SEARCHING FOR OPTIMAL ORTHOGONAL ARRAYS FOR ESTIMATING MAIN EFFECTS AND SOME SPECIFIED TWO-FACTOR INTERACTIONS

by

Donghong Wu

Bachelor of Science, Shanxi University, China, 2001

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the Department
of
Statistics and Actuarial Science

SIMON FRASER UNIVERSITY
Summer 2009

APPROVAL

Name: Donghong Wu

Degree: Master of Science

Title of Thesis: Searching for Optimal Orthogonal Arrays for Estimating Main Effects and Some Specified Two-factor Interactions

Examining Committee: Dr. Rachel Altman
Chair

_____

Dr. Boxin Tang, Senior Supervisor

_____

Dr. Derek Bingham, Supervisor

_____

Dr. Jiguo Cao, External Examiner

Date Approved: _____

Abstract

We consider the problem of finding optimal orthogonal arrays for estimating main effects and some specified two-factor interactions. Based on theoretical results from Tang and Zhou (2009), we develop a computational algorithm for this purpose. The D-efficiency and bias are considered as the criteria for design optimality. The performance of the algorithm is evaluated by comparing the results obtained by the algorithm with those from complete search. Finally, we present a useful collection of optimal orthogonal arrays with small run sizes.

Keywords: Bias; D-optimality; Hadamard matrix; Nonregular design; Requirement set

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Boxin Tang who made great effort to guide me throughout my studies. I really appreciate his dedication, continuing patience and understanding. His supervision has been a highlight of my time at SFU. It is my great honor to be his student.

I am very grateful to my examining committee, Dr. Derek Bingham and Dr. Jiguo Cao, for their time and suggestions to make my thesis better. I also want to give my gratitude to the professors who taught me various courses and showed me various statistical problems. Many thanks to my fellow graduate students for being company and growing together with me during my stay at SFU.

Last but not least I would like to thank my family who have always provided me with encouragement, endless support and unconditional love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Design of experiments is an efficient procedure used to capture the essential features of relationship between the different factors affecting a process and the output of that process by exploring related data. This method was first developed in the 1920s and 1930s, by Sir Ronald A. Fisher, the renowned mathematician and geneticist. It can be applied whenever people intend to investigate a phenomenon in order to gain understanding or improve performance. In industrial investigations, the experimenter wants to identify the most important factors from a large list of initial factors and estimate their effects to improve the production process. Two-level full factorial and fractional factorial designs are the most studied in the design literature and widely used in industrial experiments.

1.1 Factorial Designs

A two-level factorial design is a design that consists of two or more factors at two levels each. These levels are called high and low or $+1$ and -1, respectively. A factorial design permits experimenters to examine simultaneously the effects of multiple independent variables on the response. The independent variables are called factors and each factor must have at least two levels so that the effect of change in factor settings on the response can be studied. A combination of the level settings of factors is referred to as a treatment or a run. The experimental designs that deal with the arrangement of treatments are called factorial designs. A full factorial design consists of all possible treatment combinations of the independent factors. That is, if a factorial experiment contains k factors at two levels, a full factorial design requires 2^k

runs. The number of necessary runs in a full factorial design increases geometrically as the number of factors increases, so the run sizes would be quite large when there are a lot of factors of interest. As such, the full factorial design would be too costly to be used. Furthermore, a 2^k full factorial design allows us to estimate all $2^k - 1$ effects, but when k is greater than 4 or 5, most of these effects will be higher-order interaction effects that might not be significant according to the hierarchical assumption that higher-order effects are less likely to be significant. The estimation of those non-significant effects will waste our time and other resources. For economical and practical reasons, the fractional factorial designs are commonly used in practice as they only use a fraction or a subset of runs specified by the full factorial design.

By using a fraction of runs of a full factorial design, a fractional factorial design still can estimate many effects, even though estimates of some effects may not be fully distinguishable from each other, which is referred to as aliasing or confounding. Based on the aliasing pattern, fractional factorial designs can be classified into two broad categories: regular designs and nonregular designs. A regular two-level fractional factorial design is often called a 2^{k-p} design, where k is the number of factors under study and p denotes the size of the fraction of the full factorial design. Such a design can be constructed by using a defining relation. Suppose that we use "letters" to stand for the factors. Then a defining word consists of a set of letters if the product of the corresponding columns for those letters is a column of all +1's. The defining relation of a 2^{k-p} design has 2^p defining words including the grand mean I. Based on the defining relation, the aliasing pattern of this design is determined and the estimates of effects are either orthogonal or fully aliased. Nonregular designs such as Plackett-Burnman designs (Plackett and Burman, 1946) and other orthogonal arrays have a complex aliasing structure in that there exist effects that are neither orthogonal nor fully aliased, which means some effects are partially aliased.

1.2 Orthogonal Arrays

The concept of orthogonal arrays was introduced by Rao (1947) with Plackett-Burman designs included as special cases. An $N \times k$ array A with entries from $S = \{0, 1, \ldots, s-1\}$ is defined as an orthogonal array (Hedayat, Sloane and Stufken 1999) with s levels, strength t and index λ for some t in the range $(0 \leq t \leq k)$ if every $N \times t$ subarray of A contains each t-tuple based on S exactly λ times as a row. The number N of

rows is the size of the array or the number of runs. The number k of columns is the number of factors or variables, while s is the number of levels. This thesis will focus on orthogonal arrays of strength two. A wealth of orthogonal arrays can be generated from Hadamard matrices. A Hadamard matrix is a square matrix whose entries are either $+1$ or -1 and whose rows or columns are mutually orthogonal. A Hadamard matrix H of order n satisfies that $HH^T = nI_n$, where H^T is the transpose of H and I_n is an $n \times n$ identity matrix.

The order of a Hadamard matrix must be 1, 2, or a multiple of 4. A Hadamard matrix of order n can be normalized by multiplying +1 or -1 to the rows so that all entries in its first column are equal to 1. A saturated two-level orthogonal array with n runs can be obtained by removing the first column of a normalized Hadamard matrix. A design of m factors, where $m \leq n - 1$, can be generated by choosing m columns from a saturated orthogonal array of run size n.

1.3 The Problem

Very often some two factor interactions are non-negligible and important. In this case, investigators are interested in estimating all main effects plus these specified two-factor interactions (2fi's). This problem was first considered by Addelman (1962), who studied three classes of compromise plans by using regular fractional factorial designs. Greenfield (1976) considered the problem under a more general setting and used a requirement set to denote a set of effects that the experimenter is interested in estimating. Systematic procedures for estimating main effects and selected 2fi's were proposed by Greenfield (1976) and Franklin and Bailey (1977). A graph-aided method was developed by Wu and Chen (1992) for planning two-level experiments when certain interactions are important. Hamada and Wu (1992) used the complex aliasing structure of Plackett-Burman designs to entertain and estimate some interactions. For the same problem, Ke and Tang (2003) studied the selection of appropriate designs using a minimum aberration criterion, leading to the robustness of resulting designs.

For the problem of estimating all main effects and some 2fi's in the given requirement set, a desirable design should ensure that those effects of interest are not aliased with each other by using a small number of runs. For a given requirement set, a regular design is always first considered because it achieves full efficiency. However, the run sizes of regular designs must be powers of 2. If we cannot find a regular design

to estimate those effects of interest, we have to consider designs with at least doubled run sizes. Compared with regular designs, nonregular designs sacrifice orthogonality in order to consider more variables in the same number of runs. They provide more flexibility in run sizes and entertain more distinct models. The run sizes of orthogonal arrays based on Hadamard matrices are $n = 4t$, where t is an integer, so the gaps of run sizes are much smaller than those of regular designs. In this thesis, we will focus our attention on the use of orthogonal arrays for the purpose of jointly estimating all main effects and a set of specified two-factor interactions.

The purpose of this thesis is to propose an efficient algorithm for finding D-optimal designs that allow joint estimation of all main effects and these potentially important 2fi's. The motivation is that the complete search is impossible for designs of large run sizes, and optimal designs of large run sizes are often required in practice. The theoretical results of Tang and Zhou (2009) are discussed and used to develop our algorithm.

1.4 Outline

This thesis is organized as follows. In Chapter 2, we describe our problem and present the results of complete search for designs of 12 and 20 runs. The D-efficiency is used as a criterion of selection for optimal designs. The bias is introduced as a secondary criterion for comparing designs with the same highest D-efficiency. Chapter 3 reviews the theoretical results of Tang and Zhou (2009) on the existence and construction of two-level orthogonal arrays for estimating all main effects and a set of specified two-factor interactions. Based on the theoretical results, a computational algorithm is developed for searching for D-optimal designs. To evaluate the performance of our algorithm, we compare the results from the complete search with those from the algorithm. The application of our algorithm to 24 runs is also studied. From the results of 24 runs, a construction result is summarized for certain models. This thesis concludes with a summary and a discussion of future work in Chapter 4.

Chapter 2

D-Optimal Orthogonal Arrays

2.1 Problem Formulation

2.1.1 Requirement Set

A requirement set consists of all effects that the experimenter is interested in estimating. For the representation of a given requirement set, it is common for the factors to be represented by F_1, F_2, F_3, and so on, and a 2fi is denoted by pairing the letters associated with the main effects. Thus, a requirement set consisting of factors F_1, F_2 and their interaction is written as $\{F_1, F_2, F_1F_2\}$. For the corresponding model matrix, m columns are chosen from orthogonal arrays of run size n for m factors and the product of two columns for corresponding factors is included for two-factor interaction. For example, to include the interaction between factors F_1 and F_2, an extra column is added to the model matrix, consisting of the element wise products of the column vectors F_1 and F_2. The interaction column is written using the notation F_1F_2.

Example 1. Consider the requirement set $\{F_1, F_2, \ldots, F_9, F_1F_2, F_1F_3\}$. A Hadamard

matrix of order 12 is used to illustrate the model matrix construction for this requirement set. The corresponding model matrix is shown in the following array:

$$\mathbf{X} = \left[\begin{array}{c|ccccccccc|cc} \mathbf{I} & & & & & X_1 & & & & & X_2 & \\ I & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 & F_8 & F_9 & F_1F_2 & F_1F_3 \\ +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & -1 & +1 & -1 \\ +1 & +1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 \\ +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 & -1 & -1 \\ +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 & -1 & -1 & +1 \\ +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 & +1 & -1 \\ +1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & +1 & +1 \\ +1 & -1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 \\ +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & +1 \\ +1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 & -1 & +1 \end{array}\right].$$

The graph theory is used to model the pairwise relations between subjects from a certain collection in mathematics and computer science. For a requirement set, a graph can be drawn by associating a factor with a vertex and a two-factor interaction with an edge. For example, consider the requirement set $S = \{F_1, F_2, F_3, F_4, F_1F_2, F_1F_3\}$, where F_1, F_2, F_3 and F_4 represent the four factors, and F_1F_2 and F_1F_3 represent the two 2fi's, and the graph for this requirement set is shown in Figure 2.1. Because we are not interested in estimating the interactions between factor F_4 and any of the other three factors F_1, F_2 and F_3, the vertex representing factor F_4 is isolated in Figure 2.1.

2.1.2 The General Model

For a requirement set containing all main effects and some specified two-factor interactions, the corresponding model can be written as

$$Y = X\beta + \varepsilon \tag{2.1}$$

where $\beta = (\beta_1, \cdots, \beta_p)^T$ is a vector including the grand mean, m main effects and e two-factor interactions; p is the number of parameters to be estimated, $p = 1 + m + e$

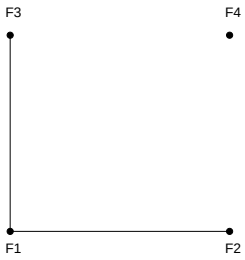


Figure 2.1: Graph for Requirement Set $S = \{F_1, F_2, F_3, F_4, F_1F_2, F_1F_3\}$

and $p \leq n$; n is the run size and X is the corresponding model matrix with component ± 1. The response Y is a column vector that includes the observed values of $Y_1, \ldots, Y_n$, and ε includes the unobserved stochastic components $\varepsilon_1, \ldots, \varepsilon_n$ and $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, where $1 \leq i \leq n$. Write the model matrix as $X = (I, X_1, X_2)$, where I is the column of all plus ones corresponding to the grand mean, X_1 is an orthogonal array of n runs for m factors, and X_2 denotes the columns corresponding the interaction terms.

