proposed generalized minimum aberration, also referred to as minimum G-aberration. Because minimum G-aberration is very stringent, Tang and Deng (1999) then introduced a relaxed version of minimum G-aberration, called minimum G_2-aberration. They justified the criterion by showing that it leads to designs that minimize the contamination of nonnegligible interactions on the estimation of main effects. Tang (2001) provided a projection justification of minimum G_2-aberration. Further, Cheng, Deng and Tang (2002) established a justification of minimum G_2-aberration from model robustness and efficiency point of view.

With the minimum G and G_2-aberration, an important problem is to obtain optimal designs with respect to one or both criteria. Deng, Li and Tang (2000) appeared to be the first attempt in this direction. They restricted their attention to the class of Hadamard matrices of orders 16, 20, and 24 and used short versions of minimum G-aberration, which they term MA-4 and MA-5 classifiers, to obtain a catalogue of top nonregular designs of 16 for all $m \leq 15$ and 20 runs for all $m \leq 19$, and 24 runs for $m \leq 8$. Deng and Tang (2002) made similar efforts except that they searched for nonregular designs as well as regular designs.

Because not every two-level orthogonal design can be embedded into a Hadamard matrix, Tang and Deng (2003) sought minimum G-aberration designs within the whole class of orthogonal designs. They were able to construct minimum G-aberration designs of 3, 4, 5 factors for any run size n that is a multiple of 4. Li, Deng and Tang (2004) further pursued the problem in this direction and obtained minimum G-aberration designs of 20, 24, 28, 32 and 36 runs and up to 6 factors. Butler (2003a, 2003b) presented some construction results which allow MA regular and minimum G_2-aberration nonregular designs to be found. The results on MA regular designs of n runs apply to the cases that $5n/16 \leq m < n$. For minimum G_2-aberration nonregular designs, the results are used to find such designs for many of the cases with the run size $n = 16, 24, 32, 48, 64, 96$ and $m \geq n/2 - 2$ factors. Ingram and Tang (2005) focused on designs of 24 runs and provided a complete table of minimum or near-minimum G aberration designs for all values of $m \leq 23$. Xu (2005) made use of the Nordstrom and Robinson (1967) code to construct nonregular designs with 32, 64, 128, and 256 runs with 7-16 factors. Many of these nonregular designs were shown to have minimum G_2-aberration among all possible designs. Xu and Wong (2007) explored the connection between nonregular designs and quaternary linear codes and presented a collection of nonregular designs with 16, 32, 64, 128, 256 runs and up to 64 factors.

In spite of the above rich results, obtaining a collection of good two-level designs remains largely unsolved. In this chapter, we aim to provide a general method for constructing good two-level FF designs of flexible run size n and all possible values of m.

4.2 Notation, and definitions and background

In this section, we will first introduce the notation and concepts used in the rest of the chapter, and then provide some background knowledge on the design construction.

4.2.1 Notation and definitions of two-level FF designs

Consider designs with n runs and m factors, each factor at two levels, denoted by 1 and -1, respectively. We use an $n \times m$ matrix $D = (d_{ij})$ to represent such a design. To assess the "goodness" of two-level designs, Deng and Tang (1999) proposed the generalized resolution and the minimum G-aberration. To introduce them, the following concepts need to be defined. For $s = \{d_1, \ldots, d_k\}$, a subset of k columns of D, define

$$J_k(s) = |\sum_{i=1}^{n} d_{i1} \cdots d_{ik}|,$$

where d_{ij} is the ith entry of column d_j. Obviously, $0 \leq J_k(s) \leq n$. In particular, when D is orthogonal, we have $J_1(s) = J_2(s) = 0$. In addition, when D is a regular design, $J_k(s)$ must equal 0 or n, with 0 corresponding to orthogonality and n to full aliasing.

The formal definition of the generalized resolution is then given as follows.

Definition 4.1. *The generalized resolution of D is defined as*

$$R(D) = r + [1 - max_{|s|=r} J_r(s)/n]$$

where r is the smallest integer such that $max_{|s|=r} J_r(s) > 0$.

It should be noted that for regular designs, the generalized resolution is the same as the usual resolution. Moreover, when D is orthogonal, we have $R(D) \geq 3$. In general, the larger generalized resolution, more desirable a design.

Many designs may have the same generalized resolution. To further characterize or discriminate between two-level designs, Deng and Tang (1999) proposed the minimum G-aberration criterion which will be defined based on the following concept.

Definition 4.2. *Let $n = 4t$. The confounding frequency vector (CFV) of D is defined to be the vector of length $(m-2)(t+1)$, $F(D) = [F_3(D); \dots; F_m(D)]$ where $F_k(D) = (f_{k1}, \dots, f_{k(t+1)})$ and f_{kj} represents the frequency of k column combinations such that $J_k(s) = 4(t+1-j)$ for $j = 1, \dots, t+1$.*

Definition 4.3. *For any two designs D_1 and D_2, let $F(D_1)$ and $F(D_2)$ be their respective CFV's and $f_i(D_1)$ and $f_i(D_2)$ be the corresponding ith entries, where $i = 1, \dots, (m-2)(t+1)$. Let l be the smallest integer such·that $f_l(D_1) \neq f_l(D_2)$. Then D_1 is said to have less G aberration than D_2 if $f_l(D_1) < f_l(D_2)$. If there is no design with less G-aberration than D_1, then D_1 has minimum G-aberration.*

Tang and Deng (1999) proposed a relaxed variant of minimum G-aberration, called minimum G_2 aberration. Let $B_k(D) = n^{-2}\sum_{|s|=k}[J_k(s)]^2$. The generalized word length pattern and minimum G_2 aberration can then be defined.

Definition 4.4. *The vector $(B_1(D), \dots, B_m(D))$ is called the generalized word length pattern.*

Definition 4.5. *For any two designs D_1 and D_2, let r be the smallest integer such that $B_r(D_1) \neq B_r(D_2)$. Then D_1 is said to have less G_2 aberration than D_2 if $B_r(D_1) < B_r(D_2)$. If no design has less G_2-aberration than D_1, then D_1 has minimum G_2-aberration.*

For regular designs, both minimum G-aberration and minimum G_2-aberration reduce to minimum aberration introduced by Fries and Hunter (1980).

Next, we will present a useful result due to Butler (2003b). Let $T = DD^T$ and $M_k = n^{-2}\sum_{p=1}^{n}\sum_{q=1}^{n} T_{pq}^k$, where $T = (T_{pq})$.

Lemma 4.1. *For designs of resolution III and more, we have*

$M_1 = 0, M_2 = m, M_3 = 6B_3, M_4 = 24B_4 + m(3m-2), M_5 = 120B_5 + (60m-120)B_3.$

Therefore, finding minimum G_2-aberration designs is equivalent to sequentially minimizing M_3, M_4, ..., M_m. Calculating M_k is computationally much easier and thus will be adopted in our work. However, it is worth mentioning that Lemma 4.1 does not help for finding minimum G-aberration designs.

4.2.2 Background on design constructions

As mentioned in Chapters 2 and 3, Bingham, Sitter and Tang (2008) proposed one basic method and two generalizations for constructing a rich class of orthogonal designs suitable for computer experiments. Because the method to be used to construct good two-level FF designs here is adapted from their constructions, we next revisit their basic method and generalizations.

In their notation, a design of n runs for m factors of s levels is denoted by $D(n, s^m)$ and represented by an $n \times m$ matrix $D = (d_{ij})$. Their choice of level setting is slightly different from ours in Chapters 2 and 3. They chose s levels to be centered at zero, equally spaced and integer valued. Thus the levels instead are $-s+1, -s+3, \ldots, -1, 1, \ldots, s-3, s-1$ when s is even. When s is odd, the levels remain $-(s-1)/2, \ldots, -1, 0, 1, \ldots, (s-1)/2$. In particular, design D becomes a two-level FF design when $s = 2$.

Let $A = (a_{ij})$ be an $n_1 \times m_1$ matrix with $a_{ij} = \pm 1$ as before. Further let D_0 be a $D(n_2, s^{m_2})$. Their basic method provides a design

$$D = A \otimes D_0, \tag{4.1}$$

which is a $D(n_1 n_2, s^{m_1 m_2})$.

For each $j = 1, \ldots, m_1$, let D_j be a $D(n_2, s^{m_2})$. Their first generalization gives

$$D \;=\; (a_{ij}D_j) = \begin{bmatrix} a_{11}D_1 & a_{12}D_2 & \ldots & a_{1m_1}D_{m_1} \\ a_{21}D_1 & a_{22}D_2 & \ldots & a_{2m_1}D_{m_1} \\ \vdots & \vdots & & \vdots \\ a_{n_1 1}D_1 & a_{n_1 2}D_2 & \ldots & a_{n_1 m_1}D_{m_1} \end{bmatrix}. \tag{4.2}$$

They then studied the orthogonality and 3-orthogonality of design D. The orthogonality is the same as the one in orthogonal LHDs in Chapter 2.

Definition 4.6. *Design D is called orthogonal if it is balanced and the inner product of any two columns of D is zero, that is, $\sum_{i=1}^{n} d_{ij} = 0$ and $\sum_{i=1}^{n} d_{ij_1} d_{ij_2} = 0$.*

Definition 4.7. *Design D is called 3-orthogonal if it simultaneously satisfies*

(i) $\sum_{i=1}^{n} d_{ij} = 0$ *for all* j;

(ii) $\sum_{i=1}^{n} d_{ij_1} d_{ij_2} = 0$ *for all* $j_1 \neq j_2$;

(iii) $\sum_{i=1}^{n} d_{ij_1} d_{ij_2} d_{ij_3} = 0$ *for all* j_1, j_2, j_3.

Note that, a two-level FF design is orthogonal if and only if $r \geq 3$ and is 3-orthogonal if and only if $r \geq 4$, where r is defined as in Definition 4.1.

The following results will be useful for the later development.

Lemma 4.2. *Let A be column-orthogonal. Design D in (4.1) is orthogonal if and only if D_0 is orthogonal.*

Lemma 4.3. *Let A be column-orthogonal. Design D in (4.2) is orthogonal if and only if $D_1, \ldots, D_{m_1}$ are all orthogonal.*

4.3 Design construction

Consider constructing two-level orthogonal FF designs of n runs for m factors. Suppose that there exist n_1, n_2, m_1 and m_{2j} ($j = 1, \ldots, m_1$) such that an $n_1 \times m_1$ column-orthogonal matrix $A = (a_{ij})$ with $a_{ij} = \pm 1$ and $n_2 \times m_{2j}$ orthogonal two-level D_j's can be obtained. Consider the following construction

$$D = (a_{ij} D_j) = \begin{bmatrix} a_{11}D_1 & a_{12}D_2 & \ldots & a_{1m_1}D_{m_1} \\ a_{21}D_1 & a_{22}D_2 & \ldots & a_{2m_1}D_{m_1} \\ \vdots & \vdots & & \vdots \\ a_{n_1 1}D_1 & a_{n_1 2}D_2 & \ldots & a_{n_1 m_1}D_{m_1} \end{bmatrix}. \tag{4.3}$$

Obviously, design D is an orthogonal two-level FF design with n runs for m factors by Lemma 4.3, where $m = \sum_{j=1}^{m_1} m_{2j}$. Compared with the first generalization (4.2) in Bingham, Sitter and Tang (2008), the construction (4.3) allows D_j's to have different numbers of factors.

There are a few important issues regarding the use of the construction above. First, the construction is applicable to any run size that is a multiple of 8. Second, for a given run size n, there may exist multiple value settings of n_1 and n_2. Unfortunately, no general theory on the optimal setting of n_1 and n_2 can be given at this moment. Instead, we consider all possible combinations of n_1 and n_2. Third, for a given m_1, the vector $(m_{21}, \ldots, m_{2m_1})$ may have different value settings. There is no clear optimal choice of the vector $(m_{21}, \ldots, m_{2m_1})$, as indeed shown in the applications of the construction in Section 4.4. Finally, we discuss the choices of D_j's for all $j = 1, \ldots, m_1$. The first possibility is $D_j \subset D^*_{\mathrm{M}}$ where $\mathrm{M} = \max\{m_{21}, \ldots, m_{2m_1}\}$. That is, after taking a design D^*_{M} from a complete catalogue of designs of n_2 runs with M factors, we can take m_{2j} columns from these M columns to form D_j. In fact, this covers the basic construction in Bingham, Sitter and Tang (2008). The second possibility is taking each D_j from a complete catalog of designs of n_2 runs for m_{2j} factors. In addition, better G or G_2 aberration designs may be attained by row-permuting D_j's for each design obtained by the two possibilities above. It should, however, be mentioned that the designs after column-permuting or sign-switching D_j's are isomorphic to the initial design and thus do not help improve the criteria of minimum G or G_2-aberration.

The above discussion leads us to consider how one can row-permute D_j's. For n_2-run D_j's, $j = 1, \ldots, m_1$, there are $(n_2!)^{m_1-1}$ possible row permutations. It is computationally infeasible to carry out all these permutations even for moderately large values of n_2 and m_1. When the complete search is impossible, random permutations become a naive solution. Here, we propose an efficient algorithm to search for good designs. The algorithm essentially adopts the two important operations, pairwise switch and exchange, in Xu's algorithm (see Section 2.4 for the details). For ease in presentation, we assume $n_1 = m_1 = 2$, $a_{11} = a_{12} = a_{21} = 1$ and $a_{22} = -1$ in the construction (4.3). In other words, we aim to construct designs

$$D = \begin{bmatrix} D_1 & D_2 \\ D_1 & -D_2 \end{bmatrix}, \tag{4.4}$$

where D_1 and D_2 are two-level orthogonal designs of n_2 runs for m_{21} and m_{22} factors, respectively. For each given D, the algorithm for seeking a better design by row-permuting D_2 works as follows.

Step 1: Randomly row permute D_2;

Step 2: For each pair of rows in D_2, make a switch and calculate the corresponding criterion. Choose the pair with the best value of the criterion and switch the pair of rows. Repeat Step 2 until no further improvement is possible;

Step 3: Repeat Step 1 and Step 2 T times.

The algorithm above is a general form. We may be able to perform fast update in Step 2 depending on the criterion used in the algorithm.

With the construction (4.3) and the above discussion, we can have a search algorithm for obtaining a collection of good designs using minimum G and G_2-aberration defined in the previous section. For simplicity, the algorithm is presented only for the simple form of the construction as in (4.4). Let Γ_i be the catalogue of non-isomorphic designs of n_2 runs for i factors.

The following procedure generates a collection of S top designs formed as in (4.4) and ranked by minimum G-aberration criterion.

Let $\mathcal{C}$ be the top designs obtained and s be the number of designs in $\mathcal{C}$, with initial values $\mathcal{C} = \varnothing$ and $s = 0$. For $m_{21} = 0, \ldots, m$, let $m_{22} = m - m_{21}$ and do the following.

Step 1: If $m_{22} = 0$, obtain a design $D = [D_1^T, D_1^T]^T$ and calculate its CFV. Set $s = s + 1$ and add this design to $\mathcal{C}$. If $s = S + 1$, then discard the design with the worst CFV in $\mathcal{C}$ and set $s = s - 1$;

Step 2: If $m_{21} = 0$, obtain a design $D = [D_2^T, -D_2^T]^T$ and calculate its CFV. Set $s = s + 1$ and add this design to $\mathcal{C}$. If $s = S + 1$, then discard the design with the worst CFV in $\mathcal{C}$ and set $s = s - 1$;

Step 3: If $m_{21} > 0$ and $m_{22} > 0$, let $M_{max} = \text{Max}\{m_{21}, m_{22}\}$. For each design D_0 in the catalogue $\Gamma_{M_{max}}$, do the following:

(a) If $m_{21} = M_{max}$, let $D_1 = D_0$ in (4.4). If $m_{22} = M_{max}$, let $D_2 = D_0$ in (4.4);

(b) Consider all possible M_{min} columns out of M_{max} columns of D_0 where $M_{min} = \text{Min}\{m_{21}, m_{22}\}$. Let the M_{min} columns be D_2 if $m_{21} = M_{max}$ and D_1 if $m_{22} = M_{max}$. Obtain a design D formed as in (4.4) and calculate its CFV. Set $s = s + 1$ and add this design to $\mathcal{C}$. If $s = S + 1$, then discard the design with the worst CFV in $\mathcal{C}$ and set $s = s - 1$;

(c) Randomly row permute D_2;

(d) For each pair of rows in D_2, make a switch and calculate the corresponding criterion. Choose the pair with the smallest CFV and switch the pair of rows. Set $s = s + 1$ and add this design to $\mathcal{C}$. If $s = S + 1$, then discard the design with the worst CFV in $\mathcal{C}$ and set $s = s - 1$;

(e) Repeat (d) until no further improvement is possible;

(f) Repeat (c), (d) and (e) T times.

Similarly, the procedure above can be adjusted to construct a collection of good designs according to minimum G_2-aberration. In the next section, we will apply this procedure to obtain a collection of good designs of 24, 32, 40 runs based on minimum G and G_2-aberration criteria.

4.4 Applications to designs of 24, 32, 40 runs

In this section, the method (4.3) is applied to designs of 24, 32 and 40 runs, using minimum G and G_2-aberration as ranking criteria. New results are presented and comparisons with the existing results are made. Due to the space consideration, the design matrices are not given here but are available upon request. Information on the generalized resolution, generalized word length pattern and CFV's of the top designs is provided in Appendices B, C and D.

4.4.1 Designs of 24 runs

For the case $n = 24$, we choose $n_1 = 2$ and $n_2 = 12$ in (4.3). Appendix B contains the three best CFV's and the three best generalized word length patterns. We also report one combination of m_{21} and m_{22} such that the corresponding CFV's or generalized word length pattern is achieved.

