Optimal Compound Orthogonal Arrays and Single Arrays for Robust Parameter Design Experiments

Yu Zhu

Department of Statistics
Purdue University
West Lafayette, IN

Peng Zeng

Department of Mathematics and Statistics
University of Auburn
Auburn, AL

Abstract

Compound orthogonal arrays (COAs) and single arrays are alternatives to the inner-outer arrays advocated by Taguchi for robust parameter design experiments. A criterion based on the wordtype patterns and strengths of COAs is proposed to select optimal COAs. Single arrays are classified into prodigal single arrays (PSAs) and economical single arrays (ESAs) according to their relative estimation capacities, and various optimality criteria again based on the wordtype patterns are proposed for selecting optimal single arrays. Useful optimal COAs, PSAs and ESAs are constructed and tabulated as convenient references for experimenters in practice.

KEY WORDS: Robust parameter design, compound orthogonal array, single array

1 INTRODUCTION

Robust parameter design (or briefly parameter design) is an engineering strategy, originally proposed by Taguchi (1986), for quality improvement in industrial systems. Factors that affect a system can be classified into two types, *control factors* and *noise factors*. Control factors refer to the variables whose levels are adjustable, while noise factors refer to the variables whose levels are hard or impossible to control in a system's normal operation state. In a parameter design experiment, both control and noise factors are varied systematically. The key idea of parameter design is to explore the effects of control factors, noise factors and their interactions, and choose control factor settings to simultaneously bring the system's mean response on target and reduce its performance variation caused by noise factors. For comprehensive reviews on parameter design, see Nair (1992) and Steinberg (1996).

Taguchi originally proposed to use inner-outer array, or *cross array,* as the experimental plan for parameter design. A cross array is the cross product of an orthogonal array for control factors (or briefly control array) and an orthogonal array for noise factors (or briefly noise array).

After data are generated from an experiment using cross array, response mean and variance at each setting of control factors are calculated, and the *location-dispersion modeling* approach is usually used to identify location and dispersion effects for parameter design optimization (Vining and Myers, 1990). Due to the concerns over the run size and flexibility of cross array, Lucas (1989), Welch et al. (1990) and Shoemake et al. (1991) proposed to use combined array, or *single array*, as an alternative to cross array. A single array is an ordinary orthogonal array that accommodate both control and noise factors and does not necessarily possess the "crossing" structure as cross array. In analyzing the data generated from an experiment using single array, the *response modeling* approach is usually used to directly model response as a function of control factors, noise factors and their interactions, then control-by-noise plots and the transmitted variance model are employed to identify location and dispersion effects. The idea of response modeling was first hinted in Easterling (1985).

The advantages and disadvantages of cross array and single array were discussed in Nair (1992) and Steinberg (1996). Recently there is a fair amount of effort devoted to the selection of optimal experimental plans for parameter design. In a series of papers, Rosenbaum (1994, 1996, 1998) extended cross array to compound orthogonal array (COA) and provided justifications for using COAs in parameter design experiments. COA relaxes the rigid crossing structure required by cross array so that different arrays can be employed for noise factors at different settings of control factors. A detailed description of COA will be given in Section 3.1. Hedayat and Stufken (1999) studied the basic properties of COAs and constructed the tables of COAs with both the numbers of control and noise factors less than six. However, the optimal selection of COAs have not been addressed in the literature.

Although combinatorially single arrays are just ordinary orthogonal arrays, their optimal selection is not straightforward due to the presence of two different types of factors. Bingham and Sitter (2003) proposed a minimum-aberration type of criterion based on modified wordlengths, and constructed the tables of small arrays applicable in split-plot parameter design experiments. The tables show that the criterion is not sensitive in discriminating single arrays when their sizes are relatively large. Wu and Zhu (2003) developed a general framework for selecting optimal single arrays, again using a minimum-aberration type of criterion. The framework becomes too complicated when high order effects are taken into consideration. Hence, a simple and direct approach is still needed for the selection of optimal single arrays

The current paper is intended to address the optimal selection of COAs and single arrays and is organized as follows. Section 2 is a brief review of fractional factorial design with two groups of factors. Section 3 introduces COAs and proposes a minimum aberration criterion for selecting optimal COAs. In Section 4, single arrays are classified into economical single arrays and prodigal single arrays, several criteria for selecting optimal single arrays are proposed, and optimal economical single arrays and prodigal single arrays are tabulated and discussed. Concluding remarks are given in Section 5. In this paper, we focus on COAs and single arrays that are regular two-level fractional factorial designs.

2 $2^{(l_1+l_2)-m}$ DESIGNS WITH TWO GROUPS OF FACTORS

Regular two-level fractional factorial designs, i.e., 2^{l-m} designs, are generated by defining relations (or defining words) among experimental factors. The collection of all the possible defining words of a design d is referred to as its defining contrast subgroup denoted by $\mathcal{G}$. Let W_i be the number of defining words of length i in $\mathcal{G}$ for $1 \leq i \leq l$ and $W = (W_0, W_1, W_2, \ldots, W_l)$. Then W is called the wordlength pattern of d. The *resolution* of d is the smallest positive integer i such that $W_i > 0$. If d has resolution R, it is well-known that the strength of d is $R-1$; see Rao (1947) for the definition of strength for general orthogonal arrays. Suppose d_1 and d_2 are two designs, d_1 is said to have less aberration than d_2 if $W_{i_0}(d_1) < W_{i_0}(d_2)$ where i_0 is the smallest integer i such that $W_i(d_1) \neq W_i(d_2)$. If there does not exist a design with less aberration than d_1, then d_1 is said to have *minimum aberration* (MA) (Fries and Hunter, 1980). A main effect or a two-factor interaction (2fi) is said to be *clear* or clearly estimable if it is not aliased with any other main effects or 2fi's, and is *eligible* if it is not clear but only aliased with some other 2fi's (Wu and Chen, 1992).

In 2^{l-m} design, experimental factors are treated symmetrically. This, however, is not proper when multiple groups, or types, of factors are involved, for example, in parameter design where factors are divided into control factors and noise factors. In this paper, we only consider the case of two groups of factors , which are denoted by Group I and Group II. Let l_1 and l_2 be the number of factors in Group I and Group II respectively. A $2_{\tau}^{(l_1+l_2)-p}$ design d is a two-level fractional factorial design with l_1 columns assigned to Group I factors and the remaining l_2 columns to Group II factors, where τ represents the assignment of the columns and is usually suppressed. d is also determined by its defining contrast subgroup $\mathcal{G}$. However, the wordlength pattern W is not a proper aliasing characterization for $\mathcal{G}$, because defining words with the same length can consist of different numbers of group I and group II factors. Zhu (2003) proposed using the *wordtype pattern* matrix, $(A_{i.j})_{0 \leq i \leq l_1, 0 \leq j \leq l_2}$, to characterize the aliasing pattern of $\mathcal{G}$, where $A_{i.j}$ is the number of defining words in $\mathcal{G}$ involving i Group I factors and j Group II factors. In a parameter design experiment involving l_c control factors and l_n noise factors, we assume Group I consists of the control factors and Group II the noise factors. If we only consider two-level regular fractional factorial designs as possible experiment plans, then cross arrays, COAs and single arrays are in fact $2^{(l_c+l_n)-m}$ designs with or without additional constraints. As a convention in the paper, we use capital letters such as A, B, etc. to represent control factors, and little letters such as a, b, etc. to represent noise factors.

3 OPTIMAL COMPOUND ORTHOGONAL ARRAYS

3.1 COAs and Maximum Strengths

Let $OA(N, l, 2, t)$ denote a two-level orthogonal array with N rows, l columns and strength t (Rao, 1947). A COA with parameters N_c, N_n, l_c, l_n, t_c, t_n and t_a is an $N_cN_n \times (l_c + l_n)$ orthogonal array with the following properties: (P1) The first l_c columns are assigned to control factors and consists of N_n replications of an $OA(N_c, l_c, 2, t_c)$; (P2) The remaining l_n columns are assigned to noise factors, and for each fixed setting of the l_c control factors, denoted by c_i for $1 \le i \le N_c$, the corresponding settings of the l_n noise factors form an $OA_{c_i}(N_n, l_n, 2, t_n)$, where the subscript c_i indicates that the array may vary from one setting to another; (P3) The entire array has strength t_a. Let $T = (t_c, t_n, t_a)$. We call T the *strength vector* of a COA. It can be verified that $\min(t_c, t_n) \le t_a \le (t_c + t_n)$ (Hedayat and Stufken, 1999). When all the $OA_{c_i}(N_n, l_n, 2, t_n)$'s are identical for $1 \le i \le N_c$, the COA is a cross array. A COA is said to be *regular* if it is a regular $2^{(l_c+l_n)-m}$ design that satisfies (P1)-(P3). As mentioned earlier, we only consider regular COAs in the current paper.

Example 1. Suppose a 32-run experiment involves four control factors (A, B, C, D) and three noise factors (a, b, c). Consider the following two $2^{(4+3)-2}$ designs: $d_1 : I = ABCD = ABabc = CDabc$ and $d_2 : I = ABCD = abc = ABCDabc$. Both d_1 and d_2 have 32 rows and seven columns with the first four assigned to the control factors and the remaining three to the noise factors. In d_1, the control factor columns consist of four copies of a 16-run 2^{4-1} resolution IV design defined by $I = ABCD$. At each fixed setting of A, B, C and D, the corresponding settings of the noise factors form a 4-run resolution III design defined by $AB = CD = abc$. For example, at the setting $(1, -1, 1, -1)$ of the control factors, the corresponding settings of a, b and c are generated by $-I = abc$, which are $(-1, -1, -1), (-1, 1, 1), (1, -1, 1), (1, 1, -1)$. So, d_1 is a COA with $N_c = 8, N_n = 4, l_c = 4, l_n = 3, t_c = 3, t_n = 2$ and $t_a = 3$, and $T(d_1) = (3, 2, 3)$. Similarly, it can be verified that d_2 is a COA with $T(d_2) = (3, 2, 2)$. Notice that, in d_2, at different settings of the control factors, the noise arrays are identical, which is generated by $I = abc$. Hence, d_2 is a cross array. It is clear that d_1 is not a cross array and it has higher overall strength than d_2. The wordtype patterns of d_1 are $A_{0.0} = 1$, $A_{2.3} = 2$, $A_{4.0} = 1$, and $A_{i.j} = 0$ otherwise; the wordtype patterns of d_2 are $A_{0.0} = 1$, $A_{0.3} = A_{4.0} = A_{4.3} = 1$, and $A_{i.j} = 0$ otherwise.

