Journal of Complexity 31 (2015) 762–772

Contents lists available at ScienceDirect

Journal of Complexity

journal homepage: www.elsevier.com/locate/jco

Construction of second-order orthogonal sliced Latin hypercube designs

Rui-Yuan Cao, Min-Qian Liu *

LPMC and Institute of Statistics, Nankai University, Tianjin 300071, China

ARTICLE INFO

Article history:
Received 5 April 2014
Accepted 12 February 2015
Available online 26 February 2015

Keywords:
Computer experiment
Correlation
Second-order orthogonality
Space-filling

ABSTRACT

Sliced Latin hypercube designs are useful for computer experiments with qualitative and quantitative factors, model calibration, cross validation, multi-level function estimation, stochastic optimization and data pooling. Orthogonality and second-order orthogonality are crucial in identifying important inputs. Besides orthogonality, good space-filling properties are also necessary for Latin hypercube designs. In this paper, a construction method for second-order orthogonal sliced Latin hypercube designs is proposed. The constructed designs are further optimized to achieve better space-filling properties. Furthermore, the method is extended to construct nearly orthogonal sliced Latin hypercube designs. The numbers of slices and columns as well as the levels of the resulting designs are more flexible than those obtained by existing methods.

1. Introduction

Sliced Latin hypercube designs (SLHDs), first proposed by Qian [14], are able to fulfill various kinds of modeling circumstances such as computer experiments with qualitative and quantitative factors, model calibration, cross validation, multi-level function estimation, stochastic optimization and data pooling. The feature of an SLHD is that the whole design is so well organized that it can be divided into several slices which are still Latin hypercube designs (LHDs)(Mckay et al. [12]) when the levels of each slice are collapsed properly. Since the whole design and each slice are LHDs, they have

* Corresponding author.
E-mail address: mqliu@nankai.edu.cn (M.-Q. Liu).

http://dx.doi.org/10.1016/j.jco.2015.02.002
0885-064X/

the attractive marginal property that the maximum uniformity in any one-dimensional projection is achieved. Besides uniformity, orthogonality is another character favored by the experimenters since it can eliminate the disturbance of other inputs on the estimation of one input, making it easy to identify the most important inputs. When a second-order effect is potentially present in the model, we would like the estimations of the main effects not to be affected by this second-order effect. Thus a second-order orthogonal design is preferred. Literatures concerning the construction of orthogonal and nearly orthogonal LHDs are rather plenty, including Ye [22], Steinberg and Lin [15], Lin et al. [11], Bingham et al. [2], Pang et al. [13], Georgiou [3], Sun et al. [16,17], Lin et al. [10], Sun et al. [18], Georgiou and Stylianou [5], Yang and Liu [19], Ai et al. [1], Georgiou and Efthimiou [4], among others. As for SLHDs, the idea of making them possess the orthogonality or projection uniformity has come to researchers' attention. Yin et al. [23] and Yang et al. [20] constructed SLHDs via both symmetric and asymmetric (resolvable) orthogonal arrays so that the resulting designs possess an attractive low-dimensional uniformity. Yang et al. [21] constructed a series of orthogonal SLHDs and Huang et al. [8] provided another method for constructing orthogonal and nearly orthogonal SLHDs.

In this paper, we propose a new construction method for second-order orthogonal SLHDs. The proposed designs and their slices not only have zero correlations among the columns, but also possess a foldover structure and the second-order orthogonality, which is not guaranteed in Huang et al. [8]. The number of slices of the proposed designs could be any positive integer, and the levels are far more flexible than those constructed by the two existing methods on orthogonal SLHDs (i.e., Yang et al. [21]; Huang et al. [8]). And for a given number of runs of each slice, the maximum number of columns is attained by the resulting second-order orthogonal SLHDs. Apart from the orthogonality, we further suggest a strategy to optimize the designs to possess better space-filling properties.

The remainder of this paper is organized as follows. Section 2 provides some useful definitions and notation. Section 3 proposes the construction method for second-order orthogonal SLHDs including the space-filling property improving strategy. In Section 4, the construction method is extended to construct nearly orthogonal SLHDs. Section 5 contains some concluding remarks.

2. Definitions and notation

An $N \times p$ matrix is called a Latin hypercube design (LHD) consisting of N runs and p factors, when each of its columns is a uniform permutation of N equally spaced levels. Such a design is denoted by LHD(N, p) and, in this paper, we take the levels to be $-(N-1)/2, -(N-3)/2, \ldots, (N-1)/2$. If an LHD$(N, p)$ with $N = mt$ can be divided into t slices and each slice forms a smaller LHD(m, p) with levels $-(m-1)/2, -(m-3)/2, \ldots, (m-1)/2$ when collapsed according to $\lceil (i+(N+1)/2)/t \rceil - (m+1)/2$ for level i, where $\lceil a \rceil$ means the smallest integer greater than or equal to a, then this is called a sliced LHD (SLHD), denoted by SLHD(m, t, p). An LHD is said to be orthogonal if the correlation between any two distinct columns is zero. If an SLHD as a whole design is orthogonal as well as each slice of it, it is called an orthogonal SLHD.

A p-factor q-degree polynomial full model is of the following form

$$Y = \mu + \sum_{1\le i\le p} \beta_i x_i + \sum_{1\le i_1\le i_2\le p} \beta_{i_1 i_2} x_{i_1} x_{i_2} + \cdots + \sum_{1\le i_1\le\cdots\le i_q\le p} \beta_{i_1\ldots i_q} x_{i_1}\cdots x_{i_q} + \varepsilon,$$

where β_i is the linear effect of x_i, $\beta_{i_1\ldots i_l}$ is the l-order interaction of $x_{i_1}, \ldots, x_{i_l}$, specially β_{ii} represents the quadratic effect of factor x_i and $\beta_{i_1 i_2}$ represents the bilinear interaction of factors x_{i_1} and x_{i_2} for $i_1 \ne i_2$. In regression analysis, it is desirable that the variables in the model are orthogonal to each other, in which case the estimates of the regression coefficients are uncorrelated. When it comes to fitting a q-degree polynomial regression model, orthogonal LHDs can guarantee the estimates of linear effects uncorrelated to each other. While sometimes second-order effects may be present, we seek designs with the following properties:

(a) each column is orthogonal to the others in the design;
(b) the sum of the elementwise product of any three columns is zero.

