Statistics and Probability Letters 93 (2014) 108–115

Contents lists available at ScienceDirect

Statistics and Probability Letters

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

Resolvable orthogonal array-based uniform sliced Latin hypercube designs

Xue Yang [a], Hao Chen [b,a], Min-Qian Liu [a,*]

[a] LPMC and Institute of Statistics, Nankai University, Tianjin 300071, China

[b] Department of Statistics, Tianjin University of Finance and Economics, Tianjin 300222, China

ARTICLE INFO

Article history:
Received 28 March 2014
Received in revised form 13 June 2014
Accepted 25 June 2014
Available online 8 July 2014

Keywords:
Computer experiment
Centered L_2 discrepancy
Resolvable orthogonal array
Space-filling design
Sliced Latin hypercube design

ABSTRACT

Sliced Latin hypercube designs, introduced by Qian (2012), are widely used for computer experiments with qualitative and quantitative factors, multiple experiments, cross-validation and stochastic optimization. In this paper, we propose a new class of sliced Latin hypercube design, called the resolvable orthogonal array-based uniform sliced Latin hypercube design. Such designs are constructed via both symmetric and asymmetric resolvable orthogonal arrays, and measured by the centered L_2 discrepancy criterion. When the construction is based on a resolvable orthogonal array with strength $w+1$, the resulting design not only possesses stratification in any w-dimensional projection for each slice, but also achieves stratification in any $(w+1)$-dimensional projection for the whole design. Furthermore, the uniformity of the resulting design is also highly improved with respect to the centered L_2 discrepancy criterion.

1. Introduction

Sliced Latin hypercube designs (SLHDs) were introduced by Qian (2012) and have been regarded as a popular choice for computer experiments with both qualitative and quantitative factors. An SLHD is a Latin hypercube design (LHD) that can be divided into slices such that each of which is a smaller LHD. Orthogonality and space-filling property are two desirable features for SLHDs. The SLHDs constructed by Qian (2012) cannot guarantee orthogonality or projective uniformity in two or more dimensions although they achieve a maximum stratification in any one dimension. Recently, Yang et al. (2013) and Huang et al. (2014) proposed methods to construct sliced orthogonal and nearly orthogonal LHDs, and Yin et al. (2014) constructed SLHDs with an attractive low-dimensional uniformity via both symmetric and asymmetric orthogonal arrays. However, the uniformity of the constructed SLHDs on the whole experimental domain was not considered by any of those references.

Uniform designs have been widely used for computer experiments since such a design scatters its design points evenly on the experimental domain under some discrepancy measure (Fang and Lin, 2003; Fang et al., 2006). The commonly used discrepancy measures include, e.g., the centered L_2 discrepancy (CD_2), the wrap-around L_2 discrepancy (Hickernell, 1998) and the discrete discrepancy (Hickernell and Liu, 2002; Fang et al., 2002, 2003). In a design space $\mathcal{D}$, a design is said to be a uniform design under some discrepancy measure if it minimizes the discrepancy among all designs in $\mathcal{D}$.

Orthogonal array (OA)-based LHDs were first proposed by Tang (1993). Such designs achieve stratification in any t dimensions when an OA of strength t is used for the construction. In this paper we provide an approach for constructing

* Corresponding author.
E-mail addresses: mqliu@nankai.edu.cn, mqliu@aliyun.com (M.-Q. Liu).

http://dx.doi.org/10.1016/j.spl.2014.06.021
0167-7152/

uniform SLHDs based on symmetric and asymmetric resolvable OAs (ROAs). The resulting uniform SLHDs from ROAs with strength $w+1$ achieve stratification in $(w+1)$-dimension and each of their slices achieves stratification in w-dimension. Moreover, all designs have a better space-filling property under the CD_2 criterion. The construction methods are easy to implement and the resulting designs have flexible run sizes and number of factors.

The remainder of this paper is organized as follows. Section 2 provides a method for constructing SLHDs based on symmetric and asymmetric ROAs. In Section 3, a modified threshold-accepting (TA) algorithm is provided to search uniform SLHDs under the CD_2 criterion. Section 4 gives some concluding remarks.

2. Generation of SLHDs based on ROAs

This section provides an approach to construct SLHDs based on symmetric and asymmetric ROAs and modified sliced permutation matrices. The resulting designs have low-dimensional stratifications and flexible run sizes. Two examples are given for illustration.

Let us first introduce some useful definitions and notation. An asymmetric orthogonal array (OA), denoted by $OA(N, s_1^{k_1}s_2^{k_2}\ldots s_v^{k_v}, w+1)$, is an array of size $N\times k$, where $k=\sum_{i=1}^{v}k_i$ is the total number of factors (columns), in which the first k_1 columns have symbols from $\{0,1,\ldots,s_1-1\}$, the next k_2 columns have symbols from $\{0,1,\ldots,s_2-1\}$, and so on, with the property that in any $N\times(w+1)$ subarray every possible $(w+1)$-tuple occurs an equal number of times as a row, here $w+1$ is called the strength of this OA. An asymmetric $OA(N, s_1^{k_1}s_2^{k_2}\ldots s_v^{k_v}, w+1)$ with $\sum_{i=1}^{v}k_i=k$ is called an asymmetric resolvable OA (ROA), denoted by $ROA_p(N, s_1^{k_1}s_2^{k_2}\ldots s_v^{k_v}, w+1)$, if it can be partitioned into p submatrices each of which is an asymmetric $OA(n, s_1^{k_1}s_2^{k_2}\ldots s_v^{k_v}, w)$, where $n=N/p$. When all s_i's are equal to s, the asymmetric OA and ROA become symmetric ones, denoted by $OA(N, s^k, w+1)$ and $ROA_p(N, s^k, w+1)$, respectively.