Let d be a $2^{(l_c+l_n)-m}$ COA with wordtype pattern matrix $(A_{i.j})$. Considering the relationship between strength and resolution, we can verify that

$$t_c = \min(l_c, \min\{i - 1 : A_{i.0} \neq 0 \text{ and } i \geq 1\}),$$

$$t_n = \min(l_n, \min\{j - 1 : A_{i.j} \neq 0 \text{ and } j \geq 1\}),$$

and

$$t_a = \min\{i + j - 1 : A_{i.j} \neq 0 \text{ and } i + j \geq 1\}.$$

Hence $(A_{i.j})$ determines $T(d)$. In the definition of COA above, there is no specific requirement for t_n, so t_n can take on any nonnegative integer values. Rosenbaum (1994, 1996) proposed COAs as an extension of cross arrays and pointed out that COAs with $t_n < 2$ are not able to identify dispersion effects and are unlikely to be useful; see Theorem 1 and Section 3.1 of Rosenbaum (1996). In other words, COAs with $t_n \leq 1$ are not a reasonable extension of cross arrays. When investigating the maximum strengths of COAs from a theoretical perspective, Hedayat and Stufken (1999) included COAs with $t_n \leq 1$. In this paper, we are more interested in COAs that are useful in parameter design, therefore, in addition to the three properties given above, we further require COAs to satisfy the following condition,

$$\text{(P4) } t_c \geq \min(l_c, 2) \text{ and } t_n \geq \min(l_n, 2).$$

For regular $2^{(l_c+l_n)-m}$ COAs with resolution III or higher, that is, $t_a \geq 2$, the condition $t_c \geq \min(l_c, 2)$ is automatically satisfied. When $l_n \geq 2$, the condition $t_n \geq \min(l_n, 2)$ requires that the noise arrays of a COA have strength at least 2. In terms of wordtype patterns, $t_n \geq \min(l_n, 2)$ is equivalent to $A_{i.1} = A_{i.2} = 0$ for $1 \leq i \leq l_c$. Clearly, cross arrays are COAs. Both d_1 and d_2 in Example 1 satisfy (P4), so they are still COAs.

For given l_c, l_n and run size 2^k, COAs do not always exist. Define

$$\mathcal{S}(2^k) = \{(i, j) : \lceil \log_2(i + 1) \rceil + \lceil \log_2(j + 1) \rceil \leq k\}, \tag{1}$$

where $\lceil x \rceil$ represents the smallest integer greater than or equal to x.

Proposition 1. The necessary and sufficient condition for the existence of a COA with l_c control factors, l_n noise factors and 2^k runs is $(l_c, l_n) \in \mathcal{S}(2^k)$

The proof of Proposition 1 is given in the Appendix. When $k = 4$, $\mathcal{S}(2^4) = \{(1, j)\}_{1 \le j \le 7} \bigcup \{(i, 1)\}_{1 \le i \le 7} \bigcup \{(2, 2), (2, 3), (3, 2), (3, 3)\}$. Because (3,4) $\notin \mathcal{S}(2^4)$, there does not exist a 16-run COA with three control factors and four noise factors. Note that the necessary and sufficient condition for the existence of cross arrays is exactly the same as that for COAs.

Suppose d is a COA with $T(d) = (t_c(d), t_n(d), t_a(d))$. d is said to have *maximum strengths* if there does not exist another COA $\tilde{d}$ with the same run size, l_c and l_n as d such that $T(\tilde{d}) \ge T(d)$ componentwisely and at least one of the inequalities is strict. The preference of COAs with maximum strengths is justified by Theorem 1 in Rosenbaum (1996) and an observation of Hedayat and Stufken (1999) that has further improved part c of Theorem 1 in Rosenbaum (1996); see the beginning of Section 2 in Hedayat and Stufken (1999). When the response modeling approach is used in analysis, COAs with maximum strengths should also be preferred according to Section 4 in Rosenbaum (1996). In Example 1, $T(d_1) = (3, 2, 3)$ and $T(d_2) = (3, 2, 2)$. Clearly, $T(d_1) \ge T(d_2)$ and $t_a(d_1) > t_a(d_2)$. Hence, d_2, a cross array, does not achieve maximum strengths. In fact, we can show that d_1 attains maximum strengths as a COA with 32-run, four control factors and three noise factors.

3.2 W_c-Aberration and Optimal COAs

The strength vector T of a COA can be considered as an extension of the resolution R of a fractional factorial design. Different COAs may share the same strength vector and need to be further discriminated.

Example 2. Let us consider a 64-run parameter design experiment that involves four control factors (A, B, C, D) and six noise factors (a, b, c, d, e, f). Four possible $2^{(4+6)-4}$ COAs are given below. Due to limited space, only a set of independent defining words is listed for each array.

$$d_3 : ABCD, Dabd, Dace, Dbcf; \quad d_4 : ABCD, abde, ABacd, ACabf;$$
$$d_5 : ABCD, abce, abdf, ACacd; \quad d_6 : ABCD, abd, ace, bcf.$$

d_3 was the design given in Table 1 in Rosenbaum (1996); d_4 is a MA 2^{10-4} design with the distinction between control factors and noise factors neglected; and d_6 is a cross array. The strength vectors of the four COAs are $T(d_3) = T(d_4) = T(d_5) = (3, 2, 3)$ and $T(d_6) = (3, 2, 2)$. It can be shown that d_3, d_4 and d_5 all have achieved the maximum strengths for COAs with 64 rows, four control factors and six noise factors. However, they are still quite different from each other as will be shown below, so further discrimination between them is necessary in order to select the best plan for the experiment.

The defining contrast subgroups and wordtype pattern matrices of the above arrays can be derived from independent defining words. It can be verified that the wordtype pattern matrices are different from each other. In the following, we further compare d_3, d_4 and d_5 in terms of their detailed aliasing patterns. For simplicity, we assume effects of order 3 or higher are negligible. So we only need to consider the defining words with length less than five, which are $\{ABCD, Dabd, Dace, Dbcf, Ddef, bcde, acdf, abef\}$ for d_3, $\{ABCD, abde\}$ for d_4, and $\{ABCD, abce, abdf, cdef\}$ for d_5. Because all the three arrays have resolution IV, their control and noise main effects are clear. Because $ABCD$ appears in d_3, d_4 and d_5, the control-by-control 2fi's in all the three arrays are eligible. Consider the 24 possible control-by-noise 2fi's. In d_3, the 18 control-by-noise 2fi's between $\{A, B, C\}$ and $\{a, b, c, d, e, f\}$ are clear, and the six control-by-noise 2fi's between $\{D\}$ and $\{a, b, c, d, e, f\}$ are aliased with some noise-by-noise 2fi's and thus are only eligible; in d_4, all the 24 control-by-noise 2fi's are clear, in addition, the noise-by-noise 2fi's ac, bc, cd, ce, cf, af, bf, df and ef are also clear, and the other six noise-by-noise 2fi's are eligible; in d_5, all the 24 control-by-noise 2fi's are clear, but all the noise-by-noise 2fi's are only eligible. Heuristically, d_4 is the best among the three arrays according to their aliasing patterns, followed by d_5 and d_3. Hence, d_4 should be selected for the experiment.

Example 2 shows that COAs with maximum strengths may not be unique and they can be further discriminated by their wordtype patterns. Let $\overbrace{C \cdots C}^{i}\overbrace{n \cdots n}^{j}$ represent a defining word consisting of i control factors and j noise factors, which is said to be of type (i, j). Note that $A_{i.j}$ is the number of defining words of type (i, j) in the defining contrast subgroup of a COA. Defining words of length k have $(k+1)$ different types, which are $(k, 0)$, $(k-1, 1)$, $\ldots$, $(1, k-1)$, and $(0, k)$. In the following, we re-arrange $A_{i.j}$ into a sequence that can be used to rank-order different COAs. First, we consider the defining words of length 3: CCC, CCn, Cnn and nnn. In COAs, CCn and Cnn are not present. Because the purposes of robust parameter design are mean response optimization and variance reduction, in terms of aliasing severity, CCC is considered more severe than nnn, which we write as $CCC \lhd nnn$, or, $A_{3.0} \lhd A_{0.3}$. Next, we consider the defining words of length 4: $CCCC$, $CCCn$, $CCnn$, $Cnnn$ and $nnnn$. Similarly, $CCCn$ and $CCnn$ are not present in COAs, and we regard $CCCC$ as being more severe than $nnnn$, that is, $CCCC \lhd nnnn$. It is reasonable that $Cnnn$ should be considered more severe than $nnnn$, however, the comparison between $CCCC$ and $Cnnn$ is not conclusive at first glance. $CCCC$ causes aliasing between control-by-control 2fi's, which may hinder the identification of *second order* location effects for response mean optimization, while $Cnnn$ causes aliasing between control-by-noise 2fi's and noise-by-noise 2fi's, which may hinder the identification of *first order* dispersion effects for variation reduction. Therefore, we regard $Cnnn$ to be more sever than $CCCC$, and $Cnnn \lhd CCCC \lhd nnnn$, or, $A_{1.3} \lhd A_{4.0}, \lhd A_{0.4}$. For the defining words of order 5, we have $CCnnn \lhd Cnnnn \lhd CCCCC \lhd nnnnn$, or, $A_{2.3} \lhd A_{1.4} \lhd A_{5.0} \lhd A_{0.5}$. In general, $A_{i_1.j_1} \lhd A_{i_2.j_2}$ if

$$\begin{aligned}
&\text{(i) } i_1 + j_1 < i_2 + j_2; \text{ or}\\
&\text{(ii) } \mid i_1 - j_1 \mid < \mid i_2 - j_2 \mid \ \text{ and } i_1 + j_1 = i_2 + j_2; \text{ or}\\
&\text{(iii) } i_1 > i_2 \text{ and } |i_1 - j_1| = |i_2 - j_2| \text{ and } i_1 + j_1 = i_2 + j_2.
\end{aligned} \tag{2}$$

Thus, following $\lhd$, all the wordtype patterns $A_{i.j}$ can be arranged into a sequence, denoted by W_c, from the most severe to the least severe, as follows:

$$W_c = (A_{3.0}, A_{0.3}, A_{1.3}, A_{4.0}, A_{0.4}, A_{2.3}, A_{1.4}, A_{5.0}, A_{0.5}, A_{3.3}, A_{2.4}, A_{1.5}, A_{6.0}, A_{0.6}, \cdots). \tag{3}$$

And we refer to W_c as the wordtype pattern sequence for COAs. Note that $A_{i.j}$ with $0 \le i+j \le 2$ are constants for designs with resolution at least III, so they are not included in W_c.

Suppose d_1 and d_2 are two COAs whose wordtype pattern sequences are $W_c(d_1)$ and $W_c(d_2)$ respectively. Let $A_{i_0.j_0}$ be the first component of W_c such that $A_{i_0.j_0}(d_1) \neq A_{i_0.j_0}(d_2)$. If $A_{i_0.j_0}(d_1) < A_{i_0.j_0}(d_2)$, then d_1 is said to have less W_c-aberration than d_2. If there does not exist a COA with less W_c-aberration than d_1, d_1 is said to have minimum W_c-aberration. Furthermore, if d_1 has minimum W_c-aberration and maximum strengths, then d_1 is said to be an optimal COA. The reason to include the requirement of maximum strengths in the optimality criterion for COAs is that it is not necessarily true that $T(d_1) \ge T(d_2)$ componentwisely when d_1 has less W_c-aberration than d_2. Though all the known minimum W_c-aberration COAs achieve maximum strengths, we are not able to prove that it is always true. So we conjecture that minimum W_c-aberration COAs also achieve maximum strengths.