We call a design satisfying these two properties a second-order orthogonal design. It is well known that if a design D has the foldover structure $D = \left(D_0', -D_0'\right)'$, where D' is the transpose of D, it naturally satisfies property (b). The LHDs constructed by Ye [22], Sun et al. [16,17], and Yang and Liu [19], for instance, possess these two properties. For a second-order orthogonal SLHD, it means each slice should be second-order orthogonal, simultaneously making the whole design second-order orthogonal.

3. Construction of second-order orthogonal SLHDs

In this section, we propose the construction method for second-order orthogonal SLHDs. First, let us construct a new kind of orthogonal matrices which will play an important role in the construction of orthogonal SLHDs.

3.1. A new construction of orthogonal matrices

In Sun et al. [16], they mentioned a matrix

$$S_c = \begin{pmatrix} S_{c-1} & -S^*_{c-1} \\ S_{c-1} & S^*_{c-1} \end{pmatrix} \quad \text{for } c > 1, \text{ with } S_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{1}$$

where c is an integer, and for a matrix $S = (A', B')'$ with A and B having the same dimension, $S^* = (A', B')'^* = (-A', B')'$.

Now, we define a new matrix $W_c(a, b)$ as

$$W_c(a, b) = \begin{pmatrix} W_{c-1} & W_{c-1} + 2^{c-1} \cdot t \cdot J_{2^{c-1}} \\ W_{c-1} + 2^{c-1} \cdot t \cdot J_{2^{c-1}} & W_{c-1} \end{pmatrix} \quad \text{for } c = 2, 3, \ldots,$$

with

$$W_1(a, b) = \begin{pmatrix} a & b \\ b & a \end{pmatrix},$$

where a, b and t are positive integers and J_n denotes an $n \times n$ matrix with all elements unity.

Take $T_c(a, b)$ to be the elementwise product of $W_c(a, b)$ and S_c, i.e., $(T_c(a, b))_{ij} = (W_c(a, b))_{ij} \cdot (S_c)_{ij}$, denoted as

$$T_c(a, b) = W_c(a, b) \odot S_c. \tag{2}$$

Then we have the following lemma.

Lemma 1. *For the S_c defined in (1) and $T_c(a, b)$ constructed in (2), we have*

$$S'_c S_c = S^{*\prime}_c S^*_c = 2^c I_{2^c}, \tag{3}$$

$$S'_c T_c(a, b) + T'_c(a, b) S_c = h_c(a, b, t) I_{2^c}, \tag{4}$$

$$S'_c T^*_c(a, b) - T'_c(a, b) S^*_c = 0, \tag{5}$$

where I_n denotes an n-order identity matrix and $h_c(a, b, t) = 2\sum_{i=0}^{2^{c-1}-1} ((a + 2it) + (b + 2it))$.

Proof. First, (3) is from Sun et al. [16]. We now prove (4) and (5) by induction. For notational simplicity, the parameters a, b and t are omitted from $T_c(a, b)$ and $h_c(a, b, t)$. For (4), it is easy to verify that $S'_1 T_1 + T'_1 S_1 = 2(a + b)I_2$. Suppose that $S'_c T_c + T'_c S_c = h_c I_{2^c}$. If we can prove that $S'_{c+1} T_{c+1} + T'_{c+1} S_{c+1} = h_{c+1} I_{2^{c+1}}$, then (4) is verified. In fact, from (2) and the meaning of operator *,

$$\begin{aligned} S'_{c+1} T_{c+1} + T'_{c+1} S_{c+1} &= \begin{pmatrix} S'_c & S'_c \\ -S^{*\prime}_c & S^{*\prime}_c \end{pmatrix} \begin{pmatrix} W_c \odot S_c & -(W_c + 2^c \cdot t \cdot J_{2^c}) \odot S^*_c \\ (W_c + 2^c \cdot t \cdot J_{2^c}) \odot S_c & W_c \odot S^*_c \end{pmatrix} \\ &\quad + \begin{pmatrix} W'_c \odot S'_c & (W'_c + 2^c \cdot t \cdot J_{2^c}) \odot S'_c \\ -(W'_c + 2^c \cdot t \cdot J_{2^c}) \odot S^{*\prime}_c & W'_c \odot S^{*\prime}_c \end{pmatrix} \begin{pmatrix} S_c & -S^*_c \\ S_c & S^*_c \end{pmatrix} \\ &= \begin{pmatrix} A & B \\ B' & C \end{pmatrix}, \end{aligned}$$

where

$$\begin{aligned} A &= C = S_c'T_c + S_c'(T_c + 2^c \cdot t \cdot S_c) + T_c'S_c + (T_c' + 2^c \cdot t \cdot S_c')S_c \\ &= 2(S_c'T_c + T_c'S_c) + 2^{c+1} \cdot t \cdot S_c'S_c, \quad \text{and} \end{aligned}$$
$$B = -S_c'(T_c^* + 2^c \cdot t \cdot S_c^*) + S_c'T_c^* - T_c'S_c^* + (T_c' + 2^c \cdot t \cdot S_c')S_c^* = 0.$$

With some calculation, we can write

$$A = C = h_{c+1}I_{2^c}.$$

Thus $S_{c+1}'T_{c+1} + T_{c+1}'S_{c+1} = h_{c+1}I_{2^{c+1}}$, and the conclusion is true.