We compare our result with Ingram and Tang (2005). In Appendix B, the '*', '**' and '***' designations correspond to the cases that our design has less aberration than, the same aberration as, and more aberration than the design found by Ingram and Tang (2005). For the case $3 \leq m \leq 12$, we found the same G-aberration designs as those obtained by Ingram and Tang (2005). Therefore, these designs are minimum G-aberration followed by Proposition 1 in Ingram and Tang (2005). In addition, we found two 24×6 designs of resolution 4.67 because there are two non-isomorphic designs of 12 runs for 6 factors. For the case $m \geq 13$, although the minimum G-aberration design obtained by Ingram and Tang (2005) has less aberration than the one found by our method, excluding the case $m = 14$, the difference is very small. For minimum G_2-aberration, we found as good designs as those by Ingram and Tang (2005). The comparison also leads us to conclude that there exist two-level designs that do not have form (4.3).

4.4.2 Designs of 32 runs

In the use of the method (4.3) for constructing 32-run designs, we have four choices of n_1 and n_2, ($n_1 = 2$, $n_2 = 16$), ($n_1 = 4$, $n_2 = 8$), ($n_1 = 16$, $n_2 = 2$) and ($n_1 = 8$, $n_2 = 4$). Our investigation indicates that the combination ($n_1 = 2$, $n_2 = 16$) produces the most comprehensive and best designs. Consequently, we choose $n_1 = 2$ and $n_2 = 16$ in (4.3) for constructing 32-run designs of m factors, where $3 \leq m \leq 31$. The three best resolution, generalized word length pattern, and CFV's are listed in Appendix C.

We compare our results with Xu and Wong (2007), yielding the last column of the tables in Appendix C. The 'G**' and 'G_2**' designations correspond to the cases our designs have the same G and G_2 aberration as those obtained by Xu and Wong (2007), respectively. The 'G*' and 'G_2*' designations indicate that our designs are better than those obtained by Xu and Wong (2007). Note that m for 32-run designs in Xu and Wong (2007) must satisfy $7 \leq m \leq 24$. Appendix C reveals that our designs are better for $m \geq 10$ and as good as theirs for $m \leq 9$ in terms of minimum G-aberration. Turning to minimum G_2-aberration, our designs are better for $m = 10$ and equally good in other cases. As a result, the method (4.3) not only allows us to construct a class of good designs with every possible number of factors, but also obtain the best or nearly-best designs in terms of both criteria.

4.4.3 Designs of 40 runs

A complete catalogue of 20-run non-isomorphic designs is available thanks to Sun, Li and Ye (2002). We construct 40-run designs of m ($3 \leq m \leq 39$) factors by using $n_1 = 2$ and $n_2 = 20$ in (4.3). However, we have not considered the row permutations of D_j's for the time being. The resulting two best generalized word length patterns and CFV's are tabulated in Appendix D. These results are new. The search incorporating row permutations of D_j's will be done in the future work.

Chapter 5

Folded Over Non-Orthogonal Designs

Folded over non-orthogonal designs for screening are studied in this chapter. The notion of minimal dependent sets (MDS) is used to introduce MDS-resolution and MDS-aberration as criteria for comparing folded over non-orthogonal designs. A fast isomorphism check is developed that uses a cyclic matrix defined on the design before it is folded over. The isomorphism check is used to obtain a catalogue of minimum MDS-aberration designs for some useful run sizes n and the number k of factors. An algorithm for obtaining "good" larger designs is discussed.

5.1 Introduction

Screening experiments are used to sift through a set of candidate factors to identify those that impact the response – these factors are referred to as being "active." For this chapter we assume that the standard linear model assumptions are valid. Further we assume that the active factors can impact the response through either a main effect (ME) or a two-factor interaction (2FI) but that all interactions involving three or more factors are negligible. Thus the model is a linear model that contains ME's and 2FI's formed using the active factors. The primary goal of a screening experiment is to identify the active factors but an important secondary goal is to provide a simple model that captures the essential features of the relationship between these active factors and the response. Clearly, if an experiment is run that allows the true model to be correctly identified, then both of these goals are achieved. Folded over non-orthogonal two-level designs were demonstrated to be useful in such screening experiments (Miller and Sitter, 2005). Such designs are our study objects in this chapter. The term "folding over" indicates that the levels of all the factors are reversed to form runs that are the mirror images of those in the original design.

To assess and compare folded over non-orthogonal designs, we introduce two criteria, MDS-resolution and MDS-aberration, both of which are based on the concept of MDS developed by Miller and Sitter (2004).

With MDS-resolution and MDS-aberration, we can obtain a catalogue of MDS-aberration folded over non-orthogonal designs. However, this is not an easy task as it involves determining whether or not two designs are in fact different. Two designs are said to be isomorphic if one can be obtained from the other by relabeling the factors, reordering the treatment combinations and/or relabeling the levels of one or more factors. Otherwise, the two designs are non-isomorphic. In other words, isomorphic designs can be changed into each other by the usual randomization of factor labels and level labels. Since isomorphic designs share the same statistical properties in classical ANOVA models and are essentially the same, it is sufficient to include only one of them in a catalogue of designs. In addition, one wants to avoid considering more than one of them in any search for optimal designs and thus avoid unnecessary computations. The identification of the isomorphism of two designs is a combinatorial problem. For two k-factor (each having two levels) n-run designs, a complete search compares $n!k!2^k$ designs based on the definition of isomorphism. It is known as an NP problem, when n and k increase. To alleviate the computational burden, we develop a fast isomorphism check that uses a cyclic matrix defined on the design before it is folded over. By doing so, the speed of checking for isomorphism is much faster than directly applying an isomorphism check to the fold-over design. This relative difference becomes greater as the design size increases. As a result, we are able to use the isomorphism check to obtain a catalogue of minimum MDS-aberration designs for some useful n and k, and we will also discuss an algorithm for obtaining "good" larger designs.

5.2 MDS-resolution and MDS-aberration

In this section, the concept of minimum dependent sets (MDS) will be reviewed. Two criteria, MDS-resolution and MDS-aberration, will then be introduced and discussed. Miller and Sitter (2004) introduced the concept of MDS. Its formal definition is given as follows.

Definition 5.1. *A minimal dependent set is a set of 2FI's such that the model that contains all of the main effects and this set of 2FI's is not estimable but if any of the 2FI's is removed the resulting model is estimable.*

A model is estimable if and only if the columns in its model matrix are linearly independent. As an illustration, we consider a 12-run Plackett-Burman design (PB12).

Example 5.1. Table 5.1 contains the design matrix of the 12-run Plackett-Burman design. Consider the first 5 columns from this design and denote them by $PB12_{5a}$. The number of MDS's of various sizes for design $PB12_{5a}$ is given in Table 5.2. Note that the smallest MDS are of size four. In addition, design $PB12_{5a}$ has 10 MDS's of size four, one of which is $\{12, 13, 24, 35\}$. This implies that it will be difficult to distinguish between the following sets of interactions: (a) $\{12, 13\}$ from $\{24, 35\}$, (b) $\{12, 24\}$ from $\{13, 35\}$ and (c) $\{12, 35\}$ from $\{13, 24\}$.

Table 5.1: The 12-run Plackett-Burman design

1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	1	1	1	1	1
-1	-1	-1	1	-1	1	1	-1	1	-1	1
1	-1	-1	-1	1	1	1	-1	-1	1	-1
1	1	-1	-1	-1	-1	1	1	-1	-1	1
1	1	1	-1	-1	1	-1	-1	1	-1	-1
-1	1	1	1	-1	-1	1	-1	-1	1	-1
1	-1	1	1	1	-1	-1	-1	-1	-1	1
-1	1	-1	1	1	1	-1	1	-1	-1	-1
-1	-1	1	-1	1	-1	1	1	1	-1	-1
1	-1	-1	1	-1	-1	-1	1	1	1	-1
-1	1	-1	-1	1	-1	-1	-1	1	1	1
-1	-1	1	-1	-1	1	-1	1	-1	1	1

Table 5.2: Minimal dependent sets

Design	Number of 2FI's in the MDS					
	1	2	3	4	5	6
$PB12_{5a}$	0	0	0	10	0	80
$PB12_{5b}$	0	0	0	15	0	15

An MDS implies that we cannot distinguish some 2FI's in the MDS from the rest of 2FI's. This implication can be better understood if we view an MDS as a word. A word is an interaction that equals the identity element I in regular fractional factorial designs. For example, in a 2^{5-1} design, $I = 2345$ means that the corresponding design is not capable of distinguishing the two effects of 5 and 234. Just as a longer word is preferred, an MDS of larger size is preferable. In other words, we would like to select designs such that their MDS's are as large as possible. If we maximize the size of the smallest MDS, we obtain the criterion of maximum MDS-resolution in obvious parallel to maximum resolution in regular FF designs. It is also evident that if we have two designs that have the same size of the smallest MDS but one has fewer

MDS's of that size than the other, then the former is preferred to the latter. Thus, we introduce the "MDS word length pattern" as follows.

Definition 5.2. *The vector* $W = (A_1, A_2, ..., A_k)$ *is called the MDS word length pattern where* A_i *is the number of MDS's of size* i.

Returning to Example 5.1, design $PB12_{5a}$ has an MDS word length pattern (0, 0, 0, 10, 0, 80). It indicates that this design has no MDS's of size ≤ 3, 10 MDS's of size 4, no MDS of size 5, and 80 MDS's of size 6. This is an obvious parallel to the usual word length pattern of the defining contrast subgroup of a regular FF design. This leads to the obvious notion of MDS-aberration.

Definition 5.3. *For two designs* d_1 *and* d_2, *let* r *be the smallest integer such that* $A_r(d_1) \neq A_r(d_2)$. *Then* d_1 *is said to have less MDS-aberration than* d_2 *if* $A_r(d_1) < A_r(d_2)$. *If there is no design with less MDS-aberration than* d_1, *then* d_1 *has minimum MDS-aberration.*

Example 5.2. Consider the last 5 columns from the design in Table 5.1. We refer to these 5 columns as design $PB12_{5b}$. The corresponding MDS word length pattern is (0, 0, 0, 15, 0, 15), given in Table 5.2. Based on the minimum MDS-aberration criterion, design $PB12_{5a}$ has less MDS-aberration than design $PB12_{5b}$, and is therefore preferred.

We have introduced maximum MDS-resolution and minimum MDS-aberration. Both of them will be used for comparing designs in Section 5.4.

5.3 Folded over non-orthogonal designs for screening

Miller and Sitter (2005) have investigated the use of folded over non-orthogonal designs for screening. Their work indicates that fold-over designs are effective in situations where the following conditions are satisfied:

(1) All interactions that involve three or more factors are negligible.

(2) At most, a small proportion of the 2FI's will be active.

(3) A 2FI that satisfies strong heredity is more apt to be active than one that satisfies weak heredity which, in turn, is more apt to be active than one that does not satisfy heredity.

In the condition (3), strong heredity assumes that an interaction can be active only when both corresponding main effects are active, and weak heredity assumes that an interaction can be active when at least one of the corresponding main effects is active (see Chipman, 1996; Chipman, Hamada and Wu, 1997).

They go on to propose a 2-stage analysis that exploits the fact that for fold-over designs there is a clear separation of the information about ME's and 2FI's. That is, for fold-over designs it is well known that every odd-order effect (ME's, 3FI's, 5FI's, etc.) is orthogonal to every even-order effect (intercept, 2FI's, 4FI's, etc.) and that the sample space of the response can be divided into two orthogonal subspaces each of dimension $n/2$ such that all the odd-order effect vectors occur in one subspace and all of the even-order effect vectors occur in the other. Given that the intercept is included in all models it is useful to adjust the 2FI's to make them all orthogonal to the intercept. Under the assumption that all interactions involving more than 2 factors are negligible, the sample space of the response $\mathbf{Y}$ can be split into three orthogonal subspaces: a subspace of dimension 1 that contains the intercept, a subspace of dimension $n/2$ that contains the ME's and a subspace of dimension $n/2-1$ that contains the 2FI's. There are two consequences of this that are important for the following discussion. First, we can evaluate how effective a design will be for identifying active ME's and for identifying active 2FI's separately. Second, although the degrees of freedom available for ME's and 2FI's are roughly the same, $n/2$ and $n/2-1$, there are typically considerably fewer ME's than 2FI's. As a result, the criteria we use to evaluate how well a design can identify ME's will differ from that used for 2FI's.

First consider ME's. In order not to restrict the maximum number of active ME's that can be identified, we only consider designs that allow the full ME model to be estimated. Thus the designs can be used for situations where the practitioner believes, a priori, that all the ME's may be active and wishes estimates for all ME's as well as for screening applications. To evaluate ME estimation/identification, we adapt the definition of efficiency used in Margolin (1969):

$$\text{ME efficiency} = k/\left[n \times \text{trace } \left(\mathbf{X}_{ME}^T\mathbf{X}_{ME}\right)^{-1}\right],$$

where $\mathbf{X}_{ME}$ contains only the columns in the model matrix $\mathbf{X}$ for ME's. This evaluates average variance of the estimated main effects and thus is closely related to A-efficiency. If ME's are defined as 1/2 the difference between the average response at the -1 and $+1$ levels then the average variance of the estimated ME's is equal to $\sigma^2/(n \times \text{efficiency})$. A design that has orthogonal ME's will have efficiency $= 1$ and designs which have non-orthogonal ME's will have efficiency < 1 which will result in the average variance of the estimated ME's being inflated by a factor of 1/efficiency. Thus the efficiency can be interpreted as a measure of how close the ME-design matrix is to being orthogonal. It is generally accepted that for screening applications the best possible situation is to have all the effects orthogonal to each other. Thus this measure of efficiency should also give a good indication of how suitable the design is for screening applications.

Now consider 2FI's. For the fold-over designs considered in this chapter, and larger designs, the degrees of freedom (dfs) available for 2FI's are not large enough to allow the full 2FI model to be estimated. For example, for the 6-factor 24-run design there are 11 dfs available for 2FI's and a total of 15 2FI's to be considered. Thus we have a situation where the design is supersaturated with respect to 2FI's and we cannot use the efficiency of estimated effects for the full 2FI model as a criterion. For this type of situation, minimum MDS-aberration as described in Section 5.2 provides a useful criterion to evaluate how well a design can screen for 2FI's.

To obtain fold-over designs that have high ME-efficiency and minimum or near-minimum MDS-aberration for the 2FI's that can entertain from 4 to 12 factors in 24 runs or less requires an extensive computer search. This can be attributed to three facts: (1) the number of possible designs is large for each combination of factor number, k, and run size, n, and (2) calculating the MDS word length pattern for a single design can consume a surprising amount of computing time; and (3) checking isomorphism (equivalence) of any two designs is computationally intensive. All of these problems become worse as k and n increase.

We address these problems by first developing a new isomorphism check that is an adaptation from Clark and Dean (2001) specifically for fold-over designs in the next section.

5.4 An isomorphism check

Let A and B be two 2-level $n \times k$ non-orthogonal design matrices, and let $D_1 = (A^T, -A^T)^T$ and $D_2 = (B^T, -B^T)^T$ be the $2n \times k$ design matrices constructed by respectively folding these over. Then we have the following definitions, the second of which follows from the first and the special structure of the design matrix of a folded over non-orthogonal design.

Definition 5.4. *D_1 and D_2 are said to be isomorphic or equivalent if one can be obtained from the other by row permutations, column permutations and relabeling the levels within one or more columns.*

Definition 5.5. *D_1 and D_2 are said to be isomorphic or equivalent if there exists an $n \times n$ row permutation matrix R and a $k \times k$ column permutation matrix C such that $A = L_1 RBCL_2$, where L_1 and L_2 are diagonal matrices with ± 1 on the diagonals.*

For $k \geq 2$, define a *cyclic* matrix A^* of a matrix $A = (a_{ij})_{n\times k}$ to have (i,j)th element

$$[A^*]_{i,j} = \begin{cases} -1, & \text{if } a_{i(j+1)} = a_{ij} \\ 1, & \text{if } a_{i(j+1)} \neq a_{ij} \end{cases}$$

for each $j = 1, 2, \ldots, k-1$, $i = 1, 2, \ldots, n$ and

$$[A^*]_{i,k} = \begin{cases} -1, & \text{if } a_{ik} = a_{i1} \\ 1, & \text{if } a_{ik} \neq a_{i1} \end{cases}$$

for each $i = 1, 2, \ldots, n$.

We call A^* a cyclic matrix for easy reference. The term cyclic refers to the fact that one compares the last element of the ith row of A to the first element when forming A^*. The cyclic matrix A^* is invariant to changes of sign within rows of A and for each row of A, if we know any one of the entries, we can obtain A from A^*. In other words, we can say that A^* uniquely determines A up to changes of sign within rows.

Define the Hamming distance matrix $h(A^*)$ of A^* to have (i,j)th element

$$[h(A^*)]_{i,j} = \begin{cases} \sum_{l=1}^{k} \delta[A^*]_{i,j}^{l}, & \text{if } i \neq j \\ 0, & \text{if } i = j, \end{cases}$$

where $\delta[A^*]_{i,j}^{l}$ is equal to 1 if in the lth column of A^*, the symbols in the ith and jth rows are different, and equal to zero if they are the same. The (i,j)th element of $h(A^*)$ counts the number of dimensions in which the ith and jth points fail to coincide. The distance matrix $h(A^*)$ is invariant to permutations of columns and relabeling of levels within columns of A^*.

Let $(AC)^*$ denote the cyclic matrix of AC and $h((AC)^*[1:q])$ be the Hamming distance matrix corresponding to the first q columns of $(AC)^*$.