Example 2 (Continued). For d_3, d_4, d_5 and d_6, $W_c(d_3) = (0,0,4,1,3,0,0,0,0,4,0,\ldots)$, $W_c(d_4) = (0,0,0,1,1,8,0,0,0,0,4,\ldots)$, $W_c(d_5) = (0,0,0,1,3,\ 8,0,0,0,0,0,\ldots)$ and $W_c(d_6) =$

$(0, 4, 0, 1, 3, 0, 0, 0, 0, 0, 0, \ldots)$. In the order of increasing W_c-aberration, we have d_4, d_5, d_3 and d_6. Complete computer search concludes that d_4 has minimum W_c-aberration and maximum strengths among all the 64-run COAs with four control factors and six noise factors. Hence, d_4 is the optimal COA and should be recommended for the experiment, and the second best is d_5.

3.3 Tables of Optimal COAs with $l_n \geq 3$

Because optimal COAs of 16-, 32- and 64-run are useful for parameter design experiments, it is valuable to tabulate them as a convenient reference for experimenters in practice. The optimal COAs with 2^k runs and one or two noise factors can be easily constructed as follows. Because $l_n \leq 2$, W_c is reduced to be $(A_{i.0})_{3 \leq i \leq l_c}$ after the zero components are removed. For $(l_c, 1)$ or $(l_c, 2)$ that belongs to $\mathcal{S}(2^k)$, the optimal COA is a cross array, where the control array is a 2^{l_c} full factorial design for $1 \leq l_c \leq k-1$, or a $2^{l_c-(l_c+l_n-k)}$ MA design for $l_c \geq k$, and the noise array is a 2^{l_n} full factorial design.

When $l_n \geq 3$, in general, how to construct optimal COAs with $(l_c, l_n) \in \mathcal{S}(2^k)$ is unknown, so we have conducted extensive computer search with complete isomorphism checking to select optimal COAs. Two $2^{(l_1+l_2)-p}$ designs d and d' are said to be isomorphic, if there exists a relabeling of the factors of d that transforms $\mathcal{G}(d)$ to $\mathcal{G}(d')$, where $\mathcal{G}(d)$ and $\mathcal{G}(d')$ are the defining contrast subgroups of d and d', respectively. Our search procedure consists of three steps. The first step is to construct the list of all nonisomorphic $2^{(l_1+l_2)-p}$ designs, the second step is to sort all the nonisomorphic designs according to the optimality criterion for COAs for any given l_c, l_n and k ($k = 4, 5, 6$), and the third step is to output the optimal COAs. For 16- and 32-run designs, we have utilized the lists of nonisomorphic 2^{l-p} designs generated by Sun et al. (1993), where factors are not separated into two groups. For 64-run designs, we have constructed the list of nonisomorphic $2^{(l_1+l_2)-p}$ designs from scratch for $l_1 + l_2$ up to 16.

The final output of our computer search is reported in Table 1 and Table 2. Table 1 contains all the 16-run and 32-run optimal COAs with $l_n \geq 3$, and Table 2 contains all the 64-run optimal COAs with $l_c + l_n \leq 16$ and $l_n \geq 3$. In both the tables, every row gives an optimal COA, and the columns, from the left to the right, give (l_c, l_n), independent defining words (or generators), strength vector $T = (t_c, t_n, t_a)$ and the number of clear control main effect, noise main effects, control-by-control 2fi's, control-by-noise 2fi's and noise-by-noise 2fi's, respectively. For example, from Table 1, the optimal 32-run COA with $l_c = 2$ and $l_n = 7$, denoted by d_1, is

$$(2,7)^{\star} \quad abce\ abdf\ acdg\ ABbcd \quad (2,2,3) \quad (2,7,1,14,0);$$

the optimal 32-run COA with $l_c = 3$ and $l_n = 6$, denoted by d_2, is

$$(3,6)^{\circ} \quad ABC\ Aabd\ Aace\ Bbcf \quad (2,2,2) \quad (0,\ 6,\ 0,\ 8,\ 1).$$

d_1 is a resolution IV design with its control array and noise arrays of strength 2. In d_1, all the control main effects, the noise main effects, the control-by-control 2fi's and the control-by-noise 2fi's are clear. The 32-run cross array with $l_c = 2$ and $l_n = 7$, denoted by d_1', is given by crossing a 2^2 control array and a 2^{7-4} noise array. Note that the noise array in d_1' is a saturated 8-run fractional factorial design. The resolution of d_1' is III, which is less than that of d_1. In d_1', the control main effects and the control-by-noise 2fi's are also clear, but the noise main effects are only eligible. Therefore, in terms of both strengths and the number of clear effects, d_1 is better than d_1', which is indicated by the $\star$ in $(2,7)^{\star}$. The same interpretation applies to other arrays marked with $\star$ in Tables 1 and 2.

The optimal COA d_2 above is a resolution III design with its control array and noise arrays of strength 2. In d_2, six noise main effects, eight control-by-noise 2fi's and one noise-by-noise 2fi are clear, and control main effects are aliased with control-by-control 2fi's and the other control-by-noise 2fi's and noise-by-noise 2fi's are only eligible. The corresponding 32-run cross array, denoted by d_2', is given by crossing a 4-run 2^{3-1} control array and a 8-run 2^{6-3} design. d_2 and d_2' have the same strength vector. In d_2', all the control-by-noise 2fi's are clear, control main effects are aliased with some control-by-control 2fi's, noise main effects are aliased with some noise-by-noise 2fi's, and the rest 2fi's are eligible. There clearly exists a trade-off between d_2 and d_2', that is, d_2 guarantees the clear estimation of the noise main effects while d_2' ensures the clear estimation of all the control-by-noise 2fi's. Due to the trade-off, the advantage of d_2' over d_2 is not as clear as that of d_1 over d_1' discussed in the proceeding paragraph. We indicate this trade-off by the $\circ$ in $(3,6)^{\circ}$. The same interpretation applies to other arrays marked with $\circ$ in Tables 1 and 2. The trade-off can also be regarded as an indication that the array for the given l_c and l_n is already too tight to simultaneously possess a crossing structure and a good estimation capability for important effects. It may suggest that single array be considered instead; see the good single arrays with $l_c = 3$ and $l_n = 6$ reported in Table 3 later. Some of the designs reported in Tables 1 and 2 were also obtained by Hedayat and Stufken (1999) and Bingham and Sitter (2003). Nonetheless, they are included in the tables for completeness.

4 OPTIMAL SINGLE ARRAYS

From the discussion in Section 3.1, we know that the smallest COA or cross array with l_c control factors and l_n noise factors requires $2^{\lceil \log_2(l_c+1)\rceil + \lceil \log_2(l_n+1)\rceil}$ runs. When both l_c and l_n increase, COAs become too large to be feasible in practice. For example, there does not exist

Table 1: 16- and 32-Run Optimal Strong Compound Orthogonal Arrays

Design	generators	strength	clear effects
16-run:			
$(1,4)^\star$	*Aabcd*	(1, 3, 4)	(1, 4, 0, 4, 6)
$(2,3)^\star$	*ABabc*	(2, 2, 4)	(2, 3, 1, 6, 3)
$(1,5)^\circ$	*Aabd Aace*	(1, 2, 3)	(1, 5, 0, 0, 0)
$(3,3)^\circ$	*ABC Aabc*	(2, 2, 2)	(0, 3, 0, 6, 0)
$(1,6)^\circ$	*Aabd Aace Abcf*	(1, 2, 3)	(1, 6, 0, 0, 0)
$(1,7)^\circ$	*Aabd Aace Abcf abcg*	(1, 2, 3)	(1, 7, 0, 0, 0)
32-run:			
$(1,5)^\star$	*Aabcde*	(1, 4, 5)	(1, 5, 0, 5, 10)
$(2,4)^\star$	*ABabcd*	(2, 3, 5)	(2, 4, 1, 8, 6)
$(3,3)^\star$	*ABCabc*	(3, 2, 5)	(3, 3, 3, 9, 3)
$(1,6)^\star$	*abce Aabdf*	(1, 3, 3)	(1, 6, 0, 6, 9)
$(2,5)^\star$	*abcd ABabe*	(2, 2, 3)	(2, 5, 1, 10, 4)
$(3,4)^\star$	*ABC Aabcd*	(2, 3, 2)	(0, 4, 0, 12, 6)
$(4,3)^\star$	*ABCD ABabc*	(3, 2, 3)	(4, 3, 0, 12, 3)
$(1,7)^\star$	*abce abdf Aacdg*	(1, 3, 3)	(1, 7, 0, 7, 6)
$(2,6)^\star$	*abce abdf ABacd*	(2, 2, 3)	(2, 6, 1, 12, 0)
$(3,5)^\circ$	*ABC Aabd Bace*	(2, 2, 2)	(0, 5, 0, 9, 4)
$(5,3)^\star$	*ABD ACE BCabc*	(2, 2, 2)	(0, 3, 0, 15, 3)
$(1,8)^\star$	*abcf abdg abeh Aacde*	(1, 3, 3)	(1, 8, 0, 8, 0)
$(2,7)^\star$	*abce abdf acdg ABbcd*	(2, 2, 3)	(2, 7, 1, 14, 0)
$(3,6)^\circ$	*ABC Aabd Aace Bbcf*	(2, 2, 2)	(0, 6, 0, 8, 1)
$(6,3)^\star$	*ABD ACE BCF ABCabc*	(2, 2, 2)	(0, 3, 0, 18, 3)
$(1,9)^\circ$	*Aabe Aacf Aadg Abcdh abcdi*	(1, 2, 3)	(1, 9, 0, 0, 0)
$(3,7)^\circ$	*ABC Aabd Aace Abcf Babcg*	(2, 2, 2)	(0, 7, 0, 8, 0)
$(7,3)^\circ$	*ABD ACE BCF ABCG Aabc*	(2, 2, 2)	(0, 3, 0, 18, 0)
$(1,10)^\circ$	*abce abdf acdg bcdh Aabi Aacj*	(1, 2, 3)	(1, 10, 0, 0, 0)
$(1,11)^\circ$	*abce abdf acdg bcdh Aabi Aacj Aadk*	(1, 2, 3)	(1, 11, 0, 0, 0)
$(1,12)^\circ$	*abcf abdg acdh bcdi abej acek bcel Aade*	(1, 2, 3)	(1, 12, 0, 0, 0)
$(1,13)^\circ$	*Aabe Aacf Abcg abch Aadi Abdj abdk Acdl acdm*	(1, 2, 3)	(1, 13, 0, 0, 0)
$(1,14)^\circ$	*Aabe Aacf Abcg abch Aadi Abdj abdk Acdl acdm bcdn*	(1, 2, 3)	(1, 14, 0, 0, 0)
$(1,15)^\circ$	*Aabe Aacf Abcg abch Aadi Abdj abdk Acdl acdm bcdn Aabcdo*	(1, 2, 3)	(1, 15, 0, 0, 0)

