The Canadian Journal of Statistics
Vol. 47, No. 2, 2019, Pages 302–314
La revue canadienne de statistique

Design selection for strong orthogonal arrays

Chenlu SHI and Boxin TANG*

Department of Statistics and Actuarial Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

Key words and phrases: Complementary design; computer experiment; Latin hypercube; second order saturated design; space-filling design.

MSC 2010: Primary 62K15; secondary 05B15.

Abstract: Strong orthogonal arrays (SOAs) were recently introduced and studied as a class of space-filling designs for computer experiments. An important problem that has not been addressed in the literature is that of design selection for such arrays. In this article, we conduct a systematic investigation into this problem, and we focus on the most useful SOA$(n, m, 4, 2+)$s and SOA$(n, m, 4, 2)$s. This article first addresses the problem of design selection for SOAs of strength 2+ by examining their three-dimensional projections. Both theoretical and computational results are presented. When SOAs of strength 2+ do not exist, we formulate a general framework for the selection of SOAs of strength 2 by looking at their two-dimensional projections. The approach is fruitful, as it is applicable when SOAs of strength 2+ do not exist and it gives rise to them when they do. *The Canadian Journal of Statistics* 47: 302–314; 2019 © 2019 Statistical Society of Canada

Résumé: Les tableaux fortement orthogonaux (TFO) ont récemment été suggérés comme classe de plans d'expérience comblant l'espace dans le cadre d'expérimentation par ordinateur. La planification des expériences pour ces structures constitue un important problème qui n'a toutefois pas encore été résolu dans la littérature. Les auteurs procèdent à une analyse systématique de ce problème en s'attardant particulièrement aux TFO$(n, m, 4, 2+)$ et TFO$(n, m, 4, 2)$ les plus utiles. Les auteurs abordent d'abord la question du choix d'un plan pour les TFO de force 2+ en examinant leurs projections en trois dimensions. Ils présentent des résultats théoriques et calculatoires. Lorsque des TFO de force 2+ n'existent pas, les auteurs formulent un cadre général pour le choix de TFO de force 2 en observant leurs projections en deux dimensions. Cette approche fort utile est applicable lorsque les TFO de force 2+ n'existent pas, tout en les révélant lorsqu'ils existent. *La revue canadienne de statistique* 47: 302–314; 2019 © 2019 Société statistique du Canada

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 & Notz (2003) and Fang, Li & Sudjianto (2006) for some detailed discussion of the ideas and methods.

We concentrate in this article on the construction of space-filling designs based on orthogonal arrays. This approach started in McKay, Beckman & Conover (1979) when they introduced Latin hypercube designs, which are orthogonal arrays of strength 1, and continued with the work of Owen (1992) and Tang (1993), who proposed the use of orthogonal arrays to construct

* *Author to whom correspondence may be addressed.*
E-mail: boxint@sfu.ca

space-filling designs. Two decades later, inspired by (t, m, s)-nets from quasi-Monte Carlo (Niederreiter, 1992), He & 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 & 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 & Tang (2018) to the introduction of SOAs of strength 2+.

Despite these theoretical developments, an important problem that has not been addressed in the literature is that of design selection for SOAs. In this article, 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 two-dimensional space-filling properties as but can accommodate many more factors than SOAs of strength 3 for given run sizes. He, Cheng & 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 the 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 three-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 & 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 two-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.

The article is organized as follows. Section 2 introduces notation and provides some background. Section 3 examines the selection of SOAs of strength 2+ while Section 4 deals with the selection of SOAs of strength 2. We conclude the article with a discussion in Section 5.

2. NOTATION AND BACKGROUND

This section introduces some necessary notation and reviews background material to prepare for the rest of the article. 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 a property that all possible combinations of levels occur with the same frequency. We denote such an array by OA$(n, m, s_1 \times \cdots \times s_m, t)$ in this article. When $s_1 = \cdots = s_m = s$, this array is symmetric and simply denoted by OA(n, m, s, t), otherwise it is asymmetric. We refer to Dey & Mukerjee (1999), Hedayat, Sloane & 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 $\left\{0, 1, \ldots, s^t - 1\right\}$ in which 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 $\lfloor a/s^{t-u_j} \rfloor$ for $a = 0, 1, \ldots, s^t - 1$ (He & Tang, 2013). Such an array is denoted by SOA(n, m, s^t, t).