The least square estimator of β is $\hat{\beta} = (X^TX)^{-1}X^TY$, which follows a joint normal distribution, $\hat{\beta} \sim N(\beta, \sigma^2(X^TX)^{-1})$. The fitted values are $\hat{Y} = X(X^TX)^{-1}X^TY$, and the hat matrix is given by $H = X(X^TX)^{-1}X^T$, which is symmetric and idempotent. The information matrix is X^TX, where X^T is the transpose of X. The parameters in the linear regression can be estimated if and only if the inverse of information matrix $(X^TX)^{-1}$ exists. In other words, a design supports a requirement set if and only if the determinant of the corresponding information matrix is strictly positive, that is, $|X^TX| > 0$.

2.1.3 *D*-Optimal Criterion

For a specified model, there are several well-known optimality criteria available for choosing optimal designs. The information matrix and hat matrix are important for the specified model, because they reflect the variability of the parameter estimates

and the fitted values, respectively. The criteria related to the information matrix X^TX include D-optimal criterion, A-optimal criterion and E-optimal criterion. The D-optimal design maximizes the determinant of the information matrix of this design, which results in minimizing the variation of the parameter estimates. The A-optimal criterion chooses a design that minimizes the sum of the variances of the estimated parameters for the model, which is the same as minimizing the sum of the diagonal elements, or the trace of the inverse of the information matrix. As a less known criterion, E-optimal criterion uses the minimum of eigenvalues of the information matrix to select optimal designs. The popular criteria concerned with prediction variances are G-optimal criterion and I-optimal criterion. The G-optimal design minimizes the maximum value in the diagonal of the hat matrix, which minimizes the maximum variance of the predicted values. The I-optimal criterion seeks the design that minimizes the average prediction variance. Among these criteria, D-optimal criterion is the most popular criterion for generating optimal designs, and it seeks the design that minimizes the variances of estimated parameters in the pre-specified model. So we employ D-optimal criterion to select optimal designs.

D-optimal designs are given by maximizing $|X^TX|$, the determinant of the information matrix X^TX. For an easy comparison of design efficiency, we use the following standardized D-efficiency:

$$D\text{-efficiency} = |X^TX/n|^{(1/p)}, \tag{2.2}$$

where p is the number of parameters in the pre-specified model, and n is the run size. The range of this measure is from 0 to 1, where 1 means that the corresponding design achieves full efficiency and 0 indicates that the parameters in our pre-specified model cannot be estimated. For a requirement set, an orthogonal design achieves full efficiency. The higher the D-efficiency of a design is, the smaller the standard errors of estimated parameters are. All effects in a design with D-efficiency equal to 1 are uncorrelated and the estimates of parameters in the linear model have the smallest standard errors. When the D-efficiency decreases, some of the parameter estimates are correlated. When the D-efficiency is low, the standard errors of parameter estimates become large, which makes our estimation of parameters less reliable.

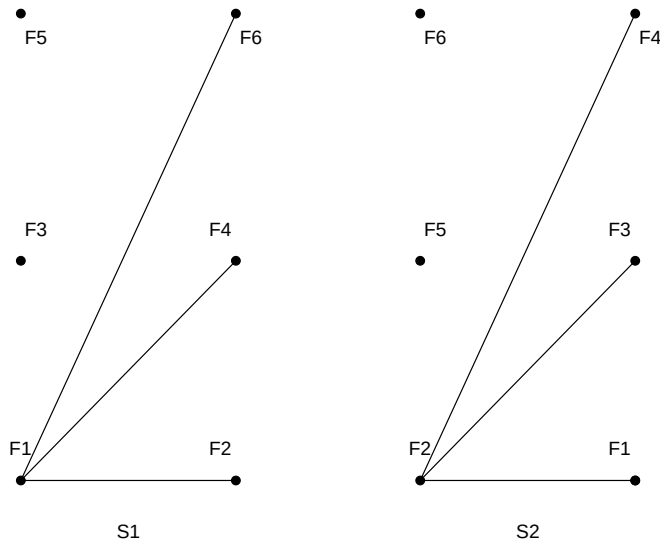


Figure 2.2: Graphs for Models S_1 and S_2

2.1.4 Models with Up to Three 2fi's

We would like to find optimal designs for all requirement sets containing all main effects and some important 2fi's. Consider the requirement sets containing m factors and e 2fi's. There are $N = \binom{m}{2}$ 2fi's in all. Since any e 2fi's out of all N 2fi's can be included, the total number of the models or requirement sets is $\binom{N}{e}$, which is a very large number even for moderate m and e. For example, for $m = 9$ and $e = 3$, we have $N = \binom{9}{2} = 36$ 2fi's and $\binom{36}{3} = 7140$ models with three 2fi's. We notice that some of those models have essentially the same structure. For example, consider two models given by requirement sets $S_1 = \{F_1, F_2, F_3, F_4, F_5, F_6, F_1F_2, F_1F_4, F_1F_6\}$ and $S_2 = \{F_1, F_2, F_3, F_4, F_5, F_6, F_1F_2, F_2F_3, F_2F_4\}$. The graph representations of these two models are shown in Figure 2.2. Figure 2.2 shows that the model for S_1 can be obtained from the model for S_2 by relabeling these 6 factors.

For the models containing up to three 2fi's, Ke and Tang (2003) provided all different models, and their graphical representations are displayed in Figure 2.3. Note

that the structure of a model only depends on those non-isolated vertices representing interacting factors. The interacting factors are defined as the factors that occur in 2fi's in the given requirement set. In Figure 2.3, the isolated vertices are deleted deliberately and only those interacting factors are included in the graphs. For any given requirement set containing up to three 2fi's, we can find a unique corresponding graphical representation from Figure 2.3.

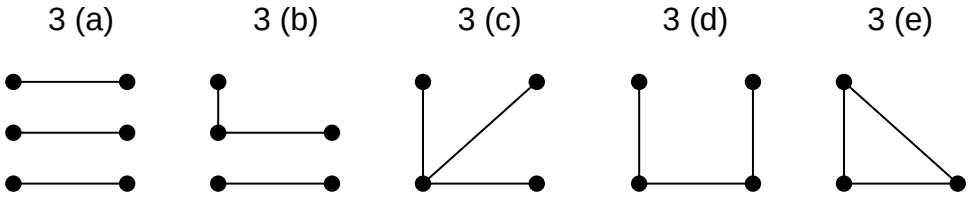
2.2 Complete Search

2.2.1 Method of Complete Search

A direct method of finding D-optimal orthogonal arrays is to search for the D-optimal design from all possible designs for a specified model. We focus our search on those models containing up to three 2fi's, as given in Figure 2.3. For a specified model, all possible designs of n runs can be constructed by assigning m factors to m columns chosen from the saturated orthogonal arrays of n runs. However, different factor assignments may lead to the same designs for a specified model. For example, consider a saturated orthogonal array of 12 runs O for a requirement set $S = \{F_1, F_2, F_3, F_4, F_1F_2, F_1F_3\}$ represented by model 2(b) in Figure 2.3. Let $O = (d_1, d_2, \ldots, d_{11})$ and four columns $\{d_1, d_2, d_3, d_4\}$ are chosen for these four factors. If we assign factor F_1, F_2, F_3 to column d_1, d_2, d_3, respectively, the model matrix would be $X = (I, d_1, d_2, d_3, d_4, d_1d_2, d_1d_3)$, where d_1d_2 denotes the product of column d_1 and d_2 and similarly for d_1d_3. The same design can also be obtained by assigning F_1, F_3, F_2 to column d_1, d_2, d_3, respectively. For a chosen set of m columns, the total number of factor assignments is $m!$, but some designs are the same even though factor assignments are different. In our complete search, we only consider all different designs obtained by factor assignments to further simplify our search of optimal designs. For example, model 2(a) in Figure 2.3 contains two 2fi's that have no common factor. There are $\binom{m}{2}$ ways to assign the first two factors occurring in one 2fi; and there are $\binom{m-2}{2}$ ways to assign the other two factors occurring in the other 2fi. The order of these two 2fi's does not matter, so the total number of assignments is $\binom{m}{2}\binom{m-2}{2}/2!$. For model 3(c) in Figure 2.3 containing three 2fi's that have one common factor, the number of ways of assigning this common factor is $\binom{m}{1}$; and the number of ways of assigning the other three interacting factors is $\binom{m-1}{3}$. So the total number of assignments that we need to consider is $\binom{m}{1}\binom{m-1}{3}$. Now we will use the following example

A. Graph for Model With One Two−Factor Interaction.

B. Graphs for Model With Two Two−Factor Interactions.



C. Graphs for Model With Three Two−Factor Interactions.

Figure 2.3: Graphs for Models Containing Up to Three 2fi's

to demonstrate the method of complete search for a specified model.

Example 2. Suppose that we want to find the D-optimal design of run size 20 for the requirement set $\{F_1, F_2, F_3, F_4, F_1F_2, F_1F_3\}$, which is represented by model 2(b) in Figure 2.3. First, three saturated two-level orthogonal arrays of 20 runs are generated from all three normalized Hadamard matrices of order 20. Then, all possible designs for this requirement set can be constructed as follows. Four columns can be chosen from 19 columns in a saturated orthogonal array in $\binom{19}{4}$ possible ways for those four factors F_1, F_2, F_3 and F_4. For the common factor F_1 in two 2fi's F_1F_2 and F_1F_3, there are $\binom{4}{1}$ ways to assign it to one column from the four selected columns, then the other two interacting factors can be assigned to the two columns chosen from the remaining three columns in $\binom{3}{2}$ different ways. Thus, the four factors are assigned in $\binom{4}{1}\binom{3}{2} = 12$ different ways. There are $3\binom{19}{4}\binom{4}{1}\binom{3}{2} = 139536$ designs constructed from all three orthogonal arrays of 20 runs. The complete search is implemented by searching from all 139536 candidate designs available for finding the optimal ones. For each candidate design, the corresponding D-efficiency is obtained by calculating $|X^TX/20|^{(1/7)}$, where X is the model matrix for this requirement set.

Using the above method, we have obtained the optimal designs of $n = 12$ and 20 runs for all models containing up to three 2fi's. The results are discussed in Section 2.2.2.

2.2.2 Results of Complete Search

Up to isomorphism, there are one unique Hadamard matrix of order 12 and three Hadamard matrices of order 20. Based on the unique Hadamard matrix of order 12, optimal designs of 12 runs for models containing m ($2 \le m \le 10$) main effects and one 2fi are tabulated in Table 2.1. Tables 2.2 and 2.3 provide optimal designs of 12 runs for models with two and three important 2fi's, respectively. We give an explanation for Table 2.2. In Table 2.2, the first column gives the number m of factors. Entries under "Model" give information on which model is under consideration. The notation in Ke and Tang (2003) is used here. For example, the entry 2(a) denotes the model 2(a) in Figure 2.3. This model requires at least 4 factors to form the two 2fi's, and a design of 12 runs can accommodate at most 9 factors because of the grand mean and two 2fi's. So the number m of factors must satisfy $4 \le m \le 9$ for model 2(a). For model 2(b) in Figure 2.3, the number m of factors must satisfy $3 \le m \le 9$ because this model only requires three factors for two 2fi's. The column of "Had" shows which Hadamard

matrix is used to generate the saturated orthogonal array. The Hadamard matrix had.12 is from N.J.A Sloane's homepage (http://www.research.att.com/ñjas/). Since there is one unique Hadamard matrix of order 12, all optimal designs of 12 runs are constructed from had.12. The entries under "Selected columns" give a set of design columns for the corresponding optimal designs. For example, $\{1, 2, 3, 4\}$ means that the first four columns of had.12 are selected to form the optimal design and $\{1, 2, 3, 4, 5, 6, 7\}^c$ means that the set of columns $\{8, 9, 10, 11\}$ is selected, which is a complement of $\{1, 2, \ldots, 7\}$ within the set $\{1, 2, \ldots, 11\}$. The column of "2fi's" shows how to assign the factors involved in the important 2fi's. The optimal design can be constructed based on the information in "Model", "Selected columns" and "2fi's." The corresponding D-efficiency and bias are calculated. The precise meaning of the bias in the tables will be given in Section 2.3.