Table 2: 64-Run Optimal Strong Compound Orthogonal Arrays

Design	generators	strength	clear effects
$(1,6)^\star$	*Aabcdef*	(1, 5, 6)	(1, 6, 0, 6, 15)
$(2,5)^\star$	*ABabcde*	(2, 4, 6)	(2, 5, 1, 10, 10)
$(3,4)^\star$	*ABCabcd*	(3, 3, 6)	(3, 4, 3, 12, 6)
$(4,3)^\star$	*ABCDabc*	(4, 2, 6)	(4, 3, 6, 12, 3)
$(1,7)^\star$	*Aabcf abdeg*	(1, 3, 4)	(1, 7, 0, 7, 21)
$(2,6)^\star$	*Aabce Babdf*	(2, 3, 4)	(2, 6, 1, 12, 15)
$(3,5)^\star$	*ABabd Cabce*	(3, 2, 4)	(3, 5, 3, 15, 10)
$(4,4)^\star$	*ABCD ABabcd*	(3, 3, 3)	(4, 4, 0, 16, 6)
$(5,3)^\star$	*ABabc ABCDE*	(4, 2, 4)	(5, 3, 10, 15, 3)
$(1,8)^\star$	*abcf Aabdg acdeh*	(1, 3, 3)	(1, 8, 0, 8, 22)
$(2,7)^\star$	*abce Aabdf Bacdg*	(2, 3, 3)	(2, 7, 1, 14, 15)
$(3,6)^\star$	*abce ACabd Bacdf*	(3, 2, 3)	(3, 6, 3, 18, 9)
$(4,5)^\star$	*ABCD ABabd ACace*	(3, 2, 3)	(4, 5, 0, 20, 10)
$(5,4)^\star$	*ABD ACE BCabcd*	(2, 3, 2)	(0, 4, 0, 20, 6)
$(6,3)^\star$	*ABCE ABDF ACDabc*	(3, 2, 3)	(6, 3, 0, 18, 3)
$(1,9)^\star$	*abcf Aabdg Aabeh acdei*	(1, 3, 3)	(1, 9, 0, 9, 24)
$(2,8)^\star$	*abcf Aabdg Aabeh Bacde*	(2, 3, 3)	(2, 8, 1, 16, 16)
$(3,7)^\star$	*abcf abdg Bacde ACabe*	(3, 2, 3)	(3, 7, 3, 21, 6)
$(4,6)^\star$	*ABCD ABabd ABace ACbcf*	(3, 2, 3)	(4, 6, 0, 24, 9)
$(5,5)^\star$	*ABD ACE BCabd ABCace*	(2, 2, 2)	(0, 5, 0, 25, 10)
$(6,4)^\star$	*ABD ACE BCF ABCabcd*	(2, 3, 2)	(0, 4, 0, 24, 6)
$(7,3)^\star$	*ABCE ABDF ACDG BCDabc*	(3, 2, 3)	(7, 3, 0, 21, 3)
$(1,10)^\star$	*abcg abdh acdei acdfj Aabef*	(1, 3, 3)	(1, 10, 0, 10, 24)
$(2,9)^\star$	*abcf abdg abeh acdei ABbcde*	(2, 2, 3)	(2, 9, 1, 18, 8)
$(3,8)^\star$	*ABC abce Aabdf Bacdg ABbcdh*	(2, 3, 2)	(0, 8, 0, 24, 16)
$(4,7)^\star$	*abce abdf ACacd BDacd ABabg*	(3, 2, 3)	(4, 7, 0, 28, 6)
$(5,6)^\star$	*ABD ACE BCabd BCace ABCbcf*	(2, 2, 2)	(0, 6, 0, 30, 9)
$(6,5)^\star$	*ABD ACE BCF ABCabd ABCace*	(2, 2, 2)	(0, 5, 0, 30, 4)
$(7,4)^\star$	*ABD ACE BCF ABCG Aabcd*	(2, 3, 2)	(0, 4, 0, 28, 6)
$(8,3)^\star$	*ABCE ABDF ACDG BCDH ABabc*	(3, 2, 3)	(8, 3, 0, 24, 3)
$(1,11)^\star$	*abcg abdh acei adfj aefk Aabcdef*	(1, 3, 3)	(1, 11, 0, 11, 10)
$(2,10)^\star$	*abcf abdg aceh adei abcdej ABacd*	(2, 2, 3)	(2, 10, 1, 20, 0)
$(3,9)^\circ$	*ABC Aabe Aacf Badg Abcdh ABabcdi*	(2, 2, 2)	(0, 9, 0, 17, 16)
$(5,7)^\star$	*ABD ACE BCabd BCace BCbcf Aabcg*	(2, 2, 2)	(0, 7, 0, 35, 6)
$(6,6)^\star$	*ABD ACE BCF ABCabd ABCace ABCbcf*	(2, 2, 2)	(0, 6, 0, 36, 0)
$(7,5)^\circ$	*ABD ACE BCF ABCG Aabd Bace*	(2, 2, 2)	(0, 5, 0, 29, 4)
$(9,3)^\star$	*ABE ACF ADG BCDH ABCDI BCabc*	(2, 2, 2)	(0, 3, 0, 27, 3)
$(1,12)^\star$	*abcg abdh abei Abcde acfj defk acdefl*	(1, 3, 3)	(1, 12, 0, 12, 0)
$(2,11)^\star$	*abcf abdg acdh abei acej adek ABbcd*	(2, 2, 3)	(2, 11, 1, 22, 0)
$(3,10)^\circ$	*ABC Aabe Aacf Bbcg Badh Abcdi ABabcdj*	(2, 2, 2)	(0, 10, 0, 10, 17)
$(6,7)^\star$	*ABD ACE BCF ABCabd ABCace ABCbcf abcg*	(2, 2, 2)	(0, 7, 0, 42, 0)
$(7,6)^\circ$	*ABD ACE BCF ABCG Aabd Bace Cbcf*	(2, 2, 2)	(0, 6, 0, 30, 3)
$(10,3)^\star$	*ABE ACF BCG ADH BCDI ABCDJ BDabc*	(2, 2, 2)	(0, 3, 0, 30, 3)
$(1,13)^\star$	*abcg abdh acei adej abfk Abcdf cefl defm*	(1, 3, 3)	(1, 13, 0, 13, 0)
$(2,12)^\star$	*abcf abdg acdh bcdi abej acek adel ABbce*	(2, 2, 3)	(2, 12, 1, 24, 0)
$(3,11)^\circ$	*ABC Aabe Aacf Bbcg Aadh Bbdi Bacdj Abcdk*	(2, 2, 2)	(0, 11, 0, 8, 6)
$(7,7)^\circ$	*ABD ACE BCF ABCG Aabd Bace Cbcf ABCabcg*	(2, 2, 2)	(0, 7, 0, 28, 0)
$(11,3)^\star$	*ABE ACF BCG ADH BDI ACDJ BCDK ABCDabc*	(2, 2, 2)	(0, 3, 0, 33, 3)
$(1,14)^\star$	*abcg abdh acei adej abcdek abfl Aacdf aefm abcefn*	(1, 3, 3)	(1, 14, 0, 14, 0)
$(2,13)^\star$	*abcf abdg acdh bcdi abej acek bcel adem ABbde*	(2, 2, 3)	(2, 13, 1, 26, 0)
$(3,12)\circ$	*ABC Aabe Aacf Bbcg Aadh Bbdi ABacdj Abcdk abcdl*	(2, 2, 2)	(0, 12, 0, 7, 3)
$(12,3)\star$	*ABE ACF BCG ADH BDI ACDJ BCDK ABCDL ABCabc*	(2, 2, 2)	(0, 3, 0, 36, 3)
$(1,15)\star$	*abcg abdh acei adej abcdek abfl Aacdf aefm abcefn abdefo*	(1, 3, 3)	(1, 15, 0, 15, 0)
$(2,14)\star$	*abcf abdg acdh bcdi abej acek bcel adem bden ABcde*	(2, 2, 3)	(2, 14, 1, 28, 0)
$(3,13)^\circ$	*ABC Aabe Aacf Abcg Babch Aadi Abdj Babdk Bcdl acdm*	(2, 2, 2)	(0, 13, 0, 6, 2)
$(13,3)^\star$	*ABE ACF BCG ABCH ADI BDJ ABDK CDL ACDM BCDabc*	(2, 2, 2)	(0, 3, 0, 39, 3)

a 32-run COA with $l_c = 4$ and $l_n = 6$. If one can only afford to conduct a 32-run experiment, he/she should consider single arrays. For given l_c, l_n and run size 2^k, formally, a single array is a $2^{(l_c+l_n)-m}$ design with l_c columns assigned to the control factors and the remaining l_n columns assigned to the noise factors, where $m = l_c + l_n - k$ (Wu and Zhu, 2003). The necessary and sufficient condition for the existence of a single array is $l_c + l_n \leq 2^k - 1$. For the experiment with four control factors (A, B, C, D) and six noise factors (a, b, c, d, e, f), for example, the single array generated by the independent defining words abd, ace, bcf, $ACabc$ and $ABDa$, can be considered.

COAs are single arrays that possess crossing structures, so they are special cases of single arrays. Based on the number of control factors (l_c), the number of noise factors (l_n), and the run size (2^k), we propose to classify single arrays into two categories, which are prodigal single arrays and economical single arrays, as follows. A single array with l_c, l_n and run size 2^k is a prodigal single array (PSA) if $(l_c, l_n) \in \mathcal{S}(2^k)$, and it is an economical single array (ESA) if $(l_c, l_n) \notin \mathcal{S}(2^k)$.

We know that 2^k-run single arrays exist for l_c control factors and l_n noise factors if $l_c + l_n \leq 2^k - 1$. For 2^k-run single arrays, the total degrees of freedom are fixed to be 2^k, but l_c and l_n can vary. Heuristically, when $l_c + l_n$ is small, the number of lower-order effects such as the control main effects and the control-by-noise 2fi's is small, so it may be possible to construct the single array in a way such that all the lower-order effects can be clearly estimable, and we claim that the single array has large relative estimation capacity with respect to l_c and l_n. When $l_c + l_n$ is large, the number of lower-order effects is large, so it may not be possible to have a single array in which all the lower-order effects are clearly estimable, and we claim that the single array has small relative estimation capacity with respect to l_c and l_n. One purpose of classifying single arrays into PSAs and ESAs is to distinguish single arrays with large relative estimation capacities, i.e., PSAs, from those with small relative estimation capacities, i.e., ESAs. As will be clear later, different criteria need to be used for selecting optimal PSAs and ESAs. Another purpose of the above classification is to facilitate the fair comparison between cross arrays and single arrays. In the literature, there is much discussion regarding the advantages and disadvantages of single arrays versus cross arrays, however, no general conclusions have been reached. We suggest that cross arrays or COAs are only comparable with PSAs, but not directly with ESAs. This greatly clarifies the discussion, and leads to various useful criteria for choosing COAs, PSAs and ESAs. In the rest of the section, we focus on proposing optimality criteria for single arrays in general in Section 4.1, then discuss the selection of optimal ESAs in Section 4.2, and the selection of optimal PSAs in Section 4.3.