By the definition of SOAs, an SOA$(n, m, s^2, 2)$ is an orthogonal array of strength 1 in itself and becomes an orthogonal array of strength 2 if its s^2 levels are collapsed into s levels using $\lfloor a/s \rfloor$. This implies that an SOA$(n, m, s^2, 2)$ achieves a stratification on an $s \times s$ grid in any two-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 two-dimensions and $s \times s \times s$ grids in three-dimensions. While achieving stratifications on $s \times s \times s$ grids in three-dimensions, an OA$(n, m, s, 3)$ only promises stratifications on $s \times s$ grids in two-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 two-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 & 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 & 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 & Mee (2003) defined 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

$$D = A + B/2 + 3/2, \tag{1}$$

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

Lemma 2. *If an SOA of strength 2+ is to be constructed through (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 two-dimensional properties, their three-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$(n, m, 4, 2+)$.

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 (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 three-dimensions. For A to have resolution IV or higher while keeping $\bar{A}$ to be SOS, we must have that $m \le n/2 - 1$. This is actually the range of m values for which SOAs of strength 3 can be constructed (He & Tang, 2013). When $m \ge n/2$, SOAs of strength 3 do not exist but SOAs of strength 2+ can be constructed. This is the case we focus on in this section.

Let $m \ge n/2$ 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 (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 $2 \times 2 \times 2$ grids, and is said to be optimal for convenience of referencing.

According to Chen & Hedayat (1996) and Tang & 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 (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 consideration to SOS designs, design C with the maximum $W_3(C)$ has been constructed in Chen & 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 a case where an optimization problem with certain constraints is often more difficult than one without constraints. Nonetheless, we are able to establish one theoretical result, to be discussed below.

He, Cheng & 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 $1 \le k_1 \le k_2 \le k$, and $k_1 + k_2 = k$. Then the first construction of SOS designs in He, Cheng & 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

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

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 (2). This completes the proof. ■

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 catalogues in Chen, Sun & 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 & Wu (1993) only obtained a complete catalogue of resolution IV designs, which is not useful for our purpose. Hongquan Xu kindly generated for us a complete catalogue 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 one of its factors is removed (Cheng, He & Tang, 2019). 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 a computer search, Davydov, Marcugini & 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 catalogue of minimal SOS designs makes it possible to conduct an exhaustive search of designs of 64 runs, at least in principle.

We have conducted an exhaustive 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 1. The SOS designs with the maximum W_3 are generally not unique up to isomorphism. To save space, Table 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 catalogue 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 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 & Tang (2018). Since no design with $21 \leq f \leq 31$ and $n = 64$ permits such a construction, our results in Table 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 & Wu (1993) and Xu (2009) to present our designs in Table 1 (and Table 2 in the next section). Each design is represented by a set of f columns with the k independent columns labelled as $1, 2, 4, \ldots, 2^{k-1}$. We omit the independent columns in Table 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 $W = (W_3, W_4)$.

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

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

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 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 have the largest W_3 value. Clearly this corresponds to $k = 4$ and $f = 6$. From Table 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 $A = (a_1, a_2, \dots, a_9)$. We then need to find $B = (b_1, b_2, \dots, b_9)$. For a_j, we can take any b from C as b_j so long as $a_j b$ is in C and thus 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 (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. 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 to be 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 & Pambianco (2006) and Cheng, He & Tang (2019). 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 $A = (a_1, \dots, a_m)$ by Lemma 1, we need to find a $B = (b_1, \dots, b_m)$ 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 $B = (b_1, \dots, b_m)$. 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) 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 two-dimensional projection properties of design D in (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 (1) respectively. 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 $A = (a_1, \dots, a_m)$ 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 a given A. The maximum value of M is given by $M = (m-2)m + m_1$.*

Proof. For a given $u = 1, \dots, m_1$, 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 $u = m_1 + 1, \dots, m$, choosing b_u from A, say $b_u = a$ where $a \in A$, will cause array (a, a_u, b_u) to have strength 1 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 $u = m_1 + 1, \dots, m$. 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 & 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 (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 & 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 possible. 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, or 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).