Tables 2.4 and 2.5 present optimal designs of 20 runs for models with one and two important 2fi's, respectively. The optimal designs for models containing three 2fi's are provided in Table 2.6. In Tables 2.4, 2.5 and 2.6, the Hadamard matrices I, II and III correspond to had.20.pal, had.20.will and had.20.toncheviv, respectively, in N.J.A Sloane's homepage (http://www.research.att.com/ñjas/). Let us look at how to construct the optimal design for a special case.

Example 3. Suppose that we want to use a two-level design of 20 runs for estimating a model with 12 factors and two 2fi's represented by model 2(b) in Figure 2.3. The optimal design for this model can be constructed by using the information in Table 2.5. From the row for $m = 12$ and model 2(b) in Table 2.5, we see that the optimal design is obtained from the Hadamard matrix II, and the set of design columns under "Selected columns" is $\{1, 2, 3, 10, 11, 14, 19\}^c$, which means that its complement $\{4, 5, 6, 7, 8, 9, 12, 13, 15, 16, 17, 18\}$ is selected. To construct the optimal design, we need to appropriately assign the three factors in the 2fi's to the three selected columns $\{7, 16, 17\}$. The common factor in these two 2fi's is assigned to column $\{7\}$, and the other two interacting factors can be arbitrarily assigned to columns $\{16, 17\}$. We can arbitrarily assign the other factors to the remaining nine columns $\{4, 5, 6, 8, 9, 12, 13, 15, 18\}$. The D-efficiency for this optimal design is 0.93 and the corresponding bias is 7.17.

For each model in the tables, the results show that the D-efficiency decreases when the number of factors increases. Since our designs are all orthogonal arrays, which guarantee that all main effects are orthogonal to each other, the loss of efficiency

Table 2.1: Optimal Designs of 12 runs for Model With One 2fi

m	Had	Selected columns	2fi	D-efficiency	Bias
2	had.12	{1,2}	(1,2)	1	0
3	had.12	{1,2,3}	(1,2)	0.98	0.47
4	had.12	{1,2,3,4}	(1,2)	0.96	1.24
5	had.12	{1,2,3,4,5}	(1,2)	0.94	2.1
6	had.12	$\{1,2,3,4,5\}^c$	(6,10)	0.93	3.00
7	had.12	$\{1,2,3,4\}^c$	(5,6)	0.91	4.26
8	had.12	$\{1,2,3\}^c$	(4,6)	0.9	5.69
9	had.12	$\{1,2\}^c$	(3,4)	0.87	7.67
10	had.12	$\{1\}^c$	(2,3)	0.83	11.15

Table 2.2: Optimal Designs of 12 Runs for Models With Two 2fi's

m	Model	Had	Selected columns	2fi	D-efficiency	Bias
3	2(a)	had.12	-	-	-	-
	2(b)	had.12	{1,2,3}	(1,2)(1,3)	0.96	0.33
4	2(a)	had.12	{1,2,3,4}	(1,2)(3,4)	0.90	1.09
	2(b)	had.12	{1,2,3,4}	(1,2)(1,3)	0.93	1.28
5	2(a)	had.12	{1,2,3,4,5}	(1,2)(3,4)	0.89	2.34
	2(b)	had.12	{1,2,3,4,5}	(1,2)(1,3)	0.89	2.29
6	2(a)	had.12	$\{1,2,3,4,5\}^c$	(6,7)(9,10)	0.87	3.55
	2(b)	had.12	$\{1,2,3,4,5\}^c$	(6,10)(7,10)	0.87	3.49
7	2(a)	had.12	$\{1,2,3,4\}^c$	(5,7)(8,10)	0.85	5.04
	2(b)	had.12	$\{1,2,3,4\}^c$	(5,6)(6,7)	0.85	4.97
8	2(a)	had.12	$\{1,2,3\}^c$	(4,5)(8,9)	0.81	7.06
	2(b)	had.12	$\{1,2,3\}^c$	(4,5)(4,6)	0.81	7.21
9	2(a)	had.12	$\{1,2\}^c$	(3,4)(5,6)	0.78	9.50
	2(b)	had.12	$\{1,2\}^c$	(3,4)(3,5)	0.78	9.50

Note: A row with - indicates the situation where the specified model does not exist for the given number m of factors.

Table 2.3: Optimal Designs of 12 Runs for Models With Three 2fi's

m	Model	Had	Selected columns	2fi	D-efficiency	Bias
3	3(a)	had.12	-	-	-	-
	3(b)	had.12	-	-	-	-
	3(c)	had.12	-	-	-	-
	3(d)	had.12	-	-	-	-
	3(e)	had.12	{1,2,3}	(1,2)(1,3)(2,3)	0.95	0
4	3(a)	had.12	-	-	-	-
	3(b)	had.12	-	-	-	-
	3(c)	had.12	{1,2,3,4}	(1,2)(1,3)(1,4)	0.90	1.19
	3(d)	had.12	{1,2,3,4}	(1,2)(1,3)(2,4)	0.88	1.11
	3(e)	had.12	{1,2,3,4}	(1,2)(1,3)(2,3)	0.90	1.19
5	3(a)	had.12	-	-	-	-
	3(b)	had.12	{1,2,3,4,5}	(1,2)(3,4)(4,5)	0.84	2.51
	3(c)	had.12	{1,2,3,4,5}	(1,2)(1,3)(1,4)	0.86	2.52
	3(d)	had.12	{1,2,3,4,5}	(1,2)1,3)(2,4)	0.84	2.49
	3(e)	had.12	{1,2,3,4,5}	(1,2)(1,3)(2,3)	0.86	2.53
6	3(a)	had.12	$\{1,2,3,4,6\}^c$	(5,7)(8,10)(9,11)	0.83	4.17
	3(b)	had.12	$\{1,2,3,4,5\}^c$	(6,7)(8,9)(9,10)	0.83	4.05
	3(c)	had.12	$\{1,2,3,4,5\}^c$	(6,10)(7,10)(8,10)	0.83	3.96
	3(d)	had.12	$\{1,2,3,4,5\}^c$	(6,7)(6,10)(8,10)	0.83	3.98
	3(e)	had.12	$\{1,2,3,4,5\}^c$	(6,7)(6,10)(7,10)	0.83	4.00
7	3(a)	had.12	$\{1,2,3,4\}^c$	(5,7)(9,10)(6,11)	0.80	5.73
	3(b)	had.12	$\{1,2,3,4\}^c$	(5,7)(8,10)(8,11)	0.80	5.70
	3(c)	had.12	$\{1,2,3,4\}^c$	(5,6)(6,7)(6,9)	0.80	5.60
	3(d)	had.12	$\{1,2,3,4\}^c$	(5,7)(5,8)(7,9)	0.80	5.68
	3(e)	had.12	$\{1,2,3,4\}^c$	(5,7)(5,9)(7,9)	0.80	5.67
8	3(a)	had.12	$\{1,2,3\}^c$	(4,5)(6,7)(8,9)	0.73	9.54
	3(b)	had.12	$\{1,2,3\}^c$	(4,5)(7,8)(8.9)	0.73	8.97
	3(c)	had.12	$\{1,2,3\}^c$	(4,5)(4,6)(4,10)	0.73	9.43
	3(d)	had.12	$\{1,2,3\}^c$	(4,5)(4,6)(6,8)	0.73	9.22
	3(e)	had.12	$\{1,2,3\}^c$	(4,5)(4,10)(5,10)	0.73	9.44

Note: A row with - indicates the situation where the specified model does not exist for the given number m of factors.

Table 2.4: Optimal Designs of 20 Runs for Model With One 2fi

m	Had	Selected columns	2fi	D-efficiency	Bias
2	I	{1,2}	(1, 2)	1	0
3	I	{1,2,3}	(1,2)	0.99	0.28
4	I	{1,2,3,4}	(1,2)	0.99	0.69
5	I	{1,2,3,4,8}	(1,2)	0.98	1.09
6	I	{1,2,3,4,8,10}	(4,10)	0.98	1.60
7	II	{1,2,3,4,5,10,11}	(3,11)	0.98	2.15
8	II	{1,2,3,4,6,7,8,9}	(6,9)	0.97	2.86
9	II	{5,6,7,8,9,10,11,12,13}	(5,6)	0.97	3.62
10	II	$\{1, 2, 3, 4, 15, 16, 17, 18, 19\}^c$	(5,6)	0.97	4.31
11	II	$\{1, 2, 3, 4, 15, 17, 18, 19\}^c$	(5,9)	0.97	5.55
12	II	$\{1, 2, 5, 6, 7, 11, 16\}^c$	(3,17)	0.96	6.56
13	III	$\{1, 2, 3, 8, 9, 10\}^c$	(4,6)	0.96	7.64
14	II	$\{1, 2, 6, 7, 11\}^c$	(4,9)	0.96	8.78
15	I	$\{1, 2, 3, 14\}^c$	(8,9)	0.96	9.88
16	III	$\{1, 10, 19\}^c$	(2,3)	0.96	11.04
17	I	$\{1, 2\}^c$	(3,14)	0.95	12.29
18	I	$\{1\}^c$	(2,13)	0.95	13.59

comes from the nonorthogonality between main effects and 2fi's or between 2fi's. The more factors we have in the specified model, the more nonorthogonality occurs between the added main effects and 2fi's. This explains why the D-efficiency is low for models containing a large number of factors. Those results also show that the more 2fi's we have in the specified model, the lower the D-efficiency is. Our search results also show that orthogonal arrays do provide good designs with high D-efficiency for models containing a small number of 2fi's, and this is expected.

Two problems arise during the process of complete search. The first problem is that the computational load is too large and becomes impractical for finding optimal designs of larger run sizes. We find that the complete search method is doable for designs up to 20 runs, but it is already very time-consuming for some cases of 20 runs. For example, for a requirement set containing 7 factors and 3 2fi's in the structure of model 3(a) shown in Figure 2.3, there are $\binom{11}{7}\binom{7}{2}\binom{5}{2}\binom{3}{2}/3! = 207,900$ possible designs of run size 12, but the total number of candidate designs of 20 runs is $3\binom{19}{7}\binom{7}{2}\binom{5}{2}\binom{3}{2}/3! = 1.5 \times 10^7$. The total number of candidate designs of 24 runs increases to $60\binom{23}{7}\binom{7}{2}\binom{5}{2}\binom{3}{2}/3! = 1.5 \times 10^9$. So optimal designs of larger run sizes

Table 2.5: Optimal Designs of 20 Runs for Models With Two 2fi's

m	Model	Had	Selected columns	2fi	D-efficiency	Bias
3	2(a)	-	-	-	-	-
	2(b)	I	{1,2,3}	(1,2) (1,3)	0.99	0.2
4	2(a)	I	{1,2,3,4}	(1,2) (3,4)	0.97	0.62
	2(b)	I	{1,2,3,4}	(1,2) (1,3)	0.98	0.68
5	2(a)	II	{1,2,3,4,10}	(1,3) (4,10)	0.96	1.11
	2(b)	I	{1,2,3,4,8}	(1,2) (1,8)	0.97	1.08
6	2(a)	I	{1,2,3,4,8,10}	(1,2) (8,10)	0.96	1.63
	2(b)	I	{1,2,3,4,8,10}	(4,10) (8,10)	0.96	1.67
7	2(a)	I	{1,2,3,4,7,8,10}	(3,8)(7,10)	0.96	2.41
	2(b)	II	{1,2,3,4,5,10,11}	(3,5) (3,11)	0.96	2.34
8	2(a)	II	{1,2,3,4,6,7,8,9}	(6,9) (7,8)	0.95	3.13
	2(b)	II	{1,2,3,6,7,8,9,14}	(8,9) (8,14)	0.95	3.12
9	2(a)	II	{5,6,7,8,9,10,11,12,13}	(5,6) (10,12)	0.95	4.08
	2(b)	II	{5,6,7,8,9,10,11,12,13}	(5,6) (5,7)	0.95	4.02
10	2(a)	II	$\{1,2,3,4,15,16,17,18,19\}^c$	(5,6) (7,8)	0.94	4.94
	2(b)	II	$\{1,2,3,4,15,16,17,18,19\}^c$	(5,6) (5,7)	0.94	4.80
11	2(a)	II	$\{1,2,3,5,6,7,11,16\}^c$	(9,14) (12,17)	0.94	6.12
	2(b)	II	$\{1,2,3,4,15,16,17,18\}^c$	(5,6) (5,7)	0.94	6.10
12	2(a)	I	$\{1,2,3,4,14,15,18\}^c$	(11,16) (12,19)	0.93	7.22
	2(b)	III	$\{1,2,3,10,11,14,19\}^c$	(7,16) (7,17)	0.93	7.17
13	2(a)	II	$\{1,2,3,8,10,13\}^c$	(4,19) (12,16)	0.93	8.34
	2(b)	II	$\{1,2,5,6,7,11\}^c$	(4,9) (4,10)	0.93	8.35
14	2(a)	III	$\{2,3,11,12,13\}^c$	(8,19) (10,17)	0.93	9.51
	2(b)	III	$\{2,3,6,12,18\}^c$	(10,15) (15,16)	0.93	9.55
15	2(a)	I	$\{1,2,3,8\}^c$	(6,12) (10,11)	0.92	10.79
	2(b)	I	$\{1,2,3,11\}^c$	(6,8) (6,12)	0.92	10.79
16	2(a)	I	$\{1,2,3\}^c$	(5,11) (8,17)	0.92	12.10
	2(b)	I	$\{1,2,5\}^c$	(8,9) (9,15)	0.92	12.12
17	2(a)	I	$\{1,2\}^c$	(3,14) (5,11)	0.91	13.49
	2(b)	I	$\{1,2\}^c$	(3,14) (3,17)	0.91	13.49

Note: A row with - indicates the situation where the specified model does not exist for the given number m of factors.