For a real number x, $\lceil x\rceil$ denotes the smallest integer greater than or equal to x, and $\lfloor x\rfloor$ denotes the largest integer less than or equal to x. Similarly define $\lceil C\rceil$ and $\lfloor C\rfloor$ for a real matrix C. For a positive integer N, let Z_N be the set $\{1,2,\ldots,N\}$. A Latin hypercube design (LHD) of N runs and k factors, denoted by $L(N,k)$, is considered as an $N\times k$ matrix in which each column consists of a uniform permutation of Z_N and all the columns are obtained independently. Moreover, an LHD is called a sliced LHD (SLHD) with $N=np$ runs, k factors and p slices, denoted by $SL(N,k,p)$, if it can be divided into p $n\times k$ subarrays, say $D_1,\ldots,D_p$, and each $\lceil D_i/p\rceil$ is an $L(n,k)$.

Qian (2012) used the sliced permutation matrix to construct SLHDs. A sliced permutation matrix $SPM(n,p)$ on Z_N with $N=np$ and n and p being strictly positive integers is defined to be an $n\times p$ matrix in which each element of Z_N appears precisely once and each column of $\lceil SPM(n,p)/p\rceil$ forms a permutation on Z_n (cf. Qian, 2012). For the construction of ROA-based SLHDs, we need to modify the generation of the sliced permutation matrix as follows. (i) Divide the entries of Z_N into an $n\times p$ matrix $B=(b_{ij})$ with $N=np$ such that $\lceil b_{ij}/p\rceil=i$ for $i=1,\ldots,n, j=1,\ldots,p$. (ii) Split B into q $t\times p$ matrices $B_l=(b_{ij}^l)$ for $l=1,\ldots,q$, such that for any $b_{ij}^l\in B_l$, $\lceil b_{ij}^l/(t\times p)\rceil=l$, where $n=tq$ and $i=1,\ldots,t, j=1,\ldots,p$. (iii) For $l=1,\ldots,q$, let B_l^* be the matrix obtained by randomly reordering the rows and columns of B_l. (iv) Let $M(n,p,q)=({B_{\tau 1}^*}',\ldots,{B_{\tau q}^*}')'$, where $(\tau 1,\ldots,\tau q)$ is a permutation on $\{1,\ldots,q\}$. Then $M(n,p,q)$ is called the modified sliced permutation matrix, and will be used to construct ROA-based SLHDs in the following.

Algorithm 1 (*Construction of SLHDs via ROAs*).

Step 1. Let $A=(A_1',\ldots,A_p')'$ be an $ROA_p(N, s_1^{k_1}s_2^{k_2}\cdots s_v^{k_v}, w+1)$ with $N=np$ and $\sum_{i=1}^{v}k_i=k$, where A_i is an $OA(n, s_1^{k_1}s_2^{k_2}\cdots s_v^{k_v}, w)$ for $i=1,\ldots,p$.

Step 2. For $u=1,\ldots,v$, independently generate k_u $M(n,p,s_u)$'s as described above, denoted by $M_1^{s_u},\ \ldots, M_{k_u}^{s_u}$.

Step 3. Randomly replace the c's in the jth s_u-level column of A_i by the entries in the ith column of the $(c+1)$th submatrix of $M_j^{s_u},\ j=1,\ldots,k_u,\ c=0,1,\ldots,s_u-1,\ i=1,\ldots,p$, and $u=1,\ldots,v$. Then a new matrix $D=({D_1}',\ldots,{D_p}')'$ is obtained.

According to Algorithm 1, we can obtain the following result.

Theorem 1. *For the D and each D_l for $l=1,\ldots,p$ constructed in Algorithm 1,*

(i) *D is an $SL(N,k,p)$ with p slices D_l for $l=1,\ldots,p$;*

(ii) *D achieves stratification on $s_1^{i_1}\times s_2^{i_2}\times\cdots\times s_v^{i_v}$ grids for $i_u\le k_u,\ u=1,\ldots,v$, with $\sum_{u=1}^{v}i_u\le w+1$, and each D_l achieves stratification on $s_1^{j_1}\times s_2^{j_2}\times\cdots\times s_v^{j_v}$ grids, for $j_u\le k_u,\ u=1,\ldots,v$, with $\sum_{u=1}^{v}j_u\le w$, $l=1,\ldots,p$.*

Proof. (i) This conclusion is obvious according to the construction of D and each $D_l,\ l=1,\ldots,p$.

