Sliced Latin hypercube designs with both branching and nested factors

Hao Chen\textsuperscript{a}, Jinyu Yang\textsuperscript{b}, Dennis K.J. Lin\textsuperscript{c}, Min-Qian Liu\textsuperscript{b,∗}

\textsuperscript{a} School of Statistics, Tianjin University of Finance and Economics, Tianjin 300222, China
\textsuperscript{b} School of Statistics and Data Science & LPMC, Nankai University, Tianjin 300071, China
\textsuperscript{c} Department of Statistics, The Pennsylvania State University, University Park, PA 16802, USA

A R T I C L E   I N F O

Article history:
Received 15 January 2018
Received in revised form 22 October 2018
Accepted 3 November 2018
Available online 16 November 2018

Keywords:
Branching and nested factor
Computer experiment
Orthogonality
Uniformity

A B S T R A C T

One special kind of sliced Latin hypercube designs (SLHDs) for computer experiments with branching and nested factors is proposed here, where not only the whole design is an SLHD, but all its slices are also SLHDs. In addition, the SLHD in the first layer has a flexible number of slices, and the slice numbers of the SLHDs in the second layer can be flexible (either the same or different). The construction method is easy to implement, and the resulting designs are orthogonal under some mild conditions. Based on the centered $L_2$-discrepancy, uniform SLHDs with branching and nested factors are further constructed.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Latin hypercube designs (LHDs) proposed by McKay et al. (1979) have been widely used for computer experiments with quantitative factors. An LHD with $n$ runs is a design in which each factor includes $n$ equally-spaced levels. Such designs uniformly spread out design points in each dimension, which makes them suitable for the complex character of computer experiments. The space-filling properties are desirable because the models for computer experiments can be highly nonlinear, and spreading all design points uniformly is likely the best option. Most times, only few important variables have dominating effects (the so-called sparsity principle). Therefore, low-dimensional projection properties are important for these designs. Besides quantitative factors, however, computer experiments can also involve qualitative factors; for example, air diffuser unit location and hot air return vent location in a computational fluid-dynamics program for studying data center thermal dynamics (Qian et al., 2008), and the force pattern in biomechanical engineering for investigating wear mechanisms of total knee replacements (Han et al., 2009). As a variant of LHDs, sliced LHDs (SLHDs) were proposed by Qian (2012) to accommodate computer experiments with both quantitative and qualitative factors. An SLHD is a special LHD that can be divided into slices, each of which is a smaller LHD and corresponds to one level-combination of the qualitative factors. So when an SLHD is collapsed onto the qualitative factors, it will obtain maximum stratification under each level-combination of the qualitative factors.

Hung et al. (2009) studied computer experiments with both quantitative and qualitative factors (with branching and nested factors), where the branching factors are qualitative, and the nested factors are quantitative. There, the factors that only exist within certain level-combinations of some other factors are called nested factors. Accordingly, a factor within which other factors are nested is called a branching factor. Take the printed circuit board manufacturing (Hung et al., 2009) as an example. The surface preparation method (i.e. the branching factor) is a qualitative factor consisting of two levels:
mechanical scrubbing and chemical treatment. Under each of the two levels, there exist different quantitative factors: pressure and micro-etch rate, respectively. Thus the pressure and micro-etch rate are nested factors. To accommodate such a situation, Hung et al. (2009) proposed branching LHDs (BLHDs), each of which includes three parts: (i) an orthogonal array (OA) for the branching factors; (ii) several LHDs for the nested factors; and (iii) an LHD for the shared factors. The shared factors are those that are common to both of the branching and nested factors.