Table 2.6: Optimal Designs of 20 Runs for Models With Three 2fi's

m	Model	Had	Selected columns	2fi	D-efficiency	Bias
3	3(a)	-	-	-	-	-
	3(b)	-	-	-	-	-
	3(c)	-	-	-	-	-
	3(d)	-	-	-	-	-
	3(e)	I	{1,2,3}	(1,2)(1,3)(2,3)	0.98	0
4	3(a)	-	-	-	-	-
	3(b)	-	-	-	-	-
	3(c)	I	{1,2,3,4}	(1,2)(1,3)(1,4)	0.97	0.67
	3(d)	I	{1,2,3,4}	(1,2)(1,3)(2,4)	0.96	0.60
	3(e)	I	{1,2,3,4}	(1,2)(1,3)(2,3)	0.98	0.64
5	3(a)	-	-	-	-	-
	3(b)	I	{1,2,3,4,10}	(1,3)(2,4)(4,10)	0.95	1.16
	3(c)	I	{1,2,3,4,8}	(1,2)(1,3)(1,8)	0.96	1.17
	3(d)	I	{1,2,3,4,5}	(1,2)(2,3)(1,4)	0.95	1.14
	3(e)	I	{1,2,3,4,10}	(1,2)(1,3)(2,3)	0.96	1.23
6	3(a)	I	{1,2,4,5,10,12}	(1,3)(2,10)(5,12)	0.94	1.74
	3(b)	I	{1,2,3,4,8,10}	(1,4)(2,10)(8,10)	0.95	1.90
	3(c)	I	{1,2,3,4,8,10}	(1,10)(4,10)(8,10)	0.95	1.81
	3(d)	I	{1,2,3,4,8,10}	(1,4)(4,10)(8,10)	0.95	1.78
	3(e)	I	{1,2,3,4,5,8}	(1,5)(1,8)(5,8)	0.95	1.86
7	3(a)	II	{1,2,5,9,13,15,19}	(1,19)(2,15)(9,13)	0.94	2.53
	3(b)	II	{1,2,3,4,5,10,11}	(2,4)(3,5)(3,11)	0.94	2.56
	3(c)	I	{1,2,3,4,7,8,10}	(4,10)(7,10)(8,10)	0.94	2.50
	3(d)	II	{1,2,3,4,5,10,11}	(3,5)(3,11)(4,11)	0.94	2.69
	3(e)	I	{1,2,3,4,7,8,12}	(3,7)(3,12)(7,12)	0.94	2.59
8	3(a)	I	{1,2,3,7,8,11,16,18}	(1,11)(2,3)(8,18)	0.93	3.38
	3(b)	II	{1,2,3,4,6,7,8,9}	(4,7)(6,9)(7,8)	0.93	3.42
	3(c)	II	{1,2,3,6,7,8,9,14}	(6,7)(6,8)(6,14)	0.93	3.43
	3(d)	II	{1,2,4,6,7,8,9,13}	(6,8)(6,13)(7,8)	0.93	3.52
	3(e)	II	{1,2,3,4,6,7,8,9}	(6,7)(6,8)(7,8)	0.93	3.48
9	3(a)	II	{1,2,3,4,5,10,11,14,17}	(1,3)(2,14)(10,11)	0.93	4.42
	3(b)	II	{1,2,3,4,6,7,8,11,19}	(1,6)(6,7)(8,11)	0.93	4.48
	3(c)	II	{5,6,7,8,9,10,11,12,13}	(5,6)(5,7)(5,12)	0.93	4.39
	3(d)	II	{1,2,3,5,9,10,11,14,17}	(2,14)(10,11)(11,14)	0.93	4.62
	3(e)	II	{1,2,3,4,5,10,11,14,17}	(1,3)(1,11)(3,11)	0.93	4.40

Note: A row with - indicates the situation where the specified model does not exist for the given number m of factors.

m	Model	Had	Selected columns	2fi	D-efficiency	Bias
10	3(a)	III	$\{1,2,3,9,10,11,15,18,19\}^c$	(4,16)(6,12)(7,14)	0.92	5.41
	3(b)	II	$\{1,2,3,9,10,13,14,16,17\}^c$	(5,6)(8,11)(11,12)	0.92	5.37
	3(c)	II	$\{1,2,3,4,15,16,17,18,19\}^c$	(5,6)(5,7)(5,8)	0.92	5.25
	3(d)	II	$\{1,2,3,4,15,16,17,18,19\}^c$	(5,6)(5,9)(6,13)	0.92	5.58
	3(e)	II	$\{1,2,3,4,15,16,17,18,19\}^c$	(6,7)(6,8)(7,8)	0.92	5.40
11	3(a)	III	$\{2,3,4,7,12,13,14,19\}^c$	(5,9)(8,10)(17,18)	0.91	6.52
	3(b)	I	$\{1,2,3,4,5,11,13,16\}^c$	(6,8)(9,10)(9,19)	0.91	6.62
	3(c)	II	$\{1,2,3,4,15,16,17,18\}^c$	(5,6)(5,7)(5,8)	0.91	6.59
	3(d)	I	$\{2,4,5,7,9,10,13,16\}^c$	(3,14)(8,11)(11,14)	0.91	6.79
	3(e)	III	$\{1,3,6,7,14,15,16,17\}^c$	(4,18)(4,19)(18,19)	0.91	6.63
12	3(a)	II	$\{1,2,3,6,8,10,18\}^c$	(4,15)(11,19)(16,17)	0.91	7.68
	3(b)	II	$\{1,2,3,5,8,10,13\}^c$	(4,19)(12,16)(12,17)	0.91	7.75
	3(c)	III	$\{1,2,3,4,11,14,15\}^c$	(7,9)(9,16)(9,17)	0.91	7.78
	3(d)	II	$\{1,3,5,6,8,12,17\}^c$	(4,10)(4,11)(7,11)	0.91	7.91
	3(e)	II	$\{1,2,3,5,6,7,8\}^c$	(9,14)(9,17)(14,17)	0.91	7.77
13	3(a)	II	$\{1,4,5,8,11,18\}^c$	(2,19)(12,17)(14,15)	0.90	8.89
	3(b)	II	$\{1,2,3,8,10,13\}^c$	(4,19)(9,16)(12,16)	0.90	8.98
	3(c)	II	$\{1,2,3,6,9,17\}^c$	(8,11)(8,13)(8,18)	0.90	9.01
	3(d)	I	$\{1,2,3,4,14,15\}^c$	(5,7)(5,9)(8,9)	0.90	9.00
	3(e)	II	$\{1,2,3,5,6,7\}^c$	(9,14)(9,17)(14,17)	0.90	8.99
14	3(a)	II	$\{1,2,6,7,17\}^c$	(3,15)(8,18)(10,13)	0.90	10.15
	3(b)	I	$\{1,2,3,4,8\}^c$	(12,16)(12,19)(13,17)	0.90	10.26
	3(c)	II	$\{1,2,3,10,13\}^c$	(9,12)(9,14)(9,16)	0.90	10.27
	3(d)	III	$\{1,2,4,6,8\}^c$	(14,16)(15,16)(15,18)	0.90	10.25
	3(e)	I	$\{1,2,3,7,10\}^c$	(5,8)(5,19)(8,19)	0.90	10.30
15	3(a)	II	$\{2,6,7,13\}^c$	(1,11)(4,17)(9,16)	0.89	10.16
	3(b)	I	$\{1,2,3,11\}^c$	(4,6)(6,12)(9,18)	0.89	11.60
	3(c)	I	$\{1,2,3,11\}^c$	(4,6)(6,8)(6,12)	0.89	11.59
	3(d)	I	$\{1,2,3,4\}^c$	(10,11)(11,12)(12,19)	0.89	11.61
	3(e)	I	$\{1,2,4,9\}^c$	(5,11)(5,15)(11,15)	0.89	11.62
16	3(a)	I	$\{1,3,16\}^c$	(2,13)(4,15)(11,14)	0.88	12.96
	3(b)	I	$\{1,2,4\}^c$	(5,16)(9,10)(10,14)	0.88	13.06
	3(c)	I	$\{1,2,3\}^c$	(6,12)(12,16)(12,19)	0.88	13.11
	3(d)	I	$\{1,2,5\}^c$	(8,9)(9,15)(11,15)	0.88	13.09
	3(e)	II	$\{1,7,8\}^c$	(4,12)(4,14)(12,14)	0.88	13.12

Note: A row with - indicates the situation where the specified model does not exist for the given number m of factors.

are computationally infeasible to obtain. Motivated by this problem, an algorithm without an exhaustive search of all possible designs is desired for designs with large run sizes. The second problem is that the complete search gives us a lot of designs with the same highest D-efficiency for a specified model. In the next section, we will use another criterion to distinguish those good designs with the highest D-efficiency.

2.3 Bias Consideration

The complete search provides many designs that have the same highest D-efficiency for a given requirement set. This offers us an opportunity of using a secondary criterion for design selection. The unbiasness and small variance are two desirable statistical properties. The variance of the parameter estimator is minimized by D-optimal criterion, our major criterion of design selection. The bias is not considered yet and will be used as a secondary criterion. In finding designs that allow joint estimation of all main effects and some important 2fi's, we have assumed that all other 2fi's and higher order interactions are negligible. Sometimes, we are not so confident that our assumption is true, so a reasonable criterion to consider is the biases of the estimates of the parameters caused by these non-negligible effects. To make our case simple, we consider the situation that all three or higher order interactions are negligible. The true model is now written as

$$Y = X\beta + W\gamma + \varepsilon, \tag{2.3}$$

where γ is the vector of those 2fi's that we are not interested in estimating but may be non-negligible and W is the corresponding matrix. The least square estimate of β from the model in equation (2.1) is $\hat{\beta} = (X^TX)^{-1}X^TY$ and the expectation of $\hat{\beta}$ taken under the true model in equation (2.3) is $E(\hat{\beta}) = \beta + B\gamma$, where $B = (X^TX)^{-1}X^TW$. So the bias of least square estimate of β is $\text{bias}(\hat{\beta}, \beta) = B\gamma$, which is the product of the unknown parameter and matrix B. To quantify the matrix $B = (b_{ij})$, we consider $||B|| \stackrel{\text{def}}{=} \sqrt{\sum_{i,j} b_{ij}^2}$ as the size measure for this matrix. In this thesis, the bias measured by $||B||$ is used as the secondary criterion in the selection of optimal designs. The entries under "Bias" in Tables 2.1, 2.2, 2.3, 2.4, 2.5 and 2.6 are the smallest values of the bias among all designs with the same highest D-efficiency.

Chapter 3

Algorithmic Search

3.1 Theoretical Results

Tang and Zhou (2009) considered using two-level orthogonal arrays for jointly estimating all main effects and a set of specified two-factor interactions. They presented some theoretical results about the existence and construction of such designs.