For (5), the proof is similar. It is obvious that $S_1'T_1^* - T_1'S_1^* = 0$. Assume that $S_c'T_c^* - T_c'S_c^* = 0$; now we prove $S_{c+1}'T_{c+1}^* - T_{c+1}'S_{c+1}^* = 0$. In fact,

$$\begin{aligned} S_{c+1}'T_{c+1}^* - T_{c+1}'S_{c+1}^* &= \begin{pmatrix} S_c' & S_c' \\ -S_c^{*\prime} & S_c^{*\prime} \end{pmatrix} \begin{pmatrix} -W_c \odot S_c & (W_c + 2^c \cdot t \cdot J_{2^c}) \odot S_c^* \\ (W_c + 2^c \cdot t \cdot J_{2^c}) \odot S_c & W_c \odot S_c^* \end{pmatrix} \\ &- \begin{pmatrix} W_c' \odot S_c' & (W_c' + 2^c \cdot t \cdot J_{2^c}) \odot S_c' \\ -(W_c' + 2^c \cdot t \cdot J_{2^c}) \odot S_c^{*\prime} & W_c' \odot S_c^{*\prime} \end{pmatrix} \begin{pmatrix} -S_c & S_c^* \\ S_c & S_c^* \end{pmatrix} \\ &= \begin{pmatrix} F & G \\ H & J \end{pmatrix}, \end{aligned}$$

where

$$\begin{aligned} &F = -S_c'T_c + S_c'(T_c + 2^c \cdot t \cdot S_c) + T_c'S_c - (T_c' + 2^c \cdot t \cdot S_c')S_c = 0, \\ &G = S_c'(T_c^* + 2^c \cdot t \cdot S_c^*) + S_c'T_c^* - T_c'S_c^* - (T_c' + 2^c \cdot t \cdot S_c')S_c^* = 2(S_c'T_c^* - T_c'S_c^*), \\ &H = S_c^{*\prime}T_c + S_c^{*\prime}(T_c + 2^c \cdot t \cdot S_c) - (T_c^{*\prime} + 2^c \cdot t \cdot S_c^{*\prime})S_c - T_c^{*\prime}S_c = 2(S_c^{*\prime}T_c - T_c^{*\prime}S_c), \quad \text{and} \\ &J = -S_c^{*\prime}(T_c^* + 2^c \cdot t \cdot S_c^*) + S_c^{*\prime}T_c^* + (T_c^{*\prime} + 2^c \cdot t \cdot S_c^{*\prime})S_c^* - T_c^{*\prime}S_c^* = 0. \end{aligned}$$

Under the assumption of $S_c'T_c^* - T_c'S_c^* = S_c^{*\prime}T_c - T_c^{*\prime}S_c = 0$, we have $G = H = 0$. Thus the proof is completed. □

With the help of this lemma, we have the following theorem.

Theorem 1. *For any positive integers a, b, c and t, the $T_c(a, b)$ constructed in (2) is orthogonal, and*

$$T_c'(a, b)T_c(a, b) = k_c(a, b, t)I_{2^c},$$

where $k_c(a, b, t) = \sum_{i=0}^{2^{c-1}-1} \left((a + 2it)^2 + (b + 2it)^2\right)$.

Proof. We prove this theorem by induction. For notational simplicity, the parameters a, b and t are omitted from $T_c(a, b)$ and $k_c(a, b, t)$. First, it is obvious that $T_1'T_1 = (a^2 + b^2)I_2$. Suppose $T_c'T_c = k_cI_{2^c}$; if we can prove $T_{c+1}'T_{c+1} = k_{c+1}I_{2^{c+1}}$, the proof is completed. By the definition of operators * and $\odot$, we can write

$$\begin{aligned} T_{c+1}'T_{c+1} &= \begin{pmatrix} W_c' \odot S_c' & (W_c' + 2^c \cdot t \cdot J_{2^c}) \odot S_c' \\ -(W_c' + 2^c \cdot t \cdot J_{2^c}) \odot S_c^{*\prime} & W_c' \odot S_c^{*\prime} \end{pmatrix} \\ &\cdot \begin{pmatrix} W_c \odot S_c & -(W_c + 2^c \cdot t \cdot J_{2^c}) \odot S_c^* \\ (W_c + 2^c \cdot t \cdot J_{2^c}) \odot S_c & W_c \odot S_c^* \end{pmatrix} \\ &= \begin{pmatrix} A & B \\ B' & C \end{pmatrix}, \end{aligned}$$

where

$$\begin{aligned} &A = T_c'T_c + (T_c' + 2^c \cdot t \cdot S_c')(T_c + 2^c \cdot t \cdot S_c), \\ &B = -2^c \cdot t \cdot T_c'S_c^* + 2^c \cdot t \cdot S_c'T_c^*, \quad \text{and} \\ &C = (T_c^{*\prime} + 2^c \cdot t \cdot S_c^{*\prime})(T_c^* + 2^c \cdot t \cdot S_c^*) + T_c^{*\prime}T_c^*. \end{aligned}$$

Sun et al. [16] proved that $A^{*\prime}B^{*} = A'B$ for any two square matrices A and B with the same even number of rows. So based on this useful fact and Lemma 1, with some calculations we have

$$\begin{aligned} A = C &= 2T_c'T_c + 2^c \cdot t \cdot (T_c'S_c + S_c'T_c) + 2^{3c} \cdot t^2 \cdot I_{2^c} \\ &= \left(\sum_{i=0}^{2^c-1} \left((a+2it)^2 + (b+2it)^2 \right) \right) I_{2^c} = k_{c+1} I_{2^c}, \quad \text{and} \\ B &= 2^c \cdot t \cdot (S_c'T_c^* - T_c'S_c^*) = 0. \end{aligned}$$

Therefore $T_{c+1}'T_{c+1} = k_{c+1}I_{2^{c+1}}$, and the proof is completed. □

3.2. An algorithm for constructing second-order orthogonal SLHDs

Suppose we intend to construct a second-order orthogonal SLHD$(2^{c+1}, t, 2^c)$, where t is the number of slices and c is a positive integer. The construction can be carried out through the following algorithm.