Although the nested factors considered in Hung et al. (2009) are all quantitative, they can be qualitative as well. For example, in the printed circuit board manufacturing, if the micro-etch rate for the chemical treatment is fixed, and one wants to compare several kinds of micro-etchant, then the chemical treatment becomes a qualitative factor. In fact, such situations where the branching factors and nested factors are both qualitative are rather common (but have not been studied). In this paper, we propose two-layer SLHDs to suit such situations and provide a construction method for the optimal designs. A two-layer SLHD is a special SLHD with two layers, where the whole SLHD is the first layer and its slices (also SLHDs) are the second layer. The sliced structures in both layers are flexible and correspond to the branching factors and the nested factors, respectively. It should be pointed out that the newly constructed designs can act as the third part of BLHDs (for the shared factors). For the first two parts, we only need to find suitable OAs since the branching and nested factors are both qualitative. Most existing OAs can be found from websites http://neilsloane.com/oadir/index.html and http://support.sas.com/techsup/technote/ts723.html.

The remainder of this paper is organized as follows. Section 2 provides the construction method of the proposed designs and discusses their orthogonality. Section 3 presents the optimization algorithm to obtain uniform SLHDs. Some concluding remarks are given in Section 4. All the proofs are deferred to the Appendix.

2. Construction of SLHDs with branching and nested factors

Before presenting the construction method, we first provide some relevant definitions and notation. For any real number $r$, $[r]$ denotes the smallest integer greater than or equal to $r$, and for a matrix $M$, $\lfloor M \rfloor$ is similarly defined for its elements. For two vectors $u = (u_1, \ldots, u_n)'$ and $w = (w_1, \ldots, w_m)'$, define

$$u \oplus w = (u_1 + w_1, \ldots, u_1 + w_m, \ldots, u_n + w_1, \ldots, u_n + w_m)'.$$

And the correlation coefficient between $u$ and $w$ is defined as

$$corr(u, w) = \frac{\sum_{i=1}^{n}(u_i - \bar{u})(w_i - \bar{w})}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2 \sum_{i=1}^{n}(w_i - \bar{w})^2}}.$$

where $\bar{u} = \sum_{i=1}^{n} u_i / n$ and $\bar{w} = \sum_{i=1}^{n} w_i / n$. An LHD is said to be an orthogonal LHD (OLHD), if the correlation coefficients between any two columns are zero; while an SLHD is called a sliced OLHD (SOLHD) if the design itself and also its slices are OLHDs. For a design $D$, let $D(i, \cdot)$, $D(\cdot, j)$ and $D(i, j)$ be its $i$th row, $j$th column and the element at $i$th row and $j$th column, respectively. Denote an LHD with $n$ runs and $q$ factors by $L(n, q)$, and an SLHD with $n$ runs, $q$ factors and $s$ slices by $SL(n, q, s)$, where the $n$ levels of each factor are $1, 2, \ldots, n$.

Without loss of generality, consider only one branching factor $z$ with $s$ levels, and under each level $k$ of $z$, there are $m_k$ nested factors $v_1, \ldots, v_{m_k}$ which are supposed to have $t_k$ level-combinations, $k = 1, \ldots, s$. In addition, $q$ shared factors are involved. Then a two-layer SLHD with an $SL(n, q, s)$ in the first layer and $s$ SLHDs, $SL(n, q, t_1), \ldots, SL(n, q, t_s)$, in the second layer, denoted by $SL((N, n); q; t_1, \ldots, t_s)$, can be constructed by Algorithm 1, where $n = N/s$, and $t_k | n$ for $k = 1, \ldots, s$.

Algorithm 1.

Step 1. Construct an $L(s, q)$, denoted by $E$, by taking $q$ random permutations on $\{1, \ldots, s\}$ independently column by column.

Step 2. Construct $s$ SLHDs $SL(n, q, t_1), \ldots, SL(n, q, t_s)$ using the method in Qian (2012), denoted by $F_1, \ldots, F_s$, respectively.

Step 3. Obtain the $i$th slice in the first layer of the SLHD, denoted by $S_i$, by

$$E(i, j) \oplus (s \ast F_s(\cdot, j)) - s \ast 1,$$

for $i = 1, \ldots, s$, $j = 1, \ldots, q$, where $1$ is an $n \times 1$ vector with all elements unity, and for an integer $s$ and a vector $f = (f_1, \ldots, f_s)', s \ast f = (sf_1, \ldots, sf_s)'$.