3.1.1 Core Requirement Set

For a given requirement set S, a new concept of core requirement set $C(S)$ was proposed to denote a subset of S, which is obtained by keeping 2fi's and all interacting factors. Let the requirement set S consist of m main effects and e two-factor interactions, and its core $C(S)$ consist of m_1 main effects out of the m main effects and all the e 2fi's in S. A graph can be drawn to represent a requirement set, and the core of a requirement set can be obtained by simply deleting all isolated vertices. Every requirement set has a unique core.

Example 4. For the requirement set $S_1 = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_1F_2, F_1F_3, F_2F_3\}$, the core of S_1 is $C(S_1) = \{F_1, F_2, F_3, F_1F_2, F_1F_3, F_2F_3\}$. To obtain $C(S_1)$, the main effects F_4, F_5, F_6 and F_7 are deleted because they do not occur in any 2fi's. Similarly, for $S_2 = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_1F_2, F_2F_3, F_3F_4\}$, its core is $C(S_2) = \{F_1, F_2, F_3, F_4, F_1F_2, F_2F_3, F_3F_4\}$ after deleting factors F_5, F_6, F_7 and F_8.

3.1.2 Existence and Construction of Designs

Since the core requirement set $C(S)$ is a subset of the corresponding requirement set S, we can always find an orthogonal array that supports $C(S)$ if an orthogonal array that supports S exists. One of the theoretical results in Tang and Zhou (2009) says that an orthogonal array that supports S must exist if there is an orthogonal array that supports $C(S)$. A method is provided for constructing designs for S from a design that supports its core $C(S)$. Let $O = (D_1, D_2, D_3)$ be a saturated orthogonal array of n runs obtained by removing the first column from a normalized Hadamard matrix. Suppose that a subarray D_1 with m_1 columns allows joint estimation of all the effects in the core requirement set $C(S)$ of a given requirement set S. Let D_2 have $m_2 = m - m_1$ columns and D_3 have $m_3 = n - 1 - m$ columns. Let X_2 denote the model matrix for the e 2fi's. The result of Tang and Zhou (2009) says that there must exist a design $D = (D_1, D_2)$ that supports the given requirement set S as long as D_1 supports $C(S)$. Such a design can be obtained by choosing D_3 such that

$$|X_2^T D_3 D_3^T X_2| > 0. \tag{3.1}$$

So we can always obtain $D = (D_1, D_2)$ that supports S by deleting D_3 which satisfies (3.1) if D_1 supports $C(S)$. In other words, a design that supports S can be generated by adding $m_2 = m - m_1$ columns chosen from (D_2, D_3) to design D_1 that supports $C(S)$. There are $\binom{n-1-m_1}{n-1-m} = \binom{n-1-m_1}{m-m_1}$ possible candidates for D_3 or D_2. An example is given to explain the implementation of this theoretical result.

Example 5. Suppose that the experimenter wants to use an orthogonal array of 12 runs for the requirement set $S = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_1F_2, F_1F_3\}$. The core requirement set here is $C(S) = \{F_1, F_2, F_3, F_1F_2, F_1F_3\}$. Consider the 12 run saturated orthogonal array $O = (d_1, d_2, \ldots, d_{11})$ generated from Hadamara matrix of order 12 in N.J.A Sloane's homepage (http://www.research.att.com/ñjas/). A design $D_1 = (d_1, d_2, d_3)$ that supports $C(S)$ can be found by checking the determinant of corresponding information matrix. Then $X_2 = (d_1d_2, d_1d_3)$ and $(D_2, D_3) = (d_4, d_5, \ldots, d_{11})$. Because $D_3 = (d_4, d_7)$ satisfies $|X_2^T D_3 D_3^T X_2| > 0$, a design $D = (D_1, D_2) = (d_1, d_2, d_3, d_5, d_6, d_8, d_9, d_{10}, d_{11})$ that supports S is obtained after deleting D_3 from O. Since D_1 supports $C(S)$, the factors F_1, F_2, F_3 are assigned to columns d_1, d_2, d_3, respectively, and factors $F_4, F_5, F_6, F_7, F_8, F_9$ can be arbitrarily assigned to $d_5, d_6, d_8, d_9, d_{10}$ and d_{11}.

3.2 Algorithm

Having completely searched for the D-optimal designs of run sizes 12 and 20 for all models containing up to three 2fi's, we have found that the method of complete search can be done for designs of small run sizes but quickly becomes cumbersome. An effective search method is desirable for selecting the D-optimal designs of run sizes larger than 20. Tang and Zhou (2009) presented a theoretical result about the construction of the optimal orthogonal array for a requirement set S, conditionally on D_1 that supports $C(S)$. Inspired by this result, we propose a sequential algorithm for searching for optimal orthogonal arrays that support S. This algorithm consists of two steps:

Step 1. Find all designs that support the core set $C(S)$ and compute the D-efficiency for each of them. The top R designs in terms of D-efficiency are kept for the next step;

Step 2. Find the designs that support the requirement set S from each of R designs kept in Step 1 and then find the design with the highest D-efficiency.

The two-step procedure takes advantage of the fact that an orthogonal array supporting S exists if we can find an orthogonal array that supports its core $C(S)$. In Step 1, an exhaustive search is done for finding the top R designs that support $C(S)$. The value of R is a non-zero integer and the choice of number R is subjective. In Step 2, all possible designs that support S are generated from each of R designs kept in Step 1. A design with the highest D-efficiency is retained. This algorithm greatly reduces the computational effort by restricting the searching scope from all possible designs for S to the possible designs generated from the top R designs supporting $C(S)$. For example, suppose that we want to find the D-optimal design of 20 runs for S_1 containing 6 factors and three 2fi's represented by model 3(b) in Figure 2.3. From each of three saturated orthogonal arrays of 20 runs, we can obtain the candidate set $\binom{19}{6}\binom{6}{5}\binom{5}{2}\binom{3}{1}\binom{2}{2} = 4,883,760$ possible designs for S_1. In complete search, we go through all those designs to find the D-optimal design for S. In contrast, our algorithm only searches from all $\binom{19}{5}\binom{5}{2}\binom{3}{1}\binom{2}{2} = 348,840$ possible designs to find the designs that support $C(S_1)$ in Step 1 and keep the top R designs in terms of D-efficiency. Let $R = 3$, then we construct all $3 \times \binom{14}{1} = 42$ possible designs for S in Step 2. The algorithm only computes the D-efficiencies of all $\binom{19}{5}\binom{5}{2}\binom{3}{1}\binom{2}{2} + 3 \times \binom{14}{1} = 348,882$ designs. Furthermore, our algorithm is especially powerful for a requirement set S

containing a large number of factors and three or fewer 2fi's because the set $C(S)$ is much smaller than S itself. For example, suppose that the requirement set S_2 contains 15 factors and also can be represented by model 3(b) in Figure 2.3. The complete search searches the D-optimal design from all $\binom{19}{15}\binom{15}{5}\binom{5}{2}\binom{3}{1}\binom{2}{2} = 349,188,840$ possible designs obtained from one saturated orthogonal array of 20 runs. Our algorithm only goes through $\binom{19}{5}\binom{5}{2}\binom{3}{1}\binom{2}{2} + 3 \times \binom{14}{10} = 351,843$ designs in two steps to find the D-optimal design. When the number of factors increases, the number of possible designs directly obtained from orthogonal arrays increases quickly, and so the computational load of complete search increases substantially. However, the computational load in two steps of our algorithm does not change too much. As we can see that a lot of computation is reduced by searching for D-optimal designs from a portion of all designs that support S in our algorithm, so our algorithm is very efficient for D-optimal design selection of large run sizes.

An issue arising from our algorithm is whether the designs obtained are D-optimal. The best designs that support the requirement set S do not necessarily come from the best designs that support the core set $C(S)$, but they are more likely to be obtained from the good designs that support the core set $C(S)$. The efficiency loss in our designs is caused by the nonorthogonality between main effects and 2fi's or between those 2fi's themselves. If a design has the highest D-efficiency, there is least nonorthogonality between those effects in this design. In our algorithm, we select good designs that support $C(S)$ in Step 1, the loss of D-efficiency is minimized by finding the designs with the highest D-efficiency. For the construction of designs that support S in Step 2, the added columns chosen from the saturated orthogonal array are orthogonal. So only a little efficiency will be lost due to the nonorthogonality between the added main effects and those 2fi's in S. The design with the highest D-efficiency is chosen from all possible designs based on those good designs that support $C(S)$. Nonorthogonality between effects is minimized again in Step 2, so the designs obtained from our algorithm should have very high D-efficiency.

The two basic steps in the above algorithm require some elaboration and discussion. Since optimal designs for S are more likely from the good designs for $C(S)$, we can improve our algorithm by choosing more designs with higher D-efficiency in the step of selecting good designs that support $C(S)$. An improved algorithm is as follows:

Step 1. Find all designs that support the core set $C(S)$ obtained from a requirement set S and compute the D-efficiency for each of them;

Step 2. Keep w_1 designs with the highest D-efficiency, w_2 designs with the second highest D-efficiency and so on, where $w_1 > w_2 \ldots > w_k$ and k is a non-zero integer;

Step 3. Find the designs that support the requirement set S for each of the selected designs in Step 2 and then find the designs with the highest D-efficiency.

In Step 2, $w_1, w_2, \ldots, w_k$ are non-zero integers and k is the number of different values of D-efficiency we want to keep. For designs with the ith highest D-efficiency, we randomly choose w_i designs for the construction of designs that support S, where $i = 1, 2, \ldots, k$. Due to the nonorthogonality between effects in the design, a reasonable restriction $w_1 > w_2, \ldots > w_k > 0$ is considered for the improvement of our algorithm. The idea here is that designs with the same D-efficiency may not be the same. Ideally, we want to keep all different designs with high D-efficiencies in Step 2 for the construction of designs for S in Step 3, but this will make our algorithm more complicated. By using the bias to distinguish the designs with the same highest D-efficiency, our algorithm is further improved as follows:

Step 1. Find the designs that support the core of a requirement set $C(S)$ and compute the D-efficiency and bias for each of them;

Step 2. Keep w_1 designs with the highest D-efficiency, w_2 designs with the second highest D-efficiency and so on, where $w_1 > w_2 \ldots > w_k$ and k is a non-zero integer; After checking the biases of those designs with the ith D-efficiency, their biases take on j different values. For w_i designs, we randomly choose $w_{i1}, \ldots, w_{ij}$ designs with all j different biases and more designs with smaller biases, where $w_{i1} > \ldots > w_{ij} > 0$ and $w_{i1} + \ldots + w_{ij} = w_i$;

Step 3. Construct the designs that support the requirement set S for each of R designs and then find the designs with the highest D-efficiency.

Step 4. For each of those designs with the highest D-efficiency, we calculate its bias defined in Section 2.3, and then select only one design with the smallest bias.

For each specified model, the above algorithm uses the bias to find the different designs with the same D-efficiency in Step 2. In Step 2, we keep more designs with

Table 3.1: Scheme of Designs Selection in Our Algorithm

Designs	D-efficiency	Bias	Number of selected designs
D_1	0.99	0.8	5
		1.2	3
		1.5	2
D_2	0.97	1.8	4
		3.5	2
D_3	0.90	5.6	3
		8.1	1

higher D-efficiencies from all designs with different D-efficiencies. For those designs with the same D-efficiency, some designs with all different biases are chosen and more designs with smaller biases are kept. For example, suppose that the D-efficiencies of designs found in Step 1 are 0.99, 0.97, 0.90 and 0.88, and we only keep some designs with top three D-efficiencies. The scheme of designs selection in Step 2 is displayed in Table 3.1. Let D_i denote the designs with the ith highest D-efficiency, where $1 \leq i \leq 3$. The corresponding D-efficiency and bias of the design are provided in the entries "D-efficiency" and "Bias", respectively. The design with the smallest bias is selected among all designs with the highest D-efficiency in Step 3.

In Section 3.3, our two-step approach will be applied to search for the D-optimal designs of 20 runs for all models containing up to three 2fi's. Those results will be used to demonstrate the performance of our algorithm. In Section 3.4, the four-step algorithm will be applied to the designs of 24 runs and the D-optimal designs of 24 runs are provided.

3.3 Performance of Our Algorithm

For each of the models in Figure 2.3, we already have the results from complete search for run sizes of 12 and 20. To demonstrate the performance of our algorithm, we display the relative positions of the best results from our algorithm in the range of results obtained from complete search and the percentages of good designs that our algorithm missed for designs of 20 runs.