Algorithm 1 (*Construction of Second-order Orthogonal SLHDs*).

Step 1. Given the number of slices t, obtain two groups of integers as $g_1 = \{1, \ldots, t\}$ and $g_2 = \{t+1, \ldots, 2t\}$.

Step 2. For $i = 1, \ldots, t$, sample from g_1 and g_2 without replacement as a_i and b_i, respectively. Then we have $\{(a_1, b_1), \ldots, (a_t, b_t)\}$.

Step 3. For $i = 1, \ldots, t$, construct $T_c(a_i, b_i)$ via (2), and let

$$D_c(a_i, b_i) = \begin{pmatrix} T_c(a_i, b_i) \\ -T_c(a_i, b_i) \end{pmatrix} - \frac{1}{2} \begin{pmatrix} S_c \\ -S_c \end{pmatrix}. \tag{6}$$

Step 4. Obtain a design D_c by stacking $D_c(a_1, b_1), \ldots, D_c(a_t, b_t)$ row by row, i.e.,

$$D_c = (D_c'(a_1, b_1), \ldots, D_c'(a_t, b_t))'. \tag{7}$$

For the design just constructed, we have following theorem.

Theorem 2. *The design D_c constructed in (7) is a second-order orthogonal SLHD$(2^{c+1}, t, 2^c)$ with t slices $D_c(a_1, b_1), \ldots, D_c(a_t, b_t)$.*

Proof. From the construction method, we can see that both the proposed design D_c and its slices are LHDs. Since any of them has a foldover structure, property (b) is satisfied unconditionally. Then we only need to prove that D_c and its slices satisfy property (a). The levels of D_c and its slices are centered because of the foldover structure. They will satisfy property (a) only if we can prove that they are column orthogonal. For $i = 1, \ldots, t$, from (6),

$$\begin{aligned} D_c'(a_i, b_i)D_c(a_i, b_i) &= 2\left(T_c(a_i, b_i) - \frac{1}{2}S_c\right)' \left(T_c(a_i, b_i) - \frac{1}{2}S_c\right) \\ &= 2\left(T_c'(a_i, b_i)T_c(a_i, b_i) - \frac{1}{2}T_c'(a_i, b_i)S_c - \frac{1}{2}S_c'T_c(a_i, b_i) + \frac{1}{4}S_c'S_c\right). \end{aligned}$$

From Lemma 1 and Theorem 1, we can see that $D_c'(a_i, b_i)D_c(a_i, b_i) = d_i I_{2^c}$ for some constant d_i. Then

$$D_c'D_c = \sum_{i=1}^{t} D_c'(a_i, b_i)D_c(a_i, b_i) = \left(\sum_{i=1}^{t} d_i\right) I_{2^c}.$$

Thus they are all column orthogonal. So the whole design D_c and its slices satisfy property (a). Thus the proof is completed. □

Remark 1. (i) Given t and c, the second-order orthogonal SLHD constructed via Algorithm 1 is not unique. This is because the sampling strategy in Step 2 can result in various sampling outcomes. (ii) Since the SLHD and each slice of it have a foldover structure, the levels of the SLHD and its slices are centered.

An illustrative example for constructing a second-order orthogonal SLHD via Algorithm 1 is given below.

Example 1. For $c = 2$ and $t = 3$, we construct a second-order orthogonal SLHD(8, 3, 4) as follows. First obtain two groups as $g_1 = \{1, 2, 3\}$ and $g_2 = \{4, 5, 6\}$, and following the sampling strategy in Step 2 of Algorithm 1, we can get $\{(1, 5), (2, 6), (3, 4)\}$. Then from (6), we have

$$D_2(1,5) = \begin{pmatrix} 0.5 & 4.5 & 6.5 & 10.5 & -0.5 & -4.5 & -6.5 & -10.5 \\ 4.5 & -0.5 & 10.5 & -6.5 & -4.5 & 0.5 & -10.5 & 6.5 \\ 6.5 & -10.5 & -0.5 & 4.5 & -6.5 & 10.5 & 0.5 & -4.5 \\ 10.5 & 6.5 & -4.5 & -0.5 & -10.5 & -6.5 & 4.5 & 0.5 \end{pmatrix}',$$

$$D_2(2,6) = \begin{pmatrix} 1.5 & 5.5 & 7.5 & 11.5 & -1.5 & -5.5 & -7.5 & -11.5 \\ 5.5 & -1.5 & 11.5 & -7.5 & -5.5 & 1.5 & -11.5 & 7.5 \\ 7.5 & -11.5 & -1.5 & 5.5 & -7.5 & 11.5 & 1.5 & -5.5 \\ 11.5 & 7.5 & -5.5 & -1.5 & -11.5 & -7.5 & 5.5 & 1.5 \end{pmatrix}', \quad \text{and}$$

$$D_2(3,4) = \begin{pmatrix} 2.5 & 3.5 & 8.5 & 9.5 & -2.5 & -3.5 & -8.5 & -9.5 \\ 3.5 & -2.5 & 9.5 & -8.5 & -3.5 & 2.5 & -9.5 & 8.5 \\ 8.5 & -9.5 & -2.5 & 3.5 & -8.5 & 9.5 & 2.5 & -3.5 \\ 9.5 & 8.5 & -3.5 & -2.5 & -9.5 & -8.5 & 3.5 & 2.5 \end{pmatrix}'.$$

Stack $D_2(1, 5)$, $D_2(2, 6)$ and $D_2(3, 4)$ row by row to get a design D_2 as

$$D_2 = (D_2'(1,5), D_2'(2,6), D_2'(3,4))'.$$

In this example, $N = 24$, $m = 8$. It is easy to verify that $D_2(1, 5)$, $D_2(2, 6)$ and $D_2(3, 4)$ are all orthogonal and they are LHD(8, 4)'s when their levels are collapsed according to $\lceil (i + 25/2)/3 \rceil - 9/2$ for level i. Since each slice has a foldover structure, the whole SLHD has a foldover structure as well. Then D_2 is a second-order orthogonal SLHD(8, 3, 4).

