Supersaturated Designs for Screening Experiments and Strong Orthogonal Arrays for Computer Experiments

by

Chenlu Shi

M.Sc., Simon Fraser University, 2015

B.Sc., St. Francis Xavier University, 2013

Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in the

Department of Statistics and Actuarial Science

Faculty of Science

SIMON FRASER UNIVERSITY

Summer 2019

Approval

Name:	**Chenlu Shi**
Degree:	**Doctor of Philosophy (Statistics)**
Title:	**Supersaturated Designs for Screening Experiments and Strong Orthogonal Arrays for Computer Experiments**
Examining Committee:	**Chair:** Jinko Graham Professor
	Boxin Tang Senior Supervisor Professor
	Richard Lockhart Supervisor Professor
	Liangliang Wang Internal Examiner Associate Professor
	Jiahua Chen External Examiner Professor Department of Statistics University of British Columbia
Date Defended:	**August 23, 2019**

Abstract

This dissertation centers on supersaturated designs and strong orthogonal arrays, which provide useful plans for screening experiments and computer experiments, respectively.

Supersaturated designs are a good choice for screening experiments. In order to use such designs, a common assumption that all interactions are negligible is made. In this dissertation, this assumption is dropped for the use of supersaturated designs. We propose and study a new class of supersaturated designs, namely foldover supersaturated designs, which allow the active main effects to be identified without making the assumption that two-factor interactions are absent. The $E(s^2)$-optimal foldover supersaturated designs are constructed, and further optimization is also considered for these $E(s^2)$-optimal supersaturated designs.

Strong orthogonal arrays were recently introduced and studied as a class of space-filling designs for computer experiments. This dissertation tackles two important problems that so far have not been addressed in the literature. The first problem is how to develop concrete constructions for strong orthogonal arrays of strength 3. We provide a systematic and comprehensive study on the construction of these arrays, with the aim at better space-filling properties. Besides various characterizing results, three families of arrays of strength 3 are presented. The other important problem is that of design selection for strong orthogonal arrays. We conduct a systematic investigation into this problem with the focus on strong orthogonal arrays of strength 2+ and 2. We first select arrays of strength 2+ by examining their 3-dimensional projections, and then formulate a general framework for the selection

of arrays of strength 2 by looking at their 2-dimensional projections. Both theoretical and computational results for arrays are presented.

Keywords: Foldover design; Hadamard matrix; Latin hypercube; regular design; second order saturated design; space-filling design

Dedication

To my beloved parents for their unconditional love and support!

Acknowledgements

I would like to convey my deepest gratitude to my senior supervisor Professor Boxin Tang who brought me to a fantastic research area, experimental design. Throughout my M.Sc. and Ph.D studies, he has always been very patient and supportive to me. I really appreciate his thoughtful guidance, unwavering support and inspirational encouragement in both my personal and professional life. I admire him for his comprehensive knowledge in statistics, his great passion for research, and his positive attitude towards life. It has been such a privilege for me to be his student. Without him this thesis would not have been possible.

I am thankful to my thesis committee members for their thorough reading of and valuable comments on my thesis. I would like to express my sincere appreciation to Professor Richard Lockhart for serving as my supervisor on both my M.Sc. and Ph.D thesis committees, and also for his great help on my job hunting. Sincere thanks go to Professor Jiahua Chen from University of British Columbia for serving as the external examiner. Many thanks to my internal examiner Professor Liangliang Wang who also served on my M.Sc. committee. I would like to thank Professor Jinko Graham for chairing my defence.

In addition, my gratitude extends to all the faculty and staff members in the Department of Statistics and Actuarial Science for offering such a wonderful studying environment. Special thanks go to Kelly, Charlene and Sadika for always being there when I need help. I would also like to thank my fellow graduate students, especially Luyao Lin, Ying Zhang, Tianyu Guan, Yi Xiong, Dongdong Li, Dongmeng Liu, Shufei Ge, Anqi Chen, Yunlong Nie, Perry Sang, Yuping Yang, Zhiyang Zhou, Trevor Thomson, Charlie Zhou, William Ruth, Shijia Wang, Cheng-Yu Sun. Thank you so much for your companionship during these years.

Table of Contents

List of Tables

Chapter 1

Introduction

Experimentation is a very useful tool to study a system, allowing an investigator to examine the changes in the outputs, when the settings for the inputs in the system are changed. To better understand the relationship between the inputs and the outputs produced by a system, a thorough investigation through data collection and analysis is needed. Statistical analysis based on a set of data including inputs and their corresponding outputs takes on this role. Design of experiments, i.e. the plan for setting inputs, enables an investigator to collect better data for conducting more efficient statistical analysis. In this thesis, we study designs for both physical and computer experiments, and our primary interest in physical experiments lies in screening experiments.

In many industrial experiments, the study for a system at early stages often involves a large number of factors, but usually only a few of them have significant effects. This is effect sparsity (Box and Meyer 1986). Under this assumption, identifying these active factors becomes the priority step for the investigation. Although there are various designs available to do this job, the most attractive ones for screening experiments are supersaturated designs, which are designs that can accommodate more factors than traditional factorial designs.

When the system under investigation is very complex, it is infeasible to conduct a physical experiment for studying such a system. With the advent of powerful computers, however, it can be done by undertaking a computer simulation. The model, represented by a computer code, is the basis for carrying out a computer simulation. Generally, it can be deterministic or stochastic. We concern ourselves with deterministic models, which means that given the same input, the output produced by running the computer code is unchanged. But in many real applications, the computer model is very complex; running the computer code is time-consuming. This calls for a cheaper model to function as a substitute for a computer simulation. Computer experiment can offer such a model, as it aims at using a set of data including the inputs and their corresponding outputs to build a statistical surrogate model for practical use. The selection for the inputs is crucial. Without much knowledge about the true model, a design that supports diverse modeling methods is desired. Space-filling designs emerge as the most appropriate designs for this purpose. A space-filling design refers to any design that spreads out its design points over the design region in some uniform fashion. One attractive approach for constructing space-filling designs is using a distance criterion (Johnson, Moore and Ylvisaker 1990) or a discrepancy criterion (Fang, Lin, Winker and Zhang 2000). More work can be found in Tang (1994), Morris and Mitchell (1995), van Dam, Husslage, den Hertog and Melissen (2007), and Moon, Dean and Santner (2011). Orthogonality also contributes to the construction of space-filling designs (Bingham, Sitter and Tang, 2009). Designs based on orthogonal arrays or a similar structure are most attractive due to their guaranteed properties. The use of such arrays dates back to McKay, Beckman and Conover (1979), which introduced a class of designs, called Latin hypercubes. A Latin hypercube achieves uniformity when projected to any 1-dimension, as there is exactly one point in each interval when the range of each input variable is divided into the same number of equally spaced intervals as the number of input points. Such a design is an orthogonal array of strength 1. Owen (1992) and Tang (1993) further investigated the use of orthogonal arrays of strength 2 or higher in the construction of space-filling designs. Motivated by (t, m, s)-nets from quasi-Monte Carlo (Niederreiter 1992), a class of new arrays, strong orthogonal arrays, was studied by He and Tang (2013). In this thesis, our focus is on the study of these arrays.

My research on supersaturated designs and strong orthogonal arrays has been published in three papers (Shi and Tang 2019a; Shi and Tang 2019b; Shi and Tang 2019c). The first of these, presented in Chapter 2, is on supersaturated designs. The other two, given in Chapters 3 and 4, are on strong orthogonal arrays. To facilitate the reading of these chapters, in Sections 1.1 and 1.2, respectively, we give broad overviews of supersaturated designs and strong orthogonal arrays and brief descriptions of our work on these two types of designs.

1.1 Supersaturated Designs

A supersaturated design refers to a factorial design where the number of factors exceeds its run size, and is very useful in screening experiments where the primary goal is to identify a few significant factors from a large number of potential factors. The idea of supersaturated designs goes back to Satterthwaite (1959), who suggested constructing such designs at random. The systematic investigation for such designs was firstly considered by Booth and Cox (1962). A measure of nonorthogonality, $E(s^2)$, was also discussed in that paper for the evaluation of supersaturated designs, and has been most commonly used in later work as a criterion for optimizing supersaturated designs. Three decades later, Lin (1993) and Wu (1993) studied supersaturated designs from the perspective of their theoretical construction. This drew researchers' attention back to supersaturated designs.

Later, more work gradually came out, and the main body includes deriving and improving the lower bound of $E(s^2)$ and constructing optimal-designs with respect to the $E(s^2)$ criterion. See Nguyen (1996), Tang and Wu (1997), Cheng (1997), Butler, Mead, Eskridge and Gilmour (2001), Bulutoglu and Cheng (2004), and Das, Dey, Chan and Chatterjee (2008) for details. Another major area for supersaturated designs is the corresponding analysis methods, which were discussed in Chipman, Hamada and Wu (1997), Westfall, Young and Lin (1998), Abraham, Chipman and Vijayan (1999), Li and Lin (2003), Zhang, Zhang and Liu (2007), Phoa, Pan and Xu (2009), and Marley and Woods (2010). Recently, a class of supersaturated designs, $UE(s^2)$-optimal supersaturated designs, was studied by Jones and Majumdar (2014). This paper expanded the family of $E(s^2)$-optimal supersaturated designs via relaxing a requirement of $E(s^2)$-optimal designs that each factor is balanced - any main effect is orthogonal to the ground mean. $UE(s^2)$-optimal supersaturated designs are intimately related to our work on supersaturated designs.

When a supersaturated design is used, there are not enough degrees of freedom to estimate all main effects. Therefore, the existing work assumes that all interactions are negligible when using supersaturated designs. In screening experiments, it is reasonable to assume that among a large number of factors, only a few factors are active. But it may not be appropriate to assume that none of interactions among these active factors exist.

In consideration of the above, in Chapter 2, we examine supersaturated designs under the presence of some interactions among active factors. We introduce and study a new class of supersaturated designs, called foldover supersaturated designs. Foldover supersaturated designs allow main effects to be identified under the existence of interactions between any two active factors. It follows that foldover supersaturated designs provide a class of designs that are robust to two-factor interactions.

1.2 Strong Orthogonal Arrays

A strong orthogonal array (SOA) of strength t is more space-filling in $2 < g < t$ dimensions than a comparable ordinary orthogonal array. This implies that an SOA enjoys a better space-filling property only when it is of strength 3 or higher. The higher the strength of an SOA is, the more expensive the SOA is. Thus, SOAs of strength 3 come to be most useful. Although the general characterization and construction for SOAs of strength 3 have been done in He and Tang (2014), concrete constructions for such arrays have not been explored so far.

Chapter 3 fills this gap. It gives a systematic and comprehensive study on the construction of SOAs of strength 3. Instead of obtaining an arbitrary array, this work is about constructions for arrays that possess more space-filling properties. More explicitly, we direct our attention to SOAs of strength 3 that enjoy some of the space-filling properties that SOAs of strength 4 can possess. Various characterizing results and three families of SOAs of strength 3 are provided in this chapter. A remarkable result is that arrays from one of these families enjoy almost all of the space-filling properties of SOAs of strength 4, while being able to accommodate many more factors than the latter.

However, for certain investigations, SOAs of strength 3 may be too expensive for experimenters to afford. This leads to the introduction of SOAs of strength 2+ (He, Cheng and Tang 2018). Compared with SOAs of strength 3, such arrays enjoy the same two-dimensional space-filling properties but have more economical run sizes for the given number of factors. Even though a series of theoretical developments have been done, an important problem that has not been addressed is that of design selection for SOAs of strength 2+.

Chapter 4 provides a systematic investigation into this problem of design selection. We address the issue by formulating a criterion based on their three-dimensional properties. When the number of factors is large relative to the run size, SOAs of strength 2+ do not exist. Although an SOA of strength 2 has no better space-filling properties in two-dimensions than an ordinary orthogonal array of strength 2, some of its two-colum subarrays enjoy better space-filling properties. We provide a general framework for the selection of SOAs of strength 2 by investigating their two-dimensional properties. This framework is applicable when SOAs of strength 2+ do not exist, and gives rise to SOAs of strength 2+ when they do exist.

Chapter 2

Supersaturated Designs Robust to Two-Factor Interactions

2.1 Introduction

Supersaturated designs are useful in screening experiments where the primary goal is to identify a few significant factors from a large number of potential factors. As the number of factors in a supersaturated designs is at least as large as its run size, the main effects cannot all be estimated simultaneously. The analysis of such designs relies on the assumption of effect sparsity (Box and Meyer 1986) that only a few factors are active. As supersaturated designs cannot be made orthogonal, a natural way of constructing such designs is to minimize some measure of nonorthogonality, with the $E(s^2)$ being the most popular criterion. Earlier work on supersaturated designs includes Booth and Cox (1962), Lin (1993), Wu (1993), Nguyen (1996), Tang and Wu (1997) and Cheng (1997). More recent work is given in Butler, Mead, Eskridge and Gilmour (2001), Bulutoglu and Cheng (2004), Das, Dey, Chan and Chatterjee (2008), Phoa, Pan and Xu (2009), Marley and Woods (2010), and Jones and Majumdar (2014).

Explicitly or implicitly, this body of work on supersaturated designs assumes the absence of all interaction effects. The reasoning is fairly natural. Since there are not even enough degrees of freedom to estimate all main effects, it seems pointless to allow the existence of significant interactions when considering supersaturated designs. It makes sense to assume that only a few factors are active in screening experiments where a large number of factors are studied - this is the sparsity argument for the use of supersaturated designs. But it does not seem always appropriate to assume that these active factors do not interact with each other.

We allow the possibility that some of the active factors can potentially interact with each other, which has never been done before to the best of our knowledge. This scenario calls for new supersaturated designs. It turns out that foldover supersaturated designs are well suited for the job. As main effects are all orthogonal to two-factor interactions in a foldover design, the existence of active two-factor interactions will not affect the identification of active main effects. This implies that foldover supersaturated designs provide a class of designs that are robust to two-factor interactions. To analyze such a design, one can first examine main-effect-only subset models that the design supports to identify active factors. Potential two-factor interactions among active factors will not affect the identification of active factors. One can then consider those models containing the main effects and some two-factor interactions of the active factors. This analysis strategy for foldover designs has been used by many authors, perhaps with the first being Miller and Sitter (2001).

Section 2.2 introduces foldover supersaturated designs and examines their $E(s^2)$-optimality. It turns that $E(s^2)$-optimal foldover supersaturated designs can be obtained from $UE(s^2)$-optimal supersaturated designs, which were recently studied by Jones and Majumdar (2014). We also present an example to illustrate the potential benefits of using foldover supersaturated designs. In Section 2.3, we further optimize $E(s^2)$-optimal foldover supersaturated designs by first minimizing the maximum correlation and then the frequency of the pairs of columns attaining the maximum correlation. Two methods are presented and small designs are tabulated for practical users. Section 2.4 concludes this chapter.

2.2 Robust Supersaturated Designs

2.2.1 Notation and background

A two-level supersaturated design of n runs for m factors can be represented by an $n \times m$ matrix $D = (d_{ij})$ with $d_{ij} = \pm 1$, where $n \leq m$. A natural requirement is that each column of design D has the same number of ± 1; designs with this property are called factor-balanced or column-balanced. As supersaturated designs cannot be orthogonal, one way of selecting good designs is to minimize a measure of nonorthogonality. Booth and Cox (1962) proposed the $E(s^2)$ criterion which is defined as $E(s^2) = \sum_{1 \leq u < v \leq m} s_{uv}^2 / \binom{m}{2}$ where $s_{uv} = d_u^T d_v$ with d_u being the uth column of design D. There is a rich literature on the study of $E(s^2)$-optimal supersaturated designs. Some representative papers are Nguyen (1996), Tang and Wu (1997), Cheng (1997), Butler, Mead, Eskridge and Gilmour (2001), and Bulutoglu and Cheng (2004). The latest results on the bounds on E(s^2)-optimality can be found in Das, Dey, Chan and Chatterjee (2008). A conservative method for selecting supersaturated designs is to minimize $\max_{u<v} |s_{uv}|$. Although this criterion can be used alone in design selection, more often than not, it is used to select designs from among $E(s^2)$-optimal supersaturated designs. We adopt this approach in this chapter.

A breakaway from $E(s^2)$-optimal supersaturated designs is the introduction of $UE(s^2)$-optimal supersaturated designs by Jones and Majumdar (2014). Such designs are selected using the same $E(s^2)$ criterion but applied to a different class of matrices D. More specifically, a $UE(s^2)$-optimal supersaturated design is the one that minimizes the $E(s^2)$ value, which is denoted by $UE(s^2)$ to mark the difference in Jones and Majumdar (2014), among all matrices D with its first column being the all-ones column. Two features stand out: (i) D is now the model matrix with the first column corresponding to the intercept and (ii) factor-balance is not required for any of the $m-1$ factors. In contrast, an $E(s^2)$-optimal supersaturated design requires factor-balance for all of its m factors. Construction of $UE(s^2)$-optimal supersaturated designs is very easy, and in fact can be done almost effortlessly. Assuming the existence of Hadamard matrices, $UE(s^2)$-optimal supersaturated designs can then be constructed for all cases of n and m as follows. Consider a Hadamard matrix of order m with its first column being all ones. Then any n rows provide a $UE(s^2)$-optimal supersaturated design. Call this design D. Adding any one column to D or deleting any one column (except for the all-ones column) from D still gives a $UE(s^2)$-optimal supersaturated design. Whereas deleting two columns from D may not lead to a $UE(s^2)$-optimal supersaturated design, adding two columns of certain form to D does produce a $UE(s^2)$-optimal supersaturated design. The required two-column matrix is such that $[n/2]$ of its rows are either $(-1,-1)$ or $(1,1)$ and the other $n-[n/2]$ rows are either $(-1,1)$ or $(1,-1)$, where $[x]$ is the greatest integer not exceeding x.