In the implementation of our two-step algorithm, we only keep three designs and D-efficiencies of these three designs are different from each other in Step 1. All designs

that support the corresponding core $C(S)$ achieve full efficiency for model containing one 2fi. So we only keep one design for the construction of the designs that support S. For model 2(a) in Figure 2.3, all designs that support the corresponding $C(S)$ have three different D-efficiencies, so each of these three designs with different D-efficiencies is randomly chosen and kept in Step 1. For model 2(b) in Figure 2.3, there are only two different D-efficiencies for all designs that support $C(S)$, so we only randomly choose two designs with different D-efficiencies for generating the designs that support S.

For the model containing one 2fi, the search results from the algorithm give us the designs with the same highest D-efficiency as those from the complete search. For the models containing two 2fi's, Tables 3.2 and 3.3 provide the comparisons of results from complete search and those obtained from the algorithm. Table 3.2 will be used as an explanation, the range of D-efficiency from complete search is shown in the entries of "D (best)" and "D (worst)." The row of "m" indicates the number of factors included in the model. The entries under "Algorithm" give the highest D-efficiency obtained from our algorithm for the corresponding model. The row of "Rank" indicates that the relative position of the algorithmic result in the results of complete search. The row of "Percent (missed designs)" shows the percentage of better designs that our algorithm missed by using the following equation:

$$\frac{\#\text{ of designs from complete search better than the one from the algorithm}}{\#\text{ of all designs for the given model}} \times 100\%. \tag{3.2}$$

The frequency distribution of D-efficiencies is obtained from complete search. We rank the n distinct values of D-efficiency in descending order, and associate the frequencies $f_1, f_2, \ldots, f_n$ respectively with them. The percentage of missed designs corresponding to rank k is then obtained by using $\frac{\sum_{i=1}^{k-1} f_i}{\sum_{i=1}^{n} f_i} \times 100\%$.

For model 2(a) in Figure 2.3, we see in Table 3.2 that most of the results from the algorithm provide the best designs and only a few of them achieve the second best designs. Even for the cases where the algorithm misses the best design, the percentages of missed design from the algorithm are very small. The results in Table 3.3 show that our algorithm finds the best designs for model 2(b), so the ranks of algorithmic results are all 1's and all corresponding percentages of missed designs are 0's.

For the models containing three 2fi's, the comparisons of the results from complete search and those from the algorithm are displayed in Tables 3.4, 3.5, 3.6, 3.7 and 3.8.

Table 3.2: Comparisons of Searching Results of 20 Runs from Complete Search and Algorithm for Model 2(a)

	m	5	6	7	8	9	10	11
Complete	D (best)	0.96	0.96	0.96	0.95	0.95	0.94	0.94
search	D (worst)	0.82	0.80	0.77	0	0	0	0
Algorithm	D-efficiency	0.96	0.96	0.96	0.95	0.95	0.94	0.94
	Rank	1	1	1	1	1	1	1
	Percent (missed designs)	0	0	0	0	0	0	0
	m	12	13	14	15	16	17	
Complete	D (best)	0.93	0.93	0.93	0.92	0.92	0.91	
search	D (worst)	0	0	0	0	0	0	
Algorithm	D-efficiency	0.93	0.93	0.92	0.92	0.91	0.89	
	Rank	1	1	2	1	2	2	
	Percent (missed designs)	0	0	1.7	0	0.76	0.25	

Table 3.3: Comparisons of Searching Results of 20 Runs from Complete Search and Algorithm for Model 2(b)

	m	4	5	6	7	8	9	10
Complete	D (best)	0.98	0.97	0.96	0.96	0.95	0.95	0.94
search	D (worst)	0.86	0.84	0.83	0.81	0.78	0	0
Algorithm	D-efficiency	0.98	0.97	0.96	0.96	0.95	0.95	0.94
	Rank	1	1	1	1	1	1	1
	Percent (missed designs)	0	0	0	0	0	0	0
	m	11	12	13	14	15	16	17
Complete	D (best)	0.94	0.93	0.93	0.93	0.92	0.92	0.91
search	D (worst)	0	0	0	0	0	0	0
Algorithm	D-efficiency	0.94	0.93	0.93	0.93	0.92	0.92	0.91
	Rank	1	1	1	1	1	1	1
	Percent (missed designs)	0	0	0	0	0	0	0

Table 3.4: Comparisons of Searching Results of 20 Runs from Complete Search and Algorithm for Model 3(a)

	m	7	8	9	10	11
Complete search	D (best)	0.94	0.93	0.93	0.92	0.91
	D (worst)	0.72	0	0	0	0
Algorithm	D-efficiency	0.94	0.93	0.92	0.92	0.91
	Rank	1	1	2	1	1
	Percent (missed designs)	0	0	0.04	0	0
	m	12	13	14	15	16
Complete search	D (best)	0.91	0.9	0.9	0.89	0.88
	D (worst)	0	0	0	0	0
Algorithm	D-efficiency	0.9	0.9	0.89	0.87	0.85
	Rank	2	1	2	3	4
	Percent (missed designs)	0.13	0	0.02	0.11	0.03

Table 3.5: Comparisons of Searching Results of 20 Runs from Complete Search and Algorithm for Model 3(b)

	m	6	7	8	9	10	11
Complete search	D (best)	0.95	0.94	0.93	0.93	0.92	0.91
	D (worst)	0.75	0.72	0	0	0	0
Algorithm	D-efficiency	0.94	0.94	0.93	0.92	0.92	0.91
	Rank	2	1	1	2	1	1
	Percent (missed designs)	1.4	0	0	0.1	0	0
	m	12	13	14	15	16	
Complete search	D (best)	0.91	0.9	0.9	0.89	0.88	
	D (worst)	0	0	0	0	0	
Algorithm	D-efficiency	0.91	0.9	0.89	0.89	0.87	
	Rank	1	1	2	1	2	
	Percent (missed designs)	0	0	0.07	0	0.02	

Table 3.6: Comparisons of Searching Results of 20 Runs from Complete Search and Algorithm for Model 3(c)

	m	5	6	7	8	9	10
Complete	D (best)	0.96	0.95	0.94	0.93	0.93	0.92
search	D (worst)	0.81	0.79	0.76	0.74	0	0
Algorithm	D-efficiency	0.96	0.95	0.94	0.93	0.93	0.92
	Rank	1	1	1	1	1	1
	Percent (missed designs)	0	0	0	0	0	0
	m	11	12	13	14	15	16
Complete	D (best)	0.91	0.91	0.9	0.9	0.89	0.88
search	D (worst)	0	0	0	0	0	0
Algorithm	D-efficiency	0.91	0.91	0.9	0.9	0.89	0.88
	Rank	1	1	1	1	1	1
	Percent (missed designs)	0	0	0	0	0	0

Table 3.7: Comparisons of Searching Results of 20 Runs from Complete Search and Algorithm for Model 3(d)

	m	5	6	7	8	9	10
Complete	D (best)	0.95	0.95	0.94	0.93	0.93	0.92
search	D (worst)	0.76	0.75	0.72	0	0	0
Algorithm	D-efficiency	0.95	0.95	0.94	0.93	0.93	0.92
	Rank	1	1	1	1	1	1
	Percent (missed designs)	0	0	0	0	0	0
	m	11	12	13	14	15	16
Complete	D (best)	0.91	0.91	0.9	0.9	0.89	0.88
search	D (worst)	0	0	0	0	0	0
Algorithm	D-efficiency	0.91	0.91	0.9	0.89	0.89	0.87
	Rank	1	1	1	2	1	2
	Percent (missed designs)	0	0	0	0.18	0	0.04

Table 3.8: Comparisons of Searching Results of 20 Runs from Complete Search and Algorithm for Model 3(e)

	m	4	5	6	7	8	9	10
Complete search	D (best)	0.97	0.96	0.95	0.94	0.93	0.93	0.92
	D (worst)	0.82	0.82	0.81	0.78	0.77	0	0
Algorithm	D-efficiency	0.97	0.96	0.95	0.94	0.93	0.93	0.92
	Rank	1	1	1	1	1	1	1
	Percent (missed designs)	0	0	0	0	0	0	0
	m	11	12	13	14	15	16	
Complete search	D (best)	0.91	0.91	0.9	0.9	0.89	0.88	
	D (worst)	0	0	0	0	0	0	
Algorithm	D-efficiency	0.91	0.91	0.9	0.9	0.89	0.88	
	Rank	1	1	1	1	1	1	
	Percent (missed designs)	0	0	0	0	0	0	

From the row of "Rank" in Table 3.4, we see that the algorithm does not find the best designs for five cases, and two of them achieve the third and fourth best designs respectively. However, the percentages of missed designs by our algorithm are quite small. Even in the worst case that the fourth best design is obtained, the algorithm only misses 0.03% designs. The algorithm gives us all designs with the highest D-efficiency for models 3(c) and 3(e) presented in Tables 3.6 and 3.8. For models 3(b) and 3(d), Tables 3.5 and 3.7 show that only two or three cases reach the second highest D-efficiency and all percentages of missed designs are less than 0.2%. The ranks of the algorithmic results demonstrate that our algorithm works very well; those values of percentages of missed design further show that the algorithm performs almost as well as the complete search.

For the designs of 12 runs, the comparisons of results from complete search and those from algorithm also show that the algorithm works very well.

3.4 Application to Designs of 24 Runs

We apply our algorithm to designs of 24 runs for searching for the D-optimal designs for models containing up to three 2fi's. There are 60 Hadamard matrices of order 24, and one of which is generated from the 12-run Plackett-Burman design, which is called a fold-over design (Box and Wilson, 1951). A fold-over design can be obtained

from a fractional factorial design by reversing the signs of all the columns of the original design matrix. It is the combination of the original design and the design with reversed signs. The Plackett-Burman designs provide orthogonal main effects, and their alias structure is complex. Diamond (1995) studied some properties of a fold-over design of the 12-run Plackett-Burman design. Partly due to those good properties of a fold-over design, we use this special orthogonal array of 24 runs for searching for the optimal designs of 24 runs. The Hadamard matrix of 12 runs from N.J.A Sloane's homepage is used for generating the orthogonal array of 24 runs. Let H_{12} be the unique Hadamard matrix of 12 runs, then a Hadamard matrix of 24 runs obtained from H_{12} can be written as $H_{24} = \begin{pmatrix} H_{12} & H_{12} \\ H_{12} & -H_{12} \end{pmatrix}$. A complete dialogue of D-optimal designs obtained from this fold-over design are tabulated by applying our four-step algorithm. For some specified models that include 2fi's sharing a common factor, a useful result is summarized from these tables and can be applied to the construction of the orthogonal designs.

3.4.1 D-Optimal Designs of 24 Runs

For the model containing one 2fi, the D-optimal designs of 24 runs with the smallest bias are presented in Table 3.9. For $2 \leq m \leq 22$, Table 3.9 provides the information on the construction of D-optimal design and the corresponding D-efficiency and bias. From Table 3.9, we see that our algorithm finds the best designs with full efficiency, so it performs as well as the method of complete search.

For model 2(a), the core of requirement set $C(S)$ includes four factors and two 2fi's. Table 3.10 shows the selected designs that support $C(S)$ in the application of our algorithm. From the complete search of the designs that support $C(S)$, we find that all designs that support $C(S)$ have four different D-efficiencies: 0.98, 0.95, 0.94 and 0.91, but we only keep 9 designs with top three D-efficiencies. The 9 designs kept in our algorithm are displayed in Table 3.10. Based on the selected designs that support $C(S)$, the algorithm gives us the searching results for model 2(a) containing $4 \leq m \leq 21$ factors and two 2fi's in Table 3.11. For all cases, the algorithm yields the designs with high efficiency. Even for the design with 21 factors, the D-efficiency of 0.91 is still quite high. When the number of factors increases, the D-efficiency decreases and the corresponding bias increases, which is expected.