(ii) Since A is an OA with strength $w+1$ which achieves stratification on $s_1^{i_1}\times\cdots\times s_v^{i_v}$ grids for $i_u\le k_u,\ u=1,\ldots,v$, with $\sum_{u=1}^{v}i_u\le w+1$ and D is an OA-based LHD based on A (cf. Tang, 1993), then D inherits the same stratification with A, i.e. D achieves stratification on $s_1^{i_1}\times\cdots\times s_v^{i_v}$ grids for $i_u\le k_u,\ u=1,\ldots,v$, with $\sum_{u=1}^{v}i_u\le w+1$. Similarly, each A_l

is an OA with strength w, therefore, each D_l achieves stratification on $s_1^{j_1} \times \cdots \times s_v^{j_v}$ grids for $j_u \leq k_u,\ u = 1, \ldots, v$, with $\sum_{u=1}^{v} j_u \leq w$, where $l = 1, \ldots, p$. This completes the proof. □

Note that Theorem 1 is obviously true for the special case of $s_1 = \cdots = s_v = s$. It is easy to verify that an $ROA_s(N, s^k, w+1)$ always exists if there exists an $OA(N, s^{k+1}, w+1)$, where $w + 1 \leq k$. We now briefly describe a general procedure for constructing symmetric ROAs. (i) Let B be an $OA(N, s^{k+1}, w+1)$, without loss of generality, suppose the entries of B come from $\{0, 1, \ldots, s-1\}$, where $s \geq 2$. (ii) Rearrange the rows of B such that the first column contains 0 in the first N/s positions, 1 in the next N/s positions and so on. Then we obtain an orthogonal array $OA(N, s^k, w+1)$ by omitting the first column of B. Moreover, the s sets of N/s consecutive rows of the $OA(N, s^k, w+1)$ are s $OA(N/s, s^k, w)$'s, i.e., this $OA(N, s^k, w+1)$ is an $ROA_s(N, s^k, w+1)$.

Now let us see an illustrative example for Algorithm 1.

Example 1. Consider the following $ROA_4(16, 2^3, 3)$:

$$\begin{pmatrix} 0 & 0 & 1 & 1 & : & 1 & 1 & 0 & 0 & : & 0 & 0 & 1 & 1 & : & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & : & 1 & 0 & 1 & 0 & : & 0 & 1 & 0 & 1 & : & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & : & 1 & 0 & 0 & 1 & : & 0 & 1 & 1 & 0 & : & 1 & 0 & 0 & 1 \end{pmatrix}',$$

which can be partitioned into four subarrays and each subarray is an $OA(4, 2^3, 2)$.

First, following Step 2 of Algorithm 1, we generate three $M(4, 4, 2)$'s, say

$$M_1^2 = \begin{pmatrix} 11 & 14 & 15 & 13 \\ 16 & 9 & 10 & 12 \\ \hdashline 3 & 1 & 8 & 4 \\ 6 & 7 & 2 & 5 \end{pmatrix}, \quad M_2^2 = \begin{pmatrix} 13 & 12 & 15 & 11 \\ 10 & 14 & 9 & 16 \\ \hdashline 5 & 1 & 6 & 7 \\ 4 & 8 & 3 & 2 \end{pmatrix}, \quad M_3^2 = \begin{pmatrix} 1 & 3 & 4 & 2 \\ 5 & 7 & 6 & 8 \\ \hdashline 12 & 11 & 9 & 10 \\ 15 & 13 & 16 & 14 \end{pmatrix}.$$

Then following Step 3 of Algorithm 1, the resulting $SL(16, 3, 4)$ is

$$\begin{aligned} D &= (D_1', D_2', D_3', D_4')' \\ &= \begin{pmatrix} 11 & 16 & 3 & 6 & : & 1 & 7 & 9 & 14 & : & 10 & 15 & 8 & 2 & : & 4 & 5 & 12 & 13 \\ 10 & 5 & 13 & 4 & : & 8 & 14 & 1 & 12 & : & 15 & 3 & 9 & 6 & : & 7 & 16 & 2 & 11 \\ 5 & 12 & 15 & 1 & : & 13 & 7 & 3 & 11 & : & 6 & 16 & 9 & 4 & : & 10 & 8 & 2 & 14 \end{pmatrix}'. \end{aligned}$$

The bivariate projections of D are displayed in Fig. 1, where the markers "•", "○", "△" and "+" denote the points from the slices D_1, D_2, D_3 and D_4, respectively. The symbol "X1" denotes the first column of D, "X2" denotes the second column of D, and so on. From Fig. 1, it can be seen that each D_l achieves maximum stratification on 2×2 grids in any two dimensions, in addition to achieving maximum stratification in any one dimension for $l = 1, 2, 3, 4$. In fact, the design D achieves not only maximum stratification in any one dimension and stratification in any two dimensions, but also stratification on $2 \times 2 \times 2$ grids in any three dimensions.

SLHDs based on asymmetric ROAs can also be constructed using Algorithm 1. Let us first briefly describe the existence of asymmetric ROAs. Agrawal and Dey (1983) presented three types of orthogonal resolution IV designs for asymmetrical factorials. Brouwer et al. (2006) and Nguyen (2008) examined the asymmetric OAs of strength 3 with run size $n \leq 100$. Jiang and Yin (2013) provided an approach to construct asymmetric OAs of strength ≥ 3 from known ones.