Cheng, Das, Singh and Tsai (2018) provided a penetrating discussion on the comparisons between $E(s^2)$- and $UE(s^2)$-optimal designs through their projection properties. They argued that if $UE(s^2)$-optimal supersaturated designs as given by Jones and Majumdar (2014) are to be used, one should choose superior $UE(s^2)$-optimal designs, those that are most level-balanced.

2.2.2 Foldover supersaturated designs and their $E(s^2)$-optimality

Let B be any $n \times m$ matrix of ± 1 with $2n \leq m$. Then a foldover supersaturated design of $N = 2n$ runs based on B is given by

$$D = \begin{bmatrix} B \\ -B \end{bmatrix}. \tag{2.1}$$

Design D in (2.1) is always factor-balanced regardless of whether or not B is factor-balanced. When a row of B is multiplied by -1, two rows of matrix D swap their positions but the same design is obtained. Thus, there is no loss of generality to assume the first column of B consists of all ones. Such a B is the model matrix of a supersaturated design considered in Jones and Majumdar (2014). This shows that

$$E_D(s^2) = 2\,UE_B(s^2).$$

This relation leads to an important result.

Proposition 1. *Design D in (2.1) is an $E(s^2)$-optimal foldover supersaturated design if and only if the base design B is a $UE(s^2)$-optimal supersaturated design.*

As simple as Proposition 1 may be, it allows all $E(s^2)$-optimal foldover supersaturated designs to be constructed easily, thanks to the easy construction of $UE(s^2)$-optimal supersaturated designs as discussed in Subsection 2.2.1.

Remark 1. The $E(s^2)$-optimality of design D in (2.1) as claimed in Proposition 1 is within the class of all foldover supersaturated designs. Obviously, a design that is an $E(s^2)$-optimal within the class of all foldover designs may not be $E(s^2)$-optimal in general. For most recent bounds on the $E(s^2)$ value in the general case, we refer to Das, Dey, Chan and Chatterjee (2008). Throughout this chapter, when we say that a design is $E(s^2)$-optimal foldover supersaturated design, we mean that it is $E(s^2)$-optimal with reference to the class of foldover supersaturated designs.

Remark 2. For $n \leq m < 2n$, while the base design B is still supersaturated, the foldover design D in (2.1) is no longer supersaturated, at least in the traditional sense that a supersaturated design has a run size no greater than the number of factors. On the other hand, design D has a flavour similar to a supersaturated design in that not all main effects can be simultaneously estimated. This is an interesting case that deserves further investigation. In this chapter, we focus on the case of supersaturated designs in the traditional sense that requires $N = 2n \leq m$, as our purpose is to provide a class of supersaturated designs that can serve as alternatives to the existing supersaturated designs when some interactions are potentially important.

Remark 3. Due to the foldover structure of design D in (2.1), all main effects are orthogonal to all two-factor interactions. This implies that the existence of two-factor interactions will not affect the identification of active main effects. Insistence on the foldover structure has some consequences. For example, in terms of the $E(s^2)$ criterion, foldover supersaturated designs do not perform as well as those without the foldover structure. This is expected as we are trading some loss in the $E(s^2)$ criterion for the property of being robust to nonnegligible interactions. Some more discussion on this issue is given at the end of Section 2.4.

Proposition 1 shows how to construct foldover supersaturated designs that are $E(s^2)$-optimal. Having achieved $E(s^2)$-optimality, in Section 2.3 we will use the secondary criterion $\max_{u<v} |s_{uv}|$ to further optimize foldover supersaturated designs. To be more specific, among all $E(s^2)$-optimal foldover supersaturated designs given in Proposition 1, we will select those which minimize the value of $\max_{u<v} |s_{uv}|$. If more than one design minimizes the value of $\max_{u<v} |s_{uv}|$, we will choose a design that minimizes the number of times that this maximal value is attained by pairs of columns. Details are given in the next section.

We give an example to illustrate the potential benefits using our proposed designs.

Example 1. We compare five designs of 12 runs for 16 factors in terms of their ability to identify active factors in the presence of two-factor interactions. The foldover design is from our results in Table 2.2 of the next section for the case $k = 4$ and $p = 2$. The other four designs are from Lin (1993), Wu (1993), Tang and Wu (1997) and Liu and Zhang (2000). The one from Wu (1993) is directly available in that paper. For the three designs from Lin (1993), Tang and Wu (1997) and Liu and Zhang (2000), we select the 16 columns from the corresponding $E(s^2)$-optimal 12-run designs with 22 factors by sequentially minimizing $E(s^2)$ values, $\max_{u<v} |s_{uv}|$, and the frequency of the pairs of columns attaining $\max_{u<v} |s_{uv}|$.

We conduct 1000 simulations for each design. Each simulation is done as follows. Randomly select m_1 active factors and randomly select m_2 two-factor interactions among these active factors to build a true model,

$$Y = \beta_0 I + X_1\beta_1 + X_2\beta_2 + \varepsilon,$$

where Y denotes the vector of 12 observations, β_0 is the grand mean, I is the all $+1$ column, β_1 is the vector of m_1 main effects, X_1 is the corresponding design matrix of β_1, β_2 is the vector of m_2 two-factor interactions, X_2 is the corresponding design matrix of β_2, and the error term ε is assumed to be normal with a mean of zero and a variance of σ^2. The response values are generated by using $\beta_0 = 0$, each main effect in β_1 being ± 1, each two-factor interaction in β_2 being ± 0.5, and $\sigma = 0.3$, with the signs of main effects and interactions assigned randomly.

T	m_1	m_2	Lin	Wu	Tang-Wu	Liu-Zhang	Authors
	2	1	0	0	3	0	581
1	3	1	0	0	1	0	179
	3	2	0	0	0	0	189
	2	1	54	18	68	11	705
2	3	1	26	15	41	32	216
	3	2	0	0	0	0	229
	2	1	366	225	339	143	846
3	3	1	214	179	207	263	274
	3	2	0	0	0	0	282

Table 2.1: Simulation results: comparisons of five designs

To identify active factors, we consider all models containing $m_1^* \le m_1 + m_2$ main effects,

$$Y = \beta_0 I + X_1^*\beta_1^* + \varepsilon,$$

where β_1^* denotes the vector of m_1^* main effects, and X_1^* is the corresponding design matrix. Then we find the best T models using the Bayesian information criterion (BIC). We use $T = 1, 2,$ and 3 in our simulations. Entries in columns 4–8 in Table 2.1 give the numbers of simulations that at least one of the best T models exactly identifies the active main effects in the true model. The results in Table 2.1 show that the foldover supersaturated design does a much better job in identifying active factors in the presence of nonnegligible interactions than the other four designs.

2.3 Design Selection

We have shown that $E(s^2)$-optimal foldover supersaturated designs can be obtained from $UE(s^2)$-optimal supersaturated designs. This section considers further optimizing foldover supersaturated designs. More specifically, we want to select a foldover supersaturated design D in (2.1) which, in addition to being $E(s^2)$-optimal, also has the following two properties:

(i) it contains no fully aliased columns, and

(ii) it minimizes $\max_{u<v} |s_{uv}|$ and then minimizes the frequency of the two-column pairs that reach this maximum value.

It is obvious that design D in (2.1) has properties (i) and (ii) if and only if the base design B has the same properties. Subsection 2.3.1 examines selection of such a base design B using regular designs while Subsection 2.3.2 considers the use of good Hadamard matrices (Shi and Tang 2018).

2.3.1 Using regular designs

Let C be a regular fractional factorial design of $m = 2^k$ runs for n factors. As C has orthogonal columns, its transpose has orthogonal rows and is thus a $UE(s^2)$-optimal supersaturated design of n runs for m factors. This implies that to construct an $E(s^2)$-optimal foldover supersaturated design D having no fully aliased columns, the base design B can be chosen to be the transpose of a regular factorial design without fully aliased rows.

Proposition 2. *Let C be a regular factorial design of $m = 2^k$ runs for n factors. Then C has no fully aliased rows if it contains k independent columns and its defining relation has at least one word of odd length.*

The proof of Proposition 2 is easy. Because C contains all k independent columns, it does not have repeated runs. Since C has a defining word of odd length, it cannot contain two runs that are mirror images of each other.

According to this proposition, $B = C^T$ is a $UE(s^2)$-optimal supersaturated design of n runs for $m = 2^k$ factors that has no fully aliased columns. Then design D in (2.1) constructed from such a B is an $E(s^2)$-optimal foldover supersaturated design of $N = 2n$ runs for $m = 2^k$ factors without fully aliased columns. From this D, one can also construct $E(s^2)$-optimal foldover supersaturated designs with $m \pm 1$ and $m+2$ factors - see Subsection 2.2.1. Note that deleting two columns from D does not necessarily give an $E(s^2)$-optimal foldover supersaturated design.

Proposition 2 allows construction of base designs B with no fully aliased columns. Our next step is to find one from all such base designs that minimizes $\max_{u<v} |s_{uv}|$ and then minimizes the frequency of the two-column pairs attaining this $\max_{u<v} |s_{uv}|$ value. We have conducted a complete search for all possible n values when $k = 3, 4, 5$ and all $n \leq 14$ when $k = 6$. The results are given in Table 2.2.

In Table 2.2, the k independent factors are labelled as $1, 2, \ldots, k$, and the $p = n - k$ additional factors are labelled as $k+1, k+2, \ldots, k+p$. The first two columns of Table 2.2 give the values k and p, and the third column tells how the additional factors are generated. For all but two cases, we are able to find a sequence of the nested designs, which allows easy presentation of our results. For example, design with $k = 4$ and $p = 1$ is generated by $5 = 34$ where independent factors are $1, 2, 3, 4$ and the additional factor is 5, whereas design

k	p	Generator	$\max_{u<v} \lvert s_{uv} \rvert$	k	p	Generator	$\max_{u<v} \lvert s_{uv} \rvert$
3	1	$4 = 23$	2	6	1	$7 = 56$	5
4	1	$5 = 34$	3		2	$8 = 1234$	4
	2	$6 = 12$	2		3	$7 = 56$	
	3	$7 = 24$	3			$8 = 23$	
	4	$8 = 14$	4			$9 = 123456$	5
5	1	$6 = 45$	4		4	$\underline{10} = 46$	6
	2	$7 = 1234$	5		5	$\underline{11} = 1345$	5
	3	$8 = 35$	4		6	$\underline{12} = 12$	4
	4	$9 = 24$	5		7	$7 = 56$	
	5	$\underline{10} = 13$	4			$8 = 46$	
	6	$\underline{11} = 25$	5			$9 = 26$	
	7	$\underline{12} = 15$	4			$\underline{10} = 13$	
	8	$\underline{13} = 34$	5			$\underline{11} = 2345$	
	9	$\underline{14} = 12$	4			$\underline{12} = 1245$	
	10	$\underline{15} = 2345$	5			$\underline{13} = 123456$	5
	11	$\underline{16} = 123$	4		8	$\underline{14} = 36$	6

Table 2.2: The best base designs from regular designs

with $k = 4$ and $p = 2$ can be obtained by adding one additional factor to the design with $k = 4$ and $p = 1$ which is given by $6 = 12$. The last column gives the minimum $\max_{u<v} |s_{uv}|$ value of the base design B which is the transpose of the regular design.

This table can be used to produce optimal foldover supersaturated designs of 8 runs for 8 factors, of $10 \leq N \leq 16$ runs for 16 factors, of $12 \leq N \leq 32$ runs for 32 factors, and of $14 \leq N \leq 28$ runs for 64 factors.

2.3.2 Using good Hadamard matrices

Subsection 2.3.1 discusses the selection of base designs B using regular fractional factorial designs. In this subsection, we examine the use of half fractions of good Hadamard matrices for this purpose.

As any n rows of a Hadamard matrix of order m with an all-ones column give a $UE(s^2)$-optimal supersaturated design, ideally, we would need to consider all possible choices of n

rows from all Hadamard matrices of order m to find the best design B. Two difficulties arise with this approach. The first is that the complete set of inequivalent Hadamard matrices is only available for order 32 or lower (Kharaghani and Tayfeh-Rezaie 2013). Even for a given Hadamard matrix, a complete search would require examination of $\binom{m}{n}$ designs. The combinatorial number $\binom{m}{n}$ can get very large unless n, m are small - this is the second difficulty. Subsection 2.3.1 deals with the first difficulty by simply using the Sylvester Hadamard matrix, the one that gives rise to all regular designs, and handles the second difficulty with the aide of Proposition 2. In this subsection, we use good Hadamard matrices as our choices of Hadamard matrices to overcome the first difficulty, and we lessen the second difficulty by considering only those designs that can be obtained by deleting rows from half fractions of good Hadamard matrices.

Let $H = (I, h_1, \ldots, h_{m-1})$ be a Hadamard matrix of order m with its first column I consisting of all ones. Deleting I from H gives a saturated orthogonal array $S = (h_1, \ldots, h_{m-1})$. Shi and Tang (2018) introduced good Hadamard matrices and examined their applications to the construction of nonregular designs and supersaturated designs. Without going into details, essentially, a good Hadamard matrix is one such that $\max_{|t|=3,4} |J_t(S)|$ is minimized, where $J_t(S)$ is the J-characteristic of design S for a subset $t \subset \{1, 2, \ldots, m-1\}$ of columns, defined as $J_t(S) = \sum_{i=1}^{m} \prod_{j \in t} h_{ij}$ with h_{ij} denoting the i-th entry of column h_j.

Let Q be obtained by selecting those $m/2$ rows of $S = (h_1, \ldots, h_{m-1})$ that have entries of $+1$ in column h_1. Write $Q = (q_1, \ldots, q_{m-1})$ where q_j is the j-th column of Q. Note that q_1 consists of all ones, and if it is deleted from Q, we obtain the design discussed by Lin (1993). Further let $s_{uv}(Q) = q_u^T q_v$. The next result establishes a link between $\max_{u<v} |s_{uv}(Q)|$ and $\max_{|t|=3} |J_t(S)|$.

Proposition 3. *For any Hadamard matrix, it must hold that*

$$\max_{u<v} |s_{uv}(Q)| \leq \max_{|t|=3} |J_t\,(S)\,|/2.$$

Proof. For $2 \leq u < v \leq m-1$, we have that

$$\sum_{i=1}^{m} h_{i1}h_{iu}h_{iv} = \sum_{h_{i1}=1} h_{iu}h_{iv} - \sum_{h_{i1}=-1} h_{iu}h_{iv},$$

$$0 = \sum_{i=1}^{m} h_{iu}h_{iv} = \sum_{h_{i1}=1} h_{iu}h_{iv} + \sum_{h_{i1}=-1} h_{iu}h_{iv}.$$

Together, the above two equations lead to

$$s_{uv}(Q) = \sum_{h_{i1}=1} h_{iu}h_{iv} = \sum_{i=1}^{m} h_{i1}h_{iu}h_{iv}/2,$$

which implies that

$$\max_{u<v} |s_{uv}(Q)| = \max_{u<v} |\sum_{i=1}^{m} h_{i1}h_{iu}h_{iv}|/2 \leq \max_{|t|=3} |J_t(S)|/2.$$

This completes the proof. □

The rationale of our search method is Proposition 3. A good Hadamard matrix that minimizes $\max_{|t|=3,4} |J_t(S)|$ should also have a small $\max_{|t|=3} |J_t(S)|$ value. In fact, all good Hadamard matrices identified for order $24, 28, 32, 36, 44, 60, 72$ and 80 in Shi and Tang (2018) minimize $\max_{|t|=3} |J_t(S)|$. By Proposition 3, the $\max_{u<v} |s_{uv}(Q)|$ value must be also small if good Hadamard matrices are chosen for these orders. If we delete a few rows from Q, the resulting design is expected to perform well in terms of the $\max_{u<v} |s_{uv}|$ criterion.

Instead of searching through all choices of n rows from all Hadamard matrices of given order m, our method is to search only all choices of n rows selected from the half fraction Q of design S from a good Hadamard matrix for given order m. For a given Hadamard matrix, the number of designs obtained by our method is $\binom{m/2}{n}$ which is much smaller than $\binom{m}{n}$.

There is exactly one good Hadamard matrix for each of the orders 24, 28 and 32, which are all given by Paley's first construction (Shi and Tang 2018). For order 36, Shi and Tang (2018) found two good Hadamard matrices. We have applied our search procedure to good Hadamard matrices of order 24, 28, 32 and 36. For order 32, the search method does not find better designs than those obtained in Subsection 2.3.1.