Step 4. Stack the $s$ slices obtained in Step 3 row by row, and obtain $D = (S'_1, \ldots, S'_s)'$.

Theorem 1. For the design $D$ constructed by Algorithm 1, we have

(i) the $i$th slice $S_i$ is an SL$(n, q, t_i)$ after the levels are collapsed according to $[j/s]$ for level $j$, where $i = 1, \ldots, s$; and

(ii) $D$ is an SL$(N, q, s)$.

Theorem 1 indicates that the whole design $D$ and its slices are all SLHDs. In addition, the SLHDs in the second layer can have different numbers of slices due to the differences of the nested factors. Note that if there are $b$ ($b > 1$) branching factors $z_1, \ldots, z_b$ each having $s_1, \ldots, s_b$ levels respectively, we only need to regard them as one branching factor with $s = \prod_{i=1}^{b} s_i$ levels. An illustrative example is given as follows.
Example 1. Consider one branching factor $z$ with two levels ($\alpha$ and $\beta$), and for simplicity, further assume that there is only one nested factor $v$ with two levels ($\alpha_1$ and $\alpha_2$) under level $\alpha$ of $z$, and three levels ($\beta_1$, $\beta_2$ and $\beta_3$) under level $\beta$ of $z$. In addition, assume that two shared factors $x_1$ and $x_2$ are involved, and 12 runs are available. So in Algorithm 1, $N = 12$, $q = 2$, $s = 2$, $n = 6$, $t_1 = 2$, $t_2 = 3$. The LHD $E$ in Step 1 is supposed to be

$$E = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix},$$

and two SLHDs $F_1$ and $F_2$ in Step 2 are supposed to be

$$F_1 = \begin{pmatrix} 5 & 1 & 3 & 6 & 2 & 4 \\ 6 & 4 & 2 & 1 & 5 & 3 \end{pmatrix}, \quad \text{and} \quad F_2 = \begin{pmatrix} 3 & 4 & 1 & 5 & 2 & 6 \\ 4 & 1 & 2 & 5 & 6 & 3 \end{pmatrix}.$$  

After carrying out Step 3, we get $S_1$ and $S_2$ as follows

$$S_1 = (E(1, 1) \oplus (2 \ast F_1(:, 1))) - 2 \ast 1, \quad E(1, 2) \oplus (2 \ast F_1(:, 2)) - 2 \ast 1), \quad \text{i.e.,}$$

$$S_1 = \begin{pmatrix} 9 & 1 & 5 & 11 & 3 & 7 \\ 12 & 8 & 4 & 2 & 10 & 6 \end{pmatrix}, \quad \text{and} \quad S_2 = \begin{pmatrix} 6 & 8 & 2 & 10 & 4 & 12 \\ 7 & 1 & 3 & 11 & 9 & 5 \end{pmatrix}.$$  

Finally, we get an SL$((12, 6); 2; 2, 3)$ by letting

$$D = (S_1, S_2)' = \begin{pmatrix} 9 & 1 & 5 & 11 & 3 & 7 \\ 12 & 8 & 4 & 2 & 10 & 6 \\ 7 & 1 & 3 & 11 & 9 & 5 \end{pmatrix}.$$  

It can be easily verified that when $D$ is collapsed onto the branching factor (i.e., under the operation $[D/s]$), each of the two slices is an LHD, so $D$ gets the maximum stratification in each dimensional projection. Furthermore, when $S_1$ and $S_2$ are collapsed onto the nested factor (i.e., under the operation $[S_i/(st_i)]$ for $i = 1, 2$), they also achieve the maximum projective stratification in each dimension. Such two-layer sliced structures can also be seen intuitively by Fig. 1, in which the solid and hollow symbols represent the first and second slices in the first layer of $D$, respectively. From Fig. 1, it can be seen that there is only one point from each slice falling into one interval of $[0, 1/6)$, $[1/6, 2/6)$, $[2/6, 3/6)$, $[3/6, 4/6)$, $[4/6, 5/6)$, $[5/6, 1)$ in each dimension. For the first slice $S_1$, it can be divided into two slices, which are represented by '■' and '♦' in Fig. 1, and each slice gets maximum stratification in any $3 \times 1$ or $1 \times 3$ grid. Similarly, the second slice $S_2$ has three slices which are represented by '△', '♢' and '◦', and each slice gets maximum stratification in any $2 \times 1$ or $1 \times 2$ grid.