Lemma 5.1. *For any given column permutation matrix C and given row permutation matrix R, the sequence of matrices $Rh((AC)^*[1:q])R^T, q = 1, 2, \ldots, k$, uniquely determines the matrix A up to the equivalence defined above.*

Proof. The following proof combines the proof of Lemma 2.2 and Theorem 2.1 of Clark and Dean (2001) with the fact that the cyclic matrix of a matrix uniquely determines the matrix up to changes of sign within rows.

Since permuting rows before or after creating the cyclic matrix is equivalent,

$$(RAC)^*[1:q] = R((AC)^*[1:q]). \tag{5.1}$$

This implies that $(RAC)^*[1:q]$ and $(AC)^*[1:q]$ are isomorphic, as (5.1) implies one can be obtained from the other via row permutations. Since a necessary condition for the isomorphism of any design matrices, D_1 and D_2, is that there exists a row permutation matrix R such that $h(D_1) = Rh(D_2)R^T$, $h((RAC)^*[1:q])=Rh\big((AC)^*[1:q]\big)R^T$. For a given R let the sequence of matrices $Rh((AC)^*[1:q])R^T$, $q=1,2,\ldots,k$, corresponding to a fixed but unknown matrix A, be fixed. Note that, for any $q \le k$,

$$\begin{aligned} [Rh((AC)^*[1:q])R^T]_{i,j} &= \sum_{p=1}^{q-1} \delta[R(AC)^*]^p_{i,j} + \delta[R(AC)^*]^q_{i,j} \\ &= [R(h((AC)^*[1:(q-1)]))R^T]_{i,j} + [R(h((AC)^*[q])R^T]_{i,j} \end{aligned}$$

where $(AC)^*[q]$ denotes the qth column of $(AC)^*$. Thus, a fixed sequence of distance matrices $Rh((AC)^*[1:q])R^T$, $q=1,2,\ldots,k$, implies a fixed sequence $Rh((AC)^*[q])R^T$, $q=1,2,\ldots,k$, and we may investigate each column of A separately. Let $\widetilde{A^*}$ be an $n \times k$ matrix with the first row $[-1,-1,\ldots,-1]$. For each $q \in \{1,2,\ldots,k\}$, we construct the qth column of $\widetilde{A^*}$ as follows. For $i = 2,3,\ldots,n$ in turn, if $[R(h((AC)^*[q])R^T]_{i,j} = 0$, for some $j = 1,2,\ldots,i-1$, then the symbol (-1 or 1) in the ith row of column q of $\widetilde{A^*}$ is identical to the symbol in the jth row, so set $[\widetilde{A^*}]_{i,q} = [\widetilde{A^*}]_{j,q}$. Otherwise, set $[\widetilde{A^*}]_{i,q}$ equal to an unused symbol. The qth column of $\widetilde{A^*}$ is then identical to the qth column of $R(AC)^*$, up to a relabeling of the symbols in the column. Let $\widetilde{A}$ be an $n \times k$ matrix with the first row $[1,1,\ldots,1]$ and the first column $[1,1,\ldots,1]^T$. Based on the definition of the cyclic matrix, we can obtain $\widetilde{A}$ from $\widetilde{A^*}$. Thus, $\widetilde{A}$ is identical to A up to row permutations, column permutations, symbol relabeling within columns and changes of sign within the rows. □

Theorem 5.1. *Designs D_1 and D_2 are isomorphic iff there exists an $n \times n$ row permutation matrix R and a column permutation matrix C such that, for every $q = 1$, $2, \ldots, k$, $h(A^*[1:q])$=$R(h((BC)^*[1:q])R^T$.*

Proof. **Necessity**: Suppose that design D_1 and D_2 are isomorphic. The distance matrix $h((BC)^*)$ is invariant to symbol relabeling in any columns of $(BC)^*$ and B^* is invariant to changes of sign within any rows of B. Hence, without loss of generality we assume that the factors in designs D_1 and D_2 have the same level labeling. Then we can write $A = L_2RBC$ which implies $A^* = R(BC)^*$, where C is the row permutation matrix and R is the permutation matrix corresponding to the row permutation. Then, for $1 \le p \le k$, we have $[h(A^*[p])]_{i,j} = \delta[R(BC)^*]^p_{i,j} = [h((BC)^*[p])]_{r_i,r_j}$. Therefore, for each $q = 1, 2, \ldots, k$,

$$\begin{aligned}
[h(A^*[1:q])]_{i,j} &= \sum_{p=1}^{q}[h(A^*[p])]_{i,j} = \sum_{p=1}^{q}[h((BC)^*[p])]_{r_i,r_j} \\
&= \sum_{p=1}^{q}[R[h((BC)^*[p])]R^T]_{i,j} = [R(h((BC)^*[1:q])R^T]_{i,j}.
\end{aligned}$$

Sufficiency: Follows from Lemma 5.1. □

Corollary 5.1. *Designs D_1 and D_2 are isomorphic iff there exists an $n \times n$ row permutation matrix R and a column permutation matrix C such that, for every $q = 1$, $2, \ldots, k$, $h(A^*[q])$=$R(h((BC)^*[q])R^T$.*

For folded over non-orthogonal designs where all the factors have two levels, A^* is still a matrix whose entries have two levels. In this case, the distance matrix can be written as $H_{A^*} = (kJ_n - A^*(A^*)^T)/2$, where J_n is an $n \times n$ matrix of unit elements. We then have the following second corollary to Theorem 5.1.

Corollary 5.2. *Designs D_1 and D_2 are isomorphic iff there exists an $n \times n$ row permutation matrix R and a column permutation matrix C such that, for every $q = 1$, $2, \ldots, k$, $A^*[q](A^*[q])^T$=$R(BC)^*[q]((BC)^*[q])^T R^T$.*

Theorem 5.1 and its corollaries simplify checking isomorphism of two-level folded over non-orthogonal designs to checking isomorphism of the original non-orthogonal designs that were folded over, yielding computational advantages. There is, however, still some thought necessary as to how to use Theorem 5.1 to establish the isomorphism of two folded over non-orthogonal designs D_1 and D_2, say. To do so, we need to determine L_1, R, C and L_2 such that $A = L_1RBCL_2$. Once the permutation matrices R and C are known, L_1 and L_2 are determined, since the negative entries on the diagonal of L_1 correspond to the elements in the first column of RBC that differ in sign from the corresponding elements in the first column of A. Likewise, the negative entries on the diagonal of L_2 correspond to the elements in the first row of RBC that differ in sign from the corresponding elements in the first row of A. Thus, the issue comes down to how to find R and C. Unfortunately, Theorem 5.1 is not enough to provide R and C. The reason is that the Hamming distance matrix of B^* is not equivalent to that of $(BC)^*$. The conventional strategy with regard to searching for possible R and C is to find R before C (see Clark and Dean, 2001; Lin and Sitter, 2008). In order to find R, we use the following procedure. We first search for the possible row permutation matrices for D_1 and D_2 using the algorithm presented in Clark and Dean (2001). Let $\tilde{R}$ be a $2n \times 2n$ possible row permutation matrix for D_1 and D_2, then $R = \tilde{R}[1:n, 1:n] + \tilde{R}[1:n, (n+1):2n]$ will be the possible row permutation matrix for A and B. For each possible row permutation matrix R, we basically take advantage of Corollary 5.2. Noting that

$$(BC)^*[q] = \begin{cases} -|B[c_q] + B[c_{q+1}]| + 1, & q < k; \\ -|B[c_1] + B[c_k]| + 1, & q = k, \end{cases}$$

we seek $c_1, c_k, c_{k-1}, \ldots, c_2$ sequentially by testing $A^*[q](A^*[q])^T{=}R(BC)^*[q]((BC)^*[q])^T R^T$. If there is no R and C satisfying Corollary 5.2, the two designs are non-isomorphic.

This new isomorphism check has some computational advantages over directly applying the Clark and Dean (2001) isomorphism check. These advantages are modest for the designs tabulated here, but become progressively greater for larger designs. To demonstrate, we perform a small numerical evaluation. For various n (number of runs) and k (number of factors) we randomly choose a set of $B = 1,500$ non-orthogonal designs with $n/2$ rows and k columns, $X_1, \ldots, X_B$, by independently generating Ber(1/2) random variables for each element. For each X_j we generate another design which is isomorphic to it and one that is not. To generate the design that is isomorphic to X_j, we randomly generate matrices R, C, L_1 and L_2 to get isomorphic design $X_j^T = L_1 R X_j C L_2$. To generate the design which is non-isomorphic to X_j we merely randomly generate non-orthogonal designs until we obtain one that is non-isomorphic to X_j. We then apply the proposed isomorphism check and the Clark and Dean isomorphism check to each pair of isomorphic designs and to each pair of non-isomorphic designs. Table 5.3 compares the speed (in seconds) of our proposed adaptation over directly applying Clark and Dean's isomorphism check to the fold-over design for isomorphic pairs and for non-isomorphic pairs, for some of the tabulated cases in the next section ($n = 20$ and 22) and for some larger cases. The table gives the average time (ET), the relative average time (RT=[ET$_{new}$-ET$_{CD}$]/ET$_{new}$, where ET$_{new}$ refers to the proposed method and ET$_{CD}$ to Clark and Dean's), and the 5th and 95th percentile of the relative times (5%, 95%). As can be seen in the first 6 rows of Table 5.3, when comparing non-isomorphic designs applying Clark and Dean isomorphism check directly is better, but in these cases both are extremely fast, while when comparing isomorphic designs, the gains of the proposed isomorphism check are greater, though still modest for these small designs. One should note, however, that there are many more comparisons necessary between isomorphic designs than between non-isomorphic designs. Overall, in the searches performed in the next section for $n = 16 - 24$ the relative gains were around 35%. In the last three rows of Table 5.3, we illustrate that the gains become much more dramatic as n and k become larger. Looking at the isomorphic cases, we can see that the proposed algorithm outperforms the Clark and Dean algorithm most of the time and by very large amounts. This bodes well for use

Table 5.3: Comparison between proposed and Clark and Dean's isomorphism check for large designs

		Isomorphic					Non-Isomorphic				
n	k	RT	5%	95%	ET$_{CD}$	ET$_{new}$	RT	5%	95%	ET$_{CD}$	ET$_{new}$
20	6	0.678	0.285	1.192	0.003	0.002	1.509	1.280	2.168	0.0001	0.0002
20	8	0.622	0.243	1.134	0.004	0.002	1.362	1.313	1.626	0.0002	0.0003
20	10	0.569	0.211	1.037	0.007	0.004	1.274	1.259	1.328	0.0002	0.0003
22	7	0.623	0.249	0.115	0.004	0.002	1.323	1.254	1.538	0.0002	0.0002
22	9	0.555	0.212	0.999	0.007	0.003	1.315	1.239	1.597	0.0002	0.0003
22	11	0.533	0.192	0.960	0.010	0.003	1.289	1.243	1.434	0.0003	0.0003
60	20	0.348	0.128	0.607	0.448	0.152	1.068	1.049	1.088	0.004	0.004
100	50	0.272	0.108	0.454	15.65	4.30	1.033	1.031	1.035	0.023	0.024
200	80	0.231	0.092	0.371	283.5	61.67	1.004	1.003	1.004	0.145	0.145

*Times given in seconds

in searching for large folded over non-orthogonal designs.

5.5 Obtaining minimum MDS-aberration designs

We are able to perform an exhaustive search using the isomorphism check of the previous section for all cases with n =10, 12, 14, 16 and 18 for which we obtain all non-isomorphic designs. We are also able to obtain all non-isomorphic designs for $n = 20$ with $k \leq 8$, $n = 22$ with $k \leq 6$, and $n = 24$ for all $5 \leq k \leq 12$ restricting to MDS(1), ..., MDS(5) all equal to zero. For $n = 20$ with $k = 9$, and 10, and for $n = 22$, $7 \leq k \leq 11$ we use the following algorithm.

1. Start with the largest k for which an exhaustive search was possible. Order the obtained designs on the basis of their MDS sequences.

2. Next search for the best designs for the same number of runs and one additional factor. To do this, perform a comprehensive search of designs that can be formed by adding one additional column to the best designs identified in Step 1,

keeping only the set of non-isomorphic designs. The logic is that if we take any $k-1$ columns from a k-factor design then the MDS for the $(k-1)$-factor design are included in the MDS for the k-factor design. Only the best 50 designs are retained.

3. Repeat Step 2 until the maximum number of factors ($k = n/2$) is reached.

This algorithm uses an idea similar to those in Loeppky, Sitter and Tang (2007) which were developed in a different context. Although we cannot guarantee that we have found the best possible design in each case, we are confident that the designs presented are among the best possible.

In Tables E.1-E.13 in Appendix E, we present the non-isomorphic minimum MDS-aberration designs obtained for each combination of k and n. In those tables, a design run $i_1 i_2 \cdots i_p$ represents a run whose i_jth setting is 1, $j = 1, \ldots, p$, and remaining settings are -1. For example, consider Table E.1, in the case of $k = 4$, a design run 12 designates a level setting (1, 1, -1, -1). There are a number of interesting aspects to the designs presented in Tables E.1-E.13:

1. There is only one design for 5 factors in 10 runs and one design for 6 factors in 12 runs that have MDS-resolution 4, and in each case these are the Margolin (1969) designs which were investigated in Miller and Sitter (2005).

2. There are 3 designs for 7 factors in 14 runs that have MDS-resolution 4 and the Margolin (1969) design is the third best in terms of both MDS-aberration and ME-efficiency.

3. For 5 factors in 16 runs, the regular FF design defined by selecting all of the runs for which the 5-factor interaction is at the +1 (or -1) level has resolution V. Although it is not a folded over design it has all ME's and 2FI's orthogonal to each other. Therefore it performs better, both for estimating ME's and for separating 2FI's, than the best fold-over designs.

Chapter 6

Conclusions and Future Research

In this thesis, we have developed methodologies for designing both computer experiments and physical experiments. Computer experiments provide a fresh and powerful approach to helping scientists understand their complex physical processes. The underlying physical mechanism in a computer experiment is represented and implemented by a computer code, which produces the response. The absence of random errors in the response necessitates new approaches to the design and analysis of experiments. Space-filling designs such as Latin hypercubes, maximin distance designs and uniform designs are commonly used to select the settings of input variables to run the computer code. This thesis studies Latin hypercube designs. Under this topic, four pieces of work are accomplished. The first is the development of a new method for constructing Latin hypercubes. The method offers new insights into the structure of Latin hypercubes. It is simple yet powerful because it allows large Latin hypercubes to be constructed using small Latin hypercubes. In addition, it has some interesting and attractive features. First, orthogonality or near orthogonality of small Latin hypercubes is carried over to large Latin hypercubes, which allows us to completely solve the problem of constructing orthogonal Latin hypercubes in terms of available run sizes. The method produces designs that are capable of entertaining more orthogonal factors than the existing methods. Second, the method can be adapted to construct cascading Latin hypercubes that provide local design points to enhance the estimation of correlation parameters (Handcock, 1991). The second piece of work is that we have established the existence of orthogonal Latin hypercubes in terms of run sizes. When orthogonal Latin hypercubes do not exist, the lower bound on the correlations is useful for both theoretical construction and computer search of the best nearly orthogonal Latin hypercubes. We have also proposed an adapted algorithm, which allows us to efficiently obtain small orthogonal and nearly orthogonal Latin hypercubes. Although the algorithm is only applied to Latin hypercubes in the thesis, it can also be used for seeking s-level designs ($2 \leq s \leq n$) or mixed-level designs. The above three pieces of work constitute Chapter 2. The fourth piece of work was presented in Chapter 3, in which we introduced and studied two generalizations of our basic method. We then exemplified that the generalizations provide designs with better projection properties.

Chapters 4 and 5 form the second topic of this thesis. They dealt with designs for physical experiments. We focus on two types of designs, two-level nonregular designs and two-level folded-over non-orthogonal designs. In spite of the important progress in the research of nonregular designs during the last decade (Xu and Wong, 2007 and the references therein), construction of minimum G and G_2-aberration designs remains largely unsolved. We have made in Chapter 4 another serious attempt in this direction. Based on the structures of designs in Bingham, Sitter and Tang (2008), we have developed a computational algorithm for searching for minimum G and G_2-aberration designs. Our method is applicable as long as the run size is a multiple of eight; in contrast, the method of Xu and Wong (2007) applies only when the run size is a power of two. Results from the application of the algorithm to designs of 24, 32 and 40 runs are obtained and presented in the thesis. Two-level folded-over non-orthogonal designs were demonstrated to be useful in screening experiments in Miller and Sitter

(2005). We here proposed two criteria, MDS-resolution and MDS-aberration, to assess and compare such designs. Obtaining a catalogue of good folded-over non-orthogonal designs is of practical interest. To this end, we proposed an isomorphism check to determine whether or not two fold-over designs are isomorphic. The isomorphism check was then demonstrated to have computational advantages through a numerical evaluation.

Next, we discuss some future work in the following five directions.

More on the proposed method in Chapter 2

The basic method and its generalizations construct large Latin hypercubes using small Latin hypercubes B and C. An obvious question is what the resulting design looks like if B, C, or both are not Latin hypercubes. In fact, we can show that when both B and C are supersaturated designs, the methods will produce multi-level supersaturated designs. A supersaturated design is a factorial design with n runs and m factors with $m > n - 1$. It can save considerable cost in situations in which the number of active factors is very small compared to the number of factors. A problem worthy of further study is whether and when supersaturated designs produced in this way have better statistical properties than the existing supersaturated designs. The examples that we have looked at show that this study is promising. Another future work is to investigate the possibility of adapting the methods to construct other space-filling designs including maximin distance designs and uniform designs. More generally, we can view the above problems as an inverse problem - determining the settings of A, B, C, D and γ for some given design L.

