Statistics and Probability Letters 146 (2019) 124–131

Contents lists available at ScienceDirect

Statistics and Probability Letters

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

Sliced Latin hypercube designs with both branching and nested factors

Hao Chen [a], Jinyu Yang [b], Dennis K.J. Lin [c], Min-Qian Liu [b,*]

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

ARTICLE INFO

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

ABSTRACT

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.

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 non-linear, 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:

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

https://doi.org/10.1016/j.spl.2018.11.007
0167-7152/

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, $\lceil r \rceil$ denotes the smallest integer greater than or equal to r, and for a matrix M, $\lceil M \rceil$ 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

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

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 $D(i, :)$, $D(:, j)$ and $D(i, j)$ be its ith row, jth column and the element at ith row and jth 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 $t_k \mid n$ 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 ith slice in the first layer of the SLHD, denoted by S_i, by

$$E(i, j) \oplus (s * F_i(:, j)) - s * \mathbf{1},$$

for $i = 1, \ldots, s,\ j = 1, \ldots, q$, where $\mathbf{1}$ is an $n \times 1$ vector with all elements unity, and for an integer s and a vector $f = (f_1, \ldots, f_n)'$, $s * f = (sf_1, \ldots, sf_n)'$.

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 ith slice S_i is an $SL(n, q, t_i)$ after the levels are collapsed according to $\lceil j/s \rceil$ 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.

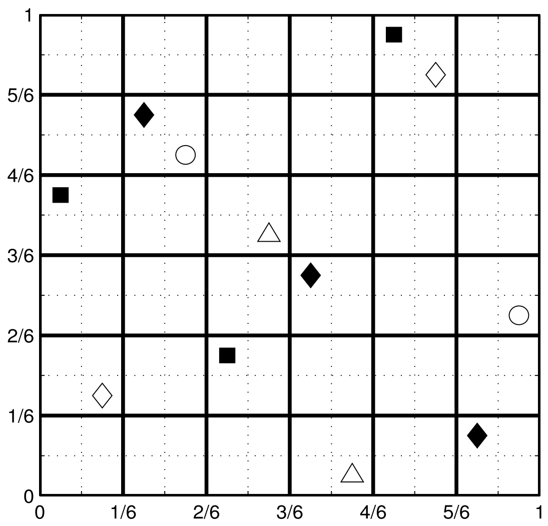

Fig. 1. Scatter plot of design points of D in Example 1.

Example 1. Consider one branching factor z with two levels (α and β), and for simplicity, further assume that there is only one nested factor v with two levels (α_1 and α_2) under level α of z, and three levels (β_1, β_2 and β_3) under level β 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

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

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

$$S_i = (E(i,1) \oplus (2 * F_i(:,1)) - 2 * \mathbf{1},\ E(i,2) \oplus (2 * F_i(:,2)) - 2 * \mathbf{1}),\ \text{i.e.,}$$

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

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

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

It can be easily verified that when D is collapsed onto the branching factor (i.e., under the operation $\lceil D/s \rceil$), 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 $\lceil S_i/(st_i) \rceil$ 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 3×1 or 1×3 grid. Similarly, the second slice S_2 has three slices which are represented by '△', '◇' and '○', and each slice gets maximum stratification in any 2×1 or 1×2 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.

Table 1
The design in Remark 1.

Run	z	v	x_1	x_2
1	α	α_1	9	12
2	α	α_1	1	8
3	α	α_1	5	4
4	α	α_2	11	2
5	α	α_2	3	10
6	α	α_2	7	6
7	β	β_1	6	7
8	β	β_1	8	1
9	β	β_2	2	3
10	β	β_2	10	11
11	β	β_3	4	9
12	β	β_3	12	5

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. For convenience, denote orthogonal $L(N, q)$, $SL(N, q, s)$ and $SL((N, n); q; t_1, \ldots, t_s)$ by $OL(N, q)$, $SOL(N, q, s)$ and $SOL((N, n); q; t_1, \ldots, t_s)$, respectively.

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.,

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

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

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

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

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 2
The obtained design in Example 2.