3.3. Improving the space-filling properties of the SLHDs

Without loss of generality, take the first two columns of the proposed design D_2 in Example 1 for an illustration and consider the bivariate projection, which is displayed in the left panel of Fig. 1 where the points marked with symbols "○", "△" and "□" correspond to the three slices $D_2(1, 5)$, $D_2(2, 6)$ and $D_2(3, 4)$, respectively. From this figure, it is observed that the resulting design possesses a clustered pattern which is undesirable when good space-filling properties are preferred. To alleviate this phenomenon, Algorithm 2 is proposed by modifying the last step of Algorithm 1.

Algorithm 2 (*Modified Construction of Second-order Orthogonal SLHDs*).

Steps 1′–3′. Same as Steps 1–3 of Algorithm 1.

Step 4′. For $i = 1, \ldots, t$, reorder the columns of $D_c(a_i, b_i) = (d_1^{(i)}, \ldots, d_{2^c}^{(i)})$ to obtain a design $E_c(a_i, b_i) = (d_{\tau_1}^{(i)}, \ldots, d_{\tau_{2^c}}^{(i)})$, where $(\tau_1, \ldots, \tau_{2^c})$ is a permutation on $\{1, \ldots, 2^c\}$, and let

$$E_c = (E_c'(a_1, b_1), \ldots, E_c'(a_t, b_t))'.$$

Note that reordering the columns of each slice does not affect the orthogonality of the slice, and thus E_c is still a second-order orthogonal SLHD.

Example 2 (Example 1 *Continued*). From Algorithm 2, we can obtain a new second-order orthogonal SLHD(8, 3, 4)

$$E_2 = (E_2'(1,5), E_2'(2,6), E_2'(3,4))'$$

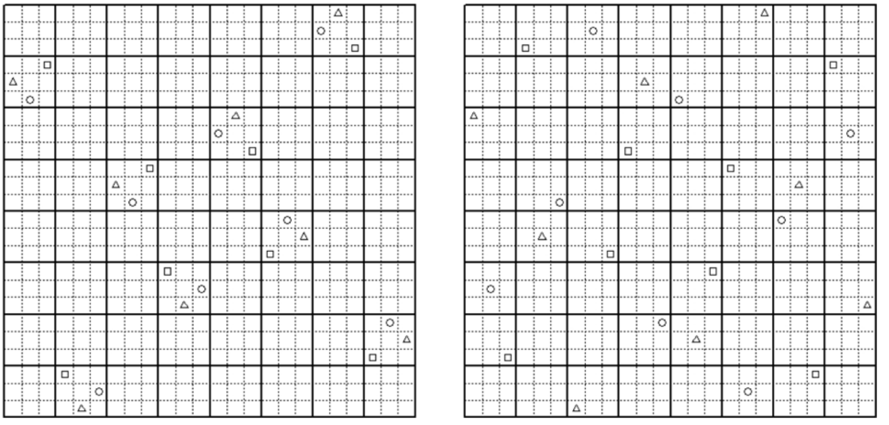

Fig. 1. Bivariate projections between the first two columns of D_2 in Example 1 (left panel) and E_2 in Example 2 (right panel).

with

$$E_2(1,5)=\begin{pmatrix} 0.5 & 4.5 & 6.5 & 10.5 & -0.5 & -4.5 & -6.5 & -10.5\\ 6.5 & -10.5 & -0.5 & 4.5 & -6.5 & 10.5 & 0.5 & -4.5\\ 10.5 & 6.5 & -4.5 & -0.5 & -10.5 & -6.5 & 4.5 & 0.5\\ 4.5 & -0.5 & 10.5 & -6.5 & -4.5 & 0.5 & -10.5 & 6.5 \end{pmatrix}',$$

$$E_2(2,6)=\begin{pmatrix} 5.5 & -1.5 & 11.5 & -7.5 & -5.5 & 1.5 & -11.5 & 7.5\\ 11.5 & 7.5 & -5.5 & -1.5 & -11.5 & -7.5 & 5.5 & 1.5\\ 7.5 & -11.5 & -1.5 & 5.5 & -7.5 & 11.5 & 1.5 & -5.5\\ 1.5 & 5.5 & 7.5 & 11.5 & -1.5 & -5.5 & -7.5 & -11.5 \end{pmatrix}', \quad \text{and}$$

$$E_2(3,4)=\begin{pmatrix} 8.5 & -9.5 & -2.5 & 3.5 & -8.5 & 9.5 & 2.5 & -3.5\\ 9.5 & 8.5 & -3.5 & -2.5 & -9.5 & -8.5 & 3.5 & 2.5\\ 3.5 & -2.5 & 9.5 & -8.5 & -3.5 & 2.5 & -9.5 & 8.5\\ 2.5 & 3.5 & 8.5 & 9.5 & -2.5 & -3.5 & -8.5 & -9.5 \end{pmatrix}'.$$

The bivariate projection of the first two columns of E_2 is displayed in the right panel of Fig. 1, where the points marked with symbols "○", "△" and "□" correspond to the three slices $E_2(1,5)$, $E_2(2,6)$ and $E_2(3,4)$, respectively. It is obvious that reordering the columns of each slice can improve the bivariate projection uniformity of the whole design, and this is also true for improving the three- or higher-dimensional projection uniformity of the whole design.