Remark 1. The SL$((12, 6); 2; 2, 3)$ in Example 1 is for the shared factors, and by combining this design with the two columns $z$ and $v$ in Table 1, which are for the branching and nested factors respectively, we get the design with four columns $z$, $v$, $x_1$ and $x_2$ in Table 1. This design can be used for a computer experiment with one qualitative branching, one qualitative nested factor, and two quantitative shared factors.
Orthogonality is a desirable property for SLHDs, since such a property guarantees that the estimates of all linear effects are uncorrelated with each other. Recently, Yang et al. (2013), Huang et al. (2014), Cao and Liu (2015), Yang et al. (2016), and Wang et al. (2017) proposed methods to construct sliced LHDs with orthogonality or near orthogonality. Due to the construction method of SLHDs in Algorithm 1, the obtained design inherits the orthogonality of $E$ and $F_i$’s, that is to say, if $E$ and $F_i$’s are orthogonal designs, then the final two-layer SLHD is also orthogonal. Here, a two-layer SLHD is called orthogonal if not only the whole design but also its slices are all SOLHDs. Such a good property is stated in Theorem 2.

**Theorem 2.** In Algorithm 1, if $E$ is an $OL(s, q)$, each $F_i$ is an $SOL(n, q, t_i)$ for $i = 1, \ldots, s$, and $N = ns$, then the obtained two-layer SLHD $D$ is an $SOL((N, n); q; t_1, \ldots, t_s)$. 

**Theorem 2** indicates that we can obtain orthogonal SLHDs by choosing $E$ and $F_i$’s in Algorithm 1 to be orthogonal. The orthogonality brings uncorrelated estimates for the effects of the shared factors, which is desired especially when they are used to fit a polynomial model. Moreover, the proposed designs make sure that the shared factors and the branching-by-nested interaction are orthogonal, under any level of the branching factor.

**Example 2.** Suppose 

$$E = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix},$$

which is an $OL(4, 2)$. Without loss of generality, for $i = 1, 2, 3, 4$, suppose $F_i$ is an $SOL(16, 2, 2)$, which can be obtained by randomly choosing two columns from the $SOL(16, 4, 2)$ in Example 1 of Yang et al. (2016), i.e.,

$$F_1 = \begin{pmatrix} 9 & 11 & 13 & 15 & 8 & 6 & 4 & 2 & 10 & 12 & 14 & 16 & 7 & 5 & 3 & 1 \\ 11 & 8 & 15 & 4 & 6 & 9 & 2 & 13 & 12 & 7 & 16 & 3 & 5 & 10 & 1 \end{pmatrix},$$

$$F_2 = \begin{pmatrix} 9 & 11 & 13 & 15 & 8 & 6 & 4 & 2 & 10 & 12 & 14 & 16 & 7 & 5 & 3 & 1 \\ 2 & 4 & 11 & 9 & 15 & 13 & 6 & 8 & 1 & 3 & 12 & 10 & 16 & 14 & 5 & 7 \end{pmatrix},$$

$$F_3 = \begin{pmatrix} 11 & 8 & 15 & 4 & 6 & 9 & 2 & 13 & 12 & 7 & 16 & 3 & 5 & 10 & 1 & 14 \\ 2 & 4 & 11 & 9 & 15 & 13 & 6 & 8 & 1 & 3 & 12 & 10 & 16 & 14 & 5 & 7 \end{pmatrix},$$