Run	z	v	x_1	x_2	Run	z	v	x_1	x_2
1	α	α_1	33	42	33	γ	γ_1	43	5
2	α	α_1	41	30	34	γ	γ_1	31	13
3	α	α_1	49	58	35	γ	γ_1	59	41
4	α	α_1	57	14	36	γ	γ_1	15	33
5	α	α_1	29	22	37	γ	γ_1	23	57
6	α	α_1	21	34	38	γ	γ_1	35	49
7	α	α_1	13	6	39	γ	γ_1	7	21
8	α	α_1	5	50	40	γ	γ_1	51	29
9	α	α_2	37	46	41	γ	γ_2	47	1
10	α	α_2	45	26	42	γ	γ_2	27	9
11	α	α_2	53	62	43	γ	γ_2	63	45
12	α	α_2	61	10	44	γ	γ_2	11	37
13	α	α_2	25	18	45	γ	γ_2	19	61
14	α	α_2	17	38	46	γ	γ_2	39	53
15	α	α_2	9	2	47	γ	γ_2	3	17
16	α	α_2	1	54	48	γ	γ_2	55	25
17	β	β_1	34	8	49	δ	δ_1	44	51
18	β	β_1	42	16	50	δ	δ_1	32	7
19	β	β_1	50	44	51	δ	δ_1	60	31
20	β	β_1	58	36	52	δ	δ_1	16	43
21	β	β_1	30	60	53	δ	δ_1	24	15
22	β	β_1	22	52	54	δ	δ_1	36	59
23	β	β_1	14	24	55	δ	δ_1	8	35
24	β	β_1	6	32	56	δ	δ_1	52	23
25	β	β_2	38	4	57	δ	δ_2	48	55
26	β	β_2	46	12	58	δ	δ_2	28	3
27	β	β_2	54	48	59	δ	δ_2	64	27
28	β	β_2	62	40	60	δ	δ_2	12	47
29	β	β_2	26	64	61	δ	δ_2	20	11
30	β	β_2	18	56	62	δ	δ_2	40	63
31	β	β_2	10	20	63	δ	δ_2	4	39
32	β	β_2	2	28	64	δ	δ_2	56	19

et al., 1990) and 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

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

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 $\mathcal{D}$ be the set containing all the $SL((N, n); q; t_1, \ldots, t_s)$'s, then the objective is to find a two-layer SLHD $D_* \in \mathcal{D}$ such that

$$CD_2(D_*) = \min_{D \in \mathcal{D}} CD_2(D),$$

here D_* 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 $SL((N, n); q; t_1, \ldots, t_s)$ 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_l for $l = 1, \ldots, L$. Set two indexes $l = 1$ and $i = 1$.

Step 2. Choose a neighbor of D_0, denoted as D_c, 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 $\lceil \cdot/s \rceil$;

(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 $\lceil \cdot/(st_i) \rceil$;

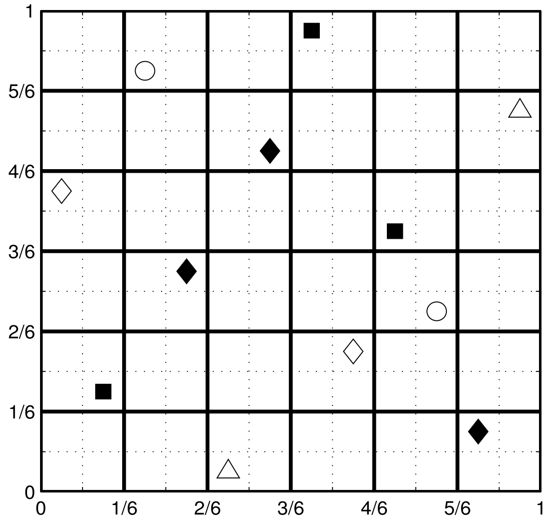

Fig. 2. Scatter plot of the obtained uniform SLHD.

(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 $CD_2(D_c) - CD_2(D_0) \le T_l$, replace D_0 by D_c; else leave D_0 unchanged.