The model 2(b) includes at least 3 factors and two 2fi's sharing one common

Table 3.9: Optimal Designs of 24 Runs from Algorithm for Model With One 2fi

m	Selected columns	2fi	D-efficiency	Bias
2	{1,2}	(1, 2)	1	0
3	{1,2,12}	(1,2)	1	0
4	{1,2,12,15}	(1,2)	1	0.33
5	{1,2,3,12,16}	(1,12)	1	0.74
6	{1,2,3,4,12,17}	(1,12)	1	1.28
7	{1,2,3,4,12,17,18}	(1,12)	1	1.81
8	{1,12,14,15,16,17,18,19}	(1,12)	1	2.21
9	{1,12,14,15,16,17,18,19,20}	(1,12)	1	2.62
10	$\{1, 12, 14, 15, 16, 17, 18, 19, 20, 21\}$	(1,12)	1	3.02
11	$\{1, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22\}$	(1,12)	1	3.43
12	$\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13\}^c$	(1,12)	1	3.83
13	$\{2, 3, 4, 5, 6, 7, 8, 9, 10, 13\}^c$	(1,12)	1	5.52
14	$\{3, 4, 5, 6, 7, 8, 9, 10, 11\}^c$	(1,2)	1	6.43
15	$\{2, 3, 4, 5, 6, 7, 8, 12\}^c$	(1,13)	1	7.94
16	$\{2, 3, 4, 5, 6, 7, 12\}^c$	(1,13)	1	8.99
17	$\{2, 3, 4, 5, 6, 12\}^c$	(1,13)	1	9.99
18	$\{2, 3, 4, 5, 12\}^c$	(1,13)	1	10.97
19	$\{2, 13, 15, 16\}^c$	(1,13)	1	11.95
20	$\{2, 3, 12\}^c$	(1,13)	1	12.95
21	$\{2, 12\}^c$	(1,13)	1	13.97
22	$\{12\}^c$	(1,13)	1	15.02

Table 3.10: Selected Designs for Model 2(a)

Selected columns	2fi	D-efficiency	Bias
{12,13,14,15}	(12,13) (14,15)	0.98	0
{12,13,15,16}	(12,13)(15,16)	0.98	0
{13,17,20,22}	(13,17)(20,22)	0.98	0
{4,12,17,19}	(4,12)(17,19)	0.98	0.47
{5,6,7,12}	(5,6)(7,12)	0.98	0.47
{3,6,21,22}	(3,21)(6,22)	0.95	0.75
{4,7,13,14}	(4,13)(7,14)	0.95	0.75
{2,13,17,23}	(2,23)(13,17)	0.95	0.93
{1,2,15,16}	(1,2)(15,16)	0.94	0.66

Table 3.11: Optimal Designs of 24 Runs from Algorithm for Model 2(a)

m	Selected columns	2fi	D-efficiency	Bias
4	{12,13,14,15}	(12,13) (14,15)	0.98	0
5	{12,13,14,15,16}	(12,13) (14,15)	0.99	0.8
6	{12,13,14,15,16,17}	(12,13) (15,16)	0.99	1.18
7	{12,13,14,15,16,17,18}	(12,13)(15,16)	0.99	1.51
8	{12,13,14,15,16,17,18,19}	(12,13) (15,16)	0.99	1.91
9	{12,13,14,15,16,17,18,19,22}	(12,13) (14,15)	0.99	2.28
10	{12,13,14,15,16,17,18,19,20,21}	(12, 13) (15,16)	0.99	2.68
11	{12,13,14,15,16,17,18,19,20,21,22}	(12,13) (14,15)	0.99	3.05
12	$\{1,2,3,4,5,6,7,8,9,10,11\}^c$	(12,13) (14,15)	0.99	3.4
13	$\{1,2,4,5,6,7,8,9,10,11\}^c$	(12,13) (14,15)	0.99	5.41
14	$\{1,4,5,6,7,8,9,10,11\}^c$	(12,13) (14,15)	0.99	6.93
15	$\{1,4,5,6,7,8,9,10\}^c$	(12,13) (14,15)	0.99	8.26
16	$\{1,4,5,6,7,8,9\}^c$	(12,13) (14,15)	0.98	9.49
17	$\{1,4,5,6,7,8\}^c$	(12,13) (14,15)	0.97	10.72
18	$\{1,4,6,8,9\}^c$	(12, 13) (14,15)	0.96	12
19	$\{1,4,9,10\}^c$	(12,13) (14,15)	0.95	13.51
20	$\{1,2,14\}^c$	(4, 12) (17,19)	0.94	15.49
21	$\{4\}^c$	(12, 13) (14,15)	0.91	19.15

Table 3.12: Optimal Designs of 24 Runs from Algorithm for Model 2(b)

m	Selected columns	2fi	D-efficiency	Bias
3	{1,2,12}	(1,2) (1,12)	1	0
4	{1,8,12,16}	(8,12) (12,16)	1	0.47
5	{1,2,8,12,16}	(8,12) (12,16)	1	0.93
6	{1,2,8,12,15,16}	(8,12) (12,16)	1	1.44
7	{1,2,3,8,12,13,16}	(8,12) (12,16)	1	1.98
8	{8,12,13,14,15,16,17,18}	(8,12) (12,16)	1	2.45
9	{8,12,13,14,15,16,17,18,19}	(8,12) (12,16)	1	2.91
10	{8,12,13,14,15,16,17,18,19,21}	(8,12) (12,16)	1	3.38
11	{8,12,13,14,15,16,17,18,19,21,22}	(8,12) (12,16)	1	3.85
12	$\{1,2,3,4,5,6,7,9,10,11,20\}^c$	(8,12) (12,16)	1	4.32
13	$\{2,3,4,5,6,7,9,10,11,20\}^c$	(8,12) (12,16)	1	5.87
14	$\{1,2,3,4,5,6,7,9,20\}^c$	(8,12) (12,16)	1	7.12
15	$\{1,2,3,4,5,6,7,20\}^c$	(8,12) (12,16)	1	8.23
16	$\{1,2,3,4,5,6,20\}^c$	(8,12) (12,16)	1	9.27
17	$\{1,2,3,4,5,20\}^c$	(8,12) (12,16)	1	10.27
18	$\{1,4,14,15,20\}^c$	(8,12) (12,16)	1	11.25
19	$\{4,13,14,20\}^c$	(8,12) (12,16)	1	12.24
20	$\{1,4,20\}^c$	(8,12) (12,16)	1	13.26
21	$\{4,20\}^c$	(8,12) (12,16)	1	14.3

factor. In the search of first step, there are two classes of designs that support $C(S)$, one class of designs have full efficiency and no bias and the other class of designs have D-efficiency 0.96 and bias 0.33. For the implementation of our algorithm, we randomly chose 6 designs with full efficiency and 4 designs with D-efficiency 0.96 to construct the designs that support S. The results for model 2(b) are presented in Table 3.12. The algorithm finds the best designs with full efficiency for $3 \leq m \leq 21$ and the corresponding biases are very small. The algorithm efficiently produces all results that are as good as those from the complete search.

For the five different models containing three 2fi's, the optimal designs are presented in Tables 3.13, 3.14, 3.15, 3.16 and 3.17. By using the algorithm, we obtain the best designs with full efficiency for model 3(c) and some cases ($3 \leq m \leq 12$) of model 3(e). Note that the D-efficiency in Tables 3.13 and 3.14 decreases as m increases for $6 \leq m \leq 12$ and $13 \leq m \leq 20$, respectively. After exploring those cases around $m = 12$, we found that the D-efficiency for the case of $m = 13$ is indeed higher than that for $m = 12$.

Table 3.13: Optimal Designs of 24 Runs from Algorithm for Model 3(a)

m	Selected columns	2fi	D-efficiency	Bias
6	{1,2,3,13,14,15}	(1,2) (3,13) (14,15)	0.97	1.91
7	{1,2,3,4,12,13,16}	(1,2) (3,12) (4,13)	0.97	3.12
8	{1,2,3,4,12,13,16,23}	(1,2) (3,12) (4,13)	0.96	3.54
9	{1,2,3,4,5,12,13,16,23}	(1,2) (3,12) (4,13)	0.95	4.16
10	{1,2,3,4,5,12,18,19,20,22}	1,2) (3,12) (4,13)	0.94	4.44
11	{1,2,3,4,5,12,18,19,20,21,22}	1,2) (3,12) (4,13)	0.94	5.40
12	{1,2,3,4,5,12,13,18,19,20,21,22}	(1,2) (3,4) (5,12)	0.94	6.43
13	{1,2,3,4,5,12,13,15,18,19,20,21,22}	(1,2) (3,4) (5,12)	0.95	7.41
14	$\{6,7,8,9,10,11,14,16,17\}^c$	(1,2) (3,4) (5,12)	0.95	8.36
15	$\{6,7,8,9,10,11,17,22\}^c$	(1,2) (3,4) (5,12)	0.95	9.33
16	$\{6,7,8,9,10,11,17\}^c$	(1,2) (3,4) (5,12)	0.95	10.27
17	$\{6,7,9,10,11,17\}^c$	(1,2) (3,4) (5,12)	0.94	11.52
18	$\{6,7,9,11,17\}^c$	(1,2) (3,4) (5,12)	0.93	13.05
19	$\{7,8,9,17\}^c$	(1,2) (3,4) (5,12)	0.90	15.14
20	$\{6,9,17\}^c$	((1,2) (3,4) (5,12)	0.88	17.66

Table 3.14: Optimal Designs of 24 Runs from Algorithm for Model 3(b)

m	Selected columns	2fi	D-efficiency	Bias
5	{1,2,3,12,16}	(1,2) (3,12) (3,16)	0.99	0.90
6	{1,2,3,5,12,16}	(1,2) (3,12) (3,16)	0.98	1.56
7	{1,2,3,4,12,17,19}	(1,2) (3,12) (4,12)	0.98	2.14
8	{1,2,3,4,12,17,22,23}	(1,2) (3,12) (4,12)	0.98	2.83
9	{1,2,3,4,12,17,18,22,23}	(1,2) (3,12) (4,12)	0.98	3.64
10	{1,2,3,4,12,17,19,20,21,22}	(1,2) (3,12) (4,12)	0.98	4.44
11	{1,2,3,4,12,17,18,19,20,21,22}	(1,2) (3,12) (4,12)	0.98	5.30
12	{1,2,3,4,12,13,17,18,19,20,21,22}	(1,2) (3,12) (4,12)	0.98	6.17
13	$\{5,6,7,8,9,10,11,14,15,16\}^c$	(1,2) (3,12) (4,12)	0.99	7.01
14	$\{5,6,7,8,9,10,11,15,16\}^c$	(1,2) (3,12) (4,12)	0.99	7.85
15	$\{5,6,7,8,9,11,15,16\}^c$	(1,2) (3,12) (4,12)	0.98	8.94
16	$\{5,6,7,9,11,15,16\}^c$	(1,2) (3,12) (4,12)	0.97	10.04
17	$\{5,7,9,11,15,16\}^c$	(1,2) (3,12) (4,12)	0.96	11.29
18	$\{5,9,11,15,16\}^c$	(1,2) (3,12) (4,12)	0.95	12.71
19	$\{5,9,15,16\}^c$	(1,2) (3,12) (4,12)	0.94	14.63
20	$\{5,15,16\}^c$	(1,2) (3,12) (4,12)	0.91	18.20

Table 3.15: Optimal Designs of 24 Runs from Algorithm for Model 3(c)

m	Selected columns	2fi	D-efficiency	Bias
4	{8,12,16,19}	(8,12) (12,16)(12,19)	1	0.57
5	{1,8,12,16,19}	(8,12) (12,16)(12,19)	1	1.04
6	{1,2,8,12,16,19}	(8,12) (12,16)(12,19)	1	1.55
7	{1,2,3,8,12,16,19}	(8,12) (12,16)(12,19)	1	2.14
8	{8,12,13,14,15,16,17,19}	(8,12) (12,16)(12,19)	1	2.66
9	{8,12,13,14,15,16,17,18,19}	(8,12) (12,16)(12,19)	1	3.18
10	{8,12,13,14,15,16,17,18,19,21}	(8,12) (12,16)(12,19)	1	3.7
11	{8,12,13,14,15,16,17,18,19,21,22}	(8,12) (12,16)(12,19)	1	4.23
12	{8,12,13,14,15,16,17,18,19,21,22,23}	(8,12) (12,16)(12,19)	1	4.75
13	$\{4, 7, 14, 15, 17, 18, 20, 21, 22, 23\}^c$	(8,12) (12,16)(12,19)	1	7.01
14	$\{1, 2, 3, 4, 5, 6, 7, 9, 20\}^c$	(8,12) (12,16)(12,19)	1	7.42
15	$\{1, 2, 3, 4, 5, 6, 7, 20\}^c$	(8,12) (12,16)(12,19)	1	8.51
16	$\{1, 2, 3, 4, 5, 7, 20\}^c$	((8,12) (12,16)(12,19)	1	9.54
17	$\{1, 2, 3, 4, 7, 20\}^c$	(8,12) (12,16)(12,19)	1	10.54
18	$\{1, 2, 4, 7, 20\}^c$	(8,12) (12,16)(12,19)	1	11.53
19	$\{1, 4, 7, 20\}^c$	(8,12) (12,16)(12,19)	1	12.53
20	$\{4, 7, 20\}^c$	(8,12) (12,16)(12,19)	1	13.56