Remark 2. There are two aspects of our construction methods allowing for improving the space-filling properties of the SLHDs. As stated in Remark 1, the sampling strategy in Step 2 of Algorithm 1 gives various outcomes where we can apply some criteria for optimizing. The other aspect lies in Step 4′ of Algorithm 2. Column reordering of the slices can also result in many designs which call for an optimization strategy to suggest a better design. In order to improve the space-filling properties of the constructed second-order orthogonal SLHDs, we can adopt some optimality criteria for evaluating designs, such as the maximin or minimax distance (Johnson et al. [9]) and various measures of uniformity, among which, for example, the centered L_2-discrepancy (Hickernell [6]) and the wrap-around L_2-discrepancy (Hickernell [7]) are two popular choices.

4. Construction of nearly orthogonal SLHDs

In the previous section, we propose a method and its modification for constructing second-order orthogonal SLHDs. In some situations, orthogonal LHDs may not exist, and thus nearly orthogonal LHDs are favored by experimenters, for example, when the number of runs m of an LHD equals $4r+2$, where r is a positive integer, the LHD cannot be orthogonal (Lin et al. [10]). As for nearly orthogonal SLHDs, few construction methods have been proposed; see e.g., Huang et al. [8]. The construction algorithms in the previous section can be extended to construct nearly orthogonal SLHDs. Though the resulting designs no longer satisfy property (a), they remain satisfying property (b).

For any positive integers c and t, let $p = 2^c$ and $m = 2^{c+1} + 2$, a nearly orthogonal SLHD(m, t, p) can be constructed through the following algorithm.

Algorithm 3 (*Construction of Nearly Orthogonal SLHDs*).

Step 1. Given the number of slices t, obtain three groups of integers as $g_1 = \{1, \ldots, t\}, g_2 = \{t+1, \ldots, 2t\}$ and $g_3 = \{2t+1, \ldots, 3t\}$.

Step 2. For $i = 1, \ldots, t$, sample from g_1, g_2 and g_3 without replacement as z_i, a_i and b_i, respectively. Then we have a set of vectors $\{(z_1, a_1, b_1), \ldots, (z_t, a_t, b_t)\}$.

Step 3. For $i = 1, \ldots, t$, construct $T_c(a_i, b_i)$ via (2) and construct two row vectors $v_c(z_i) = (\pm z_i, \ldots, \pm z_i)$ and $v_{c(i)} = v_c(z_i)/z_i$ of order p, then let

$$\tilde{D}_c(z_i, a_i, b_i) = \begin{pmatrix} T_c(a_i, b_i) \\ v_c(z_i) \\ -v_c(z_i) \\ -T_c(a_i, b_i) \end{pmatrix} - \frac{1}{2}\begin{pmatrix} S_c \\ v_{c(i)} \\ -v_{c(i)} \\ -S_c \end{pmatrix}. \tag{8}$$

Step 4. For $i = 1, \ldots, t$, reorder the columns of $\tilde{D}_c(z_i, a_i, b_i) = (e_1^{(i)}, \ldots, e_{2^c}^{(i)})$ to obtain a design $\tilde{E}_c(z_i, a_i, b_i) = (e_{\tau_1}^{(i)}, \ldots, e_{\tau_{2^c}}^{(i)})$, where $(\tau_1, \ldots, \tau_{2^c})$ is a permutation on $\{1, \ldots, 2^c\}$.

Step 5. Stack $\tilde{E}_c(z_1, a_1, b_1), \ldots, \tilde{E}_c(z_t, a_t, b_t)$ row by row to get a design $\tilde{E}_c$, i.e.,

$$\tilde{E}_c = (\tilde{E}_c'(z_1, a_1, b_1), \ldots, \tilde{E}_c'(z_t, a_t, b_t))'. \tag{9}$$

Theorem 3. *The design $\tilde{E}_c$ constructed in (9) is an SLHD$(2^{c+1}+2, t, 2^c)$ with t slices $\tilde{E}_c(z_1, a_1, b_1), \ldots, \tilde{E}_c(z_t, a_t, b_t)$, and*

(i) *for $i = 1, \ldots, t$, the absolute correlation between any two distinct columns of $\tilde{E}_c(z_i, a_i, b_i)$ is $2\left(z_i - 1/2\right)^2/\alpha_i$ with*

$$\alpha_i = 2\left(z_i - \frac{1}{2}\right)^2 + \sum_{j=0}^{2^{c-1}-1}\left(a_i + 2jt - \frac{1}{2}\right)^2 + \left(b_i + 2jt - \frac{1}{2}\right)^2;$$

(ii) *$\tilde{E}_c(z_1, a_1, b_1), \ldots, \tilde{E}_c(z_t, a_t, b_t)$ and $\tilde{E}_c$ satisfy property* (b).

Proof. From the construction method, it is obvious that design $\tilde{E}_c$ and its slices are LHDs. For (i), from (8) and the proof of Theorem 2, the near orthogonality of $\tilde{E}_c(z_i, a_i, b_i)$ comes from the part

$$\begin{pmatrix} v_c(z_i) - \dfrac{1}{2}v_{c(i)} \\ -v_c(z_i) + \dfrac{1}{2}v_{c(i)} \end{pmatrix} = \begin{pmatrix} \pm\left(z_i - \dfrac{1}{2}\right) & \cdots & \pm\left(z_i - \dfrac{1}{2}\right) \\ \mp\left(z_i - \dfrac{1}{2}\right) & \cdots & \mp\left(z_i - \dfrac{1}{2}\right) \end{pmatrix}, \quad \text{for } i = 1, \ldots, t.$$

Thus (i) can be obtained straightforwardly. As for (ii), $\tilde{E}_c(z_1, a_1, b_1), \ldots, \tilde{E}_c(z_t, a_t, b_t)$ and $\tilde{E}_c$ all keep the foldover structure, so they satisfy property (b). □

An illustrative example for constructing a nearly orthogonal SLHD via Algorithm 3 is given below.