Lemma 1 (*Jiang and Yin, 2013*). *An $ROA_p(N, s_1^{k_1} \cdots s_v^{k_v}, w)$ with $\sum_{i=1}^{v} k_i = k$ can exist only if $p|N$ and $n|(N/p)$ where*

$$n = \mathrm{lcm}\left\{ \prod_{i=1}^{w-1} s_{j_i} : 1 \leq j_1 < \cdots < j_{w-1} \leq v \text{ and } s_{j_1}, \ldots, s_{j_{w-1}} \in \{s_1, s_2, \ldots, s_v\} \right\},$$

and lcm{·} *denotes the least common multiple of the elements in the set.*

Lemma 2 (*Jiang and Yin, 2013*). *If there is an $OA(N, s_1^{k_1} \cdots s_v^{k_v}, w)$ with $\sum_{i=1}^{v} k_i = k$ and $w < k$, then there is also an $ROA_{s_i}(N, s_1^{k_1} \cdots s_{i-1}^{k_{i-1}} s_i^{k_i - 1} s_{i+1}^{k_{i+1}} \cdots s_v^{k_v}, w)$ for each $1 \leq i \leq v$.*

Example 2. Consider an $ROA_2(32, 4^2 2^3, 3)$ obtained by deleting one 2-level column from the $OA(32, 4^2 2^4, 3)$ constructed by Agrawal and Dey (1983), where the $OA(32, 4^2 2^4, 3)$ is

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 2 & 2 & 3 & 3 & 0 & 0 & 1 & 1 & 2 & 2 & 3 & 3 & 1 & 1 & 0 & 0 & 3 & 3 & 2 & 2 & 1 & 1 & 0 & 0 & 3 & 3 & 2 & 2 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}'.$$

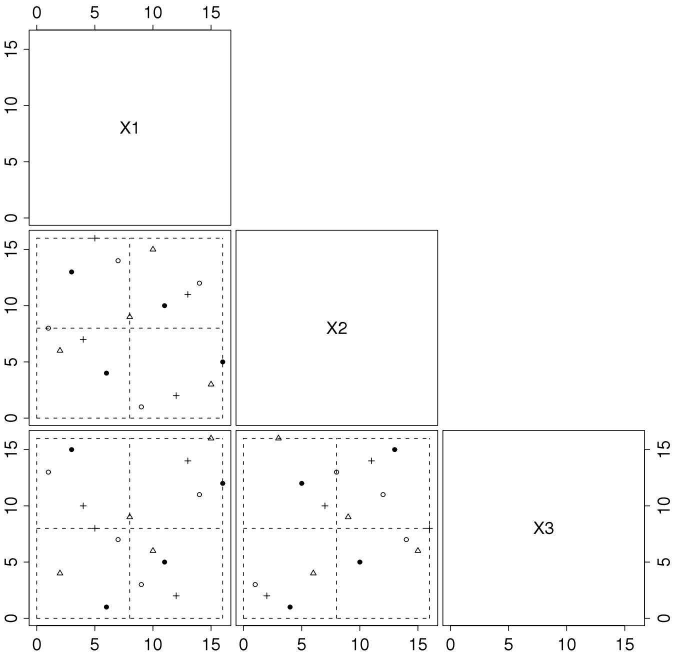

Fig. 1. Bivariate projections among columns of the $SL(16, 3, 4)$ in Example 1.

Without loss of generality, delete the fourth column and we obtain an $ROA_2(32, 4^2 2^3, 3)$ which is

$$\begin{pmatrix}
0&0&1&1&2&2&3&3&0&0&1&1&2&2&3&3&:&1&1&0&0&3&3&2&2&1&1&0&0&3&3&2&2\\
1&0&1&0&1&0&1&0&3&2&3&2&3&2&3&2&:&0&1&0&1&0&1&0&1&2&3&2&3&2&3&2&3\\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&:&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1\\
1&0&0&1&0&1&1&0&0&1&1&0&1&0&0&1&:&0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0\\
0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&:&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1
\end{pmatrix}'.$$

Obviously, this $ROA_2(32, 4^2 2^3, 3)$ can be partitioned into two subarrays and each subarray is an $OA(16, 4^2 2^3, 2)$. Two $M(16, 2, 4)$'s and three $M(16, 2, 2)$'s can be generated as

$$(M_1^4, M_2^4) = \left(\begin{array}{cc|cc}
15&11&12&16\\
10&9&15&9\\
14&16&14&13\\
12&13&10&11\\
\hdashline
29&28&23&21\\
31&30&17&18\\
27&25&22&19\\
26&32&20&24\\
\hdashline
8&5&29&30\\
1&7&26&28\\
4&2&32&31\\
6&3&27&25\\
\hdashline
17&23&2&4\\
24&19&3&1\\
21&18&6&7\\
20&22&8&5
\end{array}\right), \qquad (M_1^2, M_2^2, M_3^2) = \left(\begin{array}{cc|cc|cc}
31&18&10&8&32&31\\
26&24&12&6&29&25\\
20&19&14&2&22&18\\
23&30&1&4&26&23\\
22&32&3&11&28&21\\
17&21&7&16&17&20\\
29&27&15&9&24&30\\
28&25&5&13&19&27\\
\hdashline
6&11&18&19&4&3\\
8&4&31&28&8&11\\
2&9&27&24&1&7\\
16&1&22&17&6&15\\
14&15&29&21&10&14\\
3&13&26&30&16&5\\
10&7&23&25&12&2\\
12&5&20&32&13&9
\end{array}\right).$$

