Received: 31 October 2021 | Accepted: 15 December 2021

DOI: 10.1002/sta4.447

ORIGINAL ARTICLE

WILEY

Orthogonal designs with branching and nested factors

Qiao Wei | Min-Qian Liu | Jian-Feng Yang

School of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin, 300071, China

Correspondence
Jian-Feng Yang, School of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin 300071, China.
Email: jfyang@nankai.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 12131001, 11771220, 11871033; National Ten Thousand Talents Program of China, Fundamental Research Funds for the Central Universities, Grant/Award Number: 63211090; Natural Science Foundation of Tianjin, Grant/Award Number: 20JCYBJC01050

Computer experiments with branching and nested factors are a common class of computer experiments, but it is challenging to construct designs for this type of experiments. In this paper, we define a special type of design called branching orthogonal Latin hypercube design (BOLHD). Such a design has an appealing structure, that is, no matter at each level of a branching factor or the level-combination of branching factors, the corresponding design points of nested factors form an orthogonal Latin hypercube design (OLHD). This structure makes it a good choice for designing computer experiments with branching and nested factors. We propose several deterministic construction methods when branching factors have the same number of levels. Based on sliced Latin hypercube designs (SLHDs), the proposed methods are easy to operate. Some construction results are tabulated for practical use.

KEYWORDS
computer experiment, orthogonality, sliced latin hypercube design

1 | INTRODUCTION

Computer experiments are widely used for the design and development of products (Fang et al. 2006). Scientists are increasingly using experiments on computer simulators to help understand physical systems. In many experiments, some of the factors exist only within the level of another factor. Such factors are often called nested factors. A factor within which other factors are nested is called a branching factor. Other factors which are common to all levels of the branching factors are called shared factors. To better illustrate branching and nested factors, we refer to the example of printed circuit board manufacturing from Hung et al. (2009). Suppose that we want to test two surface preparation methods: mechanical scrubbing and chemical treatment. Mechanical scrubbing can be optimized by changing the pressure scrubbing, and chemical treatment can be optimized by changing the micro-etching rate. The surface preparation method here is a branching factor, and the pressure and microetching rate are nested factors.

Hung et al. (2009) mentioned that because nested factors differ for different levels of the branching factor, there is a need for one-dimensional balance for the nested factors within each level of the branching factor to capture the branching-by-nested interaction effects. Latin hypercube designs (LHDs) are commonly used in computer experiments (McKay et al. 1979). A desirable property of an LHD is its one-dimensional balance, that is, when an N-point design is projected onto any factor, there will be N different levels for that factor. Hence, Hung et al. (2009) incorporated LHD into the design with branching and nested factors and proposed a corresponding design called branching Latin hypercube design (BLHD). Goos and Jones (2019) discussed the modelling of data from experiments with branching and nested factors as well as the optimal design of such experiments. Chen et al. (2019) considered the case where branching factors and nested factors are both qualitative and proposed two-layer sliced Latin hypercube designs (SLHDs) to suit such situations. Chen et al. (2021) proposed the level-collapsing method to construct BLHDs having a sliced structure in the part for the shared factors.

In this paper, a more detailed case is considered, that is, when there are multiple branching factors, the hidden models under each level-combination of branching factors may be different because the corresponding nested factors differ for different level combinations of branching factors. Thus, we need a one-dimensional balance for the nested factors within each level combination of the branching factors. In addition, orthogonality is also an important property of a design, which can guarantee the independence of estimates of linear effects when a first-order model is fitted. Specifically, we need a design which can ensure that

(1) both the modelling of nested factors at a certain level of a branching factor alone and the modelling of nested factors at a level combination of branching factors can result in good parameter estimations, that is, the design is robust to the models;
(2) the design is orthogonal to ensure that significant factors can be identified.

To find designs with the aforementioned properties, we define a special type of design called branching orthogonal Latin hypercube design (BOLHD), which satisfies the following: (i) For each level of a branching factor, the corresponding design of nested factors is orthogonal and achieves maximum uniformity in any one-dimensional projection; (ii) for each level-combination of the branching factors, the corresponding design of nested factors is orthogonal and achieves maximum uniformity in any one-dimensional projection; (iii) the part of design for the shared factors is an OLHD, and for each level combination of branching factors, the corresponding design of shared factors is orthogonal to the design of nested factors and achieves maximum uniformity in any one-dimensional projection. These good properties can make (1) and (2) hold. It is worth mentioning that the proposed design is like a design between the marginally coupled design (MCD) proposed by Deng et al. (2015) and SLHD proposed by Qian (2012), but it is neither an MCD nor an SLHD. Note that the design of the nested factors in the proposed design is not, and need not to be, an LHD as a whole, since the nested factors represent different meanings for different levels of the branching factor.

The rest of this paper is organized as follows. Section 2 introduces some basic definitions and notation. Further, we give the structure of the proposed design and provide the definition of BOLHD. Section 3 provides the construction of BOLHDs with two branching factors. Section 4 provides the construction of BOLHDs with multiple branching factors and lists some of the construction results for practical use. Some concluding remarks are given in Section 5.