Cascading Latin hypercubes

Intuitively, the local points in cascading Latin hypercube designs can help provide a more accurate estimation of correlation parameters. Hence, presumably such designs will be useful in identifying important factors in the initial stage of experimentation. Handcock (1991) conducted a simulation study to demonstrate the usefulness of cascading Latin hypercubes. His simulation study offered some important insights into the potential use of such Latin hypercubes. However, the simulation study is rather limited for the following reasons: (a) only additive Gaussian process stochastic models with a Matern correlation function was considered, (b) a small run size $n = 27$ was used. A simulation study incorporating diverse models and designs of large run sizes would be beneficial to probe the further value of cascading Latin hypercubes in the context of screening experiments. This is part of our future research plan.

Designs with high projectivity

In Chapter 1, we have mentioned that one research problem in the designs for computer experiments is obtaining space-filling designs with good projection properties. Such designs are important for factor screening. In particular, those with high projectivity are desirable in practice because of the complexity of computer models. Randomized orthogonal arrays (OA's) and OA-based Latin hypercubes provide partial solutions as OA's exist only for certain run sizes. Constructing space-filling designs with high projectivity is a challenging topic and is part of our future work.

Two-level fractional factorial designs

The proposed method was applied to construct designs of 24, 32 and 40 runs. For designs of 40 runs, we have not considered the row permutations of designs D_j's. In the future, we will include row permutations of D_j's. In addition, we will continue to provide catalogues of good designs of larger run sizes. As the run size grows, the complexity and computational burden may increase considerably. Investigations on the different efficient algorithms are thus necessary. Global optimization algorithms such as genetic algorithms and simulated annealing may be useful.

Folded over non-orthogonal designs

We note that two recent papers, Bingham and Chipman (2007) and Jones, Li, Nachtsheim, and Ye (2007), presented criteria that directly evaluate the ability of a design to discriminate between competing models. These criteria could be applied to the scenarios considered in this thesis and it would be very interesting to see how the optimal designs under these criteria compare with those found using the MDS-aberration criteria. Such a comparison would require a prohibitive amount of computing since for all of the criteria involved finding an optimal design is computationally intensive. Thus we have left such a comparison for future research.

Appendix A

A 32 × 12 orthogonal Latin hypercube

Let L be a 32 × 12 orthogonal LHD. The first 16 rows of L are

−31	21	25	−19	23	27	−27	23	−25	19	−31	21
−29	17	17	29	−23	−27	27	−23	−17	−29	−29	17
−27	23	−23	−27	29	−17	−17	−29	25	−19	31	−21
−25	19	−31	21	−29	17	17	29	17	29	29	−17
−23	−27	27	−23	27	−23	23	27	21	31	−19	−25
−21	−31	19	25	−27	23	−23	−27	29	−17	−17	−29
−19	−25	−21	−31	17	29	29	−17	−21	−31	19	25
−17	−29	−29	17	−17	−29	−29	17	−29	17	17	29
17	29	29	−17	−25	19	−31	21	27	−23	23	27
19	25	21	31	25	−19	31	−21	19	25	21	31
21	31	−19	−25	−19	−25	−21	−31	−27	23	−23	−27
23	27	−27	23	19	25	21	31	−19	−25	−21	−31
25	−19	31	−21	−21	−31	19	25	−23	−27	27	−23
27	−23	23	27	21	31	−19	−25	−31	21	25	−19
29	−17	−17	−29	−31	21	25	−19	23	27	−27	23
31	−21	−25	19	31	−21	−25	19	31	−21	−25	19

The remaining 16 rows of L are

1	−11	−7	13	−9	−5	5	−9	7	−13	1	−11
3	−15	−15	−3	9	5	−5	9	15	3	3	−15
5	−9	9	5	−3	15	15	3	−7	13	−1	11
7	−13	1	−11	3	−15	−15	−3	−15	−3	−3	15
9	5	−5	9	−5	9	−9	−5	−11	−1	13	7
11	1	−13	−7	5	−9	9	5	−3	15	15	3
13	7	11	1	−15	−3	−3	15	11	1	−13	−7
15	3	3	−15	15	3	3	−15	3	−15	−15	−3
−15	−3	−3	15	7	−13	1	−11	−5	9	−9	−5
−13	−7	−11	−1	−7	13	−1	11	−13	−7	−11	−1
−11	−1	13	7	13	7	11	1	5	−9	9	5
−9	−5	5	−9	−13	−7	−11	−1	13	7	11	1
−7	13	−1	11	11	1	−13	−7	9	5	−5	9
−5	9	−9	−5	−11	−1	13	7	1	−11	−7	13
−3	15	15	3	1	−11	−7	13	−9	−5	5	−9
−1	11	7	−13	−1	11	7	−13	−1	11	7	−13

Appendix B

Top 24-run two-level designs

Table B.1: Top 24-run designs based on minimum G-aberration for $3 \leq m \leq 6$

m	Ab.	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(24 16 8 0)	R	m_{21}	m_{22}
3	**	[(0 0 0 1)$_3$, -, -]	4	0	3
3	-	[(0 0 1 0)$_3$, -, -]	3.67	1	2
3	-	[(0 1 0 0)$_3$, -, -]	3.33	1	2
4	**	[(0 0 0 4)$_3$,(0 0 1 0)$_4$, -]	4.67	0	4
4	-	[(0 0 0 4)$_3$,(1 0 0 0)$_4$, -]	4	2	2
4	-	[(0 0 1 3)$_3$,(0 0 0 1)$_4$, -]	3.67	1	3
5	**	[(0 0 0 10)$_3$,(0 0 5 0)$_4$,(0 0 0 1)$_5$]	4.67	0	5
5	-	[(0 0 1 9)$_3$,(0 0 3 2)$_4$,(0 0 1 0)$_5$]	3.67	3	2
5	-	[(0 0 2 8)$_3$,(0 0 1 4)$_4$,(0 0 0 1)$_5$]	3.67	1	4
6	**	[(0 0 0 20)$_3$,(0 0 15 0)$_4$,(0 0 0 6)$_5$]	4.67	0	6
6	-	[(0 0 0 20)$_3$,(0 0 15 0)$_4$,(0 0 0 6)$_5$]	4.67	0	6
6	-	[(0 0 4 16)$_3$,(0 0 5 10)$_4$,(0 1 2 3)$_5$]	3.67	2	4

Table B.2: Top 24-run designs based on minimum G-aberration for $7 \leq m \leq 14$

m	Ab.	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(24 16 8 0)	R	m_{21}	m_{22}
7	**	[(0 0 0 35)$_3$,(0 0 35 0)$_4$,(0 0 0 21)$_5$]	4.67	0	7
7	-	[(0 0 6 29)$_3$,(0 0 15 20)$_4$,(0 3 6 12)$_5$]	3.67	1	6
7	-	[(0 0 7 28)$_3$,(0 0 15 20)$_4$,(0 2 8 11)$_5$]	3.67	1	6
8	**	[(0 0 0 56)$_3$,(0 0 70 0)$_4$,(0 0 0 56)$_5$]	4.67	0	8
8	-	[(0 0 13 43)$_3$,(0 3 21 46)$_4$,(0 5 14 37)$_5$]	3.67	2	6
8	-	[(0 0 13 43)$_3$,(0 4 21 45)$_4$,(0 4 10 42)$_5$]	3.67	3	5
9	**	[(0 0 0 84)$_3$,(0 0 126 0)$_4$,(0 0 0 126)$_5$]	4.67	0	9
9	-	[(0 0 20 64)$_3$,(0 9 36 81)$_4$,(0 6 28 92)$_5$]	3.67	4	5
9	-	[(0 0 20 648)$_3$,(1 7 35 83)$_4$,(0 8 20 98)$_5$]	3.67	4	5
10	**	[(0 0 0 120)$_3$,(0 0 210 0)$_4$,(0 0 0 252)$_5$]	4.67	0	10
10	-	[(0 0 28 92)$_3$,(0 0 126 84)$_4$,(0 10 56 186)$_5$]	3.67	1	9
10	-	[(0 0 32 88)$_3$,(2 12 56 140)$_4$,(0 8 64 180)$_5$]	3.67	5	5
11	**	[(0 0 0 165)$_3$,(0 0 330 0)$_4$,(0 0 0 462)$_5$]	4.67	0	11
11	-	[(0 0 36 129)$_3$,(0 0 210 120)$_4$,(0 18 84 360)$_5$]	3.67	1	10
11	-	[(0 0 45 120)$_3$,(0 0 210 120)$_4$,(0 30 0 432)$_5$]	3.67	1	10
12	**	[(0 0 0,220)$_3$,(0 0 495 0)$_4$,(0 0 0 792)$_5$]	4.67	0	12
12	-	[(0 0 45 175)$_3$,(0 0 330 165)$_4$,(0 30 120 642)$_5$]	3.67	1	11
12	-	[(0 0 72 148)$_3$,(0 0 255 240)$_4$,(0 36 168 588)$_5$]	3.67	2	10
13	***	[(0 0 90 196)$_3$,(1 0 366 348)$_4$,(0 60 240 987)$_5$]	3.67	2	11
13	-	[(0 0 109 177)$_3$,(3 0 294 418)$_4$,(0 57 273 957)$_5$]	3.67	3	10
13	-	[(0 0 110 176)$_3$,(4 18 213 480)$_4$,(0 42 320 925)$_5$]	3.67	8	5
14	*	[(0 0 136 228)$_3$,(3 0 438 560)$_4$,(0 94 384 524)$_5$]	3.67	3	11
14	-	[(0 0 140 224)$_3$,(21 0 280 700)$_4$,(0 48 560 1394)$_5$]	3.67	7	7
14	-	[(0 0 146 218)$_3$,(15 0 310 676)$_4$,(0 56 510 1436)$_5$]	3.67	6	8

Table B.3: Top 24-run designs based on minimum G-aberration for $15 \leq m \leq 23$

m	Ab.	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(24 16 8 0)	R	m_{21}	m_{22}
15	***	[(0 0 182 273)$_3$,(21 0 420 9240)$_4$,(0 88 770 2145)$_5$]	3.67	7	8
15	-	[(0 0 184 271)$_3$,(6 0 547 812)$_4$,(0 136 576 2291)$_5$]	3.67	4	11
15	-	[(0 0 188 267)$_3$,(15 0 456 894)$_4$,(0 102 696 2205)$_5$]	3.67	6	9
16	***	[(0 0 224 336)$_3$,(28 0 560 1232)$_4$,(0 128 1120 3120)$_5$]	3.67	8	8
16	-	[(0 0 231 329)$_3$,(21 0 602 1197)$_4$,(0 148 1022 3198)$_5$]	3.67	7	9
16	-	[(0 0 235 325)$_3$,(10 0 695 1115)$_4$,(0 190 840 3338)$_5$]	3.67	5	11
17	***	[(0 0 280 400)$_3$,(28 0 784 1568)$_4$,(0 208 1456 4524)$_5$]	3.67	8	9
17	-	[(0 0 287 393)$_3$,(21 0 833 1526)$_4$,(0 234 1323 4631)$_5$]	3.67	7	10
17	-	[(0 0 288 392)$_3$,(21 0 833 1526)$_4$,(0 234 1316 4638)$_5$]	3.67	10	7
18	***	[(0 0 336 480)$_3$,(36 0 1008 2016)$_4$,(0 288 2016 6264)$_5$]	3.67	9	9
18	-	[(0 0 344 472)$_3$,(28 0 1064 1968)$_4$,(0 320 1848 6400)$_5$]	3.67	8	10
18	-	[(0 0 350 466)$_3$,(21 0 1121 1918)$_4$,(0 353 1680 6535)$_5$]	3.67	7	11
19	***	[(0 0 408 561)$_3$,(36 0 1344 2496)$_4$,(0 432 2520 8676)$_5$]	3.67	9	10
19	-	[(0 0 416 553)$_3$,(28 0 1408 2440)$_4$,(0 472 2304 8852)$_5$]	3.67	8	11
19	-	[(0 0 417 552)$_3$,(28 0 1408 2440)$_4$,(0 472 2296 8860)$_5$]	3.67	11	8
20	**	[(0 0 480 660)$_3$,(45 0 1680 3120)$_4$,(0 576 3360 11568)$_5$]	3.67	10	10
20	-	[(0 0 489 651)$_3$,(36 0 1752 3057)$_4$,(0 624 3096 11784)$_5$]	3.67	9	11
20	-	[(0 18 417 705)$_3$,(18 90 1554 3183)$_4$,(0 582 3264 11658)$_5$]	3.33	11	9
21	***	[(0 0 570 760)$_3$,(45 0 2160 3780)$_4$,(0 816 4080 15453)$_5$]	3.67	10	11
21	-	[(0 24 474 832)$_3$,(21 120 1896 3948)$_4$,(0 744 4368 15237)$_5$]	3.33	11	10
21	-	[(0 25 470 835)$_3$,(20 125 1885 3955)$_4$,(0 716 4480 15153)$_5$]	3.33	10	11
22	**	[(0 0 660 880)$_3$,(55 0 2640 4620)$_4$,(0 1056 5280 19998)$_5$]	3.67	11	11
22	-	[(0 30 540 970)$_3$,(25 150 2310 4830)$_4$,(0 936 5760 19638)$_5$]	3.33	11	11
22	-	[(0 40 500 1000)$_3$,(15 200 2200 4900)$_4$,(0 896 5920 19518)$_5$]	3.33	11	11
23	***	[(0 66 495 1210)$_3$,(0 330 2475 6050)$_4$,(0 1056 7920 24673)$_5$]	3.33	11	12
23	-	[(1 60 510 1200)$_3$,(5 300 2550 6000)$_4$,(0 1056 7920 24673)$_5$]	3	11	12
23	-	[(2 54 525 1190)$_3$,(10 270 2625 5950)$_4$,(0 1056 7920 24673)$_5$]	3	11	12

Table B.4: Top 24-run designs based on minimum G_2-aberration for $3 \le m \le 23$

m	WLP=(B_3 B_4 B_5)	m_{21}	m_{22}	m	WLP=(B_3 B_4 B_5)	m_{21}	m_{22}
3	0	0	3	13	6 55 40	1	12
3	0.11	1	2	13	10 41.67 53.33	2	11
				13	11.56 37.22 53.33	6	7
4	0 0.11	0	4	14	12 61 80	2	12
4	0 1	2	2	14	15.11 51.67 84.44	3	11
4	0.11 0	1	3	14	15.56 50.11 84.22	7	7
5	0 0.56 0	0	5	15	18.11 73 125.33	3	12
5	0.11 0.33 0.11	3	2	15	20.22 66.33 126	8	7
5	0.22 0.11 0	1	4	15	20.22 66.77 125.56	8	7
6	0 1.67 0	0	6	16	24.44 91.11 181.33	4	12
6[a]	0 1.67 0	0	6	16	24.89 90.22 181.33	8	8
6	0.44 1.67 0	1	5	16	25.67 87.44 179.78	9	7
7	0 3.69 0	0	7	17	31.11 115.11 254.22	8	9
7	0.67 1.67 2	1	6	17	31.11 115.56 253.33	5	12
7	0.78 1.67 1.78	1	6	17	31.11 115.56 253.78	5	12
8	0 7.78 0	0	8	18	37.33 148 352	9	9
8	1.44 3.67 3.78	2	6	18	38.22 146.22 347.56	8	10
8	3.89 3.56	1	7	18	38.33 146.67 346.67	6	12
9	0 14 0	0	9	19	45.33 185.33 472	9	10
9	2.22 7.78 6.22	1	8	19	45.78 184.89 468.89	9	10
9	2.22 7.78 6.67	1	8	19	45.89 184.89 468	7	12
10	0 23.33 0	0	10	20	53.33 231.67 629.33	10	10
10	3.11 14 10.67	1	9	20	54.22 230.78 622.22	8	12
10	3.33 14 10	1	9	20	54.33 230.67 621.33	9	11
11	0 36.7 0	0	11	21	63.33 285 816	9	12
11	4 23.33 17.33	1	10				
11	4.11 23.33 16.89	1	10	22	73.33 348.33 1056	10	12
12	0 55 0	0	12	23	84.33 421.67 1349.33	11	12
12	5 36.67 26.67	1	11				
12	8 27.44 34.67	2	10				

[a] This design has different B_6 from the first design.