Then following Step 3 of Algorithm 1, the resulting $SL(32, 5, 2)$ can be expressed as $D = ({D_1}', {D_2}')'$ with

$$D_1 = \begin{pmatrix}
15&10&31&27&4&6&21&24&14&12&26&29&8&1&20&17\\
20&12&22&15&17&14&23&10&6&32&2&26&8&29&3&27\\
31&8&10&26&20&16&3&23&22&12&2&17&29&6&14&28\\
18&10&12&20&14&26&22&1&3&27&23&7&31&15&5&29\\
32&29&22&26&4&10&13&6&16&8&12&1&28&17&24&19
\end{pmatrix}' \quad \text{and}$$

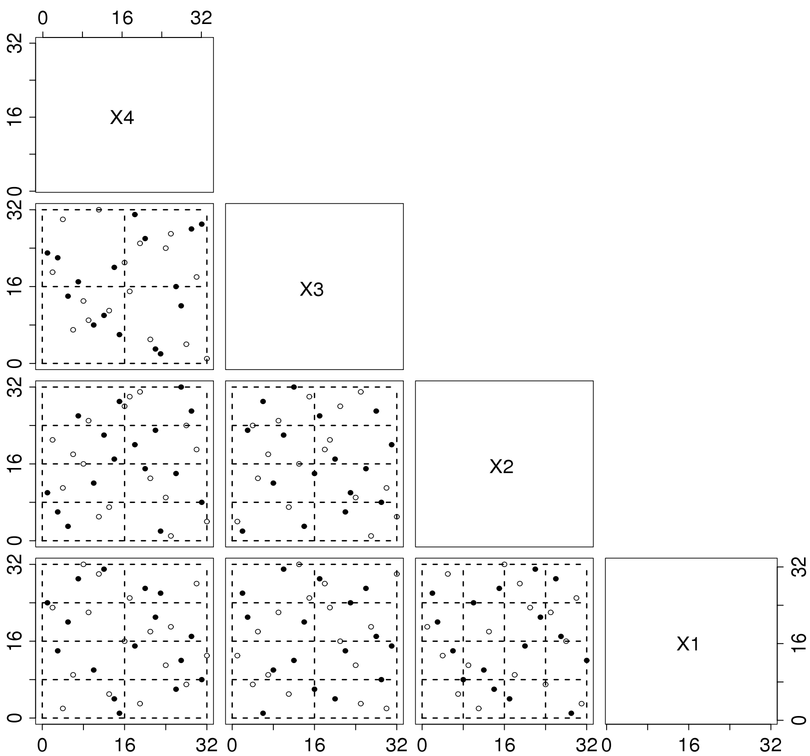

Fig. 2. Bivariate projections among the first four columns of the $SL(32, 5, 2)$ in Example 2.

$$D_2 = \begin{pmatrix} 32 & 28 & 11 & 9 & 18 & 23 & 2 & 7 & 25 & 30 & 16 & 13 & 22 & 19 & 3 & 5 \\ 16 & 19 & 9 & 18 & 13 & 21 & 11 & 24 & 30 & 5 & 28 & 4 & 25 & 1 & 31 & 7 \\ 13 & 18 & 24 & 7 & 5 & 19 & 30 & 4 & 15 & 32 & 21 & 1 & 9 & 27 & 25 & 11 \\ 8 & 30 & 24 & 6 & 21 & 2 & 4 & 28 & 17 & 11 & 16 & 32 & 9 & 25 & 19 & 13 \\ 9 & 11 & 2 & 7 & 31 & 25 & 18 & 23 & 21 & 20 & 30 & 27 & 14 & 5 & 15 & 3 \end{pmatrix}'.$$

Without loss of generality, the projections of the first four columns of $D = ({D_1}', {D_2}')'$ are presented in Fig. 2, where the markers "•" and "○" denote the points from the first four columns X1, X2, X3 and X4 of the slices D_1 and D_2, respectively. From Fig. 2, it can be seen that different level combinations of factors possess different degrees of stratification. In each D_i, $i = 1, 2$, X1 and X2, and X3 and X4 achieve stratifications on 4×4 grids and 2×2 grids in any two dimensions, respectively, and X1 and X3, X1 and X4, X2 and X3, and X2 and X4 achieve stratifications on 4×2 grids in any two dimensions, respectively. In fact, any t $(t \le 3)$ columns of the design D achieve stratification in t dimensions.

Note that if a 4-level column is deleted from the $OA(32, 4^2 2^4, 3)$, then the resulting $ROA_4(32, 4^1 2^4, 3)$ can also be used to construct an SLHD with four slices. The details for the construction are omitted here.

3. Construction of uniform SLHDs

In this section, we will construct uniform SLHDs under the centered L_2 discrepancy (CD_2). Although the SLHDs based on $ROA_p(N, s_1^{k_1} \cdots s_v^{k_v}, w + 1)$'s constructed in the previous section achieve r-dimensional stratification on $s_1^{i_1} \times s_2^{i_2} \times \cdots \times s_v^{i_v}$ grids for $i_u \le k_u$, $u = 1, \ldots, v$, with $\sum_{u=1}^{v} i_u = r \le w + 1$, the points in each $s_1^{i_1} \times s_2^{i_2} \times \cdots \times s_v^{i_v}$ grids do not carry uniformity on the whole experimental domain. Therefore, SLHDs with the uniformity being optimized under some discrepancy, e.g. CD_2, are desirable.