and

$$F_4 = \begin{pmatrix} 11 & 8 & 15 & 4 & 6 & 9 & 2 & 13 & 12 & 7 & 16 & 3 & 5 & 10 & 1 & 14 \\ 13 & 2 & 8 & 11 & 4 & 15 & 9 & 6 & 14 & 1 & 7 & 12 & 3 & 16 & 10 & 5 \end{pmatrix},$$

where the level $p$ here is corresponding to level $(2p - 17)$ in Example 1 of Yang et al. (2016). Then by Algorithm 1, an SOL((64, 16); 2; 2, 2, 2, 2) can be obtained which is listed in Table 2.

### 3. Uniform SLHDs with branching and nested factors

Another popular design property is the uniformity. As a matter of fact, the uniformity of the new design in Fig. 1 may not be ideal. In the worst case, if all $F_i$’s for $i = 1, \ldots, s$ in Algorithm 1 are the same, then the points from each slice of the first layer in the obtained design will stack together. Such a clustered structure should be avoided. Therefore, we need to find the proposed designs with better uniformity. There are several uniformity criteria, such as, the maximin distance (Johnson

**Table 1**

<table>
<thead>
<tr>
<th>Run</th>
<th>$z$</th>
<th>$v$</th>
<th>$x_1$</th>
<th>$x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\alpha$</td>
<td>$\alpha_1$</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>$\alpha$</td>
<td>$\alpha_1$</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>$\alpha$</td>
<td>$\alpha_1$</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>$\alpha$</td>
<td>$\alpha_2$</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>$\alpha$</td>
<td>$\alpha_2$</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>$\alpha$</td>
<td>$\alpha_2$</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>$\beta$</td>
<td>$\beta_1$</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>$\beta$</td>
<td>$\beta_1$</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>$\beta$</td>
<td>$\beta_2$</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>$\beta$</td>
<td>$\beta_2$</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>$\beta$</td>
<td>$\beta_3$</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>$\beta$</td>
<td>$\beta_3$</td>
<td>12</td>
<td>5</td>
</tr>
</tbody>
</table>
centered $L_2$-discrepancy ($CD_2$, Hickernell, 1998). Here, the $CD_2$ is used as the optimization criterion, as recommended by Fang et al. (2006, 2018). Other criteria, if desired, can be used as well. The $CD_2$ value of a design $D = (d_{ij})$ with $N$ runs, $q$ factors and levels in $[0, 1]$, denoted by $CD_2(D)$, can be calculated by

$$CD_2(D) = \left[ \frac{13}{12} - \frac{2}{N} \sum_{k=1}^{N} \prod_{i=1}^{q} \left( 1 + \frac{1}{2} |d_{ik} - 0.5| - \frac{1}{2} |d_{ik} - 0.5| \right) \right]^{\frac{1}{2}} \left[ \frac{1}{N^2} \sum_{k=1}^{N} \prod_{j=1}^{N} \prod_{l=1}^{q} \left( 1 + \frac{1}{2} |d_{ij} - 0.5| - \frac{1}{2} |d_{ij} - 0.5| - \frac{1}{2} |d_{ij} - d_{ij}| \right) \right].$$

Note that for a design $D = (d_{ij})$ with $N$ runs, $q$ factors and levels $1, \ldots, N$, before calculating the $CD_2$ value, its levels must be mapped into $[0, 1]$ through $(d_{ij} - 0.5)/N \to d_{ij}$ for $i = 1, \ldots, N$ and $j = 1, \ldots, q$. Let $D$ be the set containing all the $SL((N, n); q; t_1, \ldots, t_q)$'s, then the objective is to find a two-layer SLHD $D_s \in D$ such that

$$CD_2(D_s) = \min_{D \in D} CD_2(D),$$

here $D_s$ is called a uniform two-layer SLHD. The optimization steps are stated in the following Algorithm 2, which is based on the threshold accepting (TA) algorithm (Dueck and Scheuer, 1990).