Appendix C

Top 32-run two-level designs

Table C.1: Top 32-run designs for $6 \leq m \leq 10$

m	R	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(32 24 16 8 0)	WLP=(B_3,B_4,B_5)	Comparison
6	6	$[(0\ 0\ 0\ 0\ 20)_3,(0\ 0\ 0\ 0\ 15)_4,(0\ 0\ 0\ 0\ 6)_5]$	0 0 0	
6	5	$[(0\ 0\ 0\ 0\ 20)_3,(0\ 0\ 0\ 0\ 15)_4,(1\ 0\ 0\ 0.5)_5]$	0 0 1	
6	4.5	$[(0\ 0\ 0\ 0\ 20)_3,(0\ 0\ 1\ 0\ 14)_4,(0\ 0\ 2\ 0\ 4)_5]$	0 0.25 0.5	
7	4.5	$[(0\ 0\ 0\ 0\ 35)_3,(0\ 0\ 4\ 0\ 31)_4,(1\ 0\ 4\ 0\ 16)_5]$	0 1 2	G**, G_2**
7	4.5	$[(0\ 0\ 0\ 0\ 35)_3,(0\ 0\ 6\ 0\ 29)_4,(0\ 0\ 6\ 0\ 15)_5]$	0 1.5 1	
7	4.5	$[(0\ 0\ 0\ 0\ 35)_3,(0\ 0\ 8\ 0\ 27)_4,(0\ 0\ 0\ 0\ 21)_5]$	0 1.5 1.5	
8	4.5	$[(0\ 0\ 0\ 0\ 56)_3,(0\ 0\ 12\ 0\ 58)_4,(1\ 0\ 12\ 0\ 43)_5]$	0 3 4	G**, G_2**
8	4.5	$[(0\ 0\ 0\ 0\ 56)_3,(0\ 0\ 14\ 0\ 56)_4,(0\ 0\ 14\ 0\ 42)_5]$	0 3.5 3.5	
8	4.5	$[(0\ 0\ 0\ 0\ 56)_3,(0\ 0\ 20\ 0\ 50)_4,(0\ 0\ 0\ 0\ 56)_5]$	0 4 2	
9	4.5	$[(0\ 0\ 0\ 0\ 84)_3,(0\ 0\ 24\ 0\ 102)_4,(2\ 0\ 24\ 0\ 100)_5]$	0 6 8	G**, G_2**
9	4.5	$[(0\ 0\ 0\ 0\ 84)_3,(0\ 0\ 28\ 0\ 98)_4,(0\ 0\ 28\ 0\ 98)_5]$	0 7 7	
9	4.5	$[(0\ 0\ 0\ 0\ 84)_3,(0\ 0\ 42\ 0\ 84)_4,(0\ 0\ 0\ 0\ 126)_5]$	0 8 4	
10	4	$[(0\ 0\ 0\ 0\ 120)_3,(1\ 0\ 62\ 0\ 147)_4,(0\ 0\ 0\ 0\ 252)_5]$	0 10 16	G*, G_2*
10	4	$[(0\ 0\ 0\ 0\ 120)_3,(2\ 0\ 56\ 0\ 152)_4,(0\ 0\ 0\ 0\ 252)_5]$	0 15 0	G*, G_2*
10	4	$[(0\ 0\ 0\ 0\ 120)_3,(2\ 0\ 58\ 0\ 150)_4,(0\ 0\ 0\ 0\ 252)_5]$	0 15.75 0	G*, G_2**

Table C.2: Top 32-run designs for $11 \leq m \leq 17$

m	R	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(32 24 16 8 0)	WLP=(B_3,B_4,B_5)	Comparison
11	4	[(0 0 0 0 165)$_3$,(3 0 90 0 237)$_4$,(0 0 0 0 462)$_5$]	0 25 0	G^*, G_2^{**}
11	4	[(0 0 0 0 165)$_3$,(4 0 84 0 242)$_4$,(0 0 0 0 462)$_5$]	0 25.5 0	G^*
11	4	[(0 0 0 0 165)$_3$,(4 0 86 0 240)$_4$,(0 0 0 0 462)$_5$]	0 26 0	G^*
12	4	[(0 0 0 0 220)$_3$,(6 0 128 0 361)$_4$,(0 0 0 0 792)$_5$]	0 38 0	G^*, G_2^{**}
12	4	[(0 0 0 0 220)$_3$,(6 0 129 0 360)$_4$,(0 0 0 0 792)$_5$]	0 38.25 0	G^*
12	4	[(0 0 0 0 220)$_3$,(8 0 120 0 367)$_4$,(0 0 0 0 792)$_5$]	0 39 0	G^*
13	4	[(0 0 0 0 286)$_3$,(10 0 180 0 525)$_4$,(0 0 0 0 1287)$_5$]	0 55 0	G^*, G_2^{**}
13	4	[(0 0 0 0 286)$_3$,(15 0 160 0 540)$_4$,(0 0 0 0 1287)$_5$]	2 47 16	G^*
13	4	[(0 0 0 0 286)$_3$,(16 0 156 0 543)$_4$,(0 0 0 0 1287)$_5$]	3 43 22	
14	4	[(0 0 0 0 364)$_3$,(14 0 252 0 735)$_4$,(0 0 0 0 2002)$_5$]	0 77 0	G^*, G_2^{**}
14	4	[(0 0 0 0 364)$_3$,(15 0 248 0 738)$_4$,(0 0 0 0 2002)$_5$]	4 60 32	G^*
14	4	[(0 0 0 0 364)$_3$,(17 0 240 0 744)$_4$,(0 0 0 0 2002)$_5$]	4 61 32	G^*
15	4	[(0 0 0 0 455)$_3$,(21 0 336 0 1008)$_4$,(0 0 0 0 3003)$_5$]	0 105 0	G^*, G_2^{**}
15	4	[(0 0 0 0 455)$_3$,(57 0 192 0 1116)$_4$,(0 0 0 0 3003)$_5$]	6 77 62	
15	4	[(0 0 0 0 455)$_3$,(105 0 0 0 1260)$_4$,(0 0 0 0 3003)$_5$]	6 81 50	
16	4	[(0 0 0 0 560)$_3$,(28 0 448 0 1344)$_4$,(0 0 0 0 4368)$_5$]	0 140 0	G^*, G_2^{**}
16	4	[(0 0 0 0 560)$_3$,(76 0 256 0 1488)$_4$,(0 0 0 0 4368)$_5$]	7 105 84	
16	4	[(0 0 0 0 560)$_3$,(140 0 0 0 1680)$_4$,(0 0 0 0 4368)$_5$]	8 108 72	
17	3.5	[(0 0 16 64 600)$_3$,(28 0 448 0 1904)$_4$,(0 0 224 896 5068)$_5$]	8 140 112	G^*, G_2^{**}
17	3.5	[(0 0 16 64 600)$_3$,(28 0 448 0 1904)$_4$,(0 16 192 880 5100)$_5$]	11 128 134	G^*
17	3.5	[(0 0 16 64 600)$_3$,(28 0 448 0 1904)$_4$,(0 32 160 864 5132)$_5$]	12 128 132G^*	

Table C.3: Top 32-run designs for $18 \leq m \leq 24$

m	R	CFV(D)=[$F_3(D)$, $F_4(D)$, $F_5(D)$] $J_k(s)$=(32 24 16 8 0)	WLP=(B_3,B_4,B_5)	Comparison
18	3.5	[(0 0 32 128 656)$_3$,(28 0 464 64 2504)$_4$,(0 0 448 1792 6328)$_5$]	16 148 224	G*, G_2**
18	3.5	[(0 0 32 128 656)$_3$,(28 0 464 64 2504)$_4$,(0 16 416 1776 6360)$_5$]	18 140 236	G*
18	3.5	[(0 0 32 128 656)$_3$,(28 0 464 64 2504)$_4$,(0 32 384 1760 6392)$_5$]	19 136 242	G*
19	3.5	[(0 0 48 192 729)$_3$,(140 0 48 192 3496)$_4$,(0 0 688 2752 8188)$_5$]	24 164 344	G*, G_2**
19	3.5	[(0 0 48 192 729)$_3$,(140 0 48 192 3496)$_4$,(0 0 704 2688 8236)$_5$]	25 164 336	G*
19	3.5	[(0 0 48 192 729)$_3$,(140 0 48 192 3496)$_4$,(0 8 672 2744 8204)$_5$]	25.5 158 35.5	G*
20	3.5	[(0 0 64 256 820)$_3$,(140 0 96 384 4225)$_4$,(0 0 960 3840 10704)$_5$]	32 188 480	G*, G_2**
20	3.5	[(0 0 64 256 820)$_3$,(140 0 96 384 4225)$_4$,(0 0 976 3776 10752)$_5$]	32 189 480	G*
20	3.5	[(0 0 64 256 820)$_3$,(140 0 96 384 4225)$_4$,(0 8 944 3832 10720)$_5$]	33 189 480	G*
21	3.5	[(0 0 88 312 930)$_3$,(53 12 312 1260 4348)$_4$,(0 0 1296 5032 14021)$_5$]	40 221 640	G*, G_2**
21	3.5	[(0 0 88 316 926)$_3$,(53 28 280 1244 4380)$_4$,(0 0 1296 5000 14053)$_5$]	40.75 218.5 639.25	G*
21	3.5	[(0 0 88 316 926)$_3$,(53 40 256 1232 4404)$_4$,(0 16 1264 4984 14085)$_5$]	41 217 642	G*
22	3.5	[(0 0 112 352 1076)$_3$,(35 48 448 1376 5408)$_4$,(0 56 1576 6312 18390)$_5$]	48 263 832	G*, G_2**
22	3.5	[(0 0 112 352 1076)$_3$,(57 48 314 1540 5356)$_4$,(0 0 1696 6368 18270)$_5$]	48.5 262 828.5	G*
22	3.5	[(0 0 112 352 1076)$_3$,(57 72 266 1516 5404)$_4$,(0 0 1696 6368 18270)$_5$]	49 259 833	G*
23	3.5	[(0 0 132 439 1200)$_3$,(27 20 674 1586 6548)$_4$,(0 84 1910 8188 23467)$_5$]	56 315 1064	G*, G_2**
23	3.5	[(0 0 132 439 1200)$_3$,(27 34 670 1476 6648)$_4$,(0 96 1830 8400 23323)$_5$]	57 313 1057	G*
23	3.5	[(0 0 132 439 1200)$_3$,(27 45 602 1641 6540)$_4$,(0 94 1888 8194 23473)$_5$]	57.25 312.5 1056.25	G*
24	3.5	[(0 0 152 488 1384)$_3$,(31 44 805 1788 7958)$_4$,(0 128 2308 10608 29460)$_5$]	64 378 1344	G*, G_2**
24	3.5	[(0 0 152 490 1382)$_3$,(35 48 808 1664 8071)$_4$,(0 56 2440 10724 29284)$_5$]	65.5 375 1333.5	G*
24	3.5	[(0 0 153 485 1386)$_3$,(31 40 731 2097 7724)$_4$,(0 88 2458 10374 29584)$_5$]	66 374 1330	G*

Table C.4: Top 32-run designs for $25 \leq m \leq 31$

m	R	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(32 24 16 8 0)	WLP=(B_3,B_4,B_5)
25	3.5	[(0 0 180 537 1583)$_3$,(31 50 954 2232 9383)$_4$,(0 170 2846 13245 36869)$_5$]	76 442 1656
25	3.5	[(0 0 181 533 1586)$_3$,(31 51 870 2555 9143)$_4$,(0 96 3108 12871 37055)$_5$]	76.75 440 1650
25	3.5	[(0 0 181 537 1582)$_3$,(35 28 955 2350 9282)$_4$,(0 126 3036 12857 37111)$_5$]	77 440 1648
26	3.5	[(0 0 209 582 1809)$_3$,(36 62 1018 3062 10772)$_4$,(0 112 3895 15834 45939)$_5$]	88 518 2032
26	3.5	[(0 0 210 574 1816)$_3$,(36 32 1112 2964 10806)$_4$,(0 104 3940 15762 45974)$_5$]	88.375 517.25 2028.625
26	3.5	[(0 0 210 582 1808)$_3$,(36 44 1088 2936 10846)$_4$,(0 152 3726 16114 45788)$_5$]	88.5 517 2027.5
27	3.5	[(0 0 246 631 2048)$_3$,(36 56 1286 3458 12714)$_4$,(0 222 4510 19540 56458)$_5$]	100 606 2484
27	3.5	[(0 0 246 632 2047)$_3$,(40 28 1426 3084 12972)$_4$,(8 60 4926 19196 56540)$_5$]	100.375 605.625 2479.87
27	3.5	[(0 0 247 627 2051)$_3$,(42 54 1349 3128 12977)$_4$,(0 180 4663 19306 56581)$_5$]	100.44 605.625 2479.12
28	3.5	[(0 0 282 676 2318)$_3$,(42 56 1565 3864 14948)$_4$,(0 200 5674 23744 68662)$_5$]	112 707 3024
28	3.5	[(0 0 282 679 2315)$_3$,(42 90 1484 3879 14980)$_4$,(0 176 5672 23932 68500)$_5$]	112.44 706.56 3018.75
28	3.5	[(0 0 283 676 2317)$_3$,(42 98 1494 3766 15075)$_4$,(0 144 5686 24152 68298)$_5$]	112.750 706.250 3015
29	3	[(6 0 280 800 2568)$_3$,(43 92 1792 4420 17404)$_4$,(0 164 7072 28476 83043)$_5$]	126 819 3640
29	3	[(6 0 284 784 2580)$_3$,(43 64 1788 4688 17168)$_4$,(0 304 6740 28544 83167)$_5$]	
29	3	[(6 0 284 784 2580)$_3$,(43 68 1804 4588 17248)$_4$,(0 260 6884 28364 83247)$_5$]	
30	3	[(6 0 324 848 2882)$_3$,(43 76 2100 5348 19838)$_4$,(0 288 8272 34208 99738)$_5$]	140 945 4368
30	3	[(6 0 328 832 2894)$_3$,(43 118 1996 5386 19862)$_4$,(0 330 8236 33974 99966)$_5$]	
30	3	[(6 0 328 832 2894)$_3$,(45 70 2076 5466 19748)$_4$,(2 308 8344 33708 100144)$_5$]	
31	3	[(7 0 356 944 3188)$_3$,(53 68 2436 6156 22752)$_4$,(4 380 9984 39908 119635)$_5$]	155 1085 5208
31	3	[(7 0 368 896 3224)$_3$,(53 72 2416 6200 22724)$_4$,(4 504 9760 39688 119955)$_5$]	
31	3	[(7 0 376 864 3248)$_3$,(49 80 2440 6096 22800)$_4$,(8 304 10128 39952 119519)$_5$]	

Appendix D

Top 40-run two-level designs

Table D.1: Top 40-run designs for $6 \leq m \leq 9$

m	R	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(40 32 24 16 8 0)	WLP=$(B_3$,B_4,$B_5)$
4	4.8	[(0 0 0 0 0 4)$_3$,(0 0 0 0 1 0)$_4$,-]	0 0.04
4	4.4	[(0 0 0 0 0 4)$_3$,(0 0 1 0 0 0)$_4$,-]	0 0.36
5	4.8	[(0 0 0 0 0 10)$_3$,(0 0 0 0 5 0)$_4$,(0 0 0 0 0 1)$_5$]	0 0.2 0
5	4.4	[(0 0 0 0 0 10)$_3$,(0 0 1 0 4 0)$_4$,(0 0 0 0 0 1)$_5$]	0 0.52 0
6	4.8	[(0 0 0 0 0 20)$_3$,(0 0 0 0 15 0)$_4$,(0 0 0 0 0 6)$_5$]	0 0.6 0
6^a	4.8	[(0 0 0 0 0 20)$_3$,(0 0 0 0 15 0)$_4$,(0 0 0 0 0 6)$_5$]	0 0.6 0
7	4.8	[(0 0 0 0 0 35)$_3$,(0 0 0 0 35 0)$_4$,(0 0 0 0 0 21)$_5$]	0 1.4 0
7	4.4	[(0 0 0 0 0 35)$_3$,(0 0 1 0 34 0)$_4$,(0 0 0 0 0 21)$_5$]	0 1.72 0
8	4.4	[(0 0 0 0 0 56)$_3$,(0 0 2 0 68 0)$_4$,(0 0 0 0 0 56)$_5$]	0 3.44 0
8	4.4	[(0 0 0 0 0 56)$_3$,(0 0 3 0 67 0)$_4$,(0 0 0 0 0 56)$_5$]	0 3.76 0
9	4.4	[(0 0 0 0 0 84)$_3$,(0 0 6 0 120 0)$_4$,(0 0 0 0 0 126)$_5$]	0 6.96 0
9	4.4	[(0 0 0 0 0 84)$_3$,(0 0 7 0 119 0)$_4$,(0 0 0 0 0 126)$_5$]	0 7.28 0

a. This design has different F_6 and B_6 from the first design.

Table D.2: Top 40-run designs for $10 \leq m \leq 19$

m	R	CFV(D)=[$F_3(D), F_4(D), F_5(D)$] $J_k(s)$=(40 32 24 16 8 0)	WLP=(B_3,B_4,B_5)
10	4.4	[(0 0 0 0 0 120)$_3$,(0 0 10 0 200 0)$_4$,(0 0 0 0 0 252)$_5$]	0 11.6 0
10	4.4	[(0 0 0 0 0 120)$_3$,(0 0 11 0 199 0)$_4$,(0 0 0 0 0 252)$_5$]	0 11.92 0
11	4.4	[(0 0 0 0 0 165)$_3$,(0 0 18 0 312 0)$_4$,(0 0 0 0 0 462)$_5$]	0 18.96 0
11	4.4	[(0 0 0 0 0 165)$_3$,(0 0 19 0 311 0)$_4$,(0 0 0 0 0 462)$_5$]	0 19.28 0
12	4.4	[(0 0 0 0 0 220)$_3$,(0 0 27 0 468 0)$_4$,(0 0 0 0 0 792)$_5$]	0 28.44 0
12	4.4	[(0 0 0 0 0 220)$_3$,(0 0 28 0 467 0)$_4$,(0 0 0 0 0 792)$_5$]	0 28.76 0
13	4.4	[(0 0 0 0 0 286)$_3$,(0 0 41 0 674 0)$_4$,(0 0 0 0 0 1287)$_5$]	0 41.72 0
13	4.4	[(0 0 0 0 0 286)$_3$,(0 0 42 0 673 0)$_4$,(0 0 0 0 0 1287)$_5$]	0 42.04 0
14[a]	4.4	[(0 0 0 0 0 364)$_3$,(0 0 58 0 943 0)$_4$,(0 0 0 0 0 2002)$_5$]	0 58.60 0
14	4.4	[(0 0 0 0 0 364)$_3$,(0 0 58 0 943 0)$_4$,(0 0 0 0 0 2002)$_5$]	0 58.60 0
15	4.4	[(0 0 0 0 0 455)$_3$,(0 0 80 0 1285 0)$_4$,(0 0 0 0 0 3003)$_5$]	0 80.20 0
15	4.4	[(0 0 0 0 0 455)$_3$,(0 0 81 0 1284 0)$_4$,(0 0 0 0 0 3003)$_5$]	0 80.52 0
16	4.4	[(0 0 0 0 0 560)$_3$,(0 0 107 0 1713 0)$_4$,(0 0 0 0 0 4368)$_5$]	0 107.04 0
16	4.4	[(0 0 0 0 0 560)$_3$,(0 0 108 0 1712 0)$_4$,(0 0 0 0 0 4368)$_5$]	0 107.36 0
17	4.4	[(0 0 0 0 0 680)$_3$,(0 0 140 0 2240 0)$_4$,(0 0 0 0 0 6188)$_5$]	0 140 0
17	3.8	[(0 0 0 0 280 400)$_3$,(28 0 112 0 672 1568)$_4$,(0 0 0 384 1456 4348)$_5$]	0 107.04 0
18	4.4	[(0 0 0 0 0 816)$_3$,(0 0 180 0 2880 0)$_4$,(0 0 0 0 0 8568)$_5$]	0 180 0
18	3.8	[(0 0 0 0 336 480)$_3$,(36 0 144 0 864 2016)$_4$,(0 0 0 544 2016 6008)$_5$]	7.04 140 118.72
19	4.4	[(0 0 0 0 0 969)$_3$,(0 0 228 0 3648 0)$_4$,(0 0 0 0 0 11628)$_5$]	0 228 0
19	3.8	[(0 0 0 0 408 561)$_3$,(36 0 192 0 1152 2496)$_4$,(0 0 0 848 2520 8260)$_5$]	8 180 152

a. This design has different F_6 and B_6 from the second design.