Hickernell (1998) proposed the CD_2 criterion to measure the uniformity of a design. Let $X = \{X_1, \ldots, X_N\}$ be a set of N points in the k-dimensional unit cube $C^k = [0, 1]^k$, where $X_i = (x_{i1}, \ldots, x_{ik})$. The CD_2 of X can be calculated as

$$CD_2(X) = \left\{ \left(\frac{13}{12}\right)^k - \frac{2}{N} \sum_{i=1}^{N} \prod_{j=1}^{k} \left(1 + \frac{1}{2}|x_{ij} - 0.5| - \frac{1}{2}|x_{ij} - 0.5|^2\right) \right.$$
$$\left. + \frac{1}{N^2} \sum_{i,j=1}^{N} \prod_{l=1}^{k} \left(1 + \frac{1}{2}|x_{il} - 0.5| + \frac{1}{2}|x_{jl} - 0.5| - \frac{1}{2}|x_{il} - x_{jl}|\right) \right\}^{1/2}.$$

If a design has a low CD_2, the projections of the design onto subsets of $d(\leq k)$ columns also have good uniformity. Obviously, CD_2 possesses various appealing properties, such as invariant under reordering the runs, relabeling factors, or reflecting the points about any plane passing through the center of the unit cube and parallel to its faces, the projection uniformity over all low dimensions and easier to compute (cf. Fang et al., 2006). For two or more designs, the design with a smaller CD_2 value is preferred. In this paper, the threshold accepting (TA) algorithm, proposed by Dueck and Scheuer (1990), is used to search the uniform SLHDs. Note that for any SLHD $D = (d_{ij})$ constructed in this paper, before calculating $CD_2(D)$, we should first transform its N runs to N points in $C^k = [0, 1]^k$ by mapping d_{ij} to $x_{ij} = (d_{ij} - 0.5)/N$ for $i = 1, \ldots, N$ and $j = 1, \ldots, k$. The details for searching uniform SLHDs are given as follows.

Algorithm 2 (*Generation of Uniform SLHDs*)**.**

Step 1. Randomly generate an initial $SL(N, k, p)$ based on an $ROA_p(N, s_1^{k_1} \cdots s_v^{k_v}, w+1)$ with $\sum_{u=1}^{v} k_u = k$ using Algorithm 1, denoted by D_0, then calculate $CD_2(D_0)$, denoted by d_0. Give a sequence of threshold parameters $Th = (T_1, \ldots, T_L)$, where $T_1 > \cdots > T_L = 0$. Denote the iteration number by I under each T_l for $l = 1, \ldots, L$. Set two indexes $l = 1$ and $i = 1$.
Step 2. Randomly choose one column of D_0 (say an s_u-level column) and two entries (say a and b) in this column, then exchange these two entries if $\lceil as_u/N \rceil = \lceil bs_u/N \rceil$ to create a new design, denoted by D_{try}. Calculate $CD_2(D_{\mathrm{try}})$, denoted by d_{try}.
Step 3. If $d_{\mathrm{try}} - d_0 < T_l$, replace D_0 by D_{try} and set $d_0 = d_{\mathrm{try}}$; 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 deliver $D = D_0$.

Note that Algorithm 2 may find a locally optimal design, thus repeating Algorithm 2 can increase the possibility of reaching the global minimum CD_2, since we usually have different initial designs in Step 1.

Example 3 (Example 1 *Continued*)**.** As an illustration, consider the generation of a uniform $SL(16, 3, 4)$. For the D constructed in Example 1, it can be calculated that $CD_2(D) = 0.0863$. To avoid locally optimal designs, we randomly generate 100 $SL(16, 3, 4)$'s as initial designs for Algorithm 2. After carrying out Algorithm 2 for all these initial designs, choose the resulting design with the minimum CD_2. The final $SL(16, 3, 4)$, denoted by F, with $CD_2(F) = 0.0579$ which is improved by $(0.0863 - 0.0579)/0.0863 \approx 33\%$, is

$$F = (F_1', F_2', F_3', F_4')' = \begin{pmatrix} 14 & 13 & 8 & 3 & : & 6 & 7 & 11 & 10 & : & 16 & 15 & 1 & 2 & : & 5 & 4 & 9 & 12 \\ 9 & 5 & 10 & 8 & : & 2 & 13 & 7 & 16 & : & 11 & 3 & 12 & 4 & : & 6 & 15 & 1 & 14 \\ 10 & 8 & 3 & 9 & : & 6 & 14 & 16 & 7 & : & 13 & 4 & 5 & 15 & : & 1 & 11 & 12 & 2 \end{pmatrix}'.$$

The binary projections of F are displayed in Fig. 3, where the markers "•", "◦", "△" and "+" denote the points from the slices F_1, F_2, F_3 and F_4, respectively. Comparing Fig. 3 with Fig. 1, it can be seen that not only the bivariate projections of F and each F_l have the same projection properties as the design constructed in Example 1, $l = 1, 2, 3, 4$, but also the points of all slices achieve a better uniformity on $[0, 1]^2$.