Example 3. For $c = 2$ and $t = 3$, we construct a nearly orthogonal SLHD(10, 3, 4) as follows. First obtain three groups as $g_1 = \{1, 2, 3\}$, $g_2 = \{4, 5, 6\}$ and $g_3 = \{7, 8, 9\}$, then following the sampling strategy in Step 2 of Algorithm 3, we can have $\{(1, 5, 9), (2, 4, 8), (3, 6, 7)\}$. Construct $T_2(5, 9)$, $T_2(4, 8)$ and $T_2(6, 7)$ via (2) as $T_2(a_1, b_1), T_2(a_2, b_2)$ and $T_2(a_3, b_3)$ respectively, and let $v_2(z_1) = (1, -1,$

$1, -1)$, $v_2(z_2) = (-2, 2, -2, 2)$ and $v_2(z_3) = (3, 3, -3, -3)$. Then by (8), we can get three slices $\tilde{D}_2(1, 5, 9)$, $\tilde{D}_2(2, 4, 8)$ and $\tilde{D}_2(3, 6, 7)$. Reorder the columns of these three slices independently and denote the resulting slices by $\tilde{E}_2(1, 5, 9)$, $\tilde{E}_2(2, 4, 8)$ and $\tilde{E}_2(3, 6, 7)$ respectively, we can then obtain a nearly orthogonal SLHD(10, 3, 4) denoted by $\tilde{E}_2$ via (9), i.e.,

$$\tilde{E}_2 = (\tilde{E}_2'(1, 5, 9), \tilde{E}_2'(2, 4, 8), \tilde{E}_2'(3, 6, 7))'$$

with

$$\tilde{E}_2(1, 5, 9) = \begin{pmatrix} 4.5 & 8.5 & 10.5 & 14.5 & 0.5 & -0.5 & -4.5 & -8.5 & -10.5 & -14.5 \\ 10.5 & -14.5 & -4.5 & 8.5 & 0.5 & -0.5 & -10.5 & 14.5 & 4.5 & -8.5 \\ 14.5 & 10.5 & -8.5 & -4.5 & -0.5 & 0.5 & -14.5 & -10.5 & 8.5 & 4.5 \\ 8.5 & -4.5 & 14.5 & -10.5 & -0.5 & 0.5 & -8.5 & 4.5 & -14.5 & 10.5 \end{pmatrix}',$$

$$\tilde{E}_2(2, 4, 8) = \begin{pmatrix} 7.5 & -3.5 & 13.5 & -9.5 & 1.5 & -1.5 & -7.5 & 3.5 & -13.5 & 9.5 \\ 13.5 & 9.5 & -7.5 & -3.5 & 1.5 & -1.5 & -13.5 & -9.5 & 7.5 & 3.5 \\ 9.5 & -13.5 & -3.5 & 7.5 & -1.5 & 1.5 & -9.5 & 13.5 & 3.5 & -7.5 \\ 3.5 & 7.5 & 9.5 & 13.5 & -1.5 & 1.5 & -3.5 & -7.5 & -9.5 & -13.5 \end{pmatrix}', \quad \text{and}$$

$$\tilde{E}_2(3, 6, 7) = \begin{pmatrix} 11.5 & -12.5 & -5.5 & 6.5 & -2.5 & 2.5 & -11.5 & 12.5 & 5.5 & -6.5 \\ 12.5 & 11.5 & -6.5 & -5.5 & -2.5 & 2.5 & -12.5 & -11.5 & 6.5 & 5.5 \\ 6.5 & -5.5 & 12.5 & -11.5 & 2.5 & -2.5 & -6.5 & 5.5 & -12.5 & 11.5 \\ 5.5 & 6.5 & 11.5 & 12.5 & 2.5 & -2.5 & -5.5 & -6.5 & -11.5 & -12.5 \end{pmatrix}'.$$

Remark 3. The idea behind this method is adding runs to existing orthogonal LHDs while maintaining the slice and foldover structures. If we need to add more runs or it is difficult to construct an orthogonal SLHD for some given run size, we can use a similar methodology to accomplish the task by modifying the sampling procedure in Steps 1 and 2. In general, given a run size $m = 2^{c+1}+2k$ for each slice, where k is a positive integer, in Step 1 we need $k + 2$ groups of integers, then each entry in the set of Step 2 is a row vector of order $(k + 2)$. Use the largest two elements in each of these vectors to construct a $2^c \times 2^c$ orthogonal matrix by (2), and use the remaining k elements to construct a $k \times 2^c$ matrix whose k rows are different from each other and each column has exactly these k elements up to sign changes. The remaining steps are similar to (8) and Steps 3 and 4 of Algorithm 3. It is worthy to note that only even number (i.e., $2k$) of runs can be added to the second-order orthogonal design formed by the foldover of the $2^c \times 2^c$ orthogonal matrix in a similar fashion as in (8). Similarly as discussed in Remark 2, the space-filling properties of the nearly orthogonal SLHDs can also be improved under some uniformity criteria.

5. Concluding remarks

In this paper, we propose an approach for constructing second-order orthogonal SLHDs, and a strategy for improving the space-filling properties of the constructed designs. Existing methods for constructing orthogonal SLHDs include those of Yang et al. [21] and Huang et al. [8]. Compared to the method of Yang et al. [21], which is also motivated by Sun et al. [16] as in this paper, the proposed method has no limit of the number of slices. As for the construction method of Huang et al. [8], as far as parameters p and t are concerned, the proposed method is more flexible than theirs. Moreover, the levels of the resulting orthogonal SLHDs are far more flexible than those constructed by the two existing methods. This can be seen by checking the differences between the levels of each slice. In Huang et al. [8], they mentioned a drawback of their designs, which is that the points in slices of the constructed designs always appear in the same order. That is, if their SLHDs are displayed as Fig. 1, the three symbols "○", "△" and "□" will always appear in the same order. It seems like the levels in each slice have some dependence on each other. In this paper, this phenomenon is eliminated and the symbols appear more freely. Moreover, the orthogonal SLHDs constructed in this paper satisfy property (b), i.e., they are second-order orthogonal, while in Huang et al. [8], this property is not guaranteed. Given the number of runs of each slice, the maximum number of columns is attained by the resulting second-order orthogonal SLHDs (cf., Sun et al. [16]).