Table D.3: Top 40-run designs for $20 \leq m \leq 29$

m	R	CFV(D)=[$F_3(D), F_4(D), F_5(D)$] $J_k(s)$=(40 32 24 16 8 0)	WLP=(B_3,B_4,B_5)
20	4.4	[(0 0 0 0 0 1140)$_3$,(0 0 285 0 4560 0)$_4$,(0 0 0 0 0 15504)$_5$]	0 285 0
20	3.8	[(0 0 0 0 480 660)$_3$,(45 0 240 0 1440 3120)$_4$,(0 0 0 1152 3360 10992)$_5$]	9 228 192
21	3.4	[(0 0 10 0 560 760)$_3$,(45 0 240 0 1920 3780)$_4$,(0 0 80 1712 4000 14557)$_5$]	10 285 240
21	3.4	[(0 0 16 0 554 760)$_3$,(45 0 192 0 1968 3780)$_4$,(0 0 120 1696 3960 14573)$_5$]	18 237 384
22	3.4	[(0 0 20 0 640 880)$_3$,(55 0 240 0 2400 4620)$_4$,(0 0 160 2272 5120 18782)$_5$]	20 295 480
22	3.4	[(0 0 20 0 650 870)$_3$,(45 0 250 0 2480 4540)$_4$,(0 0 160 2392 4730 19052)$_5$]	27.04 255 584.64
23	3.4	[(0 0 28 0 742 1001)$_3$,(55 0 260 0 3040 5500)$_4$,(0 0 228 3040 6042 24339)$_5$]	30.04 315 729.6
23	3.4	[(0 0 30 0 740 1001)$_3$,(55 0 260 0 3040 5500)$_4$,(0 0 250 3072 6020 24307)$_5$]	30.36 315 726.4
24	3.4	[(0 0 32 0 848 1144)$_3$,(66 0 312 0 3648 6600)$_4$,(0 0 288 3840 7632 30744)$_5$]	40.16 345.04 998.4
24	3.4	[(0 0 39 0 852 1133)$_3$,(55 0 288 0 3782 6501)$_4$,(0 0 328 4040 7042 31094)$_5$]	40.16 345.36 998.4
25	3.4	[(0 0 44 0 968 1288)$_3$,(66 0 344 0 4496 7744)$_4$,(0 0 408 5040 8832 38850)$_5$]	50.4 385.2 1296
25	3.4	[(0 0 44 0 968 1288)$_3$,(66 0 344 0 4496 7744)$_4$,(0 0 408 5048 8832 38842)$_5$]	50.4 385.2 1296.16
26	3.4	[(0 0 56 0 1088 1456)$_3$,(78 0 376 0 5344 9152)$_4$,(0 0 560 6240 10880 48100)$_5$]	60.8 435.92 1632.32
26	3.4	[(0 0 56 0 1088 1456)$_3$,(78 0 376 0 5344 9152)$_4$,(0 0 560 6256 10880 48084)$_5$]	60.8 435.92 1632.48
27	3.4	[(0 0 68 0 1232 1625)$_3$,(78 0 432 0 6432 10608)$_4$,(0 0 692 7904 12464 59670)$_5$]	71.4 497.36 2017.12
27	3.4	[(0 0 68 0 1232 1625)$_3$,(78 0 432 0 6432 10608)$_4$,(0 0 692 7912 12464 59662)$_5$]	71.4 497.36 2017.44
28	3.4	[(0 0 80 0 1376 1820)$_3$,(91 0 480 0 7528 12376)$_4$,(0 0 880 9616 15136 72648)$_5$]	82.24 569.72 2461.44
28	3.4	[(0 0 80 0 1376 1820)$_3$,(91 0 488 0 7520 12376)$_4$,(0 0 880 9568 15136 72696)$_5$]	82.24 570.36 2460.16
29	3.4	[(0 0 90 0 1560 2004)$_3$,(45 0 618 0 9588 13500)$_4$,(0 0 840 13392 13080 91443)$_5$]	93.36 655.80 2971.84
29	3.4	[(0 0 92 0 1546 2016)$_3$,(91 0 564 0 8900 14196)$_4$,(0 0 1024 11936 17176 88619)$_5$]	93.36 655.80 2975.04

Table D.4: Top 40-run designs for $30 \leq m \leq 39$

m	R	CFV(D)=$[F_3(D), F_4(D), F_5(D)]$ $J_k(s)$=(40 32 24 16 8 0)	WLP=(B_3,B_4,B_5)
30	3.4	[(0 0 104 0 1716 2240)$_3$,(105 0 648 0 10272 16380)$_4$,(0 0 1248 14256 20592 106410)$_5$]	104.80 753 3563.52
30	3.4	[(0 0 104 0 1730 2226)$_3$,(91 0 660 0 10442 16212)$_4$,(0 0 1168 14716 19398 107224)$_5$]	105.44 750.44 3357.12
31	3.4	[(0 0 116 0 1914 2465)$_3$,(105 0 756 0 11984 18620)$_4$,(0 0 1404 17496 23166 127845)$_5$]	118.20 857.8 4230.72
31	3.4	[(0 0 116 0 1940 2439)$_3$,(66 0 795 0 12552 18052)$_4$,(0 0 1212 18864 19140 130695)$_5$]	118.32 856.52 4231.44
32	3.4	[(0 0 128 0 2112 2720)$_3$,(120 0 864 0 13696 21280)$_4$,(0 0 1664 20736 27456 151520)$_5$]	130.56 978.88 5015.04
32	3.4	[(0 0 131 0 2124 2705)$_3$,(105 0 872 0 13898 21085)$_4$,(0 0 1600 21164 25910 152702)$_5$]	131.36 977.28 5004.8
33	3.4	[(0 0 144 0 2336 2976)$_3$,(120 0 992 0 15808 24000)$_4$,(0 0 1888 24928 30592 179928)$_5$]	145.28 1109.44 5891.84
33	3.4	[(0 0 146 0 2334 2976)$_3$,(120 0 988 0 15812 24000)$_4$,(0 0 1912 24832 30568 180024)$_5$]	145.92 1108.16 5884.16
34	3.4	[(0 0 160 0 2560 3264)$_3$,(136 0 1120 0 17920 27200)$_4$,(0 0 2240 29120 35840 211056)$_5$]	160 1256 6899.2
34	3.4	[(0 0 160 0 2576 3248)$_3$,(120 0 1136 0 18144 26976)$_4$,(0 0 2112 29824 33968 212352)$_5$]	160.64 1254.72 6890.88
35	3.4	[(0 0 176 0 2816 3553)$_3$,(136 0 1280 0 20480 30464)$_4$,(0 0 2480 34720 39680 247752)$_5$]	176 1416 8035.2
35	3.4	[(0 0 176 0 2832 3537)$_3$,(120 0 1296 0 20720 30224)$_4$,(0 0 2352 35472 37584 249224)$_5$]	176.52 1415.52 8027.36
36	3.4	[(0 0 192 0 3072 3876)$_3$,(153 0 1440 0 23040 34272)$_4$,(0 0 2880 40320 46080 287712)$_5$]	192 1593 9331.2
36	3.4	[(0 0 193 0 3088 3859)$_3$,(136 0 1456 0 23296 34017)$_4$,(0 0 2736 41120 43776 289360)$_5$]	192.64 1592.36 9320.96
37	3.4	[(0 0 210 0 3360 4200)$_3$,(153 0 1632 0 26112 38148)$_4$,(0 0 3168 47520 50688 334521)$_5$]	210 1785 10771.2
37	3	[(17 0 193 0 3088 4472)$_3$,(136 0 1649 0 26384 37876)$_4$,(0 0 4192 41120 67072 323513)$_5$]	
38	3.4	[(0 0 228 0 3648 4560)$_3$,(171 0 1824 0 29184 42636)$_4$,(0 0 3648 54720 58368 385206)$_5$]	228 1995 12403.2
38	3	[(18 0 210 0 3360 4848)$_3$,(153 0 1842 0 29472 42348)$_4$,(0 0 4800 47520 76800 372822)$_5$]	
39	3	[(19 0 228 0 3648 5244)$_3$,(171 0 2052 0 32832 47196)$_4$,(0 0 5472 54720 87552 428013)$_5$]	247 2223 14227.2

Appendix E

Top non-isomorphic MDS designs

Table E.1: Non-isomorphic 10-run MDS designs with $k \leq 5$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
4	1 12 13 14 1234	3 0 0	0.9143
5	15 12 13 14 12345	25 102 0	0.9000

Table E.2: Non-isomorphic 12-Run MDS Designs with $k \leq 6$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
4	1 12 13 123 14 1234	1 0 0	0.8889
4	1 1 123 124 134 1234	3 0 0	0.8696
4	1 1 12 13 14 1234	3 0 0	0.8000
5	15 12 13 123 14 12345	12 18 6	0.8547
5	15 1 12 13 14 12345	15 0 15	0.7778
5	15 15 12 13 14 12345	25 102 0	0.8333
6	15 126 136 123 14 123456	45 162 1411	0.8333

Table E.3: Non-isomorphic 14-run MDS designs with $k \leq 7$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
4	1 12 13 123 14 124 1234	0 0 0	0.9143
4	1 1 12 123 124 134 1234	1 0 0	0.8649
4	1 1 12 13 123 14 1234	1 0 0	0.8067
4	1 1 123 123 124 134 1234	3 0 0	0.8649
4	1 1 1 123 124 134 1234	3 0 0	0.7857
4	1 1 12 12 13 14 1234	3 0 0	0.7857
4	1 1 1 12 13 14 1234	3 0 0	0.7033
5	15 1 125 135 123 14 12345	5 4 2	0.8193
5	15 1 125 123 124 134 12345	6 4 2	0.8494
5	15 1 125 12 13 14 12345	9 0 6	0.7635
5	15 15 12 13 123 14 12345	12 18 6	0.8193
5	15 15 1 123 124 134 12345	12 18 6	0.8036
5	15 1 1 123 124 134 12345	12 18 6	0.7635
5	15 15 1 12 13 14 12345	15 0 15	0.7418
5	15 1 1 12 13 14 12345	15 0 15	0.6786
5	15 15 12 12 13 14 12345	25 102 0	0.8036
5	15 15 15 12 13 14 12345	25 102 0	0.7418
6	15 16 1256 1356 123 14 123456	15 54 246	0.7912
6	156 16 125 135 123 14 123456	24 57 294	0.8099
6	15 16 125 135 123 14 123456	27 54 172	0.7792
6	15 16 12 13 123 14 123456	33 12 214	0.7319
6	15 1 126 136 123 14 123456	36 63 150	0.7373
6	15 16 1 12 13 14 123456	45 0 150	0.6617
6	15 15 126 136 123 14 123456	45 162 1411	0.7792
7	157 167 1256 1356 123 14 1234567	60 270 2637	0.7826
7	15 16 127 137 123 14 1234567	87 219 2145	0.7153
7	15 16 17 12 13 14 1234567	105 105 2877	0.6494

Table E.4: Top 10 Non-isomorphic 16-run MDS designs with $k \leq 8$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
5	1 12 13 124 134 125 145 12345	2 1 1	0.8400
5	1 12 13 124 134 1235 145 12345	3 0 0	0.8824
5	1 12 13 14 1234 125 135 12345	3 0 0	0.8065
5	1 12 13 123 14 125 135 12345	3 2 0	0.8120
5	1 12 123 134 135 145 1245 12345	4 0 0	0.8824
5	1 12 13 123 14 124 15 12345	5 0 2	0.7798
5	1 12 13 123 14 1234 15 12345	5 0 2	0.7500
5	1 1 123 124 1234 125 1345 12345	5 4 2	0.7955
5	1 12 13 124 125 145 145 12345	5 4 2	0.7895
5	1 1 12 13 124 125 145 12345	5 4 2	0.7474
6	16 12 13 1246 1346 1256 145 123456	6 18 112	0.8036
6	16 12 13 1246 134 1256 145 123456	10 21 52	0.7826
6	16 12 13 124 1346 1256 145 123456	10 21 98	0.8015
6	16 126 13 124 134 125 145 123456	11 20 75	0.8036
6	16 12 13 1236 14 1256 135 123456	11 24 60	0.7638
6	16 12 13 1246 134 125 145 123456	12 16 83	0.7742
6	16 126 13 124 1346 125 145 123456	12 17 82	0.8182
6	16 12 13 124 1346 125 145 123456	13 16 49	0.7768
6	1 12 136 1246 134 125 1456 123456	13 18 64	0.7795
6	16 126 136 124 134 1235 145 123456	14 17 78	0.8140
7	167 127 13 1246 1346 1256 145 1234567	27 93 799	0.7778
7	167 127 13 124 1346 1256 145 1234567	28 101 779	0.7516
7	16 12 137 12467 134 1256 1457 1234567	30 90 570	0.7438
7	167 127 137 1247 1346 125 1457 1234567	36 94 694	0.7383
7	167 127 13 1246 134 1256 145 1234567	36 116 754	0.7570
7	167 127 137 1236 14 1256 135 1234567	37 108 763	0.7711
7	167 127 13 1246 1347 1256 145 1234567	37 121 851	0.7711
7	16 127 13 124 13467 1256 1457 1234567	37 139 869	0.7538
7	16 12 137 1236 14 12567 135 1234567	39 77 658	0.7271
7	16 127 123 1347 1357 145 1245 1234567	39 90 720	0.7778
8	1678 127 138 1248 1346 12568 145 12345678	60 372 3952	0.7083
8	168 128 137 12467 134 1256 14578 12345678	76 368 3876	0.7391
8	1678 127 1378 12478 1346 1258 1457 12345678	84 364 3988	0.6881
8	167 1278 137 1247 13468 125 14578 12345678	89 329 3825	0.7109
8	17 1268 1368 1236 146 12348 15 12345678	114 348 3648	0.6842
8	17 128 138 1236 146 1234 15 12345678	114 369 3783	0.6774
8	16 18 127 13 124 125 1457 12345678	126 291 3837	0.6523
8	16 1278 1238 1347 1357 1458 1245 12345678	147 408 4832	0.7500
8	16 17 128 138 123 14 15 12345678	180 315 3645	0.6160
8	16 17 12 1348 1358 1458 1345 12345678	219 144 4800	0.6250

Table E.5: Top 10 Non-isomorphic 18-run MDS designs with $k \leq 7$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
5	1 12 13 124 134 1234 125 145 12345	0 0 1	0.8333
5	1 12 13 124 134 1234 1235 145 12345	1 0 0	0.9009
5	1 12 13 123 14 1234 125 135 12345	1 0 0	0.8547
5	1 12 13 123 14 1234 125 145 12345	1 0 0	0.7719
5	1 12 13 123 14 124 125 135 12345	1 1 0	0.8262
5	1 12 13 123 14 124 15 135 12345	2 0 1	0.7875
5	1 12 13 124 134 125 145 145 12345	2 1 1	0.8262
5	1 1 123 124 134 125 1235 1345 12345	2 1 1	0.8333
5	1 1 12 123 124 134 135 1245 12345	2 1 1	0.8124
5	1 1 12 123 134 1234 135 145 12345	2 1 1	0.7875
6	16 12 13 1246 1346 1234 1256 145 123456	2 8 46	0.8421
6	1 126 136 1246 134 1234 125 145 123456	3 6 45	0.7843
6	16 12 13 1236 146 1234 1256 135 123456	4 6 27	0.7846
6	16 12 136 1236 14 1234 1256 135 123456	4 6 30	0.8247
6	16 12 13 1246 1346 1234 125 145 123456	4 6 35	0.7846
6	16 12 13 1236 14 1234 1256 135 123456	4 7 28	0.7892
6	16 12 13 1246 134 1234 1256 145 123456	4 11 25	0.7773
6	16 12 136 1246 134 1234 125 145 123456	5 3 44	0.8421
6	16 126 13 1236 14 1234 125 145 123456	5 5 27	0.7356
6	16 126 13 1236 146 1234 125 135 123456	5 5 41	0.7892
7	17 126 136 12467 1347 1234 125 145 1234567	11 33 271	0.7398
7	167 12 137 12467 1346 1234 1256 145 1234567	11 48 325	0.7967
7	167 127 137 1236 146 12347 1256 135 1234567	12 36 261	0.7435
7	16 127 1367 12467 134 1234 125 1457 1234567	12 42 338	0.7967
7	167 1267 137 1236 146 1234 1257 135 1234567	12 47 363	0.7467
7	16 127 1367 1236 14 12347 1256 135 1234567	13 25 244	0.7456
7	16 127 137 12367 147 1234 1256 135 1234567	13 30 258	0.7754
7	167 127 137 1236 146 1234 1256 135 1234567	13 31 352	0.7943
7	16 127 137 1236 1467 1234 1256 135 1234567	13 39 285	0.7549
7	16 127 13 1236 1467 1234 1256 1357 1234567	13 41 347	0.6608