Table 2: Designs with the maximum numbers of degrees of freedom

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

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 catalogues (Chen, Sun & Wu, 1993; Xu, 2009), and are tabulated in Table 2. Designs of 128 runs for $f \leq 13$ factors with the largest ν in Table 2 are also found using Theorem 3 from the catalogue of resolution IV 128-run designs (Xu, 2009). The resolution IV design with 13 factors in Table 2 maximizes ν, because there exists no design of 13 factors with $W_3 = 1$ and $W_4 = 0$ (Draper & Lin, 1990). The results for the other cases in Table 2 are obtained by an exhaustive 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 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 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 2 also has the largest W_3 value. This is in the same spirit as the discussion in Section 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 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 (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 also wish 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 lead 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 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 $a_v b_v = a_j$, that is, 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) to not 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.

5. DISCUSSION

This article 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 & 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 article 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 & Efthimiou (2014), Liu & Liu (2015), Lin & Kang (2016), and Xiao & Xu (2018).

ACKNOWLEDGMENTS

The research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). Chenlu Shi would like to thank Alexander A. Davydov, Stefano Marcugini and Fernanda Pambianco for making minimal 1-saturating sets available. She is also grateful to Hongquan Xu for generating a complete list of resolution III designs of 64 runs for $m \leq 17$ factors.

BIBLIOGRAPHY

Block, R. M. & Mee, R. W. (2003). Second order saturated resolution IV designs. *Journal of Statistical Theory and Applications*, 2, 96–112.

Chen, H. & Hedayat, A. S. (1996). 2^{n-1} designs with weak minimum aberration. *The Annals of Statistics*, 24, 2536–2548.

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

Cheng, C. S. (2014). *Theory of Factorial Design: Single- and Multi-Stratum Experiments*. CRC Press, Boca Raton.

Cheng, C. S., He, Y., & Tang, B. (2019). *Minimal second order saturated designs and their applications to computer experiments*. Revision under review by Statistica Sinica.

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

Dey, A. & Mukerjee, R. (1999). *Fractional Factorial Plans*. Wiley, New York.

Draper, N. R. & Lin, D. K. (1990). Connections between two-level designs of resolutions Ill* and V. *Technometrics*, 32, 283–288.

Fang, K. T., Li, R. Z., & Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Chapman & Hall, New York.

Georgiou, S. D. & Efthimiou, I. (2014). Some classes of orthogonal Latin hypercube designs. *Statistica Sinica*, 24, 101–120.

He, Y., Cheng, C. S., & Tang, B. (2018). Strong orthogonal arrays of strength two plus. *The Annals of Statistics*, 46, 457–468.

He, Y. & Tang, B. (2013). Strong orthogonal arrays and associated Latin hypercubes for computer experiments. *Biometrika*, 100, 254–260.

He, Y. & Tang, B. (2014). A characterization of strong orthogonal arrays of strength three. *The Annals of Statistics*, 42, 1347–1360.

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

Lin, C. D. & Kang, L. (2016). A general construction for space-filling Latin hypercube. *Statistica Sinica*, 26, 675–690.

Liu, H. & Liu, M. Q. (2015). Column-orthogonal strong orthogonal arrays and sliced strong orthogonal arrays. *Statistica Sinica*, 25, 1713–1734.

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

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

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

Santner, T. J., Williams, B. J., & Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

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

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

Xiao, Q. & Xu, H. (2018). Construction of maximin distance designs via level permutation and expansion. *Statistica Sinica*, 28, 1395–1414.

Xu, H. (2009). Algorithmic construction of efficient fractional factorial designs with large run sizes. *Technometrics*, 51, 262–277.

Received 05 September 2018
Accepted 16 February 2019