Table 2.3 presents our search results for order 24, and provides the best base designs we have found for n runs and 23 factors where $n = 7, 8, 9, 10, 11$. These base designs can be used as in (2.1) to construct foldover supersaturated designs of $N = 2n$ runs for 23 factors, which are $E(s^2)$-optimal with good $\max_{u<v} |s_{uv}|$ properties. The top portion of Table 2.3 gives the half fraction Q of the good Hadamard matrix of order 24, and the bottom portion tells for each $n = 7, 8, 9, 10, 11$ which rows are to be selected to obtain the base design. The corresponding $\max_{u<v} |s_{uv}|$ values of the base designs are also given.

Table 2.4 presents the results from searching the half fraction of the good Hadamard matrix of order 28 and the base designs found by the method have n runs and 27 factors where $n = 7, 8, \ldots, 13$. We have used both good Hadamard matrices of order 36, and the results given in Table 2.5 are all from the first of the two given in Shi and Tang (2018). The base designs in this table have 35 factors and $n = 7, 8, \ldots, 17$ runs.

Besides the above method of deleting runs from half fractions, we have also explored an approach of considering those designs that can be obtained by adding runs to a quarter fraction of a good Hadamard matrix. A result similar to Proposition 3 has been established

Row	Half fraction Q matrix																					
1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	1	-1	1	1	1	1
2	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	1	-1	1	1	1
3	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	1	-1	1	1
4	1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	1	-1	1
5	1	1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	1	-1
6	1	-1	1	1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1	1	-1
7	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1
8	1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1
9	1	-1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1
10	1	1	-1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1
11	1	-1	-1	1	1	-1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1	1	-1
12	1	-1	1	-1	-1	1	1	-1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1

n	rows selected											$\max_{u<v} \|s_{uv}\|$
7	1	4	5	6	9	11	12					5
8	1	4	5	6	9	11	12	3				6
9	1	4	5	6	9	11	12	3	7			5
10	1	4	5	6	9	11	12	3	7	8		6
11	1	4	5	6	9	11	12	3	7	8	2	5

Table 2.3: Base designs of 7-11 runs for 23 factors

Row	Half fraction Q matrix																										
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
2	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
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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
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
12	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
13	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
14	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

n	rows selected													$\max_{u<v} \lvert s_{uv} \rvert$
7	2	6	8	9	10	13	14							5
8	2	6	8	9	10	13	14	11						6
9	2	6	8	9	10	13	14	11	7					5
10	2	6	8	9	10	13	14	11	7	3				6
11	2	6	8	9	10	13	14	11	7	3	12			7
12	2	6	8	9	10	13	14	11	7	3	12	5		6
13	2	6	8	9	10	13	14	11	7	3	12	5	4	5

Table 2.4: Base designs of 7-13 runs for 27 factors

Row	Half fraction Q matrix																																		
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
2	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
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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
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
12	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
13	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
14	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
15	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
16	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
17	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
18	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
21	-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

n	rows selected																	$\max_{u<v}\lvert s_{uv}\rvert$
7	3	4	7	8	10	13	16											5
8	3	4	7	8	13	14	15	16										6
9	4	5	7	8	9	10	11	12	13									7
10	3	4	5	7	8	10	11	13	16	17								6
11	3	4	5	7	8	10	11	13	14	16	17							7
12	3	5	6	7	8	10	11	12	13	14	17	18						6
13	3	5	6	7	8	10	11	12	13	14	16	17	18					7
14	1	2	3	4	5	6	7	8	13	14	15	16	17	18				6
15	1	3	4	5	6	7	8	10	11	12	13	14	16	17	18			7
16	1	2	3	4	5	6	7	8	10	13	14	15	16	17	18	21		6
17	1	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	7

Table 2.5: Base designs of 7-17 runs for 35 factors

for quarter fractions, which provides a basis for this alternative method. In all but one case, the quarter fraction method is no better than the half fraction method. The one exception is the case of $n = 16$ runs for 35 factors in Table 2.5. The design given there is actually from the quarter fraction method. Because the quarter fraction method only finds one better design, we decide to omit a detailed presentation of the method.

Finally, we make a remark on our choices of half fractions of Hadamard matrices. Proposition 3 gives an upper bound on the $\max_{u<v} |s_{uv}(Q)|$ value for a given Hadamard matrix, and half fractions obtained using different branching columns may give different $\max_{u<v} |s_{uv}(Q)|$ values. But this is of no concern for us as the equality in Proposition 3 actually holds for all possible half fractions when good Hadamard matrices of orders 24, 28, 32 and 36 are considered, which are what we have done.

2.4 Discussion

We propose the use of foldover supersaturated designs for screening experiments. Such designs have an appealing property of robustness in that they allow the identification of active main effects in the presence of nonnegligible two-factor interactions. We show that $E(s^2)$-optimal foldover supersaturated designs can be obtained by simply folding over $UE(s^2)$-optimal supersaturated designs. Further optimization of $E(s^2)$-optimal foldover supersaturated designs is carried out by minimizing the $\max_{u<v} |s_{uv}|$ criterion. Two methods are examined - one using regular designs and the other good Hadamard matrices. The resulting designs are tabulated for practitioners.

Superior $UE(s^2)$-optimal designs are those $UE(s^2)$-optimal supersaturated designs that are most level-balanced (Cheng, Das, Singh and Tsai 2018). Whether or not the base design B is level-balanced, the foldover design D in (2.1) is always level-balanced. It is therefore inconsequential in terms of minimizing the $\max_{u<v} |s_{uv}|$ criterion, if superior $UE(s^2)$-optimal

designs are used as the base designs B.

Consideration of foldover supersaturated designs has some undesirable consequences. If we take a closer look at the design tables, we see that the $\max_{u<v}|s_{uv}|$ value for some of the designs is fairly large relative to the corresponding run size, resulting in the high correlation among the main effects. This is a direct consequence of foldover designs in which half of the degrees of freedom are allocated to the two-factor interactions, thus leaving a space of half of the original dimension to arrange all the main effect columns. One potential strategy in the data analysis of our proposed designs is to use a few extra runs to resolve the ambiguity if the identified active main effects turn out to be among those that are highly correlated. At the design stage, one idea is to remove the restriction to $E(s^2)$-optimality. If the $\max_{u<v}|s_{uv}|$ criterion is used alone when searching for the best foldover supersaturated designs, we should be able to find designs with smaller $\max_{u<v}|s_{uv}|$ values. This topic deserves further investigation.

The robust property of foldover supersaturated designs comes at another price. A foldover supersaturated design of N runs for m factors where $N \leq m$ can estimate at most $N/2$ main effects simultaneously, and thus its MDS-resolution is at most $N/2+1$ (Miller and Tang 2012). Though undesirable, this is not unexpected as half of the degrees of freedom are allocated to two-factor interactions in a foldover design. A problem of both theoretical and practical importance would be the construction of foldover supersaturated designs with an MDS-resolution equal to or at least close to $N/2+1$. We leave it to the future.

Chapter 3

Construction Results for Strong Orthogonal Arrays of Strength Three

3.1 Introduction

Computer models are powerful tools that enable researchers to study complex systems in natural sciences, engineering, social sciences and humanities. When a computer program representing a complex model is expensive to run, it is desirable to build a more economical version of the model. Computer experiments are concerned with building these so-called surrogate models. This type of models is built using data consisting of a set of inputs and the corresponding outputs from the computer program. A vital step in the process of designing such an experiment is the selection of the right kinds of inputs to use. A widely accepted type of designs used for computer experiments is that of space-filling designs (Santner, Williams and Notz 2003; Fang, Li and Sudjianto 2006). Broadly speaking, a space-filling design refers to any design that scatters its points in the design region in some sort of uniform fashion.

There are several ways to obtain space-filling designs. An intuitively appealing approach is to use criteria based on distances or discrepancies. See Johnson, Moore and Ylvisaker (1990) and Fang, Lin, Winker and Zhang (2000) for early work, and Moon, Dean and Santner (2011), Lin and Kang (2016), Wang, Xiao and Xu (2018) for more recent developments. Orthogonality also plays an important role in the quest for space-filling designs (Ye 1998;

Bingham, Sitter and Tang 2009; Georgiou and Efthimiou 2014; Liu and Liu 2015). Use of orthogonal arrays to generate space-filling designs has a long history and goes back to McKay, Beckman and Conover (1979), Owen (1992) and Tang (1993).

Motivated by (t, m, s)-nets from quasi-Monte Carlo (Niederreiter 1992), He and Tang (2013) introduced and studied *strong orthogonal arrays* (SOAs), which are more space-filling in low dimensions than comparable ordinary orthogonal arrays. SOAs of strength three are most useful because of their economical run sizes. They are the subject of our study.

Although the existence problem of strength-three SOAs has been completely solved (He and Tang 2014), the same cannot be said about the construction of such arrays. We undertake a systematic and comprehensive study on the construction of strength-three SOAs. We do not want just any array; we want an array that possesses more space-filling properties whenever possible. More specifically, we aim at constructing strength-three SOAs that enjoy some of the space-filling properties that only strength-four SOAs can offer. Besides various characterizing results, we present three families of strength-three SOAs. The arrays in one of these families enjoy almost all of the space-filling properties of strength-four strong orthogonal arrays, and do so with much more economical run sizes than the latter. Section 3.2 introduces these better strength-three SOAs and presents illustrative examples of three families of strength-three SOAs. Characterizations and constructions will be studied under various scenarios in Section 3.3. We conclude this chapter with a discussion in Section 3.4.

3.2 Background, Preliminaries and Examples

3.2.1 Background

We use $\mathrm{OA}(n, m, s_1 \times \cdots \times s_m, t)$ to denote an orthogonal array of n runs, with m factors and having strength t, such that the jth factor has s_j levels taken from $\{0, 1, \ldots, s_j - 1\}$.

The array is symmetric if $s_1 = \cdots = s_m = s$ and asymmetric otherwise. A simple notation OA(n, m, s, t) is used for the symmetric case. Orthogonal arrays provide very useful designs in many scientific and technological investigations. Hedayat, Sloane and Stufken (1999) is devoted entirely to them; Dey and Mukerjee (1999) and Cheng (2014) also contain abundant sources of information on orthogonal arrays.

Point sets and sequences from quasi-Monte Carlo (Niederreiter 1992) have long been recognized as useful in design of experiments – see Bates, Buck, Riccomagno and Wynn (1996), Owen (1997), and Haaland and Qian (2010). Inspired by the combinatorial characterization of (t, m, s)-nets by Lawrence (1996) and Mullen and Schmid (1996), He and Tang (2013) introduced *strong orthogonal arrays* (SOAs) as space-filling designs for computer experiments. SOAs are more general than (t, m, s)-nets, and they are formulated in design language and thus more accessible to design researchers.

An $n \times m$ matrix with entries from $\{0, 1, \ldots, s^t - 1\}$ is called an SOA of n runs, m factors, s^t levels and strength t if any subarray of g columns for any g with $1 \leq g \leq t$ can be collapsed into an OA$(n, g, s^{u_1} \times \cdots \times s^{u_g}, g)$ for any positive integers $u_1, \ldots, u_g$ with $u_1 + \cdots + u_g = t$, where collapsing s^t levels into s^{u_j} levels is according to $[a/s^{t-u_j}]$ for $a = 0, 1, \ldots, s^t - 1$. We use SOA$(n, m, s^t, t)$ to denote such an array.

SOAs of strength three are most useful, since SOAs of strength two are no more space-filling than ordinary orthogonal arrays of strength two and SOAs of strength four or higher are prohibitively expensive. This chapter focuses on this class of most useful arrays.

3.2.2 Preliminaries

The existence of strength-three SOAs has been completely characterized in He and Tang (2014), but the construction of these arrays has not been explored. The objectives of this chapter are to provide a systematic and comprehensive study on the construction of strength-three SOAs. We want to find those designs that are most space-filling within this class of arrays. Our approach is to identify and construct those strength-three SOAs that enjoy some of the space-filling properties that strength-four SOAs offer.

The next result is taken from He and Tang (2013) and is needed for this chapter.

Lemma 1. *An SOA$(n, m, s^3, 3)$, say D, exists if and only if there exist three arrays $A = (a_1, \ldots, a_m)$, $B = (b_1, \ldots, b_m)$ and $C = (c_1, \ldots, c_m)$ such that (a_i, a_j, a_u), (a_i, a_j, b_j) and (a_i, b_i, c_i) are OA$(n, 3, s, 3)$s for all $i \neq j$, $i \neq u$ and $j \neq u$. These arrays are related through $D = s^2 A + sB + C$.*

An SOA$(n, m, s^3, 3)$ is collapsible into an OA$(n, 3, s, 3)$ in any three-dimension, and thus achieves stratifications on $s \times s \times s$ grids in all three-dimensions. In any two-dimension, an SOA$(n, m, s^3, 3)$ can be collapsed into an OA$(n, 2, s \times s^2, 2)$ and an OA$(n, 2, s^2 \times s, 2)$, and it therefore achieves stratifications on $s \times s^2$ and $s^2 \times s$ grids in all two-dimensions.

An SOA$(n, m, s^4, 4)$ is more space-filling, achieving

(α) stratifications on $s^2 \times s^2$ grids in all two-dimensions,

(β) stratifications on $s^2 \times s \times s$, $s \times s^2 \times s$ and $s \times s \times s^2$ grids in all three-dimensions, and

(γ) stratifications on $s^3 \times s$ and $s \times s^3$ grids in all two-dimensions.

The goal of this chapter is to construct strength-three SOAs that enjoy some or all of properties α, β and γ. The following provides a basis for later construction results. Its proof is similar to that of Proposition 2 of He and Tang (2013).

Proposition 1. *An SOA$(n, m, s^3, 3)$, as characterized in Lemma 1 through A, B and C, has* *(i) property α if and only if (a_i, b_i, a_j, b_j) is an $OA(n, 4, s, 4)$ for all $i \neq j$,*

(ii) property β if and only if (a_i, a_j, a_u, b_u) is an $OA(n, 4, s, 4)$ for all $i \neq j$, $i \neq u$ and $j \neq u$, and

(iii) property γ if and only if (a_i, a_j, b_j, c_j) is an $OA(n, 4, s, 4)$ for all $i \neq j$.

An SOA$(n, m, s^4, 4)$ has two more space-filling properties: (δ) stratifications on a set of s^4 intervals in all one-dimensions and (ϵ) stratifications on $s \times s \times s \times s$ grids in all four-dimensions. These two properties are not very interesting for strength-three SOAs. Property δ is not interesting at all because Latin hypercubes based on an SOA$(n, m, s^3, 3)$ can achieve the maximum stratifications in all one-dimensions. Property ϵ is not very interesting because an SOA$(n, m, s^3, 3)$ with this property implies the existence of an OA$(n, m, s, 4)$, requiring a large run size n for a given number m of factors.

The remainder of this chapter is devoted to the construction of strength-three SOAs with some or all of properties α, β and γ, by making use of regular 2^{m-p} designs. To our surprise, the results are far richer and more insightful than we have initially anticipated. The theory of maximal designs and doubling constructions as studied by Chen and Cheng (2006) plays a crucial role in many of our theoretical developments. In return, our construction results also bring some new insights into the theory of doubling.

3.2.3 Three examples

To help the reader appreciate the general theoretical results to be presented in Section 3.3, we provide three examples in this subsection.