Table E.6: Top 10 Non-isomorphic 18-run MDS designs with $8 \le k \le 9$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
8	1678 128 137 12467 1346 12348 1256 145 12345678	24 136 1798	0.7619
8	167 12678 1378 1236 1468 1234 1257 1358 12345678	26 145 1696	0.6772
8	178 1268 136 12467 1347 12348 125 1458 12345678	29 175 1776	0.7529
8	168 1278 1367 12467 1348 1234 125 1457 12345678	30 146 1791	0.8064
8	167 12678 1378 12368 1468 1234 12578 135 12345678	30 153 1843	0.7111
8	168 127 1378 12367 1467 1234 1256 135 12345678	31 163 1500	0.7011
8	1678 128 137 12467 13468 12348 1256 145 12345678	31 167 1727	0.7293
8	178 1268 136 12467 1347 1234 125 1458 12345678	31 182 1582	0.6895
8	168 1278 137 12368 1467 1234 1256 135 12345678	31 182 1784	0.6095
8	168 1278 137 12368 1467 1234 1256 1358 12345678	32 157 1750	0.7254
9	1689 12789 1367 12467 1348 12349 125 14579 123456789	56 518 6806	0.7732
9	1679 12678 13789 12369 14689 1234 12579 1358 123456789	62 558 7322	0.5510
9	168 1278 1367 124679 13489 1234 1259 1457 123456789	62 561 6970	0.7081
9	1689 1279 1378 12367 1467 1234 1256 1359 123456789	64 564 7012	0.6346
9	1689 1278 1379 12368 1467 12349 12569 135 123456789	65 615 6942	0.6282
9	1678 1289 1379 124679 13469 1234 12569 145 123456789	66 582 6376	0.6606
9	1689 12789 1379 12368 1467 12349 1256 1358 123456789	66 582 6532	0.6755
9	1678 1289 1379 12467 134689 12348 1256 1459 123456789	67 578 6991	0.6430
9	1789 1268 1369 124679 1347 12349 125 14589 123456789	68 546 6925	0.6552
9	167 12789 1379 12368 14689 1234 12569 135 123456789	68 594 6949	0.6100

Table E.7: Top 10 Non-isomorphic 20-run MDS designs with $k \leq 7$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
5	1 12 13 124 134 1234 125 135 145 12345	0 0 0	0.9231
5	1 12 13 123 14 124 125 135 145 12345	0 0 0	0.8955
5	1 12 13 123 14 124 134 125 135 12345	0 0 0	0.8408
5	1 12 13 123 14 124 1234 135 145 12345	0 0 0	0.7500
5	1 12 13 124 134 1234 125 135 1235 12345	0 0 0	0.6000
5	1 12 13 124 134 1234 125 145 145 12345	0 0 1	0.8333
5	1 1 12 123 124 134 135 1235 1245 12345	0 0 1	0.8227
5	1 12 13 123 14 124 15 135 145 12345	0 0 1	0.7959
5	1 1 12 13 124 134 1234 125 145 12345	0 0 1	0.7845
5	1 12 13 123 14 124 125 135 1245 12345	0 1 0	0.8333
6	1 126 136 123 146 124 125 135 145 123456	0 0 12	0.8239
6	1 126 136 124 1346 1234 1256 135 1235 123456	0 0 15	0.6429
6	16 12 13 1246 1346 1234 1256 1356 145 123456	0 0 16	0.9000
6	16 12 13 1236 14 1246 1256 1356 1245 123456	0 6 10	0.8000
6	16 12 13 1236 14 1246 15 1356 1456 123456	0 6 10	0.7385
6	1 126 136 123 146 1246 125 1356 145 123456	1 0 12	0.7826
6	1 126 136 124 1346 1234 1256 135 145 123456	1 0 16	0.8471
6	1 126 136 1236 146 124 134 125 135 123456	1 1 9	0.7685
6	16 126 136 123 146 124 134 125 135 123456	1 1 11	0.8327
6	16 1 126 123 1246 1346 1356 1235 1245 123456	1 2 10	0.7694
7	17 1267 136 1247 13467 1234 1256 1357 1235 1234567	2 2 96	0.4746
7	17 126 1367 124 1346 12347 12567 135 1235 1234567	2 4 137	0.6720
7	167 12 137 12467 1346 1234 1256 1356 145 1234567	3 3 127	0.8400
7	17 126 136 1237 1467 1247 125 1357 145 1234567	3 5 121	0.6364
7	17 1267 1367 123 1467 1246 125 1356 1457 1234567	4 5 71	0.7329
7	17 126 136 123 146 12467 1257 13567 145 1234567	4 5 96	0.7389
7	17 126 1367 1247 1346 1234 12567 1357 145 1234567	4 8 95	0.7636
7	17 126 1367 1247 1346 1234 1256 135 145 1234567	4 8 139	0.8400
7	17 126 1367 123 1467 1247 1257 1357 145 1234567	5 2 97	0.7840
7	17 1267 1367 123 1467 1246 125 1356 145 1234567	5 6 81	0.7248

Table E.8: Top 10 Non-isomorphic 20-run MDS designs with $8 \leq k \leq 10$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
8	17 1268 13678 1248 1346 12347 12567 1358 1235 12345678	6 21 589	0.6857
8	167 128 1378 12467 13468 1234 12568 1356 145 12345678	7 22 608	0.8000
8	17 1268 13678 12478 1346 1234 1256 1358 1458 12345678	8 32 602	0.8000
8	178 126 13678 1248 1346 12347 12567 1358 1235 12345678	9 32 511	0.5884
8	1678 12 1378 12467 1346 12348 12568 1356 145 12345678	9 42 472	0.7200
8	17 1268 13678 124 1346 12347 12567 1358 1235 12345678	10 28 500	0.6247
8	178 1267 1368 12478 13467 12348 12568 1357 1235 12345678	10 40 468	0.3529
8	178 12678 1368 1247 13467 12348 1256 1357 1235 12345678	11 31 506	0.4258
8	178 1268 1367 124 1346 12347 12567 1358 1235 12345678	11 32 559	0.6621
8	178 12678 1368 1247 13467 12348 12568 1357 1235 12345678	11 40 465	0.4692
9	179 12689 13678 1248 1346 123479 12567 13589 1235 123456789	12 68 2055	0.6750
9	179 12689 13678 1248 1346 12347 12567 13589 1235 123456789	21 100 1830	0.6353
9	179 12689 13678 1248 13469 123479 12567 13589 1235 123456789	22 105 1901	0.6750
9	179 12689 13678 1248 13469 12347 12567 1358 1235 123456789	22 109 1952	0.6353
9	1679 1289 1378 12467 134689 1234 12568 1356 145 123456789	22 114 1855	0.7043
9	1789 12689 1367 124 13469 123479 12567 1358 12359 123456789	22 122 2197	0.5838
9	1789 12679 1368 1247 13467 123489 12569 13579 1235 123456789	22 164 1896	0.4500
9	17 1268 136789 12489 1346 12347 12567 1358 12359 123456789	23 105 1908	0.6353
9	178 12689 13679 1249 1346 123479 12567 13589 1235 123456789	23 127 2213	0.5838
9	1679 128 1378 12467 13468 12349 125689 13569 145 123456789	24 96 1877	0.5400
10	17910 12689 1367810 124810 1346 123479 12567 13589 123510 12345678910	20 170 5990	0.6000
10	17910 1268910 13678 1248 1346 123479 12567 13589 123510 12345678910	37 265 5546	0.6207
10	17910 1268910 13678 1248 13469 123479 12567 13589 123510 12345678910	40 294 5922	0.6000
10	17910 1268910 1367810 1248 1346 123479 12567 13589 123510 12345678910	41 265 5734	0.6000
10	167910 1289 137810 12467 134689 123410 1256810 1356 145 12345678910	44 306 5768	0.6429
10	1789 12679 1368910 1247910 13467 12348 1256 135710 12359 12345678910	44 399 5883	0.3947
10	1789 1267910 136810 124710 13467 123489 12569 1357910 1235 12345678910	45 360 5921	0.3979
10	17910 12689 13678 1248 1346 123479 12567 13589 123510 12345678910	46 287 5688	0.6429
10	17910 1268910 13678 124810 134610 12347 12567 13589 123510 12345678910	46 307 5526	0.5455
10	178910 12689 136710 12410 13469 123479 12567 1358 12359 12345678910	46 336 5774	0.5455

Table E.9: Top 10 Non-isomorphic 22-run MDS designs with $5 \leq k \leq 8$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
5	1 12 13 124 134 1234 125 135 1235 145 12345	0 0 0	0.9545
5	1 12 13 124 134 1234 125 135 145 145 12345	0 0 0	0.9545
5	1 12 13 123 14 124 134 125 135 145 12345	0 0 0	0.9404
5	1 12 13 123 14 124 134 15 125 135 12345	0 0 0	0.9036
5	1 1 123 124 134 1234 125 135 1245 1345 12345	0 0 0	0.9036
5	1 1 12 123 124 134 135 1235 145 1245 12345	0 0 0	0.9036
5	1 1 12 13 124 134 1234 125 135 145 12345	0 0 0	0.8864
5	1 1 12 123 124 134 1234 135 1235 145 12345	0 0 0	0.8708
5	1 12 13 123 14 124 134 15 125 1235 12345	0 0 0	0.8708
5	1 1 12 13 123 124 134 125 1235 145 12345	0 0 0	0.8658
6	1 126 136 123 146 124 134 156 125 135 123456	0 0 0	0.9028
6	16 1 1236 1246 1346 1234 1256 1356 1235 1245 123456	0 0 0	0.8300
6	16 12 13 1236 14 1246 1346 15 1256 1235 123456	0 0 0	0.8233
6	16 12 13 1236 14 1246 1346 15 1256 1356 123456	0 0 0	0.7897
6	16 1 12 1236 1246 1346 1234 1256 1356 1245 123456	0 0 0	0.7805
6	16 1 12 1236 1246 1346 1234 1256 1356 1235 123456	0 0 0	0.6783
6	16 12 13 1236 14 1246 1346 1235 1245 1345 123456	0 0 0	0.6089
6	16 1 126 136 123 146 124 125 135 145 123456	0 0 3	0.8138
6	16 1 126 136 124 134 12346 125 135 145 123456	0 0 3	0.7405
6	16 1 123 1246 1346 1234 1256 1356 1245 1345 123456	0 0 3	0.9028
7	16 17 1267 13 1246 1346 12347 1256 13567 1235 1234567	0 0 33	0.4438
7	16 17 1236 1247 13467 1234 12567 1357 1235 1245 1234567	0 0 45	0.7636
7	16 17 1267 123 1246 1346 13567 1235 145 12457 1234567	0 0 52	0.7636
7	16 17 1236 124 13467 12347 12567 1357 1235 12457 1234567	0 0 56	0.6519
7	167 17 1267 13 1246 1346 12347 1256 13567 1235 1234567	0 0 57	0.4438
7	16 1 127 137 1246 13467 1234 12567 1356 1235 1234567	0 0 58	0.2246
7	16 17 126 1237 1246 13467 1356 1235 145 12457 1234567	0 0 60	0.6519
7	17 126 136 1247 13467 1234 12567 1357 1235 145 1234567	0 0 70	0.9091
7	17 1267 136 124 1346 12347 1256 1357 1235 145 1234567	0 0 76	0.9091
7	16 17 1267 1367 124 1347 12346 1257 135 1457 1234567	1 0 34	0.7428
8	16 178 1267 1238 1246 13468 13567 1235 145 124578 12345678	0 0 248	0.2297
8	178 1268 136 1247 13467 12348 12567 13578 1235 145 12345678	0 0 310	0.8727
8	178 1267 1368 124 1346 123478 12568 1357 1235 1458 12345678	0 0 310	0.8727
8	168 17 12678 1238 1246 1346 13567 1235 1458 12457 12345678	2 3 244	0.7589
8	168 18 127 1378 1246 13467 12348 125678 1356 1235 12345678	2 6 297	0.2297

Table E.10: Top 10 Non-isomorphic 22-run MDS designs with $8 \leq k \leq 11$

k	*Design Runs*	$(MDS(4), MDS(5), MDS(6))$	*Efficiency*
8	168 178 1267 123 12468 1346 13567 12358 145 12457 12345678	2 7 248	0.7589
8	16 178 12368 124 13467 123478 12567 1357 1235 124578 12345678	3 4 254	0.5289
8	16 178 1267 1238 1246 13468 13567 1235 1458 12457 12345678	3 6 203	0.4441
8	178 126 136 1247 13467 12348 12567 13578 1235 145 12345678	3 6 244	0.7589
8	178 126 1368 1247 13467 12348 125678 1357 1235 145 12345678	3 6 250	0.8875
9	1789 1268 1369 12479 13467 12348 12567 13578 12359 145 123456789	0 0 993	0.8182
9	1789 1267 1368 1249 13469 123478 12568 13579 1235 1458 123456789	4 16 829	0.7403
9	169 1789 1267 1238 12469 13468 13567 12359 145 124578 123456789	4 20 905	0.2231
9	1789 1268 136 12479 13467 12348 12567 13578 12359 145 123456789	5 22 865	0.7403
9	16 1789 1267 1238 12469 13468 13567 12359 145 124578 123456789	6 16 881	0.2455
9	1789 1268 1369 1247 13467 123489 125679 13578 1235 145 123456789	6 24 841	0.8497
9	1789 1267 13689 1249 1346 123478 12568 1357 1235 1458 123456789	6 25 905	0.8182
9	1789 1268 1369 12479 13467 12348 12567 13578 1235 145 123456789	6 27 856	0.8182
9	1689 18 1279 1378 1246 13467 123489 125678 13569 1235 123456789	6 27 941	0.2231
9	19 123678 123 1468 13479 1567 12589 13569 12457 13458 123456789	6 28 946	0.6818
10	178910 123679 12310 146910 13478 1567 12589 1356810 1245710 13459, 12345678910	0 0 2730	0.7273
10	178910 123679 12310 146910 13478 1567 12589 13568 1245710 13459, 12345678910	8 52 2396	0.7006
10	123910 1246810 1348 1234679 1256 135689 12357810 145910 1245789, 13456710 12345678910	8 56 2510	0.1983
10	178910 123679 12310 146910 13478 1567 12589 1356810 1245710 1345910, 12345678910	9 57 2382	0.7792
10	17810 1236710 1239 146910 134789 15679 1258910 13568910 12457 134510, 12345678910	10 74 2633	0.1983
10	1237 12467910 1349 12346810 1256 13567910 12358910 14510 1245789 1345678, 12345678910	12 48 2574	0.2537
10	18910 12367 123810 146810 1347910 156710 125910 135689 124578 1345 12345678910	12 68 2494	0.6572
10	1789 123679 12310 14689 1347810 156710 1258910 13568 12457 13459 12345678910	12 80 2494	0.7982
10	19 123678 12310 146810 1347910 156710 1258910 13569 12457 13458 12345678910	12 83 2716	0.6818
10	169 1789 126710 123810 12469 13468 13567 12359 14510 124578 12345678910	13 65 2610	0.2372
11	178910 126811 13691011 1247911 13467 1234810 1256710 1357811 12359 1451011 1234567891011	0 0 6765	0.5455
11	178910 126811 13691011 1247911 13467 1234810 1256710 1357811 12359 14510 1234567891011	12 108 5871	0.6207
11	178910 126811 1369 1247911 13467 1234810 1256710 1357811 12359 1451011 1234567891011	16 160 6377	0.1395
11	178910 126811 13691011 1247911 13467 234810 1256710 1357811 12359 145 1234567891011	18 174 6327	0.6667
11	178910 126711 13681011 12491011 13469 123478 125681011 1357911 123510 145811 1234567891011	20 148 6479	0.1935
11	16911 178911 12671011 123810 12469 1346811 13567 1235911 1451011 124578 1234567891011	22 130 6515	0.2400
11	1691011 178911 126710 12381011 1246911 13468 1356711 12359 1451011 124578 1234567891011	22 178 6900	0.1935
11	16911 178911 126710 12381011 1246911 13468 1356711 12359 1451011 124578 1234567891011	24 160 6960	0.2400
11	178910 1268 136911 1247911 13467 1234810 1256710 1357811 12359 1451011 1234567891011	24 200 5578	0.5185
11	178910 126811 136911 1247911 13467 123481011 1256710 1357811 12359 1451011 1234567891011	24 201 5689	0.5185