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

Step 5. Update $l = l + 1$, if $l \le L$, 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.

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

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

Table 3
Uniform SLHD in Example 3.

Run	B	N	x_1	x_2
1	α	α_1	2	3
2	α	α_1	7	12
3	α	α_1	9	7
4	α	α_2	6	9
5	α	α_2	4	6
6	α	α_2	11	2
7	β	β_1	12	10
8	β	β_1	5	1
9	β	β_2	1	8
10	β	β_2	8	4
11	β	β_3	3	11
12	β	β_3	10	5

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), Yang et al. (2016), and Wang et al. (2017). So the achievement of orthogonality of the proposed designs is rather straightforward. Moreover, based on the CD_2, 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 jth 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 jth column $D(:, j)$ for any $j = 1, \ldots, q$. It is obvious that $D(:, j)$ is a permutation on $\{S_1(:, j), \ldots, S_s(:, j)\} = \{1, 1 + s, \ldots, 1 + (n - 1)s, \ldots, s, s + s, \ldots, s + (n - 1)s\} = \{1, \ldots, N\}$, 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 $\mathrm{corr}(d_1, d_2) = 0$. 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 $e_i = (e_{1i}, \ldots, e_{si})'$, and $f_i = (f_{1i}, \ldots, f_{ni})'$ for $i = 1, 2$. Then

$$\begin{aligned} d_1 &= (e_{11} + sf_{11} - s, \ldots, e_{11} + sf_{n1} - s, \ldots, e_{s1} + sf_{11} - s, \ldots, e_{s1} + sf_{n1} - s)', \\ d_2 &= (e_{12} + sf_{12} - s, \ldots, e_{12} + sf_{n2} - s, \ldots, e_{s2} + sf_{12} - s, \ldots, e_{s2} + sf_{n2} - s)'. \end{aligned}$$

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

$$\sum_{i=1}^{s}(e_{1i} - \bar{e}_1)(e_{2i} - \bar{e}_2) = 0, \quad \sum_{j=1}^{n}(f_{1j} - \bar{f}_1)(f_{2j} - \bar{f}_2) = 0,$$

where $\bar{e}_1 = \frac{1}{s}\sum_{i=1}^{s} e_{1i}$, and $\bar{e}_2,\ \bar{f}_1,\ \bar{f}_2$ are similarly defined, then the numerator of $\mathrm{corr}(d_1, d_2)$ equals

$$\begin{aligned}
&\sum_{i=1}^{s}\sum_{j=1}^{n}((e_{1i}+sf_{1j}-s)-(\bar{e}_1+s\bar{f}_1-s))((e_{2i}+sf_{2j}-s)-(\bar{e}_2+s\bar{f}_2-s))\\
&=\sum_{i=1}^{s}\sum_{j=1}^{n}((e_{1i}-\bar{e}_1)+s(f_{1j}-\bar{f}_1))((e_{2i}-\bar{e}_2)+s(f_{2j}-\bar{f}_2))\\
&=\sum_{i=1}^{s}\sum_{j=1}^{n}(e_{1i}-\bar{e}_1)(e_{2i}-\bar{e}_2)+s\sum_{i=1}^{s}\sum_{j=1}^{n}(e_{1i}-\bar{e}_1)(f_{2j}-\bar{f}_2)\\
&+s\sum_{i=1}^{s}\sum_{j=1}^{n}(f_{1j}-\bar{f}_1)(e_{2i}-\bar{e}_2)+s^2\sum_{i=1}^{s}\sum_{j=1}^{n}(f_{1j}-\bar{f}_1)(f_{2j}-\bar{f}_2)\\
&=0,
\end{aligned}$$

thus $\mathrm{corr}(d_1, d_2) = 0$, 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