Example 1. Consider the following array:

$$
\begin{array}{ccccccccc}
7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
3 & 7 & 5 & 7 & 5 & 1 & 1 & 3 & 1 \\
6 & 2 & 6 & 4 & 6 & 0 & 2 & 2 & 4 \\
2 & 2 & 4 & 4 & 4 & 6 & 4 & 6 & 2 \\
7 & 5 & 3 & 5 & 5 & 3 & 7 & 5 & 3 \\
3 & 5 & 1 & 5 & 7 & 5 & 1 & 1 & 5 \\
6 & 0 & 2 & 6 & 4 & 4 & 2 & 0 & 0 \\
2 & 0 & 0 & 6 & 6 & 2 & 4 & 4 & 6 \\
4 & 6 & 4 & 2 & 6 & 4 & 2 & 4 & 2 \\
0 & 6 & 6 & 2 & 4 & 2 & 4 & 0 & 4 \\
5 & 3 & 5 & 1 & 7 & 3 & 7 & 1 & 1 \\
1 & 3 & 7 & 1 & 5 & 5 & 1 & 5 & 7 \\
4 & 4 & 0 & 0 & 4 & 0 & 2 & 6 & 6 \\
0 & 4 & 2 & 0 & 6 & 6 & 4 & 2 & 0 \\
5 & 1 & 1 & 3 & 5 & 7 & 7 & 3 & 5 \\
1 & 1 & 3 & 3 & 7 & 1 & 1 & 7 & 3 \\
5 & 5 & 7 & 7 & 3 & 5 & 5 & 3 & 3 \\
1 & 5 & 5 & 7 & 1 & 3 & 3 & 7 & 5 \\
4 & 0 & 6 & 4 & 2 & 2 & 0 & 6 & 0 \\
0 & 0 & 4 & 4 & 0 & 4 & 6 & 2 & 6 \\
5 & 7 & 3 & 5 & 1 & 1 & 5 & 1 & 7 \\
1 & 7 & 1 & 5 & 3 & 7 & 3 & 5 & 1 \\
4 & 2 & 2 & 6 & 0 & 6 & 0 & 4 & 4 \\
0 & 2 & 0 & 6 & 2 & 0 & 6 & 0 & 2 \\
6 & 4 & 4 & 2 & 2 & 6 & 0 & 0 & 6 \\
2 & 4 & 6 & 2 & 0 & 0 & 6 & 4 & 0 \\
7 & 1 & 5 & 1 & 3 & 1 & 5 & 5 & 5 \\
3 & 1 & 7 & 1 & 1 & 7 & 3 & 1 & 3 \\
6 & 6 & 0 & 0 & 0 & 2 & 0 & 2 & 2 \\
2 & 6 & 2 & 0 & 2 & 4 & 6 & 6 & 4 \\
7 & 3 & 1 & 3 & 1 & 5 & 5 & 7 & 1 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 7
\end{array}
$$

As one can easily verify, this is an $\mathrm{SOA}(32, 9, 8, 3)$. But it is a special $\mathrm{SOA}(32, 9, 8, 3)$, as any subarray of two columns becomes an $\mathrm{OA}(32, 2, 4, 2)$ when the eight levels are collapsed into four levels according to $0, 1 \to 0$; $2, 3 \to 1$; $4, 5 \to 2$; $6, 7 \to 3$. Because of this, the array achieves stratifications on 4×4 grids in all two-dimensions, i.e., it possesses property α.

Example 2. The array below

$$
\begin{array}{cccccccc}
7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
6 & 0 & 4 & 2 & 4 & 2 & 6 & 0 \\
6 & 4 & 2 & 0 & 6 & 4 & 2 & 0 \\
7 & 3 & 1 & 5 & 5 & 1 & 3 & 7 \\
6 & 4 & 4 & 6 & 2 & 0 & 0 & 2 \\
7 & 3 & 7 & 3 & 1 & 5 & 1 & 5 \\
7 & 7 & 1 & 1 & 3 & 3 & 5 & 5 \\
6 & 0 & 2 & 4 & 0 & 6 & 4 & 2 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
2 & 4 & 0 & 6 & 0 & 6 & 2 & 4 \\
2 & 0 & 6 & 4 & 2 & 0 & 6 & 4 \\
3 & 7 & 5 & 1 & 1 & 5 & 7 & 3 \\
2 & 0 & 0 & 2 & 6 & 4 & 4 & 6 \\
3 & 7 & 3 & 7 & 5 & 1 & 5 & 1 \\
3 & 3 & 5 & 5 & 7 & 7 & 1 & 1 \\
2 & 4 & 6 & 0 & 4 & 2 & 0 & 6 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 6 & 2 & 4 & 2 & 4 & 0 & 6 \\
0 & 2 & 4 & 6 & 0 & 2 & 4 & 6 \\
1 & 5 & 7 & 3 & 3 & 7 & 5 & 1 \\
0 & 2 & 2 & 0 & 4 & 6 & 6 & 4 \\
1 & 5 & 1 & 5 & 7 & 3 & 7 & 3 \\
1 & 1 & 7 & 7 & 5 & 5 & 3 & 3 \\
0 & 6 & 4 & 2 & 6 & 0 & 2 & 4 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
4 & 2 & 6 & 0 & 6 & 0 & 4 & 2 \\
4 & 6 & 0 & 2 & 4 & 6 & 0 & 2 \\
5 & 1 & 3 & 7 & 7 & 3 & 1 & 5 \\
4 & 6 & 6 & 4 & 0 & 2 & 2 & 0 \\
5 & 1 & 5 & 1 & 3 & 7 & 3 & 7 \\
5 & 5 & 3 & 3 & 1 & 1 & 7 & 7 \\
4 & 2 & 0 & 6 & 2 & 4 & 6 & 0
\end{array}
$$

is an $\mathrm{SOA}(32, 8, 8, 3)$, which has properties both α and β. The array has property β since any subarray of three columns becomes an $\mathrm{OA}(32, 3, 4 \times 2 \times 2, 3)$, an $\mathrm{OA}(32, 3, 2 \times 4 \times 2, 3)$, or an $\mathrm{OA}(32, 3, 2 \times 2 \times 4, 3)$ when the levels of one factor are collapsed into four levels, and the levels of the other two factors are collapsed into two levels. Collapsing eight levels into two levels is according to $0, 1, 2, 3 \to 0$; $4, 5, 6, 7 \to 1$.

Example 3. The following array

$$\begin{array}{ccccccc}
7 & 7 & 7 & 7 & 7 & 7 & 7 \\
6 & 0 & 4 & 2 & 4 & 2 & 6 \\
6 & 4 & 2 & 0 & 6 & 4 & 2 \\
7 & 3 & 1 & 5 & 5 & 1 & 3 \\
6 & 4 & 4 & 6 & 2 & 0 & 0 \\
7 & 3 & 7 & 3 & 1 & 5 & 1 \\
7 & 7 & 1 & 1 & 3 & 3 & 5 \\
6 & 0 & 2 & 4 & 0 & 6 & 4 \\
2 & 2 & 2 & 2 & 2 & 2 & 2 \\
3 & 5 & 1 & 7 & 1 & 7 & 3 \\
3 & 1 & 7 & 5 & 3 & 1 & 7 \\
2 & 6 & 4 & 0 & 0 & 4 & 6 \\
3 & 1 & 1 & 3 & 7 & 5 & 5 \\
2 & 6 & 2 & 6 & 4 & 0 & 4 \\
2 & 2 & 4 & 4 & 6 & 6 & 0 \\
3 & 5 & 7 & 1 & 5 & 3 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 7 & 3 & 5 & 3 & 5 & 1 \\
1 & 3 & 5 & 7 & 1 & 3 & 5 \\
0 & 4 & 6 & 2 & 2 & 6 & 4 \\
1 & 3 & 3 & 1 & 5 & 7 & 7 \\
0 & 4 & 0 & 4 & 6 & 2 & 6 \\
0 & 0 & 6 & 6 & 4 & 4 & 2 \\
1 & 7 & 5 & 3 & 7 & 1 & 3 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 \\
4 & 2 & 6 & 0 & 6 & 0 & 4 \\
4 & 6 & 0 & 2 & 4 & 6 & 0 \\
5 & 1 & 3 & 7 & 7 & 3 & 1 \\
4 & 6 & 6 & 4 & 0 & 2 & 2 \\
5 & 1 & 5 & 1 & 3 & 7 & 3 \\
5 & 5 & 3 & 3 & 1 & 1 & 7 \\
4 & 2 & 0 & 6 & 2 & 4 & 6
\end{array}$$

is an SOA$(32, 7, 8, 3)$ with all of the properties α, β and γ. It has property γ because any subarray of two columns becomes an OA$(32, 2, 2 \times 8, 2)$ or an OA$(32, 2, 8 \times 2, 2)$ when the levels of one factor are collapsed into two levels.

The arrays in Examples 1, 2 and 3 enjoy some or all of properties α, β and γ. Strength-four SOAs automatically have these properties, but for 32 runs, an SOA$(32, m, 16, 4)$ can be constructed only for $m \leq 3$ (He and Tang 2013, Theorem 1).

3.3 Construction Results

We now concentrate on SOA$(n, m, s^3, 3)$s with $s = 2$, and consider their constructions using regular 2^{m-p} designs. This means that in applying Lemma 1 to obtain an SOA, the columns of A, B and C are all selected from a saturated regular two-level design. With k independent factors, one can obtain a saturated design of $n = 2^k$ runs for $m = n - 1$ factors. Let S denote this saturated design. If S is viewed as a collection of columns, then any subset of S is a design of resolution III or higher. Designs with repeated columns have resolution II and are also needed in presenting our construction results.

Two-level designs are commonly and conveniently studied with the two levels denoted by ± 1. When A, B and C all have two levels ± 1, we can transform them into levels $0, 1$ by $(A+1)/2$, $(B+1)/2$ and $(C+1)/2$. This implies that $D = 4A+2B+C$ in Lemma 1 should be replaced by

$$D = 2A + B + C/2 + 7/2. \tag{3.1}$$

3.3.1 Designs with property α

When strength-three SOAs are constructed using regular 2^{m-p} designs, the following result can be obtained.

Theorem 1. *If an SOA$(n, m, 8, 3)$ is to be constructed using regular A, B and C with their columns selected from a saturated design S, then it has property α if and only if A is of* *resolution IV or higher and* (A, B, B') *has resolution III or higher, where* $B' = (b_1', \ldots, b_m')$ *with* $b_j' = a_j b_j$.

Proof. According to Lemma 1 and Proposition 1(i), we only need to show that (a_i, b_i, a_j, b_j) where $i \neq j$ has strength four if and only if (A, B, B') has resolution III or higher. That (a_i, b_i, a_j, b_j) has strength four means that the four columns are independent, thus without any defining words among them. That (A, B, B') has resolution III or higher simply says that (A, B, B') has no repeated columns. Therefore, to prove Theorem 1, it remains to assert that (a_i, b_i, a_j, b_j) where $i \neq j$ has no defining words if and only if (A, B, B') has no repeated columns. This assertion can be easily verified. □

Designs A, B and B' in Theorem 1 produce a collection of mutually exclusive triplets $(a_j, b_j, b_j' = a_j b_j)$ with $j = 1, \ldots, m$. For a different purpose, such mutually exclusive triplets are also wanted in Wu (1989). There is, however, a major difference, being that we require A to have resolution IV or higher whereas there is no such a requirement in Wu (1989). This implies that the method of Wu (1989) would not help us to construct A, B and B' in Theorem 1. New methods have to be developed for this purpose.

Theorem 1 leads to an important result on the number of factors in an SOA$(n, m, 8, 3)$ with property α.

Proposition 2. *If an SOA*$(n, m, 8, 3)$ *with property* α*, as characterized in Theorem 1, exists, then it must hold that* $m \leq 5n/16$.

Proof. By Theorem 1, design A has resolution IV or higher. A resolution IV or higher design is either even or even/odd. A design is even if the lengths of the words in its defining relation are all even, and it is even/odd if its defining relation contains words of both even and odd lengths. If A is even/odd, then we have that $m \leq 5n/16$ (Butler 2003; Chen and Cheng

2006). Now suppose that A is even. Then A is a subset of a saturated resolution IV design, say Q, which has form $Q = (e_k, e_k Q_0)$, where $e_1, \ldots, e_k$ are independent factors and Q_0 consists of $e_1, \ldots, e_{k-1}$, and all their interaction columns. Clearly, $S = (Q_0, Q)$. For each $j = 1, \ldots, m$, as $a_j = b_j b'_j$, one of b_j and b'_j must be in Q_0 and the other in Q. Since A does not share any column with B or B', we therefore have that $2m \leq n/2$, with $n/2$ being the number of columns in Q. This shows that $m \leq n/4$ if A is even. The proof is completed. $\square$

The proof of Proposition 2 actually reveals that, in order to construct an SOA$(n, m, 8, 3)$ with property α, if an even A is used, it is impossible to obtain more than $n/4$ factors, and one has to consider an even/odd A in order to break this barrier.

We next present a recursive construction of designs A, B and B' needed in Theorem 1. Recall that $B' = (b'_1, \ldots, b'_m)$ is determined by $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_m)$ via $b'_j = a_j b_j$. Let A_k, B_k and B'_k, based on k independent factors $e_1, \ldots, e_k$, satisfy the condition in Theorem 1 that A_k is of resolution IV or higher and (A_k, B_k, B'_k) is of resolution III or higher. Then A_{k+2}, B_{k+2} and B'_{k+2}, based on $k+2$ independent factors $e_1, \ldots, e_{k+2}$, can be constructed to satisfy the requirement in Theorem 1. This is done by defining

$$
\begin{aligned}
A_{k+2} &= (A_k, e_{k+1}A_k, e_{k+2}A_k, e_{k+1}e_{k+2}A_k), \\
B_{k+2} &= (B_k, e_{k+2}B_k, e_{k+1}e_{k+2}B_k, e_{k+1}B_k).
\end{aligned}
\tag{3.2}
$$

Then $B'_{k+2} = (B'_k, e_{k+1}e_{k+2}B'_k, e_{k+1}B'_k, e_{k+2}B'_k)$. It is straightforward to verify that A_{k+2} has resolution IV or higher and $(A_{k+2}, B_{k+2}, B'_{k+2})$ has resolution III or higher.

Essentially, A_{k+2}, B_{k+2} and B'_{k+2} are obtained by doubling A_k, B_k and B'_k twice, respectively. However, their columns are re-arranged in order for $(A_{k+2}, B_{k+2}, B'_{k+2})$ to have resolution III or higher. The above construction of A_{k+2} and B_{k+2} from A_k and B_k gives a recursive construction of an SOA$(n, m, 8, 3)$ with property α.

Proposition 3. *Suppose that an SOA$(n, m, 8, 3)$ with property α is available. Then an SOA$(4n, 4m, 8, 3)$ with property α can be constructed.*

For $k = 4$ and $n = 16$, one can easily see that $A_4 = (e_1, e_2, e_3, e_4, e_1e_2e_3e_4)$ and $B_4 = (e_3e_4, e_1e_4, e_1e_2, e_2e_3, e_1e_3)$ satisfy the requirement in Theorem 1. For $k = 7$ and $n = 128$, the required A_7 and B_7 have been found to have $m = 40$ factors, and are presented in Table 3.1.

Combining the construction of A_k and B_k for $k = 4$ and 7 with Propositions 2 and 3, we obtain another major result.

Theorem 2. *With the exception of $k = 5$, we have that*

(i) an SOA$(n = 2^k, m, 8, 3)$ with property α can be constructed for every $k \geq 4$ and for $m = 5n/16$ factors, and

(ii) the SOA$(n, 5n/16, 8, 3)$ with property α given in (i) has the maximum number of factors.

For $k = 5$, the maximum number m of factors for desired A and B is 9. This follows from Wu (1989) or by a direct check on a computer. We have that

$$A_5 = (e_1, e_2, e_3, e_4, e_5, e_1e_2e_3, e_1e_2e_4, e_1e_2e_5, e_1e_3e_4e_5),$$

$$B_5 = (e_4e_5, e_3e_5, e_1e_4, e_2e_3, e_1e_3, e_1e_2e_4e_5, e_1e_5, e_3e_4, e_1e_2),$$

Table 3.1: Designs A and B for constructing an SOA$(128, 40, 8, 3)$ with property α

A				B			
e_1	e_1e_6	e_1e_7	$e_1e_6e_7$	$e_2e_4e_6$	$e_3e_4e_5e_7$	$e_2e_3e_5e_6$	$e_3e_4e_5e_6$
e_2	e_2e_6	e_2e_7	$e_2e_6e_7$	$e_1e_3e_6$	$e_1e_3e_7$	$e_3e_4e_5$	e_3e_4
e_3	e_3e_6	e_3e_7	$e_3e_6e_7$	$e_1e_4e_5$	$e_1e_2e_3e_5$	$e_1e_2e_3$	e_1e_4
e_4	e_4e_6	e_4e_7	$e_4e_6e_7$	e_1e_2	e_2e_3	e_1e_3	$e_1e_3e_4$
$e_1e_2e_3e_4$	$e_1e_2e_3e_4e_6$	$e_1e_2e_3e_4e_7$	$e_1e_2e_3e_4e_6e_7$	$e_1e_3e_5e_6e_7$	$e_1e_2e_5e_7$	$e_1e_4e_5e_6$	$e_2e_4e_5e_6$
e_1e_5	$e_1e_5e_6$	$e_1e_5e_7$	$e_1e_5e_6e_7$	$e_2e_4e_7$	$e_2e_4e_5e_7$	$e_1e_3e_4e_5e_6$	$e_1e_2e_3e_5e_6$
e_2e_5	$e_2e_5e_6$	$e_2e_5e_7$	$e_2e_5e_6e_7$	$e_1e_4e_6$	$e_1e_4e_7$	$e_3e_4e_6$	$e_2e_3e_4e_5e_6$
e_3e_5	$e_3e_5e_6$	$e_3e_5e_7$	$e_3e_5e_6e_7$	e_2e_4	$e_1e_2e_3e_7$	$e_2e_3e_5$	$e_1e_2e_6$
e_4e_5	$e_4e_5e_6$	$e_4e_5e_7$	$e_4e_5e_6e_7$	$e_2e_3e_4$	$e_1e_3e_5$	$e_1e_2e_5$	$e_1e_2e_4e_5$
$e_1e_2e_3e_4e_5$	$e_1e_2e_3e_4e_5e_6$	$e_1e_2e_3e_4e_5e_7$	$e_1e_2e_3e_4e_5e_6e_7$	$e_2e_3e_6e_7$	$e_2e_3e_5e_7$	$e_1e_3e_5e_6$	$e_2e_3e_6$

based on which an SOA$(32, 9, 8, 3)$ with property α can be constructed. This array was displayed earlier in Example 1.