We also propose a method for constructing nearly orthogonal SLHDs. Although the proposed designs no longer satisfy property (a), they still satisfy property (b) which is a merit compared to the nearly orthogonal SLHDs by Huang et al. [8].

Before ending this section, we would like to talk about the computational cost of Algorithms 1 and 2 with regard to parameters c and t. As for Algorithm 3, the idea is rather similar to that of these two algorithms and thus the discussion is omitted here. Given parameters c and t, designs with $t2^{c+1}$ runs and 2^c columns can be constructed from the two algorithms. For each of these designs, denoted by D, it has a foldover structure (up to the order of the rows) which consists of two parts where the second part equals the first part with a minus sign. In the first part of D, from (2) and (6), it can be easily seen that the computation mainly comes from the recursive construction of $W_c(a_i, b_i)$ for $i = 1, \ldots, t$, which is rather quick even for large c and t. As stated in Remarks 1 and 2, since the construction procedures involve a sampling strategy and a column reordering, the designs constructed via Algorithms 1 and 2 are not unique. There are t^2 different designs constructed from Algorithm 1 and $t^2(2^c!)^{t-1}$ different designs constructed from Algorithm 2. Algorithm 1 gives SLHDs with a clustered pattern for points from different slices, while Algorithm 2 gives designs with better space-filling properties. Furthermore, we can adopt some criterion to select a favorable design from these candidate designs. The specific optimization process is not the prime concern of this paper, since it depends on the criterion adopted to assess the designs and the optimization algorithm employed to find the solution to the objective function which may have multiple local extremum values.

Acknowledgments

The authors thank Professor Fred J. Hickernell and two anonymous referees for their valuable comments. This work was supported by the National Natural Science Foundation of China (Grant Nos. 11271205 and 11431006), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20130031110002), and the "131" Talents Program of Tianjin.

References

[1] M.Y. Ai, Y.Z. He, S.M. Liu, Some new classes of orthogonal Latin hypercube designs, J. Statist. Plann. Inference 142 (2012) 2809–2818.
[2] D. Bingham, R.R. Sitter, B. Tang, Orthogonal and nearly orthogonal designs for computer experiments, Biometrika 96 (2009) 51–65.
[3] S.D. Georgiou, Orthogonal Latin hypercube designs from generalized orthogonal designs, J. Statist. Plann. Inference 139 (2009) 1530–1540.
[4] S.D. Georgiou, E. Efthimiou, Some classes of orthogonal Latin hypercube designs, Statist. Sinica 24 (2014) 101–120.
[5] S.D. Georgiou, S. Stylianou, Block-circulant matrices for constructing optimal Latin hypercube designs, J. Statist. Plann. Inference 141 (2011) 1933–1943.
[6] F.J. Hickernell, A generalized discrepancy and quadrature error bound, Math. Comp. 67 (1998) 299–322.
[7] F.J. Hickernell, Lattice rules: how well do they measure up? in: P. Hellekalek, G. Larcher (Eds.), Random and Quasi-Random Point Sets, Springer-Verlag, New York, 1998, pp. 106–166.
[8] H. Huang, J.F. Yang, M.Q. Liu, Construction of sliced (nearly) orthogonal Latin hypercube designs, J. Complexity 30 (2014) 355–365.
[9] M.E. Johnson, L.M. Moore, D. Ylvisaker, Minimax and maximin distance designs, J. Statist. Plann. Inference 26 (1990) 131–148.
[10] C.D. Lin, D. Bingham, R.R. Sitter, B. Tang, A new and flexible method for constructing designs for computer experiments, Ann. Statist. 38 (2010) 1460–1477.
[11] C.D. Lin, R. Mukerjee, B. Tang, Construction of orthogonal and nearly orthogonal Latin hypercubes, Biometrika 96 (2009) 243–247.
[12] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245.
[13] F. Pang, M.Q. Liu, D.K.J. Lin, A construction method for orthogonal Latin hypercube designs with prime power levels, Statist. Sinica 19 (2009) 1721–1728.
[14] P.Z.G. Qian, Sliced Latin hypercube designs, J. Amer. Statist. Assoc. 107 (2012) 393–399.
[15] D.M. Steinberg, D.K.J. Lin, A construction method for orthogonal Latin hypercube designs, Biometrika 93 (2006) 279–288.

[16] F.S. Sun, M.Q. Liu, D.K.J. Lin, Construction of orthogonal Latin hypercube designs, Biometrika 96 (2009) 971–974.
[17] F.S. Sun, M.Q. Liu, D.K.J. Lin, Construction of orthogonal Latin hypercube designs with flexible run sizes, J. Statist. Plann. Inference 140 (2010) 3236–3242.
[18] F.S. Sun, F. Pang, M.Q. Liu, Construction of column-orthogonal designs for computer experiments, Sci. China Math. 54 (2011) 2683–2692.
[19] J.Y. Yang, M.Q. Liu, Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs, Statist. Sinica 22 (2012) 433–442.
[20] X. Yang, H. Chen, M.Q. Liu, Resolvable orthogonal array-based uniform sliced Latin hypercube designs, Statist. Probab. Lett. 93 (2014) 108–115.
[21] J.F. Yang, C.D. Lin, P.Z.G. Qian, D.K.J. Lin, Construction of sliced orthogonal Latin hypercube designs, Statist. Sinica 23 (2013) 1117–1130.
[22] K.Q. Ye, Orthogonal column Latin hypercubes and their applications in computer experiments, J. Amer. Statist. Assoc. 93 (1998) 1430–1439.
[23] Y. Yin, D.K.J. Lin, M.Q. Liu, Sliced Latin hypercube designs via orthogonal arrays, J. Statist. Plann. Inference 149 (2014) 162–171.