Ai, M.Y., He, Y.Z., Liu, S.M., 2012. Some new classes of orthogonal Latin hypercube designs. J. Statist. Plann. Inference 142, 2809–2818.
Cao, R.Y., Liu, M.Q., 2015. Construction of second-order orthogonal sliced Latin hypercube designs. J. Complexity 31, 762–772.
Chen, H., Liu, M.Q., 2015. Nested Latin hypercube designs with sliced structures, Comm. Statist. Theory Methods 44, 4721–4733.
Dueck, G., Scheuer, T., 1990. Threshold accepting: A general purpose optimization algorithm superior to simulated annealing. J. Comput. Phys. 90, 161–175.
Fang, K.T., Li, R., Sudjianto, A., 2006. Design and Modeling for Computer Experiments. CRC Press, New York.
Fang, K.T., Liu, M.Q., Qin, H., Zhou, Y.D., 2018. Theory and Application of Uniform Experimental Designs. Springer, Singapore.
Guo, B., Chen, X.P., Liu, M.Q., 2017. Construction of Latin hypercube designs with nested and sliced structures. Statist. Papers, http://dx.doi.org/10.1007/s00362-017-0959-8, available online Oct. 24, 2017.
Han, G., Santner, T.J., Notz, W.I., Bartel, D.L., 2009. Prediction for computer experiments having quantitative and qualitative input variables. Technometrics 51, 278–288.
Hickernell, F.J., 1998. A generalized discrepancy and quadrature error bound. Math. Comp. 67, 299–322.
Huang, H.Z., Yang, J.F., Liu, M.Q., 2014. Construction of sliced (nearly) orthogonal Latin hypercube designs. J. Complexity 30, 355–365.
Hung, Y., Joseph, V.R., Melkote, S.N., 2009. Design and analysis of computer experiments with branching and nested factors. Technometrics 51, 354–365.
Johnson, M., Moore, L., Ylvisaker, D., 1990. Minimax and maximin distance design. J. Statist. Plann. Inference 26, 131–148.
Lin, C.D., Bingham, D., Sitter, R.R., Tang, B., 2010. A new and flexible method for constructing designs for computer experiments. Ann. Statist. 38, 1460–1477.
Lin, C.D., Mukerjee, R., Tang, B., 2009. Construction of orthogonal and nearly orthogonal Latin hypercubes. Biometrika 96, 243–247.
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.
Pang, F., Liu, M.Q., Lin, D.K.J., 2009. A construction method for orthogonal Latin hypercube designs with prime power levels. Statist. Sinica 19, 1721–1728.
Qian, P.Z.G., 2012. Sliced Latin hypercube designs. J. Amer. Statist. Assoc. 107, 393–399.
Qian, P.Z.G., Wu, H.Q., Wu, C.F.J., 2008. Gaussian process models for computer experiments with qualitative and quantitative factors. Technometrics 50, 383–396.
Steinberg, D.M., Lin, D.K.J., 2006. A construction method for orthogonal Latin hypercube designs. Biometrika 93, 279–288.
Sun, F.S., Liu, M.Q., Lin, D.K.J., 2009. Construction of orthogonal Latin hypercube designs. Biometrika 96, 971–974.
Sun, F.S., Liu, M.Q., Lin, D.K.J., 2010. Construction of orthogonal Latin hypercube designs with flexible run sizes. J. Statist. Plann. Inference 140, 3236–3242.
Wang, L., Sun, F.S., Lin, D.K.J., Liu, M.Q., 2018. Construction of orthogonal symmetric Latin hypercube designs. Statist. Sinica 28, 1503–1520.
Wang, X.L., Zhao, Y.N., Yang, J.F., Liu, M.Q., 2017. Construction of (nearly) orthogonal sliced Latin hypercube designs. Statist. Probab. Lett. 125, 174–180.
Yang, J.Y., Chen, H., Lin, D.K.J., Liu, M.Q., 2016. Construction of sliced maximin-orthogonal Latin hypercube designs. Statist. Sinica 26, 589–603.
Yang, J.F., Lin, C.D., Qian, P.Z.G., Lin, D.K.J., 2013. Construction of sliced orthogonal Latin hypercube designs. Statist. Sinica 23, 1117–1130.
Yang, J.Y., Liu, M.Q., 2012. Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs. Statist. Sinica 22, 433–442.