Algorithm 2.

Step 1. Generate an $SL((N, n); q; t_1, \ldots, t_q)$ by Algorithm 1 as the initial design, denoted by $D_0$, and calculate $CD_2(D_0)$. Set a sequence of threshold parameter $T = (T_1, \ldots, T_L)$, where $T_1 > \cdots > T_L = 0$. Denote the maximum iteration number by $I$ under each $T_I$ for $I = 1, \ldots, L$. Set two indexes $I = 1$ and $i = 1$.

Step 2. Choose a neighbor of $D_0$, denoted as $D_i$, by the following three steps.

(a) Randomly choose one column of $D_0$ and two slices in the first layer, exchange some two elements that are equal after level-collapsing $\lfloor \cdot /5 \rfloor$;

(b) Within each $S_i$ for $i = 1, \ldots, s$, in the second layer of the selected column in (a), exchange some two elements that are equal after level-collapsing $\lfloor \cdot /\lfloor s_i \rfloor \rfloor$;

(c) Set the sequence of threshold parameter as $T = (T_1, \ldots, T_L)$ and calculate $CD_2(D_i)$. If $CD_2(D_i) < CD_2(D_0)$, then assign $D_i$ to the current $D_0$ and update the threshold parameter $T = (T_1, \ldots, T_L)$; otherwise, increase $I$ by 1 and return to Step 2.
(c) Exchange any two elements in one common slice of $S_i$ in the selected column in (a) for $i = 1, \ldots, s$.

**Step 3.** Calculate $CD_2(D_c)$. If $CD_2(D_c) - CD_2(D_0) \leq T_i$, replace $D_0$ by $D_c$; else leave $D_0$ unchanged.

**Step 4.** Update $i = i + 1$, if $i \leq I$, go to **Step 2**.

**Step 5.** Update $l = l + 1$, if $l \leq L$, reset $i = 1$ and go to **Step 2**; else set $D_{\text{best}} = D_0$.

Due to the limitation of the TA algorithm itself, we recommend that several initial designs $D_0$ should be tried in **Algorithm 2** in order to achieve a global optimal design. The three steps in **Step 2** focus on different exchanges in different parts of the design: Step (a) makes exchanges in the first layer, Step (b) in the second layer and Step (c) in the smallest slice of the design. Therefore, a better neighbor can be obtained by carrying out all three steps, but cannot by any single step alone. An illustrative example is given as follows.

**Example 3** *(Example 1 Continued)*. In this example, we optimize the two-layer SLHD $D$ in **Example 1** by **Algorithm 2**, where the parameters are set to be $T = (0.0001, 0.00009, \ldots, 0)$ and $I = 10000$. The $CD_2$ value of the initial $D_0 = D$ is 0.0540, and all its neighbors can be obtained as follows: for example, suppose the first column of $D_0$ is selected, then we can exchange elements 9 and 10 by Step (a) since $\lceil 9/2 \rceil = \lceil 10/2 \rceil$, exchange elements 5 and 7 by Step (b) since $\lceil 5/(2 \times 2) \rceil = \lceil 7/(2 \times 2) \rceil$, and exchange elements 11 and 3 by Step (c) since they belong to the same slice of $S_1$ in the second layer.

$$D_0 = \left( \frac{S_1}{S_2} \right) = \begin{pmatrix} 9 & 12 \\ 1 & 8 \\ 5 & 4 \\ 11 & 2 \\ 3 & 10 \\ 7 & 6 \\ 6 & 7 \\ 8 & 1 \\ 2 & 3 \\ 10 & 11 \\ 4 & 9 \\ 12 & 5 \end{pmatrix} \rightarrow D_c = \begin{pmatrix} 10 & 12 \\ 1 & 8 \\ 7 & 4 \\ 3 & 2 \\ 11 & 10 \\ 5 & 6 \\ 6 & 7 \\ 8 & 1 \\ 2 & 3 \\ 9 & 11 \\ 4 & 9 \\ 12 & 5 \end{pmatrix}$$

After carrying out **Algorithm 2**, the $CD_2$ value of the final design is reduced to 0.0470. The uniform two-layer SLHD is given in **Table 3**, and its scatter plot is presented in **Fig. 2**. Obviously, the design points in **Fig. 2** spread much more evenly than those in **Fig. 1**.