Example 4 (Example 2 *Continued*)**.** For the $SL(32, 5, 2)$ D constructed in Example 2, we have $CD_2(D) = 0.0981$. Similar to Example 3, by Algorithm 2, the final $SL(32, 5, 2)$ with minimum $CD_2(F) = 0.0734$ which is improved by about 25%, can be given as $F = ({F_1}', {F_2}')'$ with

$$F_1 = \begin{pmatrix} 10 & 16 & 8 & 6 & 17 & 18 & 25 & 29 & 15 & 11 & 4 & 1 & 19 & 21 & 27 & 26 \\ 21 & 6 & 17 & 7 & 18 & 8 & 20 & 5 & 31 & 15 & 28 & 12 & 30 & 13 & 26 & 16 \\ 25 & 3 & 2 & 31 & 32 & 14 & 1 & 23 & 30 & 6 & 16 & 22 & 21 & 15 & 12 & 26 \\ 18 & 1 & 13 & 25 & 11 & 17 & 24 & 5 & 9 & 29 & 19 & 14 & 28 & 4 & 12 & 22 \\ 28 & 18 & 24 & 19 & 1 & 8 & 6 & 4 & 15 & 3 & 2 & 10 & 30 & 26 & 29 & 20 \end{pmatrix}' \quad \text{and}$$

$$F_2 = \begin{pmatrix} 5 & 7 & 14 & 9 & 28 & 32 & 24 & 22 & 3 & 2 & 12 & 13 & 30 & 31 & 20 & 23 \\ 3 & 23 & 1 & 22 & 4 & 19 & 2 & 24 & 9 & 27 & 11 & 32 & 14 & 25 & 10 & 29 \\ 10 & 18 & 19 & 11 & 9 & 17 & 29 & 13 & 5 & 27 & 20 & 8 & 7 & 28 & 24 & 4 \\ 10 & 31 & 23 & 2 & 30 & 6 & 15 & 20 & 21 & 3 & 8 & 26 & 16 & 27 & 32 & 7 \\ 14 & 16 & 7 & 5 & 23 & 22 & 27 & 17 & 31 & 21 & 32 & 25 & 12 & 9 & 13 & 11 \end{pmatrix}'.$$

Without loss of generality, the bivariate projections of the first four columns of F are presented in Fig. 4, where the markers "•" and "◦" denote the points from the first 16 rows and the next 16 rows of F, respectively.

From Fig. 4, it can be seen that the design points not only have the same projection properties as the design D constructed in Example 2, but also scatter more evenly on the experimental domain, see e.g., the uniformity of the 32 points in the projection of X3 and X4 has been highly improved by comparing to the corresponding projection in Fig. 2.

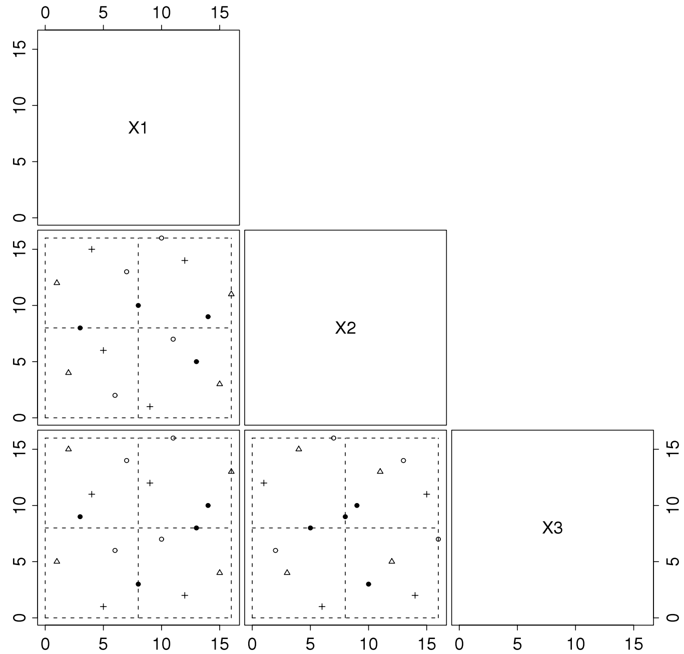

Fig. 3. Bivariate projections among columns of the uniform $SL(16, 3, 4)$ in Example 3.