4.1 Optimality Criteria for Single Arrays

4.1.1 W_s-Aberration

The wordtype patterns $A_{i.j}$ of single arrays are not subject to the restrictions $A_{i.1} = A_{i.2} = 0$. The arguments used to arrange $A_{i.j}$ with $j \geq 3$ into W_c in Section 3.2 are still valid for single arrays. To obtain a complete sequence of wordtype patterns for single arrays, we need a proper way to insert $A_{i.j}$ with $j = 1, 2$ into W_c. Let us first consider $A_{2.1}$ and $A_{1.2}$, or equivalently, CCn and Cnn. The worst aliasing relations induced by CCn and Cnn are $C = Cn$ and $Cn = n$ respectively. Because C and Cn are of primary importance for parameter design, $C = Cn$ is more severe than $Cn = n$. So $CCn \lhd Cnn$. Clearly, both CCn and Cnn are more severe in terms of aliasing severity than CCC and nnn. Hence, we have $CCn \lhd Cnn \lhd CCC \lhd nnn$, or, $A_{2.1} \lhd A_{1.2} \lhd A_{3.0} \lhd A_{0.3}$. Next, we consider $A_{4.1}$ and $A_{4.2}$, or $CCCn$ and $CCnn$. The worst aliasing relation induced by $CCCn$ is $CC = Cn$ and the worst by $CCnn$ is $Cn = Cn$. Both CC and Cn are effects of order 2, but Cn plays a more crucial role in parameter design than CC. So $Cn = Cn$ is considered to be more severe than $CC = Cn$, which implies $CCnn \lhd CCCn$. In a similar way, we have $CCCn \lhd Cnnn$. Hence, $CCnn \lhd CCCn \lhd Cnnn \lhd CCCC \lhd nnnn$, or $A_{2.2} \lhd A_{3.1} \lhd A_{1.3} \lhd A_{4.0} \lhd A_{0.4}$. We can further compare defining words with length larger than four. This leads to the same scheme as stated in (3) for ranking two different wordtype patterns $A_{i_1.j_1}$ and $A_{i_2.j_2}$ in terms of aliasing severity. We denote the resulted sequence by W_s, which is

$$W_s = (A_{2.1}, A_{1.2}, A_{3.0}, A_{0.3}, A_{2.2}, A_{3.1}, A_{1.3}, A_{4.0}, A_{0.4}, A_{3.2}, A_{2.3}, A_{4.1}, A_{1.4}, A_{5.0}, A_{0.5}, \ldots). \quad (4)$$

Note that W_c is a subsequence of W_s.

The hierarchical ordering principle states that (i) effects of lower order are more important than those of higher order; (ii) effects of the same order are equally important. In W_s, (i) is preserved, but (ii) does not hold. In fact, effects with the same order are further distinguished according to their relative importance for parameter design. Based on W_s, W_s-aberration and the minimum W_s-aberration criterion can be defined and proposed for selecting optimal single arrays. In the derivation of W_s, we have only considered the aliasing relations implied by the wordtype patterns and have not taken into account the run size of a single array, or to be more precise, the relative estimation capacity of a single array. When the relative estimation capacity of a single array is much limited, for example, in an economical single array, it is impossible to guarantee that all the lower-order effects can be clearly estimated. Hence, in selecting practically useful single arrays, one needs to prioritize the estimation of important effects versus less important effects. In extreme cases, the estimation of the less important effects may need to be compromised entirely.

4.1.2 Split Wordtype Patterns and (W_{sm}, W_{sn})-Aberration

The major advantage of using ESAs is run size economy. As mentioned in the end of the previous subsection, the relative estimation capacities of ESAs are already small. In other words, ESAs usually do not have enough capacities to accommodate all the low-order effects in a balanced way. Nonetheless, the use of ESAs in practice can be justified by the effects sparsity principle and the effects asymmetry existing in parameter design (Shoemaker, et al., 1991). The effects sparsity principle states that the number of effects that are significant in a factorial experiment is relatively small (Box and Meyer, 1986), and the effects asymmetry refers to the fact that control-by-noise 2fi's as well as control main effects are more important than noise main effects and other 2fi's. If an ESA ensures the clear estimation of the important effects, e.g., control main effects and control-by-noise 2fi's, while sacrificing other less important effects, e.g., noise effects, it is still a practically useful experimental plan. To reflect the emphasis on the important effects and the discrimination against the less important effects, we first split W_s into two separate sequences:

$$W_{sm} = (A_{2.1}, A_{1.2}, A_{3.0}, A_{2.2}, A_{3.1}, A_{1.3}, A_{4.0}, A_{3.2}, A_{2.3}, A_{4.1}, A_{1.4}, A_{5.0}, A_{3.3}, \ldots), \tag{5}$$

and

$$W_{sn} = (A_{0.3}, A_{0.4}, A_{0.5}, A_{0.6}. \ldots). \tag{6}$$

Notice that W_{sm} consists of $A_{i.j}$ with $i > 1$ that involve at least one control factor; W_{sn} consists of $A_{i.j}$ with $i = 0$ that involve only noise factors. We call W_{sm} the mixed wordtype pattern sequence and W_{sn} the noise wordtype pattern sequence. Next we append W_{sn} in the end of W_{sm} to form a longer sequence (W_{sm}, W_{sn}), which is called the split wordtype pattern sequence.

Based on (W_{sm}, W_{sn}), (W_{sm}, W_{sn})-aberration and the minimum (W_{sm}, W_{sn})-aberration criterion can be defined. Note that, in (W_{sm},W_{sn}), noise effects are treated as secondary, and are sacrificed to ensure more estimation capacity for control effects and control-by-noise interactions. Thus both statements (i) and (ii) in the hierarchical ordering principle are violated. It is known that defining words that contain only noise factors can also induce aliasing between important effects. For instance, nnn induces $Cn = Cnn$. As will be discussed in the following subsection, the split wordtype pattern sequence is one of the possible compromising schemes aimed at allocating most estimation capacity to important effects. It usually does not guarantee the clear estimation of all important effects, neither do other schemes. We propose (W_{sm}, W_{sn}) first because it is a systematic scheme that possesses clear combinatorial structure and often leads to overall good single arrays, especially good ESAs.

4.1.3 Shifted Wordtype Patterns and W_{ss}-Aberration

In contrast to (W_{sm}, W_{sn}), milder compromising schemes can be introduced by shifting $A_{0.j}$ with $j \geq 3$ rightward to new positions in W_s, instead of appending all of them at the end of W_{sm}. We propose one possible shifting scheme next. Because the estimation of noise effects is to be compromised, we ignore the aliasing between them in the following discussion. We follow the worst-case argument used by Bingham and Sitter (2003). Consider $A_{0.3}$ or nnn. The worst alias relation induced by nnn is $Cn = Cnn$, similar to $C_1n = C_2nn$, which is the worst aliasing relation induced by $CCnnn$. This implies that nnn and $CCnnn$ are comparable in terms of aliasing severity. So we can have either $nnn \lhd CCnnn$ or $CCnnn \lhd nnn$. We decide to select $nnn \lhd CCnnn$ and move $A_{0.3}$ to the position between $A_{3.2}$ and $A_{2.3}$. Next, consider $A_{0.4}$ or $nnnn$. The worst case induced by $nnnn$ is $Cnn = Cnn$, similar to $C_1nn = C_2nn$, which is induced by $CCnnnn$. Hence, we shift $A_{0.4}$ to the position between $A_{4.2}$ and $A_{2.4}$. In general, following the same argument, we have $A_{i.2} \lhd A_{0.i} \lhd A_{2.i}$ for $i \geq 3$. Shifting all $A_{0.i}$ rightward as described above, W_s becomes

$$W_{ss} = (A_{2.1}, A_{1.2}, A_{3.0}, A_{2.2}, A_{3.1}, A_{1.3}, A_{4.0}, A_{3.2}, \mathbf{A_{0.3}}, A_{2.3}, A_{4.1}, A_{1.4}, A_{5.0}, A_{3.3}, A_{4.2}, \mathbf{A_{0.4}}, A_{2.4}, A_{5.1}, A_{1.5}, A_{6.0}, A_{4.3}, \ldots). \tag{7}$$

$A_{0.3}$ and $A_{0.4}$ are highlighted in W_{ss} to indicate their new positions. W_{ss} is called the shifted wordtype pattern sequence.

The difference between W_{ss} and a list of rank-ordered effects given in Bingham and Sitter (2003) is that W_{ss} further distinguishes wordtype patterns with the same modified wordlength. For example, CCn and Cnn have the same modified wordlength 2.5, but $CCn \lhd Cnn$ in W_{ss}; CCC and $CCnn$ have the same modified wordlength 3, but $CCC \lhd CCnn$ in W_{ss}. Bingham and Sitter (2003) treated nnn as equally important as $CCCC$, $CCCnn$ and $CCnnn$, while in W_{ss}, $CCCC \lhd CCCnn \lhd nnn \lhd CCCnn$. If the wordtypes with the same modified wordlength were combined, W_{ss} is reduced to be

$$W_{DR} = (A_{2.1} + A_{1.2}, A_{3.0} + A_{2.2}, A_{3.1} + A_{1.3}, A_{4.0} + A_{3.2} + \mathbf{A_{0.3}} + A_{2.3}, A_{4.1} + A_{1.4}, A_{5.0} + A_{3.3} + A_{4.2}$$

$$+\mathbf{A_{0.4}} + A_{2.4}, A_{5.1} + A_{1.5}, A_{6.0} + A_{4.3}, \ldots), \tag{8}$$

which was exactly the sequence proposed in Bingham and Sitter (2003). The W_{ss}-aberration and the W_{DR}-aberration can be defined in a standard fashion, so are the minimum W_{ss}-aberration criterion and the W_{DR}-aberration criterion. Bingham and Sitter (2003) reported minimum W_{DR}-aberration designs. Because W_{DR} is relatively a coarse sequence, it may not be able to discriminate designs that may have different aliasing and structural properties.

Example 3. Consider a 32-run experiment involving two control factors (A, B) and five noise factors (a, b, c, d e). Three minimum W_{DR}-aberration arrays were reported in Table 4 of Bingham and Sitter (2003), which are $d_1 : I = abcd = ABabe = ABcde$, $d_2 : I = abc = ABade = ABbcde$, and $d_3 : I = abc = ade = bcde$. d_1, d_2 and d_3 share the same $W_{DR} = (0, 0, 0, 2, 0, 1)$, and all of them are in fact COAs. Their strength vectors are $T(d_1) = (2, 2, 3)$, $T(d_2) = (2, 2, 2)$ and $T(d_3) = (2, 2, 2)$, so d_1 possesses higher strengths than d_2 and d_3. All of the designs guarantee the clear estimation of control main effects and control-by-noise 2fi's. In addition, d_1 guarantees the clear estimation of noise main effects. Therefore, d_1 is overall better than the other two arrays. The shifted wordtype pattern sequences of d_1, d_2 and d_3 are $W_{ss}(d_1) = (0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, \ldots)$, $W_{ss}(d_2) = (0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, \ldots)$, and $W_{ss}(d_3) = (0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, \ldots)$. Hence, d_1 is the minimum W_{ss}-aberration single array, followed by d_2 and d_3.

In summary, on one hand, W_{ss} is more elaborate than W_{DR}, therefore it is more sensitive in discriminating different designs. On the other hand, W_{DR} is more conservative, because the minimum W_{DR}-aberration criterion usually leads to a group of designs so that good designs would not be missed.

4.2 Optimal Economical Single Arrays

Because PSAs and ESAs are fairly different from each other in terms of their relative estimation capacities, different optimality criteria should be employed to select optimal PSAs and ESAs. In this section, we will focus on the selection of optimal ESAs. For ESAs, because their relative estimation capacities are already limited, the minimum W_{ss}-aberration and (W_{sm}, W_{sn})-aberration criteria are more appropriate than the minimum W_s-aberration criterion. We have also found that W_{ss}-aberration and (W_{sm}, W_{sn})-aberration often lead to the same optimal ESAs. Hence, the minimum (W_{sm}, W_{sn})-aberration criterion more genuinely reflects the structures of optimal ESAs. In this paper, we only report the ESAs with minimum (W_{sm}, W_{sn})-aberrations.