4. Concluding remarks

In this paper, we propose a method for constructing SLHDS with both branching and nested factors, which are a special class of SLHDS with two-layer sliced structures, and the method can be easily extended to construct SLHDS with more
layers. Orthogonality of the proposed SLHDs is considered. As in Theorem 2, we use OLHDs and SOLHDs to obtain two-layer orthogonal SLHDs. Fortunately, there are many literatures on OLHDs, such as Steinberg and Lin (2006), Lin et al. (2009, 2010), Pang et al. (2009), Sun et al. (2009, 2010), Ai et al. (2012), Yang and Liu (2012), Wang et al. (2018), among others; and on SOLHDs, including Yang et al. (2013), Huang et al. (2014), Cao and Liu (2015), Yan et al. (2016), and Wang et al. (2017). So the achievement of orthogonality of the proposed designs is rather straightforward. Moreover, based on the CD2, uniform two-layer SLHDs can be obtained by Algorithm 2.

Under re-parameter, the proposed designs may have the same structures with the designs in Guo et al. (2017) and Chen and Liu (2015). However, the construction methods and resulting designs are different (e.g., the number of the slices in the second layer are not flexible as in Chen and Liu 2015). Furthermore, Guo et al. (2017) only considered the orthogonality property, but here we consider both orthogonality and uniformity.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 11431006, 11471239, 11601367, 11771219, and 11771220), National Ten Thousand Talents Program, China, Tianjin Development Program for Innovation and Entrepreneurship, China, Tianjin “131” Talents Program, China, and Project 61331903. Dennis Lin is supported by National Security Agency, USA via Grant No. H98230-15-1-0253. The first two authors contributed equally to this work. The authors thank the Co-Editor-in-Chief and two referees for their valuable comments.

Appendix

Proof of Theorem 1.

To prove part (i), we only need to show that $S_i$ is an LHD, because it is obvious that the sliced structure of $S_i$ follows the sliced structure of $F_i$ for $i = 1, \ldots, s$. Without loss of generality, we only consider one column of $S_i$, say the $j$th column $S_i(:,j)$ for any $j = 1, \ldots, q$. Since $E(i,j)$ is any one element from $\{1, \ldots, s\}$, say $e$, and $F_i(:,j)$ for any $i = 1, \ldots, s$ is a permutation on $\{1, \ldots, n\}$, it can be easily verified that $S_i(:,j)$ is a permutation on $\{e, e+s, \ldots, e+(n-1)s\}$. This proves that $S_i$ is an LHD, and more explicitly, an $L(n, q)$, thus $S_i$ is an $SL(n, q, t_i)$.

To prove part (ii), we only need to prove that $D$ is an LHD, since $S_i$ for $i = 1, \ldots, s$ are already LHDs. Similarly, we only consider the $j$th column $D(:,j)$ for any $j = 1, \ldots, q$. It is obvious that $D(:,j)$ is a permutation on $\{S_1(:,j), \ldots, S_s(:,j)\} = \{1, 1+s, \ldots, 1+(n-1)s, s+s, \ldots, s+(n-1)s\} = \{1, \ldots, N\}$, where $N = ns$. This shows that $D$ is an LHD, thus an SLHD, and more explicitly, an $SL(N, q, s)$.

Proof of Theorem 2.