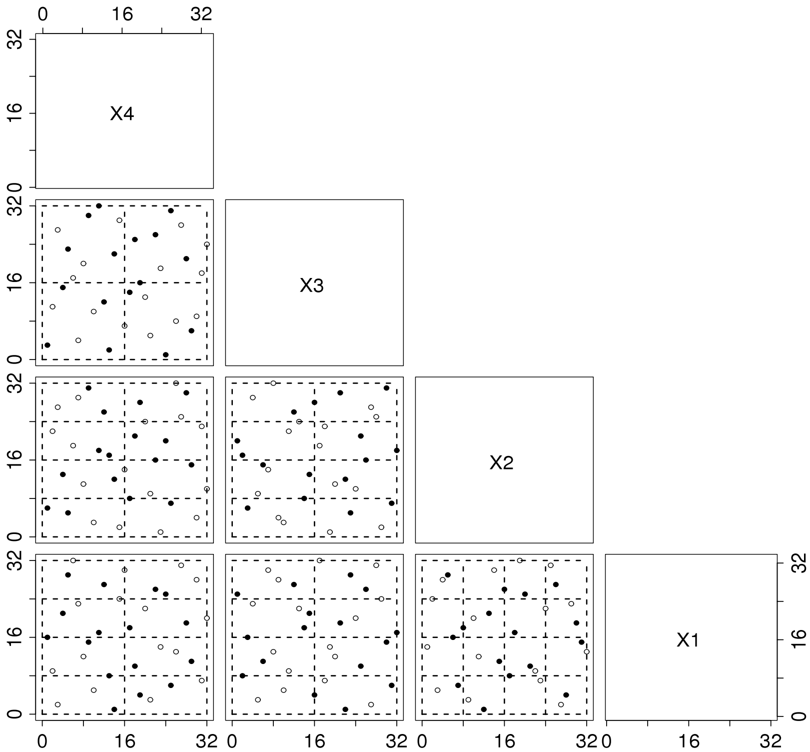

Fig. 4. Bivariate projections among the first four columns of the uniform $SL(32, 5, 2)$ in Example 4.

4. Discussion and concluding remarks

In this paper, we have proposed a new type of SLHD that is useful for computer experiments with both qualitative and quantitative factors. By using symmetric and asymmetric ROAs, we can construct ROA-based SLHDs. For the existing SLHDs, those constructed by Qian (2012) only achieve one-dimensional maximum stratification although they can accommodate any number of factors, and those of Yang et al. (2013) and Huang et al. (2014) possess the exact or near orthogonality among the columns. While the novel feature of the newly constructed SLHD is that the whole LHD achieves $r(\leq w+1)$-dimensional stratification and each slice achieves $r(\leq w)$-dimensional stratification, where $w+1$ is the strength of the corresponding parent ROA. Moreover, the CD_2 criterion is used to evaluate the uniformity of the SLHDs, and uniform SLHDs are then generated with the help of a modified TA algorithm.

Following the discussion in Section 2, we know that it is easy to obtain a symmetric or asymmetric ROA from an OA. Thus, the construction method is easy to implement. The existence of OAs can be found from Hedayat et al. (1999) and the websites maintained by Dr. N.J.A. Sloane (http://neilsloane.com/oadir/) and Dr. W.F. Kuhfeld (http://support.sas.com/techsup/technote/ts723.html).

Furthermore, it is possible to find optimal ROA-based SLHDs under some other uniformity measure, such as the wrap-around discrepancy, the minimax distance or the maximin distance criterion (cf. Fang et al., 2006).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 11271205), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20130031110002), and the ''131'' Talents Program of Tianjin. The first two authors contribute equally to this work.

References

Agrawal, V., Dey, A., 1983. Orthogonal resolution-IV designs for some asymmetrical factorials. Technometrics 25, 197–199.
Brouwer, A.E., Cohen, A.M., Nguyen, M.V.M., 2006. Orthogonal arrays of strength 3 and small run sizes. J. Statist. Plann. Inference 136, 3268–3280.
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., Ge, G.N., Liu, M.Q., 2002. Uniform supersaturated design and its construction. Sci. China Ser. A 45, 1080–1088.
Fang, K.T., Li, R., Sudjianto, A., 2006. Design and Modeling for Computer Experiments. CRC Press, New York.
Fang, K.T., Lin, D.K.J., 2003. Uniform experimental design and its applications in industry. In: Rao, C.R., Khattree, R. (Eds.), Handbook of Statistics in Industry, Vol. 22. North Holland, New York, pp. 131–170.
Fang, K.T., Lin, D.K.J., Liu, M.Q., 2003. Optimal mixed-level supersaturated design. Metrika 58, 279–291.
Hedayat, A.S., Sloane, N.J.A., Stufken, J., 1999. Orthogonal Arrays: Theory and Applications. Springer, New York.
Hickernell, F.J., 1998. A generalized discrepancy and quadrature error bound. Math. Comp. 67, 299–322.
Hickernell, F.J., Liu, M.Q., 2002. Uniform designs limit aliasing. Biometrika 89, 893–904.
Huang, H.Z., Yang, J.F., Liu, M.Q., 2014. Construction of sliced (nearly) orthogonal Latin hypercube designs. J. Complexity 30, 355–365.
Jiang, L., Yin, J.X., 2013. An approach of constructing mixed-level orthogonal arrays of strength ≥ 3. Sci. China Math. 56, 1109–1115.
Nguyen, M.V.M., 2008. Some new constructions of strength 3 mixed orthogonal arrays. J. Statist. Plann. Inference 138, 220–233.
Qian, P.Z.G., 2012. Sliced Latin hypercube designs. J. Amer. Statist. Assoc. 107, 393–399.
Tang, B., 1993. Orthogonal array-based Latin hypercubes. J. Amer. Statist. Assoc. 88, 1392–1397.
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.
Yin, Y.H., Lin, D.K.J., Liu, M.Q., 2014. Sliced Latin hypercube designs via orthogonal arrays. J. Statist. Plann. Inference 149, 162–171.