Table E.11: Top 10 Non-isomorphic 24-run MDS designs with $5 \leq k \leq 7$

k	*Design Runs*	$(MDS(6),..,MDS(8))$	*Efficiency*
5	1 12 13 123 14 124 134 1234 125 135 145 12345	0 0 0	0.9524
5	1 12 13 123 14 124 134 15 125 135 145 12345	0 0 0	0.9524
5	1 1 123 124 134 1234 125 135 1235 145 1245 12345	0 0 0	0.9524
5	1 1 12 123 124 134 134 135 1235 145 1245 12345	0 0 0	0.9524
5	1 12 13 123 14 124 134 15 125 135 1245 12345	0 0 0	0.9315
5	1 1 12 123 124 134 1234 125 135 1235 145 12345	0 0 0	0.9315
5	1 12 13 124 134 1234 125 135 145 145 145 12345	0 0 0	0.9259
5	1 1 123 124 134 1234 125 135 1235 1245 1345 12345	0 0 0	0.9259
5	1 1 12 13 124 134 1234 125 135 1235 145 12345	0 0 0	0.9259
5	1 1 12 13 124 134 1234 125 135 145 145 12345	0 0 0	0.9259
6	16 1 126 13 1236 1246 1346 1234 1235 1456 1245 123456	0 0 2	0.8381
6	16 1 12 1236 1246 1346 134 1234 1356 1245 1345 123456	0 0 2	0.8195
6	16 1 12 1236 1246 134 1356 1235 1456 1245 1345 123456	0 0 4	0.9130
6	16 1 12 13 1236 1246 1346 1234 1256 1356 145 123456	0 0 4	0.8768
6	16 1 12 1236 1246 1346 1234 1256 1356 1245 1345 123456	0 0 4	0.8555
6	16 1 12 1236 1246 1346 1234 1256 1356 1235 1345 123456	0 0 4	0.8246
6	16 1 12 13 1236 1246 1346 1234 1256 1456 1345 123456	0 0 4	0.8173
6	16 1 12 13 1236 1246 1346 1234 1256 1356 1245 123456	0 0 4	0.8011
6	16 1 126 12 13 1236 14 1346 1356 1235 1456 123456	0 0 4	0.7845
6	16 1 12 13 1236 1246 1346 1234 1256 1456 1245 123456	0 0 4	0.7825
7	16 17 12 137 1236 146 12467 1256 1257 13567 1245 1234567	11 40 134	0.7065
7	16 17 12 13 12367 1467 1247 1346 1256 1235 1345 1234567	11 41 165	0.7242
7	16 17 1267 1236 1237 1246 13467 1234 1356 1245 13457 1234567	11 42 151	0.8383
7	16 17 1267 13 1236 1246 1346 12347 1235 1456 12457 1234567	12 22 114	0.7347
7	167 1 127 137 1236 1467 1246 12567 125 13567 12457 1234567	12 36 139	0.6923
7	16 17 12 13 12367 1467 1246 1234 1257 1357 1245 1234567	12 42 157	0.6954
7	16 17 1267 13 1237 1246 134 123467 1257 1245 13457 1234567	12 47 132	0.5918
7	16 17 12 1367 1236 12467 1346 1234 12357 12456 1345 1234567	13 28 114	0.8167
7	16 17 127 1367 123 14 1246 1346 1256 12357 1345 1234567	13 29 136	0.7824
7	16 17 1267 1367 123 1467 1247 134 1257 1357 145 1234567	13 32 137	0.8128

Table E.12: Top 10 Non-isomorphic 24-run MDS designs with $8 \leq k \leq 10$

k	*Design Runs*	$(MDS(6),..,MDS(8))$	*Efficiency*
8	1678 17 12 1368 12367 124678 1346 12347 12358 12456 13457 12345678	82 191 1450	0.7381
8	167 18 1268 123 1248 13478 123467 13578 123568 1456 1345 12345678	89 207 1302	0.4349
8	167 178 126 12368 12478 13467 134 12348 1358 1245 13456 12345678	89 268 1679	0.1975
8	17 18 1278 1236 12468 13467 123478 12567 13568 123578 145678 12345678	96 270 1563	0.2128
8	17 18 1236 1246 134678 123478 12568 13568 12357 14567 12458 12345678	97 178 1392	0.6108
8	17 18 12367 124678 13468 12347 125678 13567 12358 1456 1245 12345678	97 194 1417	0.6996
8	167 18 1268 1238 12478 1347 12346 13578 123567 14568 1345 12345678	101 194 1396	0.5444
8	167 18 12367 124678 13468 1234 12568 1356 1235 1457 12458 12345678	102 265 1572	0.5781
8	16 17 1268 138 1236 12478 13467 1234 12567 13578 1235 12345678	103 231 1435	0.4468
8	17 18 1236 1246 134678 123478 1256 13568 12357 14567 12458 12345678	103 266 1508	0.3540
9	168 179 12369 124789 1348 123467 125678 1357 123589 1245 134578 123456789	425 948 12030	0.5625
9	1689 1789 1278 1379 1236789 12469 13467 12348 12567 13568 12359 123456789	442 947 11370	0.7105
9	179 18 1268 1369 1237 14679 12489 13478 15678 12579 13589 123456789	453 1298 13396	0.2195
9	1689 1 1278 1379 1236 12469 13467 12348 12567 13568 12359 123456789	496 1011 12453	0.7105
9	179 18 1236 124678 134689 123479 1256789 13567 123589 14569 1245 123456789	499 950 11158	0.8182
9	179 18 1236 124689 13467 1234789 125679 135689 123578 145678 1245 123456789	506 896 11518	0.7826
9	189 1268 1369 12379 1467 12489 13478 156789 1257 1358 1459 123456789	508 910 11878	0.8761
9	17 189 12369 12468 134678 123479 1256789 1356 123578 145679 1245 123456789	508 930 12176	0.8182
9	16789 17 126 1237 1246 13469 13478 1359 123568 1458 124579 123456789	508 1372 14111	0.2195
9	17 189 12369 12468 13467 1234789 125679 13568 123578 1456789 1245 123456789	515 863 10787	0.7826
10	179 1810 1236 12467810 134689 12347910 1256789 1356710 12358910 1456910 1245 12345678910	1383 4176 67524	0.7143
10	17910 1268 136910 12379 146710 1248910 13478 156789 125710 135810 1459 12345678910	1402 3908 60341	0.8621
10	179 1810 1236 124678 13468910 12347910 125678910 1356710 123589 14569 124510 12345678910	1402 4152 60267	0.7143
10	179 126810 1369 1237910 146710 12489 13478 156789 1257 135810 145910 12345678910	1404 4278 72633	0.7813
10	179 1810 123610 124689 13467 123478910 12567910 135689 123578 14567810 1245 12345678910	1406 4036 63679	0.7143
10	179 1810 1236 124689 1346710 123478910 12567910 13568910 123578 14567810 1245 12345678910	1410 4044 60561	0.6875
10	179 1810 123610 124678 134689 12347910 125678910 13567 123589 1456910 1245 12345678910	1417 4114 64188	0.7812
10	17910 126810 1369 1237910 146710 12489 13478 156789 1257 135810 145910 12345678910	1467 3766 57734	0.8621
10	17910 18 123610 124689 13467 123478910 125679 13568910 123578 14567810 124510 12345678910	1473 3834 60230	0.6875
10	17910 18 123610 124678 13468910 123479 125678910 13567 123589 14569 124510 12345678910	1480 4036 60373	0.7812

Table E.13: Top 10 Non-isomorphic 24-run MDS designs with $11 \leq k \leq 12$

k	*Design Runs*	$MDS(6)$	*Efficiency*
11	179 181011 1236 12467810 13468911 1234791011 125678911 135671011 12358910 1456910 124511 1234567891011	3357	0.5238
11	179 181011 123611 12467810 134689 1234791011 125678911 1356710 12358910 145691011 1245 1234567891011	3378	0.6111
11	17910 126811 13691011 12379 146710 1248910 1347811 156789 12571011 135810 145911 1234567891011	3390	0.7333
11	1791011 126811 13691011 12379 146710 1248910 1347811 156789 12571011 135810 145911 1234567891011	3453	0.8462
11	179 181011 123610 124689 1346711 12347891011 1256791011 13568911 123578 14567810 124511 1234567891011	3525	0.5238
11	17911 1810 12361011 124678 13468911 12347910 12567891011 13567 123589 1456910 124511 1234567891011	3528	0.7333
11	179 181011 123611 124678 13468910 1234791011 12567891011 1356710 123589 1456911 124510 1234567891011	3542	0.6111
11	17891011 126811 13691011 123710 14679 1248910 1347811 1567810 1257911 13589 1451011 1234567891011	3825	0.9346
11	1 126811 13691011 12379 146710 1248910 1347811 156789 12571011 135810 145911 1234567891011	4175	0.5238
11	17891011 1 1236811 12468910 134671011 123479 12567911 1356910 12357810 145678 12451011 1234567891011	4182	0.6111
12	1791012 12681112 13691011 1237912 14671012 1248910 11347811 156789 12571011 13581012 14591112 123456789101112	7578	0.6667
12	179 18101112 123612 12467810 13468911 123479101112 12567891112 135671011 12358910 145691012 124511 123456789101112	7624	0.1463
12	1789101112 12681112 13691011 12371012 1467912 1248910 1347811 1567810 1257911 1358912 145101112 123456789101112	8470	0.9317
12	17912 181011 12361112 1246781012 134689 1234791011 125678911 1356710 1235891012 14569101112 1245 123456789101112	8893	0.4286
12	179101112 12681112 13691011 1237912 14671012 1248910 1347811 156789 12571011 13581012 14591112 123456789101112	9145	0.8276
12	17911 181012 12361011 124678 1346891112 1234791012 1256789101112 1356712 123589 1456910 12451112 123456789101112	9501	0.4286
12	1 12681112 13691011 1237912 14671012 1248910 1347811 156789 12571011 13581012 14591112 123456789101112	9600	0.1463

Bibliography

Banerjee, K. S. and Federer, W. T. (1967). On a special subset giving an irregular fractional replicate of a 2^n factorial experiment. *Journal of the Royal Statistical Society, Series B* **29**, 292–299.

Bayarri, M., Berger, J. O., Higdon, D., Kennedy, M., Kottas, A., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J. and Tu, J. (2002). A framework for the validation of computer models. *Proceedings of the Workshop on Foundations for V&V in the 21st Century* , D. Pace and S. Stevenson, eds., Society for Modeling and Simulation International.

Ben-Ari, E. N. and Steinberg, D. M. (2007). An empirical comparison of kriging with mars and projection pursuit regression in modeling data from computer experiments. *Quality Engineering* **19**, 327–338.

Bingham, D. and Chipman, H. A. (2007). Incorporating prior information in optimal design for model selection. *Technometrics* **49**, 155–163.

Bingham, D., Sitter, R. R. and Tang, B. (2008). Orthogonal and nearly orthogonal designs for computer experiments. *Accepted by Biometrika* .

Box, G. E. P. and Hunter, W. G. (1961a). The 2^{k-p} fractional factorial designs. *Technometrics* **3**, 311–352.

Box, G. E. P. and Hunter, W. G. (1961b). The 2^{k-p} fractional factorial designs. *Technometrics* **3**, 449–458.

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

Butler, N. A. (2003a). Some theory for constructing minimum aberration fractional factorial designs. *Biometrika* **90**, 233–238.

Butler, N. A. (2003b). Minimum aberration construction results for nonregular two-level fractional factorial designs. *Biometrika* **90**, 891–898.

Cheng, C. S. (1995). Some projection properties of orthogonal arrays. *Annals of Statistics* **23**, 1223–1233.

Cheng, C. S., Deng, L. Y. and Tang, B. (2002). Generalized minimum aberration and design efficiency for nonregular fractional factorial designs. *Statistica Sinica* **12**, 991–1000.

Cheng, C. S. and Mukerjee, R. (1998). Regular fractional factorial designs with minimum estimation capacity. *Annals of Statistics* **26**, 2289–2300.

Cheng, C. S., Steinberg, D. M. and Sun, D. X. (1998). Minimum aberration and model robustness for two-level fractional factorial designs. *Journal of the Royal Statistical Society: Series B* **61**, 85–93.

Chipman, H. (1996). Bayesian variable selection with related predictors. *Canadian Journal of Statistics* **24**, 17–36.

Chipman, H., Hamada, M. and Wu, C. F. J. (1997). A bayesian variable selection approach for analyzing designed experiments with complex aliasing. *Technometrics* **39**, 372–381.

Cioppa, T. M. and Lucas, T. M. (2007). Efficient nearly orthogonal and space-filling latin hypercubes. *Technometrics* **49**, 45–55.

Clark, J. B. and Dean, A. M. (2001). Equavalence of fractional factorial designs. *Statistica Sinica* **11**, 537–547.

Deng, L. Y., Li, Y. and Tang, B. (2000). Catalogue of nonregular designs with small runs from Hadamard matrices based on generalized minimum aberration criterion. *Communication in Statistics - Theory and Methods* **29**, 1379–1395.

Deng, L. Y. and Tang, B. (1999). Generalized resolution and minimum aberration criteria for Plackett-Burman and other nonregular factorial designs. *Statistica Sinica* **9**, 1071–1082.

Deng, L. Y. and Tang, B. (2002). Design selection and classification for Hadamard matrices using generalized minimum aberration criteria. *Technometrics* **44**, 173–184.

Fang, K., Li, R. and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. CRC Press.

Fang, K. T. (1980). The uniform design: Application of number-theoretic methods in experimental design. *Acta Mathematicae Applicatae Sinica* **3**, 363–372.

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

Geramita, A. V. and Seberry, J. (1979). *Orthogonal designs*. Marcel Dekker.

Handcock, M. S. (1991). On cascading latin hypercube designs and additive models for experiments. *Communication Statistics - Theory and Method* **20**, 417–439.

Iman, R. L. and Conover, W. J. (1982). A distribution-free approach to inducing rank correlation among input variables. *Communication in Statistics, Part B - Simulation and Computation* **11**, 311–334.

Ingram, D. K. and Tang, B. (2005). Construction of minimum G-aberration designs via efficient computational algorithms. *Journal of Quality Technology* **37**, 101–114.

John, P. W. M. (1962). Three-quarter replicates of 2^n designs. *Biometrics* **18**, 172–184.

John, P. W. M. (1964). Blocking of $3(2^n - k)$ designs. *Technometrics* **6**, 371–376.

Johnson, M. E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26**, 131–148.

Jones, B. A., Li, W., Nachtsheim, C. J. and Ye, K. Q. (2007). Model discrimination – another perspective on model-robust designs. *Journal of Statistical Planning and Inference* **137**, 1576–1583.

Li, Y., Deng, L. Y. and Tang, B. (2004). Design catalog based on minimum G-aberration. *Journal of Statistical Planning and Inference* **124**, 219-230.

Lin, C. D. and Sitter, R. R. (2008). An isomorphism check for two-level fractional factorial designs. *Journal of Statistical Planning and Inference* **134**, 1085–1101.

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

Margolin, B. H. (1969). Results on factorial designs of resolution IV for the 2^n and 2^n3^m series. *Technometrics* **11**, 431–444.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245.

Miller, A. and Sitter, R. R. (2004). Choosing columns from the 12-run plackett-burman design. *Statistics and Probability Letters* **67**, 193–201.

Miller, A. and Sitter, R. R. (2005). Using folded over non-orthogonal designs. *Technometrics* **47**, 502–513.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for conputational experiments. *Journal of Statistical Planning and Inference* **43**, 381–402.

Nordstrom, A. W. and Robinson, J. P. (1967). An optimum nonlinear code. *Inform Control* **11**, 613–616.

Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration, and visualization. *Statistica Sinica* **2**, 439–452.

Owen, A. B. (1994). Controlling correlations in latin hypercube samples. *Journal of the American Statistical Association* **89**, 1517-1522.

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

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Sciences* .

Santner, T. J., Williams, B. J. and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Spring-Verlag.

Simpson, T. W., Peplinski, J. D., Koch, P. N. and Allen, J. K. (2001). Meta-models for computer-based engineering design: Survey and recommendations. *Engineering With Computers* .

Steinberg, D. M. and Lin, D. K. J. (2006). A construction method for orthogonal latin hypercube designs. *Biometrika* **93**, 279-288.

Sun, D. X. (1993). *Estimation Capacity and Related Topics in Experimental Design.* Ph.D. thesis, University of Waterloo.

Sun, D. X., Li, W. and Ye, Q. (2002). An algorithm for sequentially constructing non-isomorphic orthogonal designs and its applications. *Technical Report, SUNYSB-AMS* .

Tang, B. (1993). Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association* **88**, 1392–1397.

Tang, B. (1998). Selecting latin hypercubes using correlation criteria. *Statistica Sinica* **8**, 965–977.

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

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

Tang, B. and Deng, L. Y. (2003). Construction of generalized minimum aberration designs of 3, 4, and 5 factors. *Journal of Statistical Planning and Inference* **113**, 335–340.

Vartak, M. N. (1955). On an application of kronecker product of matrices to statistical designs. *The Annals of Mathematical Statistics* **36**, 420–438.

Wang, Y. and Fang, K. T. (1981). A note on uniform distribution and experimetnal design. *Kexue TongBao* **26**, 485–489.

Webb, S. (1968). Non-orthogonal designs of even resolution. *Technometric* **10**, 291–299.

Wu, C. F. J. and Chen, Y. (1992). A graph-aided method for planning two-level experiments when certain interactions are important. *Technometric* **34**, 162–174.

Wu, C. F. J. and Hamada, M. (2000). *Experiments Planning, Analysis, and Parameter Designs Optimization*. Wiley-Interscience Publication.

Xu, H. (2002). An algorithm for constructing orthogonal and nearly-orthogonal arrays with mixed levels and small runs. *Technometric* **44**, 1430–1439.

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

Xu, H. (2005). Some nonregular designs from the Nordstrom and Robinson code and their statistical properties. *Biometrika* **92**, 385–397.

Xu, H. and Wong, A. (2007). Two-level nonregular designs from quaternary linear codes. *Statistica Sinica* **17**, 1191–1213.

Ye, Q. (1998). Orthogonal column latin hypercubes and their application in computer experiments. *Journal of American Statistics Association* **93**, 1430–1439.