Similar to the selection of optimal COAs, we have carried out exhaustive computer search with isomorphism checking to select ESAs with minimum (W_{sm}, W_{sn})-aberration, and have obtained all the optimal 16-, 32- and 64-run (up to 16 factors) ESAs. Due to limited space, only some of the optimal ESAs are chosen and presented in Table 3. There are two general rules to guide the choice of optimal ESAs. First, Table 3 only includes optimal ESAs with $l_c + l_n \leq 2^{k-1}$ for $k = 4, 5$; $l_c + l_n \leq 16$ for $k = 6$. Second, if an optimal ESA with l_c and l_n is also an ordinary MA $2^{(l_c+l_n)-m}$ design ($m = l_c + l_n - k$), it is not listed in Table 3. The complete tables of 16-, 32- and 64-run optimal ESAs can be requested from the authors.

4.3 Optimal Prodigal Single Arrays

For fixed l_c and l_n, compared with ESAs, PSAs have larger relative estimation capacities. Hence, discrimination against less important effects in the selection of optimal PSAs may not be as necessary as in the selection of optimal ESAs. Among the three optimality criteria proposed in Sections 4.1.2-3, it appears that the minimum W_s-aberration criterion is the most suitable for PSAs. Although, for most l_c and l_n, the optimal PSAs according to the minimum W_s-aberration criterion are much better than the optimal PSAs according to the other two criteria in terms of strengths and the number of clearly estimable effects, there exist cases where the latter two criteria select better arrays. These cases occur usually when (l_c, l_n) are on the boundary of $\mathcal{S}(2^k)$ in the sense that there does not exist $(l'_c, l'_n) \in \mathcal{S}(2^k)$ such that $l'_c \geq l_c$, $l'_n \geq l_n$ and $l'_c + l'_n > l_c + l_n$. Hence, when constructing the tables of optimal PSAs of 16-, 32- and 64-run, we have considered all the three criteria. In the cases where the criteria lead to different optimal PSAs, if one optimal PSA is apparently better than the others, it will be selected only; if they are comparable with each other, we keep all of them in the tables. The obtained optimal PSAs are reported in Table 4 including 16- and 32-run arrays and in Table 5 including 64-run arrays with up to 16 factors.

In both Tables 4 and 5, from the left to the right, the first two columns give (l_c, l_n) and the independent defining words (or Generators), the next three columns indicate whether an array is optimal according to W_s-, (W_{sm}, W_{sn})- and W_{ss}-aberration, and the last two columns indicate whether the array is also a COA and whether it is a MA design with the distinction between control and noise factors neglected. For example, in Table 4, there are two optimal 32-run PSAs for $l_c = 4$ and $l_n = 3$, which are

Table 3: 16- 32- and 64-Run Optimal ESAs

Design	generator	clear 2fi
16-run:		
$(2,4)$	$abc\ ABad$	(2, 1, 0, 4, 2)
$(2,5)$	$abd\ ace\ ABbc$	(2, 0, 0, 2, 0)
$(2,6)$	$abd\ ace\ bcf\ ABabc$	(2, 0, 0, 0, 0)
32-run:		
$(4,5)$	$ABac\ ABbd\ Aabe\ BCDab$	(4, 5, 5, 10, 0)
$(5,4)$	$abd\ ACac\ BDac\ ABEb$	(5, 1, 0, 7, 2)
$(2,8)$	$abd\ ace\ bcf\ abcg\ ABah$	(2, 1, 0, 12, 6)
$(4,6)$	$abd\ ace\ bcf\ ACabc\ ABDa$	(4, 0, 0, 8, 0)
$(5,5)$	$abd\ ace\ ACbc\ BDbc\ ABEa$	(5, 0, 0, 4, 0)
$(6,4)$	$abc\ ABDa\ ACEa\ BCFa\ ABCbd$	(6, 1, 0, 6, 2)
$(2,9)$	$abe\ acf\ bcg\ abch\ adi\ ABbd$	(2, 0, 0, 10, 0)
$(3,8)$	$abd\ ace\ bcf\ Aabcg\ Babch\ ABCa$	(3, 2, 0, 4, 0)
$(4,7)$	$abd\ ace\ bcf\ ACabc\ Babcg\ ABDa$	(4, 1, 0, 4, 0)
$(5,6)$	$abd\ ace\ bcf\ ACabc\ BDabc\ ABEa$	(5, 0, 0, 4, 0)
$(6,5)$	$abd\ ace\ ACbc\ BDbc\ ABEa\ ABFabc$	(6, 0, 0, 0, 0)
$(7,4)$	$ABac\ ACDa\ BCEa\ ABCF\ Aabd\ BGab$	(7, 4, 0, 0, 0)
$(8,3)$	$ABDa\ ACEa\ BCFa\ ABCG\ Aabc\ BHab$	(8, 3, 0, 0, 0)
$(2,10)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ ABcd$	(2, 0, 0, 8, 0)
$(3,9)$	$abf\ acg\ adh\ Abcd\ aei\ Bbce\ Cbde$	(3, 0, 0, 3, 0)
$(4,8)$	$abd\ ace\ bcf\ ACabc\ Babcg\ ABDa\ ABbch$	(4, 2, 0, 0, 0)
$(5,7)$	$abd\ ace\ bcf\ ACabc\ BDabc\ ABEa\ ABbcg$	(5, 1, 0, 0, 0)
$(6,6)$	$abd\ ace\ bcf\ ACabc\ BDabc\ ABEa\ ABFbc$	(6, 0, 0, 0, 0)
$(8,4)$	$ACab\ Dabc\ Aacd\ AEbc\ BFab\ ABGa\ ABHb$	(8, 4, 0, 0, 0)
$(2,11)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ abdk\ ABcd$	(2, 0, 0, 6, 0)
$(3,10)$	$abe\ acf\ bcg\ adh\ bdi\ ABabc\ Aabdj\ ACcd$	(3, 1, 0, 0, 0)
$(4,9)$	$abf\ acg\ adh\ Abcd\ aei\ Bbce\ Cbde\ Dcde$	(4, 0, 0, 4, 0)
$(2,12)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ abdk\ cdl\ ABacd$	(2, 0, 0, 4, 0)
$(3,11)$	$abe\ acf\ bcg\ adh\ bdi\ acdj\ ABabc\ ACabd\ Acdk$	(3, 1, 0, 0, 0)
$(2,13)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ abdk\ cdl\ acdm\ ABbcd$	(2, 0, 0, 2, 0)
$(3,12)$	$abe\ acf\ bcg\ adh\ bdi\ acdj\ bcdk\ ABabc\ ACabd\ Acdl$	(3, 1, 0, 0, 0)
$(2,14)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ abdk\ cdl\ acdm\ bcdn\ ABabcd$	(2, 0, 0, 0, 0)
$(3,13)$	$abe\ acf\ bcg\ adh\ bdi\ acdj\ bcdk\ abcdl\ ABabc\ ACabd\ Acdm$	(3, 1, 0, 0, 0)
64-run:		
$(4,8)$	$abe\ acf\ bcg\ Aabch\ ACad\ BDabcd$	(4, 2, 5, 20, 4)
$(8,4)$	$Cabc\ Dabd\ AEacd\ BFacd\ ABGab\ ABHbcd$	(8, 4, 12, 24, 0)
$(4,9)$	$abe\ acf\ bcg\ adh\ Aabci\ ACbd\ ABDacd$	(4, 1, 5, 20, 2)
$(5,8)$	$abd\ ace\ bcf\ abcg\ ABDa\ ACEb\ BCch$	(5, 1, 2, 26, 5)
$(8,5)$	$Aabd\ Aace\ ADbc\ BEabc\ CFabc\ ABCG\ ABCHabc$	(8, 5, 6, 30, 0)
$(9,4)$	$ABCE\ ABDF\ ACDG\ ABHa\ BCDbc\ Aabd\ ABCDIab$	(9, 4, 6, 16, 0)
$(4,10)$	$abf\ acg\ bch\ adi\ bdj\ Babce\ Cabde\ ADcde$	(4, 1, 5, 20, 0)
$(5,9)$	$abe\ acf\ bcg\ abch\ adi\ ACbd\ BDabd\ ABEc$	(5, 0, 2, 23, 0)
$(6,8)$	$abd\ ace\ bcf\ abcg\ ABDa\ ACEb\ BCFc\ ABCabch$	(6, 1, 0, 24, 4)
$(8,6)$	$abd\ ace\ bcf\ ADabc\ ABEa\ ACFa\ BCGabc\ ABCHbc$	(8, 0, 0, 24, 0)
$(9,5)$	$abd\ ace\ ADbc\ BEbc\ ABFa\ ACGa\ BCHabc\ ABCIbc$	(9, 0, 0, 20, 0)
$(10,4)$	$ABEa\ ACFa\ BDGa\ CDHa\ ABCDIa\ Aabc\ BCabd\ ABCJb$	(10, 4, 4, 12, 0)
$(4,11)$	$abe\ acf\ bcg\ adh\ bdi\ acdj\ ACabc\ Aabdk\ ABDcd$	(4, 1, 5, 22, 0)
$(5,10)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ ACcd\ BDacd\ ABEb$	(5, 0, 2, 20, 0)
$(6,9)$	$abe\ acf\ bcg\ abch\ adi\ ACbd\ BDbd\ ABEc\ ABFabcd$	(6, 0, 0, 20, 0)
$(7,8)$	$abe\ acf\ adg\ Cbcd\ ADbc\ Abdh\ BEabcd\ ABFb\ ABGacd$	(7, 1, 0, 14, 1)
$(8,7)$	$abe\ acf\ adg\ Cbcd\ ADbc\ AEbd\ BFabcd\ ABGb\ ABHacd$	(8, 0, 0, 15, 0)
$(9,6)$	$abd\ ace\ bcf\ ADabc\ BEabc\ ABFc\ ACGa\ BCHb\ ABCIab$	(9, 0, 4, 12, 0)
$(10,5)$	$abd\ ace\ ADbc\ BEbc\ ABFa\ CGbc\ ACHa\ BCIabc\ ABCJ$	(10, 0, 0, 16, 0)
$(11,4)$	$abd\ ADac\ BEac\ ABFb\ CGac\ ACHb\ BCIabc\ ABCJa\ ABCKc$	(11, 1, 0, 8, 2)
$(4,12)$	$abe\ acf\ bcg\ adh\ bdi\ acdj\ bcdk\ ACabc\ Aabdl\ ABDcd$	(4, 1, 5, 24, 0)
$(5,11)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ abdk\ ACcd\ BDacd\ ABEb$	(5, 0, 2, 17, 0)
$(6,10)$	$abe\ acf\ bcg\ abch\ adi\ bdj\ ACcd\ BDcd\ ABEa\ ABFbcd$	(6, 0, 0, 14, 0)
$(7,9)$	$abf\ acg\ adh\ Bbcd\ aei\ Cbce\ Dbde\ AEbc\ AFabde\ AGacde$	(7, 0, 0, 10, 0)
$(8,8)$	$abe\ acf\ adg\ Cbcd\ Abch\ ADbd\ AEacd\ BFabcd\ ABGa\ ABHbcd$	(8, 1, 0, 14, 1)
$(9,7)$	$abe\ acf\ bcg\ Cabcd\ ADabc\ AEad\ BFabc\ BGad\ ABHb\ ABIabd$	(9, 1, 0, 8, 0)
$(10,6)$	$abd\ ace\ bcf\ ADabc\ BEabc\ ABFa\ CGabc\ ACHb\ BCIc\ ABCJ$	(10, 0, 3, 12, 0)
$(11,5)$	$abd\ ace\ ADbc\ BEbc\ ABFa\ CGbc\ ACHa\ BCIabc\ ABCJ\ ABCKbc$	(11, 0, 0, 12, 0)
$(12,4)$	$ABCE\ ABDF\ ACDG\ BCDH\ ABIa\ ACJa\ ADKa\ ABbc\ CDabd\ ABCDLab$	(12, 4, 2, 6, 0)

Design	Generators	W_s	(W_{sm}, W_{sn})	W_{ss}	COA	MA
$(4,3)$	$ABCD\ ABabc$	$\surd$			$\surd$	$\surd$
$(4,3)$	$abc\ ABCDa$		$\surd$	$\surd$		.