In addition to A and B, to construct an SOA$(n, m, 8, 3)$ as in (3.1), design C is also required. For the SOAs discussed in this subsection, design C can be trivially obtained. For given $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_m)$, one can use $C = (c_1, \ldots, c_m)$ by taking c_j to be any column other than a_j, b_j and $a_j b_j$ for $j = 1, \ldots, m$.

3.3.2 Designs with property β

For this class of arrays, the following characterization can be obtained.

Theorem 3. *If an SOA$(n, m, 8, 3)$ is to be constructed using regular A, B and C, then it has property β if and only if A is of resolution IV or higher, $(B, B') \subseteq \bar{A}$ and (B, B') does not contain any interaction column involving two factors from A, where $\bar{A} = S \setminus A$.*

Proof. We only need to give a proof for the sufficiency part; as will be seen, all of our arguments are reversible. According to Lemma 1 and Proposition 1(ii), we need to show that (a_i, a_j, a_u, b_u) does not have a defining word for any distinct i, j, u. The three columns a_i, a_j, a_u do not form a word of length three because A has resolution IV or higher, Columns a_i, a_j, b_u do not form a word of length three because B does not contain an interaction column involving two factors from A. Columns a_i, a_u, b_u do not form a word of length three because $(B, B') \subseteq \bar{A} = S \setminus A$. Finally, a_i, a_j, a_u, b_u together cannot give a word of length four, since B' has no column that is an interaction involving two factors from A. We have thus completed the proof. $\square$

Note that (B, B') in Theorem 3 is allowed to have repeated columns. Based on Theorem 3 and the theory of doubling (Chen and Cheng 2006), the next result is established.

Proposition 4. *If an SOA$(n, m, 8, 3)$ with property β, as characterized in Theorem 3, exists, we must have that $m \leq n/4$.*

Proof. The basic idea of the proof is by contradiction. We will show that if $m \geq n/4 + 1$, then it is impossible to find designs A and B that satisfy the conditions as required in Theorem 3.

Now suppose $m \geq n/4 + 1$. As A has resolution IV or higher, it is either maximal or a projection design of a maximal design (Chen and Cheng 2006, Proposition 3.1). If A is maximal, then the two-factor interactions of A use up all the columns in $\bar{A}$. Therefore, there does not exist $B \subseteq \bar{A}$ that has no interaction column involving two factors of A.

Suppose that A is a projection design of a maximal design of resolution IV or higher, i.e., A is obtained by deleting columns from a maximal design. Let W be this maximal design with w columns. Obviously, $w \geq m + 1$. Then by Theorems 3.4 and 3.5 of Chen and Cheng (2006), we must have that $w = n/4 + 2^j$, where j is an integer satisfying $w \geq m + 1$, and that W can be obtained by doubling, repeatedly j times, a maximal design of $n/2^j$ runs for $n/2^{j+2} + 1$ factors. For W, consider its alias sets that do not contain main effects. Let $l_1, \ldots, l_f$ where $f = 2^k - 1 - w$ be the sizes of these alias sets. Then Theorems 2.2 and 3.2 of Chen and Cheng (2006) imply that $l_i \geq 2^j$ for all $i = 1, \ldots, f$. When one column is deleted from W, the size of each of these alias sets decreases at most by one, since the two-factor interactions associated with this deleted column are mutually orthogonal and thus no more than one of them can belong to the same alias set. Therefore, if less than 2^j columns are deleted from W, then none of these alias sets will become empty. Recall that A is obtained from W by deleting $w-m$ columns and $w-m = n/4+2^j-m < 2^j$ as $m > n/4$. This means that each of these alias sets contains at least one two-factor interaction of A. Because of this, the columns of B in Theorem 3 can only be selected from $W \setminus A$ for otherwise B will contain a two-factor interaction of A. Because W has resolution IV or higher, the columns of B' must be all outside of W, and thus are two-factor interactions of A. This contradicts to the requirement for B'. The proof is finally completed. $\square$

A construction of designs A and B required in Theorem 3 is now presented. Again, let $e_1, \ldots, e_k$ be independent factors. Let P_0 consist of $e_3, \ldots, e_k$ and all of their interactions, and let $P = (I, P_0)$ where I is the all-ones column. Then $S = (P_0, e_1P, e_2P, e_1e_2P)$. Now take $A = e_1P$ and $B = e_2P$. It can be easily seen that such A and B meet the requirements in Theorem 3. Note that A and B have $m = n/4$ factors.

Theorem 4. *With the above choice of A and B, we have that*

(i) an SOA($n = 2^k, m, 8, 3$) with property β can be constructed for $m = n/4$ factors, and

(ii) the SOA($n, n/4, 8, 3$) given in (i) has the maximum number of factors among all SOA($n, m, 8, 3$) with property β.

Theorem 4 is a direct consequence of Proposition 4 and Theorem 3.

The remark about C at the end of subsection 3.3.1 equally applies here.

3.3.3 Designs with more than one of properties α, β and γ

The problem of constructing SOA$(n, m, 8, 3)$s with property α or β has been completely solved in the previous two subsections. We now consider the construction of SOA$(n, m, 8, 3)$s with property γ. SOA$(n, m, 8, 3)$s with property γ achieve stratifications on 2×8 and 8×2 grids in all two-dimensions. Unlike those with property α or β, a simple characterization as in Theorems 1 or 3 is not available for arrays with property γ, but a sufficient condition can be given.

Proposition 5. *Suppose that an SOA*$(n, m, 8, 3)$ *is to be constructed using regular* A, B *and* C. *If* A *and* B *satisfy that (i)* A *is resolution IV or higher, (ii)* $(B, B') \subseteq \bar{A}$ *shares no common column with* $A^{(2)}$ *where* $A^{(2)}$ *collects all the two-factor interactions of* A, *and (iii) the set of distinct columns in* $(B, B', A^{(2)})$ *is a subset of* $\bar{A}$ *but is unequal to* $\bar{A}$, *then an SOA*$(n, m, 8, 3)$ *with property* γ *can be constructed. This array also has property* β.

Proof. Take $C = (c_1, \ldots, c_m)$ where $c_j = c$ is a column in $\bar{A}$ but not any column of $(B, B', A^{(2)})$. Then we can easily check that (a_i, a_j, b_j, c_j) is an orthogonal array of strength four for any $i \neq j$. This shows that array D constructed in (3.1) is an SOA$(n, m, 8, 3)$ with property γ. This array also has property β because A and B meet the requirements in Theorem 3. $\square$

Let us go back to $S = (P_0, e_1P, e_2P, e_1e_2P)$ where $P = (I, P_0)$ and P_0 consists of $e_3, \ldots, e_k$ and all their interactions. If we take $A = e_1P_0$ and $B = e_2P_0$, then the conditions in Proposition 5 are all met.

Proposition 6. *An SOA*$(n = 2^k, m, 8, 3)$ *with properties* β *and* γ *can be constructed for* $m = n/4 - 1$ *factors.*

Proposition 6 inspires us to consider SOA$(n, m, 8, 3)$s with all properties α, β and γ. It is intriguing and somewhat surprising that they can be constructed for $m = n/4 - 1$ factors.

Let X, Y, Z be three copies of P_0, which have the same set of columns as P_0 but with their columns ordered differently. Suppose that they can be found such that $x_jy_j = z_j$ where x_j, y_j, z_j are the jth column of X, Y, Z, respectively, for $j = 1, \ldots, m = n/4 - 1$. Then $S = (P_0, e_1, e_2, e_1e_2, e_1X, e_2Y, e_1e_2Z)$. Now if we take

$$A = e_1X \text{ and } B = e_2Y.$$

Then all the conditions in Theorems 1 and 3, and Proposition 5 are satisfied.

Theorem 5. *The above construction gives an SOA$(n, n/4-1, 8, 3)$ with all three properties α, β and γ.*

If we take

$$A = (e_1, e_1X) \text{ and } B = (e_2, e_2Y)$$

instead, then Theorems 1 and 3 both apply.

Corollary 1. *An SOA$(n, m, 8, 3)$ with properties α and β can be constructed for $m = n/4$ factors.*

It remains to establish the existence of X, Y and Z satisfying that $x_j y_j = z_j$. Suppose that X_k, Y_k and Z_k have this property and are based on k independent factors $e_1, \ldots, e_k$. Then X_{k+2}, Y_{k+2} and Z_{k+2} can be constructed recursively to have the same property as follows:

$$X_{k+2} = (X_k, e_{k+1}, e_{k+1}X_k, e_{k+2}, e_{k+2}X_k, e_{k+1}e_{k+2}, e_{k+1}e_{k+2}X_k),$$

$$Y_{k+2} = (Y_k, e_{k+2}, e_{k+2}Y_k, e_{k+1}e_{k+2}, e_{k+1}e_{k+2}Y_k, e_{k+1}, e_{k+1}Y_k),$$

$$Z_{k+2} = (Z_k, e_{k+1}e_{k+2}, e_{k+1}e_{k+2}Z_k, e_{k+1}, e_{k+1}Z_k, e_{k+2}, e_{k+2}Z_k).$$

For $k = 2$, X_k, Y_k, Z_k are given by $X_2 = (e_1, e_2, e_1e_2)$, $Y_2 = (e_2, e_1e_2, e_1)$ and $Z_2 = (e_1e_2, e_1, e_2)$. For $k = 3$, they are given by $X_3 = (e_1, e_2, e_1e_2, e_3, e_1e_3, e_2e_3, e_1e_2e_3)$, $Y_3 = (e_1e_2e_3, e_1e_3, e_2, e_1, e_2e_3, e_3, e_1e_2)$ and $Z_3 = (e_2e_3, e_1e_2e_3, e_1, e_1e_3, e_1e_2, e_2, e_3)$. We thus obtain the next result.

Theorem 6. *For any $k \geq 2$, X_k, Y_k, Z_k can be constructed to satisfy that $x_j y_j = z_j$.*

3.3.4 Three families of strength-three SOAs

Various characterizing and construction results have been presented in Subsections 3.3.1, 3.3.2 and 3.3.3. We now highlight three families of strength-three SOAs and summarize their constructions. This should be helpful to those readers who are mainly interested in the final products, namely better strength-three SOAs, and are less concerned with the theoretical characterizations of these arrays.

The first family is given by Theorem 2, which asserts that an SOA$(n = 2^k, m, 8, 3)$ can be constructed to have property α for $m = 5n/16$ with the exception for $k = 5$ in which case $m = 9$ instead of $m = 10$. The construction of this array D is via (3.1) that requires arrays A, B and C. Arrays A and B can be constructed recursively by (3.2), with A_4 and B_4 given right after Proposition 3, and A_7 and B_7 given in Table 3.1. Once $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_m)$ are available, array $C = (c_1, \ldots, c_m)$ can easily be obtained by taking c_j to be any column other than a_j, b_j and $a_j b_j$ for $j = 1, \ldots, m$. Obviously, any array given by deleting columns from the SOA$(n = 2^k, 5n/16, 8, 3)$ still has property α. But if $m \leq n/4$, better arrays exist, which we discuss next.

The second family of strength-three SOAs has properties both α and β. This is given by Corollary 1, which says that an SOA$(n = 2^k, m, 8, 3)$ with properties both α and β can be constructed for $m = n/4$ factors. If one needs an array with less than $n/4$ factors, one can simply delete some columns from the SOA$(n = 2^k, n/4, 8, 3)$ and the resulting array still has properties both α and β. This is unnecessary, however, due to the availability of the third family of strength-three SOAs.

The third family of strength-three SOAs enjoys all properties α, β and γ. According to Theorem 5, we can construct an SOA$(n = 2^k, m, 8, 3)$ with all properties α, β and γ for

$m = n/4 - 1$ factors. Immediately available is an SOA$(n = 2^k, m, 8, 3)$ with all properties α, β and γ for $m < n/4 - 1$, which can be obtained by deleting columns from the one for $m = n/4 - 1$.

We have seen that the first, second and third families of strength-three SOAs enjoy increasingly better space-filling properties.

The constructions for the second and third families of SOAs are very much related, of which we now give a summary. The saturated design S based on k independent factors $e_1, \ldots, e_k$ can be written as

$$S = (P_0, e_1, e_2, e_1e_2, e_1X, e_2Y, e_1e_2Z),$$

where P_0 consists of $e_3, \ldots, e_k$ and all their interactions, and X, Y and Z are three copies of P_0 obtained by permuting the columns of P_0 such that $x_jy_j = z_j$ with x_j, y_j and z_j being the jth column of X, Y and Z, respectively. The existence of X, Y and Z having this property is guaranteed by Theorem 6, with their constructions provided right before that theorem.

Recall the construction of strength-three SOAs using (3.1), which needs three arrays A, B and C. The second family of SOAs chooses

$$A = (e_1, e_1X) \text{ and } B = (e_2, e_2Y),$$

with array $C = (c_1, \ldots, c_m)$ given by taking c_j to be any column other than a_j, b_j and a_jb_j where a_j and b_j are the jth columns of A and B, respectively. The third family of SOAs uses

$$A = e_1X \text{ and } B = e_2Y,$$

along with array $C = (c_1, \ldots, c_m)$ given by taking $c_j = e_1$ for all $j = 1, \ldots, m$.

Table 3.2: Maximum numbers of factors SOAs of strength three and four

k	$n = 2^k$	Family 1	Family 2	Family 3	
			strength three		strength four
4	16	5	4	3	2
5	32	9	8	7	3
6	64	20	16	15	4
7	128	40	32	31	5
8	256	80	64	63	8

Table 3.2 provides a numerical comparison of the maximum numbers of factors for three families of SOAs of strength three and SOAs of strength four.

Theorem 5 is a remarkable result, as SOA$(n, m, 8, 3)$s with all three properties α, β and γ achieve stratifications on (i) 4×4, 8×2 and 2×8 grids in all two-dimensions and (ii) $4 \times 2 \times 2$, $2 \times 4 \times 2$ and $2 \times 2 \times 4$ grids in all three-dimensions. The only property they do not have, when compared with SOAs of strength four, is to achieve stratifications on $2 \times 2 \times 2 \times 2$ grids in four-dimensions. SOAs of strength four are very expensive to construct. According to He and Tang (2013, Theorem 1), SOA$(n, m, 16, 4)$ can be constructed for up to $m = [M(k)/2]$ factors where $M(k)$ is the maximum number of factors for resolution V designs with $n = 2^k$ runs. Note that $M(k)/2$ is much smaller than $n/4 - 1$ as $M(k)$ is in the order of $O(\sqrt{n})$.

3.4 Discussion

This chapter introduces and constructs several families of strength-three SOAs that enjoy some of the space-filling properties of strength-four SOAs. Various characterizing and construction results are presented. The theory of maximal designs and their doubling constructions plays a crucial role in many of the theoretical arguments. Strength-three SOAs constructed in this chapter should provide very useful space-filling designs for computer experiments.

The current chapter focuses on the construction using regular 2^{m-p} designs. One interesting direction worth pursuing is to consider the construction using regular s^{m-p} designs. Some results are possible although it seems unlikely that we can obtain many rich results as what has been done using 2^{m-p} designs. Another research direction is to examine the use of non-regular two-level designs. We feel that some of the constructions in this chapter could be generalized to include non-regular situations. This deserves further investigation.

Chapter 4

Design Selection for Strong Orthogonal Arrays of Strength $2+$ and 2

4.1 Introduction

Computer experiments call for space-filling designs. Roughly speaking, a space-filling design refers to any design that spreads out its points in the design region in some uniform fashion. One attractive method of constructing such designs is to use orthogonal arrays or similar structures. Alternatively, space-filling designs can also be generated using discrepancy and distance criteria. We refer to Santner, Williams and Notz (2003) and Fang, Li and Sudjianto (2006) for some detailed discussion of the ideas and methods.

We concentrate in this chapter on the construction of space-filling designs based on orthogonal arrays. This approach started in McKay, Beckman and Conover (1979) when they introduced Latin hypercube designs, which are orthogonal arrays of strength one, and continued with the work of Owen (1992) and Tang (1993), who proposed the use of orthogonal arrays to construct space-filling designs. Two decades later, inspired by (t, m, s)-nets from quasi-Monte Carlo (Niederreiter 1992), He and Tang (2013) introduced strong orthogonal arrays (SOAs) and studied their applications to the construction of space-filling Latin hypercubes. These arrays of strength t are more space-filling in g dimensions for any $g < t$ than comparable ordinary orthogonal arrays. As discussed in He and Tang (2014), SOAs of strength 3 are most useful because they possess better space-filling properties than ordinary orthogonal arrays of strength 3 yet can be constructed from the latter at almost no cost. For certain investigations, however, SOAs of strength 3 may be too expensive for the experimenter to afford. This leads He, Cheng and Tang (2018) to the introduction of SOAs of strength 2+.

Despite of these theoretical developments, an important problem that has not been addressed in the literature is that of design selection for SOAs. In this chapter, we conduct a systematic investigation into this problem. Our focuses are on SOA$(n, m, 4, 2+)$s and SOA$(n, m, 4, 2)$s as these represent most useful situations.