To prove that $D$ is an orthogonal SLHD with two layers, we need to show that not only $D$ but also its slices $S_i$’s for $i = 1, \ldots, s$ are all orthogonal. First, we prove that $D$ is an SOLHD. It is obvious that we only need to consider the orthogonality of any two columns of $D$, without loss of generality, say the first two columns, denoted by $d_1$ and $d_2$. The objective is to show $\text{corr}(d_1, d_2) = 0$. Assume that they are generated by the first two columns $e_1$ and $e_2$ from $E$ and the first two columns $f_1$ and $f_2$ from $F$, respectively, where $e_i = (e_{i1}, \ldots, e_{is})'$ and $f_i = (f_{i1}, \ldots, f_{is})'$ for $i = 1, 2$. Then

$$
\begin{align*}
\begin{cases}
   d_1 = (e_{11} + sf_{11} - s, \ldots, e_{11} + sf_{1n} - s, \ldots, e_{11} + sf_{11} - s, \ldots, e_{11} + sf_{1n} - s')' \\
   d_2 = (e_{12} + sf_{12} - s, \ldots, e_{12} + sf_{12} - s, \ldots, e_{12} + sf_{12} - s, \ldots, e_{12} + sf_{12} - s')' 
\end{cases}
\end{align*}
$$

Since $\text{corr}(e_1, e_2) = 0$ and $\text{corr}(f_1, f_2) = 0$, i.e.,

$$
\sum_{i=1}^{s} (e_{i1} - \bar{e}_1)(e_{i2} - \bar{e}_2) = 0,
\sum_{j=1}^{n} (f_{j1} - \bar{f}_1)(f_{j2} - \bar{f}_2) = 0,
$$

Table 3

<table>
<thead>
<tr>
<th>Run</th>
<th>$B$</th>
<th>$N$</th>
<th>$x_1$</th>
<th>$x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\alpha$</td>
<td>$\alpha_1$</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>$\alpha$</td>
<td>$\alpha_1$</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>$\alpha$</td>
<td>$\alpha_1$</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>$\alpha$</td>
<td>$\alpha_2$</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>$\alpha$</td>
<td>$\alpha_2$</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>$\alpha$</td>
<td>$\alpha_2$</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>$\beta$</td>
<td>$\beta_1$</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>$\beta$</td>
<td>$\beta_1$</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>$\beta$</td>
<td>$\beta_2$</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>$\beta$</td>
<td>$\beta_2$</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>$\beta$</td>
<td>$\beta_3$</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>$\beta$</td>
<td>$\beta_3$</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>
where \( \tilde{e}_1 = \frac{1}{s} \sum_{i=1}^{s} e_{1i} \), and \( \tilde{e}_2, \tilde{f}_1, \tilde{f}_2 \) are similarly defined, then the numerator of \( \text{corr}(d_1, d_2) \) equals
\[
\sum_{i=1}^{s} \sum_{j=1}^{n} ((e_{1i} + sf_{1j} - s) - (\tilde{e}_1 + sf_{1})((e_{2i} + sf_{2j} - s) - (\tilde{e}_2 + sf_{2} - s))
\]
\[
= \sum_{i=1}^{s} \sum_{j=1}^{n} ((e_{1i} - \tilde{e}_1) + sf_{1j} - \tilde{f}_1)((e_{2i} - \tilde{e}_2) + sf_{2j} - \tilde{f}_2)
\]
\[
= \sum_{i=1}^{s} \sum_{j=1}^{n} (e_{1i} - \tilde{e}_1)(e_{2i} - \tilde{e}_2) + s \sum_{i=1}^{s} \sum_{j=1}^{n} (e_{1i} - \tilde{e}_1)(f_{2j} - \tilde{f}_2)
\]
\[
+ s \sum_{i=1}^{s} \sum_{j=1}^{n} (f_{1j} - \tilde{f}_1)(e_{2i} - \tilde{e}_2) + s^2 \sum_{i=1}^{s} \sum_{j=1}^{n} (f_{1j} - \tilde{f}_1)(f_{2j} - \tilde{f}_2)
\]
\[
= 0.
\]
thus \( \text{corr}(d_1, d_2) = 0 \), which means that \( D \) is an SOLHD. Similarly, we can prove that each slice of \( D \) is an SOLHD. This completes the proof of Theorem 2.

References