We refer to the first array above as d_1 and the second as d_2. d_1 is optimal according to the minimum W_s-aberration criterion and it is in fact an optimal COA, while d_2 is optimal according to the minimum (W_{sm}, W_{sn})-aberration and W_{ss}-aberration criteria. Note that for a 32-run parameter design experiment involving four control factors and three noise factors, the optimal COA coincides with the optimal PSA according to W_s-aberration. However, this is not generally true in other cases as shown in Tables 4 and 5. It is not difficult to see that d_2 is not a COA. d_2 guarantees the clear estimation of all the control main effects , the control-by-control 2fi's and the control-by-noise 2fi's, but the noise main effects and noise-by-noise 2fi's are aliased with each other. In d_1, the noise main effects and noise-by-noise 2fi's are clearly estimable, but the control-by-control 2fi's are aliased with each other. As a single array, d_2 is better than d_1 if the noise 2fi's are assumed to be negligible.

When $l_c = 3$ and $l_n = 5$, there are also two optimal 32-run PSAs reported in Table 4, which are

Design	Generators	W_s	(W_{sm}, W_{sn})	W_{ss}	COA	MA
$(3,5)$	$Aabd\ Aace\ ABCbc$	$\surd$				$\surd$
$(3,5)$	$abd\ ace\ ABCbc$		$\surd$	$\surd$		

We denote the first array above as d_3 and the second array as d_4. d_3 is optimal according to the minimum W_s-aberration criterion while d_4 is optimal according to the minimum (W_{sm}, W_{sn})-aberration and W_{ss}-aberration criteria. Neither of d_3 and d_4 is COA. The optimal 32-run COA for three control factors and four noise factors given in Table 1 is generated by ABC, $Aabd$ and $Bace$ and we denote it as d_5. In terms of the number of clearly estimable effects, d_3 and d_4 are better than d_5.

5 CONCLUDING REMARKS

In this paper, we suggest that structural constraints and relative estimation capacities should be taken into consideration when selecting optimal plans for parameter design experiments. We have proposed various optimality criteria for selecting useful COAs, ESAs and PSAs. Using the tables provided in this paper, experimenters can consider and compare possible good experimental plans, which are optimal in one way or another, and choose the one that best fits their experimental constraints, capacity and goals. For example, suppose there are four control

Table 4: 16- and 32-Run Optimal PSAs with $(l_c, l_n) \in \mathcal{S}(2^k)$

Design	Generators	W_s	(W_{sm}, W_{sn})	W_{ss}	COA	MA
16-Run:						
(1, 4)	*Aabcd*	√			√	√
(2, 3)	*ABabc*	√		√	√	√
(3, 2)	*ABCab*	√	√	√		√
(4, 1)	*ABCDa*	√	√	√		√
(1, 5)	*abd ace*		√	√	√	
(3, 3)	*ABCa ABbc*	√				√
(3, 3)	*ABCa abc*		√	√		
(5, 1)	*ABDa ACEa*	√	√	√		√
(1, 6)	*abd ace bcf*		√	√	√	
(6, 1)	*ABDa ACEa BCFa*	√	√	√		√
(1, 7)	*abd ace bcf abcg*		√	√	√	
(7, 1)	*ABDa ACEa BCFa ABCG*	√	√	√		√
32-Run:						
$(1, 5)$	*Aabcde*	√			√	√
$(2, 4)$	*ABabcd*	√		√	√	√
$(3, 3)$	*ABCabc*	√		√	√	√
$(4, 2)$	*ABCDab*	√	√	√		√
$(5, 1)$	*ABCDEa*	√	√	√		√
$(1, 6)$	*abce Aabdf*	√			√	√
$(2, 5)$	*abcd ABabe*	√		√	√	√
$(3, 4)$	*abcd ABCab*	√				√
$(4, 3)$	*ABCD ABabc*	√			√	√
$(4, 3)$	*abc ABCDa*		√	√		
$(5, 2)$	*ABCD ABEab*	√	√	√		√
$(6, 1)$	*ABCE ABDFa*	√	√	√		√
$(1, 7)$	*abce abdf Aacdg*	√			√	√
$(2, 6)$	*abce abdf ABacd*	√		√	√	√
$(3, 5)$	*Aabd Aace ABCbc*	√				√
$(3, 5)$	*abd ace ABCbc*		√	√		
$(5, 3)$	*ABDa ACEa BCabc*	√				√
$(6, 2)$	*ABCE ABDF ACDab*	√	√	√		√
$(7, 1)$	*ABCE ABDF ACDGa*	√	√	√		√
$(1, 8)$	*abcf abdg abeh Aacde*	√			√	√
$(2, 7)$	*abce abdf acdg ABbcd*	√		√	√	
$(3, 6)$	*Aabd Aace Abcf BCabc*	√				
$(3, 6)$	*abd ace bcf ABCabc*		√	√		
$(6, 3)$	*ABDa ACEa BCFa ABCbc*	√				
$(7, 2)$	*ABCE ABDF ACDG BCDab*	√	√	√		√
$(8, 1)$	*ABCF ABDG ABEH ACDEa*	√	√	√		√
$(1, 9)$	*abe acf adg bcdh abcdi*		√	√	√	
$(3, 7)$	*abd ace bcf abcg ABCa*		√	√		
$(7, 3)$	*abc ABDa ACEa BCFa ABCGb*		√	√		
$(9, 1)$	*ABEa ACFa ADGa BCDHa ABCDI*	√	√	√		√
$(1, 10)$	*abe acf bcg adh bcdi abcdj*		√	√	√	
$(10, 1)$	*ABCE ABDF ACDG BCDH ABIa ACJa*	√	√	√		√
$(1, 11)$	*abe acf bcg adh bdi acdj bcdk*		√	√	√	
$(11, 1)$	*ABCE ABDF ACDG BCDH ABIa ACJa ADKa*	√	√	√		√
$(1, 12)$	*abe acf bcg adh bdi acdj bcdk abcdl*		√	√	√	
$(12, 1)$	*ABCF ABDG ACDH BCDI ABEJ ACEK BCEL ADEa*	√	√	√		√
$(1, 13)$	*abe acf bcg abch adi bdj abdk cdl acdm*		√	√	√	
$(13, 1)$	*ABEa ACFa BCGa ABCH ADIa BDJa ABDK CDLa ACDM ACDM*	√	√	√		√
$(1, 14)$	*abe acf bcg abch adi bdj abdk cdl acdm bcdn*		√	√	√	
$(14, 1)$	*ABEa ACFa BCGa ABCH ADIa BDJa ABDK CDLa ACDM BCDN*	√	√	√		√
$(1, 15)$	*abe acf bcg abch adi bdj abdk cdl acdm bcdn abcdo*		√	√	√	
$(15, 1)$	*ABEa ACFa BCGa ABCH ADIa BDJa ABDK CDLa ACDM BCDN ABCDOa*	√	√	√		√