SOAs of strength 2+ enjoy the same 2-dimensional space-filling properties as, but can accommodate many more factors than, SOAs of strength 3 for given run sizes. He, Cheng and Tang (2018) considered the construction of SOAs of strength 2+ using regular 2^{m-p} designs, and found that such arrays can all be constructed from second order saturated (SOS) designs. As numerous SOS designs are available for a given run size and number of factors, the question as to which one to use naturally arises. We address this issue of design selection by formulating a criterion based on the 3-dimensional properties of SOAs of strength 2+. It turns out that SOS designs with the maximum numbers of defining words of length three give the best SOAs of strength 2+ according to this criterion. One of the four constructions in He, Cheng and Tang (2018) is proved to be optimal. Search results for designs of up to 64 runs are obtained and tabulated.

When the number of factors is large relative to the run size, SOAs of strength 2+ do not exist. Although an SOA of strength 2 only guarantees the same space-filling properties in all 2-dimensions as an ordinary orthogonal array of strength 2, some of its subarrays of two factors can have better space-filling properties. We find that the number of such better subarrays of two columns is directly linked to the number of degrees of freedom of a related design for estimating main effects and two-factor interactions. Therefore, SOAs of strength 2 can be selected by maximizing this number of degrees of freedom, thus providing a general framework for the selection of SOAs of strength 2. The framework is practically useful as it is applicable when SOAs of strength 2+ do not exist, and conceptually helpful as it gives rise to SOAs of strength 2+ when they do exist.

This chapter is organized as follows. Section 4.2 introduces notation and provides some background. Section 4.3 examines the selection of SOAs of strength 2+ while Section 4.4 deals with the selection of SOAs of strength 2. We conclude this chapter with a discussion in Section 4.5.

4.2 Notation and Background

This section introduces some necessary notation and reviews background material to prepare for the rest of this chapter. An orthogonal array of n runs, m factors, and strength t is an $n \times m$ matrix with the jth column having levels $0, 1, \ldots, s_j - 1$ in which any $n \times t$ submatrix has the property that all possible combinations of levels occur with the same frequency. We denote such an array by $\mathrm{OA}(n, m, s_1 \times \cdots \times s_m, t)$. When $s_1 = \ldots = s_m = s$, this array is symmetric and simply denoted by $\mathrm{OA}(n, m, s, t)$, otherwise it is asymmetric. We refer to Dey and Mukerjee (1999), Hedayat, Sloane and Stufken (1999), and Cheng (2014) for general references on orthogonal arrays. An SOA of n runs, m factors, s^t levels, and strength t is an $n \times m$ matrix with entries from $\{0, 1, \ldots, s^t - 1\}$ in which any subarray of g columns for any g with $1 \leq g \leq t$ can be collapsed into an $\mathrm{OA}(n, g, s^{u_1} \times \cdots \times s^{u_g}, g)$ for any positive integers $u_1, \ldots, u_g$ with $u_1 + \cdots + u_g = t$, where collapsing s^t levels into s^{u_j} levels is according to $\lfloor a/s^{t-u_j} \rfloor$ for $a = 0, 1, \ldots, s^t - 1$ (He and Tang, 2013). Such an array is denoted by $\mathrm{SOA}(n, m, s^t, t)$.

By the definition of SOAs, an SOA$(n, m, s^2, 2)$ is an orthogonal array of strength one in itself and becomes an orthogonal array of strength 2 if its s^2 levels are collapsed into s levels using $[a/s]$. This implies that an SOA$(n, m, s^2, 2)$ achieves a stratification on an $s \times s$ grid in any 2-dimension just like an OA$(n, m, s, 2)$. On the other hand, an SOA$(n, m, s^3, 3)$ achieves stratifications on $s^2 \times s$ and $s \times s^2$ grids in 2-dimensions and $s \times s \times s$ grids in 3-dimensions. While achieving stratifications on $s \times s \times s$ grids in 3-dimensions, an OA$(n, m, s, 3)$ only promises stratifications on $s \times s$ grids in 2-dimensions.

Intermediate between SOAs of strength 2 and 3 are SOAs of strength 2+, which in a nutshell are SOAs of strength 2 but achieve stratifications on $s^2 \times s$ and $s \times s^2$ grids in 2-dimensions just like SOAs of strength 3. More precisely, an $n \times m$ matrix with entries from $\{0, 1, \ldots, s^2 - 1\}$ is called an SOA of strength 2+, and with n runs and m factors of s^2 levels, if any subarray of two columns can be collapsed into an OA$(n, 2, s^2 \times s, 2)$ and an OA$(n, 2, s \times s^2, 2)$. We denote this array by SOA$(n, m, s^2, 2+)$. He, Cheng and Tang (2018) presented a characterization for SOA$(n, m, s^2, 2+)$s.

Lemma 1. *An SOA$(n, m, s^2, 2+)$, say D, exists if and only if there exist two arrays A and B where $A = (a_1, \ldots, a_m)$ is an OA$(n, m, s, 2)$ and $B = (b_1, \ldots, b_m)$ is an OA$(n, m, s, 1)$ such that (a_j, a_u, b_u) is an orthogonal array of strength 3 for any $j \neq u$. The three arrays are linked through $D = sA + B$.*

He, Cheng and Tang (2018) considered the construction of SOA$(n, m, 4, 2+)$s using regular 2^{m-p} designs with levels ± 1. A regular saturated design S of $n = 2^k$ runs for $n - 1$ factors can be obtained by first writing down a full factorial for k factors and then adding all possible interaction columns. If we regard S as a set of $n-1$ columns, then a subset C of m columns is a 2^{m-p} design where $p = m - k$. The columns outside C form the complementary design of C, denoted by $\bar{C} = S \setminus C$. Because S is regular, it follows that $ab \in S$ for any $a, b \in S$, $a \neq b$, where ab stands for the interaction column. Block and Mee (2003) defined second order saturated (SOS) designs as those in which all degrees of freedom can be used to estimate main effects or two-factor interactions. In our notation, a design C is an SOS design if any $d \in \bar{C}$ can be written as $d = ab$ for some $a, b \in C$.

When the two levels of A and B are ± 1, we should use

$$D = A + B/2 + 3/2 \tag{4.1}$$

in Lemma 1 instead of $D = 2A + B$ to construct SOA$(n, m, 4, 2+)$s. He, Cheng and Tang (2018) obtained the following result.

Lemma 2. *If an SOA of strength* $2+$ *is to be constructed through (4.1) where* A *is of resolution III or higher and* B *of resolution II or higher with their columns selected from* S*, a saturated regular design, then it is necessary and sufficient that* $\bar{A}$ *is an SOS design.*

For given n and m, many SOS designs are available and can all be used to construct SOA$(n, m, 4, 2+)$s. Although these SOA$(n, m, 4, 2+)$s have the same 2-dimensional properties, their 3-dimensional properties can be very different from one choice of an SOS design to another. The next section investigates how to choose an SOS design in order to obtain the best SOA$(n, m, 4, 2+)$.

4.3 Design Selection for SOAs of Strength $2+$

In Lemma 2, we only require A to be of resolution III or higher in order for design D from (4.1) to be an SOA$(n, m, 4, 2+)$. If A has resolution IV or higher, then the resulting design D also achieves stratifications on $2 \times 2 \times 2$ grids in all 3-dimensions. For A to have resolution IV or higher while keeping $\bar{A}$ to be SOS, we must have that $m \leq n/2 - 1$. This is actually the range of m values for which SOAs of strength 3 can be constructed (He and

Tang 2013). When $m \geq n/2$, SOAs of strength 3 do not exist but SOAs of strength 2+ can be constructed. This is the case we focus in this section.

Let $m \geq n/2$ and A have resolution III such that $\bar{A}$ is SOS. If three columns a_i, a_j, a_k of A do not form a defining word, then the subarray of D in (4.1) consisting of the corresponding columns d_i, d_j, d_k achieves a stratification on a $2 \times 2 \times 2$ grid. Let $W_3(A)$ be the number of words of length 3 in the defining relation of A. Then our criterion for selecting the best A to use is to minimize $W_3(A)$ from among all designs A's such that $\bar{A}$'s are SOS. By doing so, the resulting SOA$(n, m, 4, 2+)$ maximizes the number of subarrays of three columns that enjoy stratifications on $2\times2\times2$ grids, and is said to be optimal for convenience of referencing.

According to Chen and Hedayat (1996) and Tang and Wu (1996), we have that $W_3(A) = \text{constant} - W_3(\bar{A})$, which means that minimizing $W_3(A)$ can be done by maximizing $W_3(\bar{A})$. Thus, the problem of finding a best SOA$(n, m, 4, 2+)$ becomes that of finding an SOS design with the maximum number of defining words of length 3. We summarize the above development in a proposition.

Proposition 1. *The SOA$(n, m, 4, 2+)$ constructed through (4.1) in Lemma 2 has the most three-column subarrays that achieve stratifications on $2 \times 2 \times 2$ grids if and only if $\bar{A}$ maximizes $W_3(\bar{A})$ among all SOS designs.*

Let $C = \bar{A}$. Without restricting to SOS designs, design C with the maximum $W_3(C)$ has been constructed in Chen and Hedayat (1996). Finding C with the maximum $W_3(C)$ among all SOS designs appears to be a much harder problem as it is no easy task to find all SOS designs. This is analogous to that an optimization problem with certain constraints is often more difficult than that without constraints. Nonetheless, we are able to establish one theoretical result, to be discussed below.

He, Cheng and Tang (2018) presented four constructions of SOS designs. As before, S is a saturated design generated by k independent factors. Let P be a subset of S consisting of k_1 independent factors and all their interactions and Q be a subset of S consisting of the remaining k_2 independent factors and all their interactions, where $1 \leq k_1 \leq k_2 \leq k$, and $k_1 + k_2 = k$. Then the first construction of SOS designs in He, Cheng and Tang (2018) is given by $C^* = P \cup Q$.

Theorem 1. *The SOS design C^* given above maximizes $W_3(C)$ among all SOS designs C of $n = 2^k$ runs for $f = 2^{k_1} + 2^{k_2} - 2$ factors.*

Proof. Let $C = (c_1, \ldots, c_f)$ be SOS with c_j being the jth column of C. Because C is regular, we must have $c_j c_u \in C$ or $c_j c_u \in \bar{C}$ for any $j < u$. This implies that

$$f(f-1)/2 = 3W_3(C) + W_{2,1}(C),$$

where $W_{2,1}(C)$ is the number of pairs (c_j, c_u) such that $c_j c_u \in \bar{C}$. Thus maximizing $W_3(C)$ is equivalent to minimizing $W_{2,1}(C)$. Since C is SOS, for any $c \in \bar{C}$ there must exist a pair (c_j, c_u) such that $c = c_j c_u$. This shows that

$$W_{2,1}(C) \geq n - 1 - f. \tag{4.2}$$

Now consider a pair of columns c_j and c_u from design $C^* = P \cup Q$. They must satisfy that $c_j c_u \in P$ if both c_j and c_u are from P, $c_j c_u \in Q$ if both c_j and c_u are from Q, and $c_j c_u \in \bar{C^*}$ if $c_j \in P$ and $c_u \in Q$. We therefore have that

$$W_{2,1}(C^*) = (2^{k_1} - 1)(2^{k_2} - 1) = 2^{k_1+k_2} - 1 - (2^{k_1} + 2^{k_2} - 2) = n - 1 - f,$$

and thus $W_{2,1}(C^*)$ attains the lower bound for $W_{2,1}$ of SOS designs as given in (4.2). This completes the proof. $\square$

By taking $k_1 = 1, \ldots, [k/2]$ for given k, Theorem 1 gives $[k/2]$ SOS designs with the maximum numbers of defining words of length 3. Although Theorem 1 applies to $k_1 = 1$, this is not a case of interest because $\bar{C}^*$ is of resolution IV and has $m = n/2 - 1$ factors - see the opening paragraph of this section. Thus for any given k, Theorem 1 can be used to construct $[k/2] - 1$ optimal SOA$(2^k, m, 4, 2+)$'s with $m = (2^{k_1} - 1)(2^{k_2} - 1)$ factors. For $k = 8$, they are an SOA$(256, 189, 4, 2+)$, an SOA$(256, 217, 4, 2+)$ and an SOA$(256, 225, 4, 2+)$.

We next conduct a computer search of SOS designs with the maximum values of W_3 for up to 64 runs. For designs of 16 and 32 runs, the complete catalogs in Chen, Sun and Wu (1993) can be used to first identify all the SOS designs and then to find those with the maximum W_3. For designs of 64 runs, Chen, Sun and Wu (1993) only obtained a complete catalog of resolution IV designs, which is not useful for our purpose. Hongquan Xu kindly generated for us a complete catalog of resolution III designs of 64 runs for up to 17 factors, which we have used in our computer search.

Alternatively, our computer search can be based on minimal SOS designs. An SOS design is said to be minimal if it is no longer SOS when any of its factors is removed (Cheng, He & Tang, 2018). Any SOS design is either minimal or can be obtained by adding factors to a minimal SOS design. Thus, all SOS designs can be obtained if all minimal SOS designs are available. By computer search, Davydov, Marcugini and Pambianco (2006) found all minimal SOS designs (called minimal 1-saturating sets in projective geometry) for up to 64 runs, and have made them available at https://arxiv.org/abs/1802.04214. This complete catalog of minimal SOS designs makes it possible to conduct a complete search of designs of 64 runs, at least in principle.

We have conducted a complete search for all possible $f \leq n/2 - 1$ when $n = 16$ and 32 and all possible $f \leq 20$ when $n = 64$, where f is the number of factors. The resulting SOS designs with the maximum W_3 are presented in Table 4.1. The SOS designs with the maximum W_3 are generally not unique up to isomorphism. To save space, Table 4.1 only includes one design for each combination of n and f.

For $n \leq 32$ with $f \leq n/2 - 1$ and $n = 64$ with $f \leq 17$, these SOS designs are found by searching through the complete catalog of resolution III designs. For $n = 64$ with $f = 18$, 19 and 20, the complete list of SOS designs with f factors is obtained by adding $f - 17$ factors to all SOS designs with $f = 17$ plus all minimal SOS designs of f factors.

Table 4.1 also provides results for $n = 64$ with $f \geq 21$. Our search is incomplete because the number of all SOS designs becomes exceedingly large even for computer to handle. Our designs are found by maximizing W_3 among all SOS designs obtained by adding factors to the SOS design with $f = 18$ given by Theorem 1. According to the results of the complete search, we find that the SOS designs with the largest W_3 are either from Theorem 1 or can be obtained by adding factors to the designs given by Theorem 1 except the following four cases: $n = 16$ with $f = 5$, $n = 32$ with $f = 9$, and $n = 64$ with $f = 13$ and 17. Interestingly, the best SOS designs for these cases are all given by the second construction of He, Cheng and Tang (2018). Since no design with $21 \leq f \leq 31$ and $n = 64$ permits such a construction, our results in Table 4.1 for $f \geq 21$ and $n = 64$ should be very close to the designs with the maximum W_3 if they are not the best.

We use the same idea as that in Chen, Sun and Wu (1993) and Xu (2009) to present our designs in Table 4.1 (and Table 4.2 in the next section). Each design is represented by a set of f columns with the k independent columns labeled as $1, 2, 4, \ldots, 2^{k-1}$. We omit the independent columns in Table 4.1, and only provide the interaction columns which are given under the heading 'Additional Columns'. We also give the numbers of words of length 3 and 4 in column $W = (W_3, W_4)$.

k	f	Additional Columns	(W_3, W_4)
4	5	15	0 0
	6	3 12	2 0
	7	15 3 7	3 2
5	9	15 19 17 18	4 3
	10	7 24 3 5 6	8 7
	11	15 19 7 17 18 3	9 11
	12	7 11 19 5 24 6 3	11 16
	13	7 11 19 5 24 6 27 3	14 23
	14	31 7 11 13 14 3 5 6 15	18 42
	15	31 7 11 13 14 3 5 9 6 15	23 60
6	13	15 19 39 17 40 18 47	8 6
	14	7 56 3 24 5 40 6 48	14 14
	15	15 19 39 17 40 18 47 3 7	15 20
	16	15 19 39 21 40 3 47 18 17 7	17 29
	17	63 7 11 13 14 3 5 10 12 6 9	28 77
	18	3 5 6 7 9 10 11 12 13 14 15 48	36 105
	19	3 5 6 7 9 10 11 12 13 14 15 48 17	37 113
	20	3 5 6 7 9 10 11 12 13 14 15 48 17 18	39 128
	21	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19	42 151
	22	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20	46 180
	23	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21	51 218
	24	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22	57 265
	25	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22 23	64 322
	26	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22 23 24	72 379
	27	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22 23 24 25	81 447
	28	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22 23 24 25 26	91 526
	29	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22 23 24 25 26 27	102 617
	30	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22 23 24 25 26 27 28	114 718
	31	3 5 6 7 9 10 11 12 13 14 15 48 17 18 19 20 21 22 23 24 25 26 27 28 29	127 832

Table 4.1: SOS designs with the largest W_3, where basic columns $1, 2, 4, 8, 16$ and 32 are omitted.