Table 3.16: Optimal Designs of 24 Runs from Algorithm for Model 3(d)

m	Selected columns	2fi	D-efficiency	Bias
4	{1,2,12,15}	(1,2)(1,12)(2,15)	0.99	0.33
5	{1,2,3,12,16}	(1,2)(1,12)(3,12)	0.99	0.90
6	{1,2,3,12,16,17}	(1,2)(1,12)(3,12)	0.99	1.55
7	{1,2,3,12,15,16,17}	(1,2)(1,12)(3,12)	0.99	2.23
8	{1,2,3,12,16,17,18,23}	(1,2)(1,12)(3,12)	0.99	2.93
9	{1,2,3,12,16,17,18,19,23}	(1,2)(1,12)(3,12)	0.99	3.73
10	{1,2,3,12,16,17,18,19,20,21}	(1,2)(1,12)(3,12)	0.99	4.50
11	{1,2,3,12,16,17,18,19,20,21,22}	(1,2)(1,12)(3,12)	0.99	5.25
12	$\{4, 5, 6, 7, 8, 9, 10, 11, 14, 22, 23\}^c$	(1,2)(1,12)(3,12)	0.99	5.97
13	$\{4, 5, 6, 7, 8, 9, 10, 11, 13, 15\}^c$	(1,2) (1,3) (2,12)	0.99	6.69
14	$\{4, 5, 6, 7, 8, 9, 10, 11, 14\}^c$	(1,2) (1,3) (2,12)	0.99	7.69
15	$\{4, 5, 6, 7, 8, 9, 13, 15\}^c$	(1,2)(1,12)(3,12)	0.98	8.93
16	$\{4, 5, 6, 7, 8, 13, 15\}^c$	(1,2)(1,12)(3,12)	0.97	10.04
17	$\{4, 5, 7, 10, 13, 15\}^c$	(1,2)(1,12)(3,12)	0.94	11.29
18	$\{4, 5, 8, 13, 15\}^c$	(1,2)(1,12)(3,12)	0.95	12.72
19	$\{6, 7, 13, 15\}^c$	(1,2)(1,12)(3,12)	0.94	14.63
20	$\{4, 13, 15\}^c$	(1,2)(1,12)(3,12)	0.91	18.21

Table 3.17: Optimal Designs of 24 Runs from Algorithm for Model 3(e)

m	Selected columns	2fi	D-efficiency	Bias
3	{1,2,12}	(1,2) (1,12) (2,12)	1	0
4	{1,2,12,15}	(1,2) (1,12) (2,12)	1	0.57
5	{1,2,12,15,16}	(1,2) (1,12) (2,12)	1	1.19
6	{1,2,12,15,16,17}	(1,2) (1,12) (2,12)	1	1.81
7	{1,2,12,15,16,17,18}	(1,2) (1,12) (2,12))	1	2.42
8	{1,2,12,15,16,17,18,19}	(1,2) (1,12) (2,12)	1	3.04
9	{1,2,12,15,16,17,18,19,20}	(1,2) (1,12) (2,12)	1	3.66
10	{1,2,12,15,16,17,18,19,20,21}	(1,2) (1,12) (2,12)	1	4.28
11	{1,2,12,15,16,17,18,19,20,21,22}	(1,2) (1,12) (2,12)	1	4.89
12	$\{3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14\}^c$	(1,2) (1,12) (2,12)	1	5.51
13	$\{3, 4, 5, 6, 8, 9, 10, 11, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.99	6.75
14	$\{3, 4, 5, 6, 7, 8, 9, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.99	7.88
15	$\{3, 4, 5, 6, 7, 8, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.98	8.97
16	$\{3, 4, 5, 6, 7, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.97	10.06
17	$\{3, 4, 5, 7, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.96	11.31
18	$\{3, 8, 9, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.95	12.72
19	$\{3, 4, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.94	14.64
20	$\{3, 13, 14\}^c$	(1,2) (1,12) (2,12)	0.91	18.16

3.4.2 One Construction Result of Orthogonal Designs

A design from the fold-over design of 24 runs containing up to 12 factors has some nice properties and is of resolution IV. For the designs with resolution IV, all main effects are orthogonal to each other and to any 2fi. The 2fi columns containing one common factor are always orthogonal to each other.

Result 1. An orthogonal array of 24 runs from the fold-over design has a special structure $D_{24} = \{a_1, a_2, \ldots, a_{11}, C, Ca_1, Ca_2, \ldots, Ca_{11}\}$, where $C = c(+ + + + + + - - - - - -)^T$. If those 2fi's in the given requirement set S share a common factor, an orthogonal design that supports the given requirement set S can be constructed by assigning column C to the common factor, columns $a_1, a_2, \ldots, a_i$ to the other i factors involved in 2fi's, where $1 \leq i \leq 11$. The remaining columns $\{a_{i+1}, a_{i+2}, \ldots, a_{11}, Ca_{i+1}, Ca_{i+2}, \ldots, Ca_{11}\}$ can be arbitrarily assigned to the other factors in the requirement set.

This result provides us with a method of constructing orthogonal designs for the requirement sets in which the 2fi's share one common factor. The orthogonal designs for all requirement sets represented by models A, 2(b) and 3(c) in Figure 2.3 are displayed in Tables 3.9, 3.12 and 3.15, respectively.

Example 6. Suppose that we want to estimate 15 factors and four 2fi's sharing one common factor. An orthogonal design can be constructed from the fold-over design of 24 runs. Consider the orthogonal array of 24 runs generated from the unique Hadamard matrix of 12 runs from N.J.A Sloane's homepage. The 12th column C in D_{24} is assigned as the common factor in those four 2fi's. We can assign columns a_1, a_2, a_3 and a_4 to the other four factors occurring in 2fi's. The remaining 10 factors can be arbitrarily assigned to any 10 columns from the complement of set $\{a_1, a_2, a_3, a_4, C, Ca_1, Ca_2, Ca_3, Ca_4\}$.

Chapter 4

Conclusions

4.1 Summary

This thesis studies the problem of joint estimation of main effects and some important two-factor interactions in industrial experiments. Our goal is to find D-optimal orthogonal arrays for specified models. The D-efficiency is used as the optimality criterion of design selection and the bias is employed to further distinguish the designs with the same D-efficiency. We have obtained optimal designs of 12 and 20 runs for all different models containing up to three 2fi's via complete search. The theory developed by Tang and Zhou (2009) provides us with a simple way to construct the designs that support a requirement set and narrow the searching scope to a smaller class of designs. Based on this theoretical result, a computational algorithm has been developed for searching for optimal designs of large run sizes. Although there is no guarantee that optimal designs can be found from a small class of designs that we go through in our algorithm, good designs that support the core set generally give us good designs that support the requirement set due to the property of orthogonality. We choose more designs with higher D-efficiency and use the bias to check different designs to further improve our algorithm. The comparison of results from complete search with those obtained from our algorithm shows that our algorithm performs very well. We use our algorithm to search for the D-optimal designs of 24 runs for all models containing up to three 2fi's. The fold-over design of 24 runs is used for illustration. We give a complete collection of optimal designs for all the models in Figure 2.3 and provide a method on how to construct orthogonal designs for the requirement sets in which the 2fi's share a common factor.

4.2 Future work

To further improve our algorithm, we could employ $E(S^2)$ as a surrogate of D-optimal criterion to increase the computational efficiency of our algorithm. The $E(S^2)$ criterion was proposed by Booth and Cox (1962) to compare supersaturated designs. Let s_{ij} be the element in the ith row and jth column of the information matrix X^TX, which is a measure of the degree of non-orthogonality between two columns i and j. If two columns i and j are orthogonal, then $s_{ij} = 0$. If they are fully aliased, then $s_{ij} = \pm n$. The $E(S^2)$ is defined by

$$E(S^2) = \sum_{1 \le i \le j \le m} s_{ij}^2 / \binom{m}{2}, \tag{4.1}$$

where m is the number of columns of the design. As an optimality criterion, it minimizes the average of s_{ij}^2 over all pairs of columns. This criterion requires less calculation than D-optimal criterion, so it will further reduce the computational effort on finding optimal designs and improve the efficiency of our algorithm.

Another direction we could pursue is to remove the restriction of the orthogonality for all main effects. Since orthogonal arrays guarantee that all main effects are mutually orthogonal, all designs in this thesis are constructed based on orthogonal arrays. When the number of factors is large and the number of 2fi's is relatively small in the given requirement set, the designs from orthogonal arrays have high efficiency due to the small degree of nonorthogonality. However, optimal designs might come from other designs that are not orthogonal arrays. This is an interesting topic for future research.

Bibliography

[1] Addelman, S. (1962). Symmetrical and Asymmetrical Fractional Factorial Plans. *Technometrics* **1**, 47-58.

[2] Booth, K.H.V. and Cox, D.R. (1962). Some Systematic Supersaturated Designs. *Technometrics* **4**, 489-495.

[3] 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.

[4] 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.

[5] Cheng, C. S. and Tang, B. (2005). A General Theory of Minimum Aberration and Its Applications. *The Annals of Statistics* **33**, 944-958.

[6] 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.

[7] Deng, L. Y. and Tang, B. (2002). Design Selection and Classification for Hadamard Matrices Using Generalized Minimum Aberration Criteria. *Technometrics* **44**, 173-184.

[8] Dimond, N. T. (1995). Some Properties of a Foldover Design. *Australian Journal of Statistics* **37**, 345-352.

[9] Franklin, M. F. (1985). Selecting Defining Contrasts and Confounded Effects in p^{n-m} Factorial Experiments. *Technometrics* **27**, 165-172.

[10] Franklin, M. F. and Bailey, R. A. (1977). Selection of Defining Contrasts and Confounded effects in Two-Level Experiments. *Applied Statistics* **26**, 321-326.

[11] Greenfield, A. A. (1976). Selection of Defining Contrasts in Two-Level Experiments. *Applied Statistics* **25**, 64-67

[12] Hamada, M. and Wu, C. F. J. (1992). Analysis of Designed Experiments With Complex Aliasing. *Journal of Quality Technology* **24**, 130-137.

[13] Hedayat, A. S. and Pesotan, H. (1992). Two-level Factorial Designs for Main Effects and Selected Two-factor Interactions. *Statistica Sinica* **2**, 453-464.

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

[15] Ke, W. and Tang, B. (2003). Selecting 2^{m-p} Designs Using a Minimum Abbration Criterion When Some Two-Factor Interactions Are Important. *Technometrics* **45**, 325-360.

[16] Miller, A. and Sitter, R. R. (2001). Using the Folded-Over 12-Run Plackett-Burman Design to Consider Ineractions. *Technometrics* **43**, 44-55.

[17] Plackett, R. L. and Burman, J. P. (1946). The Design of Optimum Multifactorial Experiments. *Biometrika* **33**, 305-325.

[18] Rao, C. R. (1947). Fractional Experiments Derivable from Combinatorial Arrangements of Arrays. *Journal of the Statistical Society* (Supp.) **9**, 128-139.

[19] Tang, B. and Deng, L. Y. (1999). Minimum G_2-Aberration for Nonregular Fractional Factorial Designs. *The Annals of Statistics* **27**, 1914-1926.

[20] Tang, B. and Wu, C. F. J. (1997). A Method for Constructing Supersaturated Designs and Its Es^2 Optimality. *The Canadian Journal of Statistics* **25**, 191-201.

[21] Tang, B. and Zhou, J. (2009). Existence and Construction of Two-level Orthogonal Arrays for Estimating Main Effects and Some Specified Two-factor Interactions. *Statistica Sinica*, To appear.

[22] Wu, C. F. J. and Chen, Y. (1992). A Graph-aided Method for Planning Two-Level Experiments When Certain Interactions Are Important. *Technometrics* **34**, 162-175.