Table 5: 64-Run Optimal PSAs with $(l_c, l_n) \in \mathcal{S}(2^k)$

Design	Generators	W_s	(W_{sm}, W_{sn})	W_{ss}	COA	MA
(1, 6)	Aabcdef	✓			✓	✓
(2, 5)	ABabcde	✓		✓	✓	✓
(3, 4)	ABCabcd	✓		✓	✓	✓
(4, 3)	ABCDabc	✓		✓	✓	✓
(5, 2)	ABCDEab	✓	✓	✓		✓
(6, 1)	ABCDEFa	✓	✓	✓		✓
(1, 7)	Aabcf abdeg	✓			✓	✓
(2, 6)	Aabce Babdf	✓			✓	✓
(3, 5)	ABabd Cabce	✓			✓	✓
(4, 4)	Aabcd ABCDa	✓				✓
(5, 3)	ABabc ABCDE	✓		✓	✓	✓
(6, 2)	ABCEa ABDFb	✓	✓	✓		✓
(7, 1)	ABCFa ABDEG	✓	✓	✓		✓
(1, 8)	abcf Aabdg acdeh	✓			✓	✓
(2, 7)	abce Aabdf Bacdg	✓			✓	✓
(3, 6)	abce ACabd Bacdf	✓			✓	✓
(4, 5)	abce ACabd ABCDbc	✓		✓		✓
(5, 4)	abcd ABDab ACEac	✓				✓
(6, 3)	ABCE ABabc ACDFa	✓				✓
(6, 3)	abc ABCEa ABDFb		✓	✓		
(7, 2)	ABCE ABDFa ACDGb	✓	✓	✓		✓
(8, 1)	ABCF ABDGa ACDEH	✓	✓	✓		✓
(1, 9)	abcf Aabdg Aabeh acdei	✓			✓	✓
(2, 8)	abcf Aabdg Aabeh Bacde	✓			✓	✓
(3, 7)	abcf abdg Bacde ACabe	✓			✓	
(4, 6)	abce abdf ACacd BDbcd	✓		✓		
(5, 5)	abce ACabd BDabd ABEac	✓				✓
(5, 5)	abd ace ABDbc ACEabc		✓	✓		
(6, 4)	abcd ABDab ACEab BCFac	✓				✓
(7, 3)	ABCE ABDF ACDGa ABabc	✓				
(7, 3)	abc ABCE ABDFa ACDGb		✓	✓		
(8, 2)	ABCF ABDGa ABEHa ACDEb	✓	✓	✓		✓
(9, 1)	ABCF ABDGa ABEHa ACDEI	✓	✓	✓		✓
(1, 10)	abcg abdh acdei acdfj Aabef	✓			✓	✓
(2, 9)	abcf abdg abeh acdei ABbcde	✓		✓	✓	
(3, 8)	abcf abdg abeh Bacde ACbcde	✓				
(4, 7)	abce abdf ACacd BDacd ABabg	✓			✓	✓
(4, 7)	abe acf adg ACbcd BDabcd			✓		
(5, 6)	abce abdf ACacd BDacd ABEab	✓		✓		
(6, 5)	ABCE ABDF ACDac ACDbd ABabe	✓				✓
(6, 5)	abd ace ABDbc ACEbc BCFabc		✓	✓		
(7, 4)	ABCE ABDF ACDac ACDbd ABGab	✓				✓
(8, 3)	ABCF ABDG ABEH ACDEb BCDEac	✓				
(8, 3)	abc ABCE ABDFa ACDGb BCDHab		✓	✓		
(9, 2)	ABCF ABDG ABEH ACDEI BCDEab	✓	✓	✓		
(10, 1)	ABCG ABDH ACDEI ACDFJ ABEFa	✓	✓	✓		✓
(1, 11)	abcg abdh acei adfj aefk Aabcdef	✓			✓	
(2, 10)	abcf abdg aceh adei abcdej ABacd	✓			✓	
(2, 10)	abcf abdg acdh bcdi abej ABace			✓	✓	
(3, 9)	abe acf bcg adh bcdi ABCabcd			✓		
(3, 9)	abe acf bcg abch adi ABCabd		✓			
(5, 7)	ABCE ABDc ACDad ACDbe ABabf BCDabg	✓				✓
(5, 7)	abe acf adg ACbcd BDabcd ABEb			✓		
(5, 7)	abd ace bcf abcg ABDa ACEb		✓			
(6, 6)	abce abdf ACacd BDacd ABEab ABFbcd	✓		✓		✓
(7, 5)	abd ace ABDbc ACEbc BCFbc ABCGa		✓	✓		
(9, 3)	ABCG ABDH ABEI BCDEa ACFb DEFc	✓				
(9, 3)	abc ABEa ACFa ADGb BCDHa ABCDIab		✓	✓		
(10, 2)	ABCF ABDG ACEH ADEI ABCDEJ ACDab	✓	✓	✓		
(11, 1)	ABCG ABDH ACEI ADFJ AEFK ABCDEFa	✓	✓	✓		
(1, 12)	abcg abdh abei Abcde acfj defk acdefl	✓			✓	
(2, 11)	abcf abdg acdh abei acej adek ABbcd	✓			✓	
(3, 10)	abe acf bcg adh bdi acdj ABCbcd			✓		
(3, 10)	abe acf bcg abch adi bdj ABCcd		✓			
(6, 7)	ABDa ACEa BCFa ABCbd ABCce abcf ABCabcg	✓				
(7, 6)	abd ace bcf ABDabc ACEabc BCFabc ABCG			✓		
(10, 3)	ABCG ABDH ACEI ADEa ABFJ BCDFb CEFc	✓				
(10, 3)	abc ABEa ACFa BCGb ADHb BCDIa ABCDJab		✓	✓		
(11, 2)	ABCF ABDG ACDH ABEI ACEJ ADEK BCDab	✓	✓	✓		
(12, 1)	ABCG ABDH ABEI BCDEa ACFJ DEFK ACDEFL	✓	✓	✓		
(1, 13)	abcg abdh acei adej abfk Abcdf cefl defm	✓			✓	
(2, 12)	abcf abdg acdh bcdi abej acek adel ABbce	✓			✓	
(3, 11)	abe acf bcg adh bdi acdj bcdk ABCabcd			✓		
(11, 3)	ABCF ABDG ACEH ADEI ABCDEb ABJa ACDac AEKa	✓				
(11, 3)	abc ABEa ACFa BCGb ADHa BDIb ACDJb BCDKa		✓	✓		
(12, 2)	ABCF ABDG ACDH BCDI ABEJ ACEK ADEL BCEab	✓	✓	✓		
(13, 1)	ABCG ABDH ACEI ADEJ ABFK BCDFa CEFL DEFM	✓	✓	✓		
(1, 14)	abcg abdh acei adej abcdek abfl Aacdf aefm abcefn	✓			✓	
(2, 13)	abcf abdg acdh bcdi abej acek bcel adem ABbde	✓		✓	✓	
(3, 12)	abe acf bcg abch adi bdj abdk cdl ABCacd		✓	✓		
(12, 3)	ABCG ABDH ACDI BCDJ ABEK ACEa ABFL ADFb ABCDEFc	✓				
(12, 3)	abc ABEa ACFa BCGb ADHa BDIb ACDJab BCDKa ABCDL		✓	✓		
(13, 2)	ABCF ABDG ACDH BCDI ABEJ ACEK BCEL ADEM BDEab	✓	✓	✓		
(14, 1)	ABCG ABDH ACEI ADEJ ABCDEK ABFL ACDFa AEFM ABCEFN	✓	✓	✓		
(1, 15)	abcg abdh acei adej abcdek abfl Aacdf aefm abcefn abdefo	✓			✓	
(2, 14)	abcf abdg acdh bcdi abej acek bcel adem bden ABcde	✓		✓	✓	
(3, 13)	abe acf bcg abch adi bdj abdk cdl acdm ABCbcd		✓	✓		
(13, 3)	ABCF ABDG ACDH BCDI ABEJ ACEK ADEb ABLa ACMa ABCDEac	✓				
(13, 3)	abc ABEa ACFa BCGa ABCHb ADIa BDJa ABDKb CDLb ACDM		✓	✓		
(14, 2)	ABCF ABDG ACDH BCDI ABEJ ACEK BCEL ADEM BDEN CDEab	✓	✓	✓		
(15, 1)	ABCG ABDH ACEI ADEJ ABCDEK ABFL ACDFa AEFM ABCEFN ABDEFO	✓	✓	✓		

factors (A, B, C, D) and five noise factors (a, b, c, d, e) in a parameter design experiment. Because $(4,5) \notin \mathcal{S}(2^5)$, there does not exist a 32-run COA with $l_c = 4$ and $l_n = 5$. The optimal 64-run COA d' can be found in Table 2, which is generated by $ABCD$, $ABabd$ and $ACace$. The experimenter may also consider if the optimal 64-run PSA can be better. Table 5 lists the optimal 64-run PSA d'' with $l_c = 4$ and $l_n = 5$, which is generated by $abce$, $ACabd$ and $ABCDbc$. If the crossing structure is not crucial in experiment and analysis, it appears that d'' is a better choice than d' because d'' guarantees the clear estimation of the control-by-control 2f.i.'s. If the experimenter cannot afford to run a 64-run design, he/she may consider using 32-run ESAs. The optimal 32-run ESA d''' with $l_c = 4$ and $l_n = 5$ can be found in Table 3, which is generated by $ABac$, $ABbd$, $Aabe$, $BCDab$. Due to limited capacity, d''' does not guarantee the clear estimation of all the important effects.

6 APPENDIX

Proof of Proposition 1: *Sufficiency.* Because $(l_c, l_n) \in \mathcal{S}(2^k)$, $\lceil \log_2(l_c+1) \rceil + \lceil \log_2(l_n+1) \rceil \leq k$. Let $k_c = \lceil \log_2(l_c+1) \rceil$ and $k_n = \lceil \log_2(l_n+1) \rceil$. It is clear that $l_c \leq 2^{k_c} - 1$ and $l_n \leq 2^{k_n} - 1$. Hence two fractional factorial designs, denoted by d_1 and d_2, can be constructed for the control factors and the noise factors, respectively. Crossing d_1 and d_2 gives an COA with run size $2^{k_c} \times 2^{k_n} \leq 2^k$. *Necessity.* Suppose d is a COA with l_c control factors, l_n noise factors and 2^k runs. Let N_c and N_c be the run sizes of the control array and the noise arrays. $N_c N_n = 2^k$. Because both N_c and N_n are powers of 2, there exist k_c and k_n such that $k_c + k_n = k$, $N_c = 2^{k_c}$ and $N_n = 2^{k_n}$. According to the definition, $t_c \geq \min(l_c, 2)$ and $t_n \geq \min(l_n, 2)$. Hence the control array and the noise arrays are either full factorial designs or have resolutions higher than three. Hence, one has $l_c \leq 2^{k_c} - 1$ and $l_n \leq 2^{k_n} - 1$, which implies that $(l_c, l_n) \in \mathcal{S}(2^k)$.

7 ACKNOWLEDGMENTS

Yu Zhu's research was supported by National Science Foundation grant DMS-0405694. The authors want to thank the editor, the associate editor and the referees for constructive comments and suggestions that helped improve an early version of the paper.

8 REFERENCES

Bingham, D. and Sitter, R. R. (2003), "Fractional Factorial Split-Plot Designs for Robust Parameter Experiments," *Technometrics*, 45, 80-89.

Box, G. E. P., and Meyer, R. D. (1986), "Dispersion Effects from Fractional Designs (Corr: V29 p250)", *Technometrics*, 28, 19-27.

Easterling, R. G. (1985), "Discussion of the Paper by Karkar (1985)," *Journal of Quality Technology*, 17, 191-192.

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

Hedayat, A. S. and Stufken, J. (1999), "Compound Orthogonal Arrays," *Technometrics*, 41, 57-61.

Karkar, R. N. (1985), "Off-line Quality Control, Parameter Design, and the Taguchi method (with discussion)," *Journal of Quality Technology*, 17, 176-209.

Lucus, J. M. (1989), "Achieving a Robust Process Using Response Surface Methodology," in Proceedings of the Sesquicentennial Invited Sessions, American Statistical Association, pp. 579-593.

Nair, V. N. (1992), "Taguchi's Parameter Design: A Panel Discussion," *Technometrics*, 34, 127-161.

Rao, C. R. (1947), "Factorial Experiments Derivable From Combinatorial Arrangements of Arrays," *Journal of the Royal Statistical Society*, Supplement, 9, 128-139.

Rosenbaum, P. R. (1994), "Dispersion Effects From Fractional Factorials in Taguchi's Method of Quality Design," *Journal of the Royal Statistical Society, Ser. B*, 56, 641-652.

Rosenbaum, P. R. (1996), "Some Useful Compound Dispersion Experiments in Quality Design," *Technometrics*, 38, 354-364.

Rosenbaum, P. R. (1999), "Blocking in Compound Dispersion Experiments," *Technometrics*, 41, 125-134.

Shoemaker, A. C., Tsui, K. L., and Wu, C. F. J. (1991), "Economical Experimentation Methods for Robust Design," *Technometrics*, 33, 415-427.

Steinberg, D. (1996), "Robust Design: Experiments for Improving Quality," in *Handbook of Statistics 13: Design and Analysis of Experiments*, eds. S. Ghosh and C. R. Rao, New York: North-Holland, pp. 199-240.

Steinberg, D. M. and Bursztyn, D. (1998), "Noise Factors, Dispersion Effects and Robust Design," *Statistica Sinica*, 8, 67-85.

Taguchi, G. (1986), *Introduction to Quality Engineering: Designing Quality into Products and Process*, Tokyo: Asian Productivity Organization.

Vining, G. G., and Myers, R. H. (1990), "Combining Taguchi and Response Surface Philosophies: A Dual Response Approach, " *Journal of Quality Technology*, 22, 38-45.

Welch, W. J., Yu, T. K., Kang, S. M., and Sacks, J. (1990), "Computer Experiments for Quality Control by Parameter Design," *Journal of Quality Technology*, 22, 15-22.

Wu, C. F. J. and Hamada, M. S. (2000), *Experiments: Planning, Analysis and Parameter Design Optimization*, New York: John Wiley.

Wu, C. F. J. and Zhu, Y. (2003), "Optimal Selection of Single Arrays for Parameter Design Experiments", *Statistica Sinica*, 13, 1179-1199.

Wu, C. F. J. and Chen, Y. Y. (1992), "A Graph-aided Method for Planning Two-Level Experiments when Certain Interactions are Important", *Technometrics*, 34, 162-175.

Zhu, Y. (2003), "Structure Function for Aliasing Patterns in 2^{l-n} Design with Multiple Groups of Factors," *Annals of Statistics*, 31, 995-1011.