By Proposition 1, an SOS design of n runs and f factors with the maximum W_3 can be used to construct an optimal SOA$(n, m, 4, 2+)$ for $m = n - 1 - f$ factors. Thus, Table 4.1 allows the construction of optimal SOA$(n, m, 4, 2+)$s of 16 runs for $8 \leq m \leq 10$ factors, of 32 runs for $16 \leq m \leq 22$ factors, and of 64 runs for $32 \leq m \leq 50$ factors. For $m \leq n/2 - 1$ when $n = 16$, 32 and 64, SOAs of strength 3 can be constructed. For $m \geq 11$ when $n = 16$, $m \geq 23$ when $n = 32$ and $m \geq 51$ when $n = 64$, SOAs of strength 2+ do not exist because the corresponding SOS designs do not exist. This is the subject of the next section.

We provide an example to illustrate the results in this section.

Example 1. To construct an optimal SOA$(16, 9, 4, 2+)$, we need two designs A and B where A is such that $C = \bar{A}$ must be SOS and has the largest W_3 value. Clearly this corresponds to $k = 4$ and $f = 6$. From Table 4.1, we see that $C = (1, 2, 4, 8, 3, 12)$, collecting additional columns 3 and 12 besides the independent columns 1, 2, 4, 8. This means that $A = (5, 6, 7, 9, 10, 11, 13, 14, 15)$. Write $A = (a_1, a_2, \ldots, a_9)$. We then need to find $B = (b_1, b_2, \ldots, b_9)$. For a_j, we can take any b from C as b_j so long as $a_j b$ is in C and such b must exist because C is SOS. One choice is $B = (4, 4, 4, 8, 8, 8, 12, 12, 12)$. Full displays of designs A and B are given below

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix}.$$

Now using $D = A + B/2 + 3/2$ as in (4.1), we obtain design D below, which is an optimal SOA$(16, 9, 4, 2+)$:

$$D = \begin{bmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 1 & 3 & 1 & 1 & 3 & 1 & 1 & 3 & 1 \\ 3 & 1 & 1 & 3 & 1 & 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 & 1 & 3 & 1 & 1 & 3 \\ 0 & 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 \\ 2 & 0 & 2 & 1 & 3 & 1 & 2 & 0 & 2 \\ 0 & 2 & 2 & 3 & 1 & 1 & 0 & 2 & 2 \\ 2 & 2 & 0 & 1 & 1 & 3 & 2 & 2 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 1 & 2 & 0 & 2 & 2 & 0 & 2 \\ 3 & 1 & 1 & 0 & 2 & 2 & 0 & 2 & 2 \\ 1 & 1 & 3 & 2 & 2 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 \\ 2 & 0 & 2 & 2 & 0 & 2 & 1 & 3 & 1 \\ 0 & 2 & 2 & 0 & 2 & 2 & 3 & 1 & 1 \\ 2 & 2 & 0 & 2 & 2 & 0 & 1 & 1 & 3 \end{bmatrix}.$$

.

4.4 Design Selection for SOAs of Strength 2

Let f_k be the size of the smallest SOS design with $n = 2^k$ runs. For $k = 4, 5, 6$ and 7, the exact value of f_k is known and is equal to 5, 9, 13 and 19, respectively. For $k \geq 8$, the exact value of f_k is still unknown but useful bounds are available. Results on f_k, both theoretical and computational, can be found in Davydov, Marcugini and Pambianco (2006) and Cheng, He and Tang (2018). When $f \leq f_k - 1$, an SOS design of $n = 2^k$ runs for f factors does not exist. Correspondingly, an SOA$(2^k, 2^k - 1 - f, 4, 2+)$ does not exist. But SOAs of strength 2 can always be constructed. This section is devoted to the selection of SOAs of strength 2.

To construct an SOA$(2^k, m, 4, 2+)$ from a given $A = (a_1, \ldots, a_m)$ by Lemma 1, we need to find a $B = (b_1, \ldots, b_m)$ such that (a_j, a_u, b_u) is an orthogonal array of strength 3 for all $j \neq u$. Lemma 2 says that such a B can always be found if $\bar{A}$ is SOS. In fact, it can be obtained as follows. Because $\bar{A}$ is SOS, any a_j can be written as $a_j = b_j b_j'$ for some b_j and b_j' from $\bar{A}$. We then simply take $B = (b_1, \ldots, b_m)$. Such a choice of B guarantees that (a_j, a_u, b_u) is of strength 3 for all $j \neq u$.

When $\bar{A}$ is not SOS, it is impossible to find a B such that (a_j, a_u, b_u) has strength 3 for all $j \neq u$. There are in total $m(m-1)$ arrays of form (a_j, a_u, b_u) for $j \neq u$. Although we cannot make all of them have strength 3, we can choose A and B so that as many of them as possible have strength 3. Let M be the number of arrays (a_j, a_u, b_u)'s that have strength 3. We want to maximize M in our choices of A and B.

A more direct justification for maximizing M is obtained by looking at the 2-dimensional projection properties of design D in (4.1) constructed using A and B. For fixed $j \neq u$, we know that array (d_j, d_u) achieves a stratification on a 2×4 grid if and only if array (a_j, a_u, b_u) has strength 3, where d_j and d_u are the jth and uth columns of D in (4.1). This means that M also represents the number of arrays (d_j, d_u) that achieve stratifications on 2×4 grids. Therefore, when maximizing M, we are maximizing the number of arrays (d_j, d_u) achieving stratifications on 2×4 grids.

To maximize M over all choices of A and B, our strategy is to first maximize M over all choices of B for given A and then maximize M over all choices of A.

Consider a given $A = (a_1, \ldots, a_m)$ such that $\bar{A}$ is not SOS. Any column a of A either has the property that it can be written as $a = bb'$ for some b, b' in $\bar{A}$ or does not have this property. Let A_1 collect all columns having this property and A_2 collect the remaining columns. We then have $A = (A_1, A_2)$. Let m_1 and m_2 be the numbers of the columns in A_1 and A_2, respectively. Now we construct array B as follows:

(i) for a given column a_u from A_1, choose the corresponding b_u where $a_u = b_u b'_u$ with b_u, b'_u both from $\bar{A}$,

(ii) for a given column a_u from A_2, choose any b from $\bar{A}$ as b_u.

Proposition 2. *The above construction of B maximizes M for given A. The maximum value of M is given by $M = (m-2)m + m_1$.*

Proof. For given $u = 1, \ldots, m_1$, our choice of b_u is such that $a_u = b_u b'_u$ where b_u, b'_u are both in $\bar{A}$, which means that $a_u b_u = b'_u \neq a_j$ for all $j \neq u$. Thus, array (a_j, a_u, b_u) has strength 3 for all $j \neq u$. For $u = m_1 + 1, \ldots, m$, choosing b_u from A, say $b_u = a$ where $a \in A$, will cause array (a, a_u, b_u) to have strength one and therefore at most $(m-2)$ arrays of form (a_j, a_u, b_u) with $j \neq u$ can have strength 3. Now consider choosing b_u from $\bar{A}$ for $u = m_1 + 1, \ldots, m$. Because a_u is from A_2, we have that $a_u b_u = a_{j'}$ for some j'. This means that (a_j, a_u, b_u) has strength 2 if $j = j'$ and has strength 3 if $j \neq j'$ and $j \neq u$. With such a choice of b_u, exactly $(m-2)$ arrays (a_j, a_u, b_u), $j \neq u$, have strength 3. Thus, choosing b_u from $\bar{A}$ gives at least as many strength 3 arrays (a_j, a_u, b_u) as choosing b_u from A. This shows that our construction of B maximizes M for the given A. The argument also shows that the number of strength 3 arrays (a_j, a_u, b_u) is given by $M = (m-1)m_1 + (m-2)m_2 = (m-2)m + m_1$. □

The above deals with how to choose B for given A. From Proposition 2, it is evident that one should choose A with the largest value of m_1. Let ν be the number of degrees of freedom of design $\bar{A}$ for estimating main effects and two-factor interactions as defined in Block and Mee (2003). In other words, ν is the number of alias sets that contain main effects or two-factor interactions for design $\bar{A}$. Then we have that $\nu = f + m_1$ where $f = 2^k - 1 - m$ is the number of columns of $\bar{A}$. This establishes the following result.

Theorem 2. *With the construction for B for given A, the array D, an SOA$(n = 2^k, m, 4, 2)$, constructed in (4.1) maximizes the number of arrays (d_j, d_u) achieving stratifications on 2×4 grids if and only if design $\bar{A}$ maximizes ν, its number of degrees of freedom.*

If $\bar{A}$ is SOS, then ν is maximized at $\nu = n - 1$ and correspondingly $m_1 = m$ and $m_2 = 0$. In this case, all arrays (d_j, d_u) achieve stratifications on 2×4 grids. Thus, any array of two columns achieves stratifications on both 2×4 and 4×2 grids. The resulting array D is an SOA of strength 2+. Therefore, Theorem 2 generalizes a result of He, Cheng and Tang (2018, Theorem 1) that characterizes SOAs of strength 2+ through SOS designs.

When SOAs of strength 2+ do not exist, Theorem 2 provides a method of constructing designs that are close to SOAs of strength 2+. Exactly how close the array D in Theorem 2 is to an SOA of strength 2+ can be measured by π, the proportion of the two-column subarrays (d_j, d_u) that achieve stratifications on 2×4 grids out of all the $m(m-1)$ arrays of two columns. Corollary 1 below is immediate.

Corollary 1. *We have that $\pi = 1 - m_2/(m(m-1))$.*

We see that $\pi \geq 1 - 1/(m-1)$ as $m_2 \leq m$, and π is often much higher because Theorem 2 selects A such that $\bar{A}$ maximizes $\nu = n - 1 - m_2$.

To apply Theorem 2, we need to find array A such that $\bar{A}$ has the most degrees of freedom. Let $C = \bar{A}$. We then want to find design C that has the largest ν. This is no easy task in general, but a very useful result can be obtained and is given in the next theorem.

As before, we use W_3 and W_4 to denote the numbers of words of length 3 and 4, respectively.

Theorem 3. *Design C maximizes its ν, the number of degrees of freedom, if it minimizes $W_3 + W_4$ and the minimum value of $W_3 + W_4$ is 0, 1, and 2.*

Proof. If a design C has $W_3 + W_4 = 0$, then it either contains all independent columns or has resolution V or higher. In both cases, all main effects and two-factor interactions are clear and ν is thus maximized at $f + f(f-1)/2$. A defining word of length 3 causes three two-factor interactions aliased with three main effects, thus a loss of three degrees of freedom. Similarly, a defining word of length 4 also causes a loss of three degrees of freedom. If the minimum value of $W_3 + W_4$ is 1, then design C has a minimum loss of three degrees of freedom. When the minimum value of $W_3 + W_4$ is 2, all three possible cases: (a) $W_3 = 2$ and $W_4 = 0$, (b) $W_3 = 1$ and $W_4 = 1$ and (c) $W_3 = 0$ and $W_4 = 2$ result in a minimum loss of six degrees of freedom, since overlapping is impossible in the loss of degrees of freedom caused by the two defining words. □

Remark 1. No general conclusion can be drawn if the minimum value of $W_3 + W_4$ is equal to or greater than 3. Not only does ν depend on the value of $W_3 + W_4$ but also on the structure of these defining words. We give one example to illustrate. Consider two cases (i) $F_1F_2F_3 = F_4F_5F_6 = F_7F_8F_9F_{10} = I$ and (ii) $F_1F_2F_3 = F_3F_4F_5 = F_1F_2F_4F_5 = I$, both of which have $W_3 + W_4 = 3$. Here $F_1, \ldots, F_{10}$ denote the factors and I is the column of all plus ones. Nine degrees of freedom are lost in case (i) whereas seven degrees of freedom are lost in case (ii).

By Theorem 3, designs C of 32 runs for $f \leq 7$ factors and of 64 runs for $f \leq 10$ factors that have the largest ν can be immediately found from the complete catalogs (Chen, Sun and Wu 1993 and Xu 2009), and are tabulated in Table 4.2. Designs of 128 runs for $f \leq 13$ factors with the largest ν in Table 4.2 are also found using Theorem 3 from the catalog of

k	f	Additional Columns	(W_3, W_4)	ν	π
5	6	31	0 0	21	0.9833
	7	31 3	1 0	25	0.9891
	8	15 19 17	2 1	28	0.9941
6	7	63	0 0	28	0.9886
	8	31 39	0 0	36	0.9909
	9	31 39 9	1 0	42	0.9927
	10	31 39 41 18	1 1	49	0.9949
	11	31 39 41 51 29	1 2	57	0.9977
	12	31 39 41 51 42 20	2 4	62	0.9996
7	8	127	0 0	36	0.9935
	9	31 103	0 0	45	0.9941
	10	31 103 43	0 0	55	0.9947
	11	31 103 43 85	0 0	66	0.9954
	12	31 103 43 85 121	0 1	75	0.9960
	13	31 103 43 85 44 86	0 2	85	0.9967

Table 4.2: Designs with the maximum numbers of degrees of freedom

resolution IV 128-run designs (Xu 2009). The resolution IV design with 13 factors in Table 4.2 maximizes ν, because there exists no design of 13 factors with $W_3 = 1$ and $W_4 = 0$ (Draper and Lin 1990). The results for the other cases in Table 4.2 are obtained by a complete search.

Designs with $f \leq k$ factors are omitted from this table, because, according to Theorem 3, they are trivially given by any set of f independent factors. Theorem 3 and Table 4.2 can be used to produce SOAs of strength 2 of 16 runs for $11 \leq m \leq 15$ factors, of 32 runs for $23 \leq m \leq 31$ factors, of 64 runs for $51 \leq m \leq 63$ factors, and of 128 runs for $114 \leq m \leq 127$ factors.

In addition to the design columns and $W = (W_3, W_4)$, Table 4.2 also provides information on ν and π, where π is given in Corollary 1. We see that the π values are all very close to one. In almost all cases, there is more than one nonisomorphic design that maximizes ν, and the one given in Table 4.2 also has the largest W_3 value. This is in the same spirit as the discussion in Section 4.3.

Example 2. We construct an optimal SOA$(32, 23, 4, 2)$ in this example. First, we have that $k = 5$ and $f = 8$. From Table 4.2, we obtain $\bar{A} = C = (1, 2, 4, 8, 16, 15, 19, 17)$, collecting additional columns 15, 19 and 17 besides the independent columns 1, 2, 4, 8, 16. This gives $A = (3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31)$. For this A, design B is then constructed using the method given right before Proposition 2. One version is $B = (2, 4, 4, 8, 8, 8, 15, 8, 15, 15, 19, 16, 17, 1, 19, 16, 17, 1, 19, 19, 1, 17, 16)$. We then obtain design D, an optimal SOA$(32, 23, 4, 2)$, via $D = A + B/2 + 3/2$ as in (4.1). To save space, we only display design D below:

$$D = \begin{bmatrix}
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
1 & 1 & 3 & 1 & 1 & 3 & 0 & 3 & 0 & 2 & 2 & 3 & 0 & 2 & 0 & 3 & 0 & 2 & 0 & 2 & 0 & 2 & 1 \\
0 & 3 & 1 & 1 & 3 & 1 & 0 & 3 & 2 & 0 & 0 & 3 & 3 & 1 & 0 & 3 & 3 & 1 & 0 & 2 & 3 & 1 & 1 \\
2 & 1 & 1 & 3 & 1 & 1 & 3 & 3 & 1 & 1 & 1 & 3 & 0 & 0 & 3 & 3 & 0 & 0 & 3 & 3 & 0 & 0 & 3 \\
3 & 0 & 0 & 1 & 3 & 3 & 2 & 1 & 0 & 0 & 3 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 \\
1 & 2 & 0 & 3 & 1 & 3 & 1 & 1 & 3 & 1 & 2 & 1 & 2 & 0 & 2 & 3 & 0 & 2 & 0 & 0 & 2 & 0 & 3 \\
0 & 0 & 2 & 3 & 3 & 1 & 1 & 1 & 1 & 3 & 0 & 1 & 1 & 3 & 2 & 3 & 3 & 1 & 0 & 0 & 1 & 3 & 3 \\
2 & 2 & 2 & 1 & 1 & 1 & 2 & 1 & 2 & 2 & 1 & 1 & 2 & 2 & 1 & 3 & 0 & 0 & 3 & 1 & 2 & 2 & 1 \\
3 & 3 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 3 & 0 & 2 & 0 & 3 & 0 & 3 & 1 & 2 & 3 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 2 & 0 & 3 \\
0 & 3 & 1 & 0 & 0 & 2 & 3 & 0 & 1 & 3 & 0 & 3 & 3 & 1 & 0 & 1 & 1 & 3 & 2 & 0 & 1 & 3 & 3 \\
2 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 1 & 3 & 0 & 0 & 3 & 1 & 2 & 2 & 1 & 1 & 2 & 2 & 1 \\
3 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 \\
1 & 2 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 2 & 1 & 2 & 0 & 2 & 1 & 2 & 0 & 2 & 2 & 0 & 2 & 1 \\
0 & 0 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 1 & 1 & 3 & 2 & 1 & 1 & 3 & 2 & 2 & 3 & 1 & 1 \\
2 & 2 & 2 & 0 & 2 & 2 & 1 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 & 1 & 3 & 0 & 0 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 3 & 1 & 1 & 3 & 0 & 3 & 0 & 2 & 1 & 0 & 3 & 0 & 3 & 0 & 3 & 0 & 3 & 1 & 2 & 1 & 2 \\
0 & 3 & 1 & 1 & 3 & 1 & 0 & 3 & 2 & 0 & 3 & 0 & 0 & 3 & 3 & 0 & 0 & 3 & 3 & 1 & 1 & 2 & 2 \\
2 & 1 & 1 & 3 & 1 & 1 & 3 & 3 & 1 & 1 & 2 & 0 & 3 & 2 & 0 & 0 & 3 & 2 & 0 & 0 & 2 & 3 & 0 \\
3 & 0 & 0 & 1 & 3 & 3 & 2 & 1 & 0 & 0 & 0 & 2 & 2 & 3 & 2 & 0 & 0 & 1 & 0 & 2 & 3 & 2 & 2 \\
1 & 2 & 0 & 3 & 1 & 3 & 1 & 1 & 3 & 1 & 1 & 2 & 1 & 2 & 1 & 0 & 3 & 0 & 3 & 3 & 0 & 3 & 0 \\
0 & 0 & 2 & 3 & 3 & 1 & 1 & 1 & 1 & 3 & 3 & 2 & 2 & 1 & 1 & 0 & 0 & 3 & 3 & 3 & 3 & 0 & 0 \\
2 & 2 & 2 & 1 & 1 & 1 & 2 & 1 & 2 & 2 & 2 & 2 & 1 & 0 & 2 & 0 & 3 & 2 & 0 & 2 & 0 & 1 & 2 \\
3 & 3 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 2 \\
1 & 1 & 3 & 0 & 2 & 0 & 3 & 0 & 3 & 1 & 1 & 0 & 3 & 0 & 3 & 2 & 1 & 2 & 1 & 3 & 0 & 3 & 0 \\
0 & 3 & 1 & 0 & 0 & 2 & 3 & 0 & 1 & 3 & 3 & 0 & 0 & 3 & 3 & 2 & 2 & 1 & 1 & 3 & 3 & 0 & 0 \\
2 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 2 & 0 & 3 & 2 & 0 & 2 & 1 & 0 & 2 & 2 & 0 & 1 & 2 \\
3 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 3 & 0 & 2 & 2 & 3 & 2 & 2 & 2 & 3 & 2 & 0 & 1 & 0 & 0 \\
1 & 2 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 1 & 2 & 1 & 2 \\
0 & 0 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 0 & 3 & 2 & 2 & 1 & 1 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 2 \\
2 & 2 & 2 & 0 & 2 & 2 & 1 & 2 & 1 & 1 & 2 & 2 & 1 & 0 & 2 & 2 & 1 & 0 & 2 & 0 & 2 & 3 & 0
\end{bmatrix}.$$

Our construction of array D via choices of arrays A and B maximizes M, the number of subarrays (d_j, d_u) of two columns that achieve stratifications on 2×4 grids out of $m(m-1)$ subarrays of two columns. Just how many two-dimensions (d_j, d_u) with $j < u$ out of $m(m-1)/2$ that achieve stratifications on both 2×4 and 4×2 grids is a question we would also like to have an answer for. Let M^* be the number of such two-dimensions. Although M^* cannot be determined exactly under our construction of A and B, good bounds can be obtained and are presented in the next result.

Theorem 4. *We have that*

$$m(m-1)/2 - m_2 \leq M^* \leq m(m-1)/2 - m_2 + [m_2/2].$$

Proof. For array (d_j, d_u) to achieve stratifications on 2×4 and 4×2 grids, both arrays (a_j, a_u, b_u) and (a_j, b_j, a_u) need to have strength 3. For fixed j, consider $m - 1$ arrays (a_j, b_j, a_u) for all $u \neq j$. Clearly, (a_j, b_j, a_u) for any $u \neq j$ has strength 3 whenever a_j is from A_1. However, when a_j belongs to A_2, exactly one of $m-1$ arrays (a_j, b_j, a_u) cannot have strength 3 because it must be true that $a_j b_j = a_v$ for some v. Therefore, out of $m(m-1)$ arrays (a_j, b_j, a_u), all but m_2 of them have strength 3. The above arguments plus a moment of thought leads to the conclusion that at most m_2 out of $m(m-1)/2$ arrays (d_j, d_u) where $j < u$ do not achieve stratifications on both 2×4 and 4×2 grids. This establishes the lower bound on M^*.

Let us take a closer look at array (a_j, b_j, a_u) for the case a_j is from A_2. We know that $a_j b_j = a_v$ for some a_v. If a_v belongs to A_2, and simultaneously b_v is chosen such that $a_v b_v = a_j$, ie, we choose $b_v = b_j$, then both (a_j, b_j, a_v) and (a_v, b_v, a_j) have strength 2. This means that array (d_j, d_v) does not achieve a stratification on 4×2 nor on 2×4 grids. In this case, a_j and a_v cause the same array (d_j, d_v) not to have stratifications on both 4×2 and 2×4 grids. There are at most $[m_2/2]$ pairs (a_j, a_v) of columns in A_2 that are possible to have the property that $a_j b_j = a_v$ and $a_v b_v = a_j$. This establishes the upper bound in Theorem 4. □

The bounds in Theorem 4 are quite tight, as we have found examples showing that both the lower and upper bounds can be attained.

4.5 Discussion

This chapter presents some solutions to the problem of selecting SOAs of strength 2 and 2+ that enjoy better space-filling properties. We show that optimal SOAs of strength 2+ can be constructed using SOS designs with the maximum numbers of words of length 3. The first construction of SOS designs in He, Cheng and Tang (2018) is shown to be optimal. Our computer search results suggest that their second construction might also be optimal. This deserves further investigation. Construction of the best SOAs of strength 2 requires designs that have the most degrees of freedom. We have established one theoretical result but the problem is far from being completely solved especially for designs of large run sizes. More research is needed.

The SOAs considered in this chapter can be straightforwardly used to construct Latin hypercubes by a level expansion as done in Tang (1993). The Latin hypercubes so obtained inherit all the space-filling properties of the underlying SOAs. The fact that many Latin hypercubes can be constructed from one given SOA provides us with opportunities of finding designs with other desirable properties. Criteria based on orthogonality, distance and discrepancy can all be used for this purpose. Some related recent work includes Georgiou and Efthimiou (2014), Liu and Liu (2015), Lin and Kang (2016), and Xiao and Xu (2018).

Bibliography

[1] B. Abraham, H. Chipman, and K. Vijayan. Some risks in the construction and analysis of supersaturated designs. *Technometrics*, 41:135–141, 1999.

[2] R. A. Bates, R. J. Buck, E. Riccomagno, and H. P. Wynn. Experimental design and observation for large systems. *Journal of the Royal Statistical Society, Series B (Methodological)*, 58:77–94, 1996.

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

[4] R. M. Block and R. W. Mee. Second order saturated resolution IV designs. *Journal of Statistical Theory and Applications*, 2:96–112, 2003.

[5] K. H. Booth and D. R. Cox. Some systematic supersaturated designs. *Technometrics*, 4:489–495, 1962.

[6] G. E. Box and R. D. Mayer. An analysis for unreplicated fractional factorials. *Technometrics*, 28:11–18, 1986.

[7] D. A. Bulutoglu and C. S. Cheng. Construction of $E(s^2)$-optimal supersaturated designs. *The Annals of Statistics*, 32:1662–1678, 2004.

[8] N. A. Butler. Some theory for constructing minimum aberration fractional factorial designs. *Biometrika*, 90:233–238, 2003.

[9] N. A. Butler, R. Mead, K. M. Eskridge, and S. G. Gilmour. A general method of constructing $E(s^2)$-optimal supersaturated designs. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63:621–632, 2001.

[10] H. Chen and A. S. Hedayat. 2^{n-l} designs with weak minimum aberration. *The Annals of Statistics*, 24:2536–2548, 1996.

[11] H. H. Chen and C. S. Cheng. Doubling and projection: a method of constructing two-level designs of resolution IV. *The Annals of Statistics*, 34:546–558, 2006.

[12] J. Chen, D. X. Sun, and C. F. J. Wu. A catalogue of two-level and three-level fractional factorial designs with small runs. *International Statistical Review*, 61:131–145, 1993.

[13] C. S. Cheng. $E(s^2)$-optimal supersaturated designs. *Statistica Sinica*, 7:929–939, 1997.

[14] C. S. Cheng. *Theory of factorial design: Single- and multi-stratum experiments*. CRC Press, 2014.

[15] C. S. Cheng, A. Das, R. Singh, and P. W. Tsai. $E(s^2)$- and $UE(s^2)$-optimal supersaturated designs. *Journal of Statistical Planning and Inference*, 196:105–114, 2018.

[16] C. S. Cheng, Y. He, and B. Tang. Minimal second order saturated designs and their applications to computer experiments. *Submitted*, 2018.

[17] H. Chipman, M. Hamada, and C. F. J. Wu. A Bayesian variable-selection approach for analyzing designed experiments with complex aliasing. *Technometrics*, 39:372–381, 1997.

[18] A. Das, A. Dey, L. Y. Chan, and K. Chatterjee. On $E(s^2)$-optimal supersaturated designs. *Journal of Statistical Planning and Inference*, 138:3749–3757, 2008.

[19] A. A. Davydov, S. Marcugini, and F. Pambianco. Minimal 1-saturating sets and complete caps in binary projective spaces. *Journal of Combinatorial Theory, Series A*, 113:647–663, 2006.

[20] A. Dey and R. Mukerjee. *Fractional factorial plans*. New York: Wiley, 1999.

[21] N. R. Draper and D. K. Lin. Connections between two-level designs of resolutions Ill* and V. *Technometrics*, 32:283–288, 1990.

[22] K. T. Fang, R. Z. Li, and A. Sudjianto. *Design and modeling for computer experiments*. New York: Chapman & Hall, 2006.

[23] K. T. Fang, D. K. Lin, P. Winker, and Y. Zhang. Uniform design: theory and application. *Technometrics*, 42:237–248, 2000.

[24] S. D. Georgiou and I. Efthimiou. Some classes of orthogonal Latin hypercube designs. *Statistica Sinica*, 24:101–120, 2014.

[25] B. Haaland and P. Z. Qian. An approach to constructing nested space-filling designs for multi-fidelity computer experiments. *Statistica Sinica*, 20:1063–1075, 2010.

[26] Y. He, C. S. Cheng, and B. Tang. Strong orthogonal arrays of strength two plus. *The Annals of Statistics*, 46:457–468, 2018.

[27] Y. He and B. Tang. Strong orthogonal arrays and associated Latin hypercubes for computer experiments. *Biometrika*, 100:254–260, 2013.

[28] Y. He and B. Tang. A characterization of strong orthogonal arrays of strength three. *The Annals of Statistics*, 42:1347–1360, 2014.

[29] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. *Orthogonal arrays: theory and applications*. Springer, New York, 1999.

[30] M. Johnson, L. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26:131–148, 1990.

[31] B. Jones and D. Majumdar. Optimal supersaturated designs. *Journal of the American Statistical Association*, 109:1592–1600, 2014.

[32] H. Kharaghani and B. Tayfeh-Rezaie. Hadamard matrices of order 32. *Journal of Combinatorial Designs*, 21:212–221, 2013.

[33] K. M. Lawrence. A combinatorial characterization of (t, m, s)-nets in base b. *Journal of Combinatorial Designs*, 4:275–293, 1996.

[34] R. Li and D. K. Lin. Analysis methods for supersaturated design: some comparisons. *Journal of Data Science*, 1:249–260, 2003.

[35] C. D. Lin and L. Kang. A general construction for space-filling Latin hypercubes. *Statistica Sinica*, 26:675–690, 2016.

[36] D. K. Lin. A new class of supersaturated designs. *Technometrics*, 35:28–31, 1993.

[37] H. Liu and M. Q. Liu. Column-orthogonal strong orthogonal arrays and sliced strong orthogonal arrays. *Statistica Sinica*, 25:1713–1734, 2015.

[38] M. Liu and R. Zhang. Construction of $E(s^2)$ optimal supersaturated designs using cyclic BIBDs. *Journal of Statistical Planning and Inference*, 91:139–150, 2000.

[39] C. J. Marley and D. C. Woods. A comparison of design and model selection methods for supersaturated experiments. *Computational Statistics and Data Analysis*, 54:3158–3167, 2010.

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

[41] A. Miller and R. R. Sitter. Using the folded-over 12-run Plackett-Burman design to consider interactions. *Technometrics*, 43:44–55, 2001.

[42] A. Miller and B. Tang. Minimal dependent sets for evaluating supersaturated designs. *Statistica Sinica*, 22:1273–1285, 2012.

[43] H. Moon, A. Dean, and T. Santner. Algorithms for generating maximin Latin hypercube and orthogonal designs. *Journal of Statistical Theory and Practice*, 5:81–98, 2011.

[44] M. D. Morris and T. J. Mitchell. Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43:381–402, 1995.

[45] G. L. Mullen and W. Ch. Schmid. An equivalence between (t, m, s)-nets and strongly orthogonal hypercubes. *Journal of Combinatorial Theory, Series A*, 76:164–174, 1996.

[46] N. K. Nguyen. An algorithmic approach to constructing supersaturated designs. *Technometrics*, 38:69–73, 1996.

[47] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods.* SIAM CBMS-NSF Regional Conference Series in Applied Mathematics. Philadelphia, 1992.

[48] A. B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452, 1992.

[49] A. B. Owen. Scrambled net variance for integrals of smooth functions. *The Annals of Statistics*, 25:1541–1562, 1997.

[50] F. K. Phoa, Y. H. Pan, and H. Xu. Analysis of supersaturated designs via the Dantzig selector. *Journal of Statistical Planning and Inference*, 139:2362–2372, 2009.

[51] T. J. Santner, B. J. Williams, and W. I. Notz. *The design and analysis of computer experiments.* New York: Springer, 2003.

[52] F. E. Satterthwaite. Random balance experimentation. *Technometrics*, 1:111–137, 1959.

[53] C. Shi and B. Tang. Designs from good Hadamard matrices. *Bernoulli*, 24:661–671, 2018.

[54] C. Shi and B. Tang. Supersaturated designs robust to two-factor interactions. *Journal of Statistical Planning and Inference*, 200:119–128, 2019a.

[55] C. Shi and B. Tang. Construction results for strong orthogonal arrays of strength three. *Bernoulli*, Accepted, 2019b.

[56] C. Shi and B. Tang. Design selection for strong orthogonal arrays. *Canadian Journal of Statistics*, 47:302–314, 2019c.

[57] B. Tang. Orthogonal array-based Latin hypercubes. *Journal of the American statistical association*, 88:1392–1397, 1993.

[58] B. Tang. A theorem for selecting OA-based Latin hypercubes using a distance criterion. *Communications in Statistics-Theory and Methods*, 23:2047–2058, 1994.

[59] B. Tang and C. F. J. Wu. Characterization of minimum aberration 2^{n-k} designs in terms of their complementary designs. *The Annals of Statistics*, 24:2549–2559, 1996.

[60] B. Tang and C. F. J. Wu. A method for constructing supersaturated designs and its $E(s^2)$-optimality. *Canadian Journal of Statistics*, 25:191–201, 1997.

[61] E. R. van Dam, B. Husslage, D. den Hertog, and H. Melissen. Maximin Latin hypercube designs in two dimensions. *Operations Research*, 55:158–169, 2007.

[62] L. Wang, Q. Xiao, and H. Xu. Optimal maximin l_1-distance Latin hypercube designs based on good lattice point designs. *The Annals of Statistics*, 46:3741–3766, 2018.

[63] P. H. Westfall, S. S. Young, and D. K. Lin. Forward selection error control in the analysis of supersaturated designs. *Statistica Sinica*, 8:101–117, 1998.

[64] C. F. J. Wu. Construction of 2^m4^n designs via a grouping scheme. *The Annals of Statistics*, 17:1880–1885, 1989.

[65] C. F. J. Wu. Construction of supersaturated designs through partially aliased interactions. *Biometrika*, 80:661–669, 1993.

[66] Q. Xiao and H. Xu. Construction of maximin distance designs via level permutation and expansion. *Statistica Sinica*, 28:1395–1414, 2018.

[67] H. Xu. Algorithmic construction of efficient fractional factorial designs with large run sizes. *Technometrics*, 51:262–277, 2009.

[68] K. Q. Ye. Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, 93:1430–1439, 1998.

[69] Q. Z. Zhang, R. C. Zhang, and M. Q. Liu. A method for screening active effects in supersaturated designs. *Journal of Statistical Planning and Inference*, 137:235–248, 2007.