2 | NOTATION AND DESIGN STRUCTURE

An $n \times p$ LHD is a matrix of n rows and p columns in which each column is a permutation of n equally spaced levels. For convenience, we take the n levels to be $-(n-1)/2, -(n-3)/2, \dots, (n-3)/2, (n-1)/2$. Qian (2012) defined an $n \times q$ SLHD with $n = mv$ runs and v slices, denoted by SLHD(n, q, v), to be an $n \times q$ LHD that can be divided into v smaller LHDs of m levels. The m levels of each slice correspond to the m equally spaced intervals $\{[-n/2+(j-1)v, -n/2+jv] : 1 \le j \le m\}$. For an integer s and a vector $f=(f_1,\dots,f_n)^T$, let $s*f=(sf_1,\dots,sf_n)^T$ and $s \oplus f=(s+f_1,\dots,s+f_n)^T$. For two vectors $u=(u_1,\dots,u_n)^T$ and $w=(w_1,\dots,w_n)^T$, let $u \oplus w=(u_1+w_1,\dots,u_1+w_n,\dots,u_n+w_1,\dots,u_n+w_n)^T$, and define the correlation coefficient between u and w as

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

where $\bar{u}=\sum_{i=1}^{n}u_i/n$ and $\overline{w}=\sum_{i=1}^{n}w_i/n$. Two columns are said to be orthogonal if their correlation coefficient is zero. An LHD is called an OLHD if any two columns are orthogonal. An SLHD is called orthogonal if any two columns of each slice are orthogonal. For convenience, we denote an orthogonal SLHD(n,q,v) by SOLHD(n,q,v). For a design D, let $D(i,:)$, $D(:,j)$ and $D(i,j)$ be its ith row, jth column and (i,j)th entry, respectively.

We assume that there are q branching factors, denoted by $\mathbf{z}=(z_1,\dots,z_q)^T$, and for each branching factor z_u, the m_u nested factors are denoted by $\mathbf{v}^{z_u}=\left(v_1^{z_u},\dots,v_{m_u}^{z_u}\right)^T, 1 \le u \le q$. We further assume that, in addition to the branching and nested factors, there are t shared quantitative factors, denoted by $\mathbf{x}=(x_1,\dots,x_t)^T$. Let $\mathbf{v}=\left((\mathbf{v}^{z_1})^T,\dots,(\mathbf{v}^{z_q})^T\right)^T$ then, $\mathbf{w}=\left(\mathbf{z}^T,\mathbf{v}^T,\mathbf{x}^T\right)^T$ represents all of the p factors involved in the experiment, where $p=q+\sum_{u=1}^{q}m_u+t$. Then, an N-run BLHD can be denoted by $\mathbf{W}=(\mathbf{w}_1,\dots,\mathbf{w}_N)^T$.

Our basic idea is to incorporate the structure of SLHD into the structure of BLHD to ensure a one-dimensional balance for the nested factors within each level-combination of the branching factors. To better demonstrate the structure of the newly proposed design, let us see a simple example. Suppose there are two branching factors, called z_1, z_2, with s levels and m_i nested factors at each level of z_i. The nested factors are different at different levels of each branching factor. More generally, we also assume there are t quantitative shared factors. For the simplest case where each branching factor has two levels ($s=2$), Table B1 shows the structure of the proposed design. For comparison, Table B2 shows the BLHD constructed by Hung et al. (2009). It can be seen that, compared to the BLHD, the proposed design can not only achieve a one-dimensional balance for the nested factors within each level of each branching factor but also achieve a one-dimensional balance for the nested factors within each level-combination of the branching factors. Table B3 shows the structure of the proposed designs with two branching factors each of \$s\$ levels.

It is well known that orthogonality is critical to a design, which can guarantee the independence of estimates of linear effects when a first-order model is fitted. So, at the end of this section, on the basis of the structure of SLHD-based BLHD, we will introduce a new type of BLHD called branching orthogonal Latin hypercube design (BOLHD). The specific definition is as follows.

Definition 1. A BLHD is called a BOLHD if

(1) for each level of each branching factor, the corresponding design points of nested factors form an OLHD;
(2) for each level-combination of branching factors, the corresponding design points of nested factors form an OLHD;
(3) the part of design for the shared factors is an OLHD, and for each level-combination of branching factors, the corresponding design points of shared factors form a small LHD which is orthogonal to the design of nested factors.

This definition incorporates the three properties described in the introduction. An N-run BOLHD with q branching factors (each branching factor corresponds to m_i nested factors, $i = 1,\ldots,q$) and t shared factors can be expressed in terms of BOLHD(N, q, ($m_1, \ldots, m_q$), t). For convenience of presentation, we divide the structure of a BOLHD into three parts: part D_1 corresponding to branching factors, part D_2 corresponding to nested factors and part D_3 corresponding to shared factors. Next, we will show how to construct BOLHDs with branching factors having the same number of levels.

3 | CONSTRUCTION OF BOLHDS WITH TWO BRANCHING FACTORS

To better understand our construction method, we start with the simplest case with only two branching factors. This section provides two algorithms for constructing BOLHDs when there are two branching factors each of s levels. Because branching factors are qualitative factors, we can choose a full factorial design as D_1, that is,

$$D_1 = \begin{pmatrix} 1 & \cdots & 1 & \cdots & s & \cdots & s \\ 1 & \cdots & s & \cdots & 1 & \cdots & s \end{pmatrix}^T,$$

where the first column of D_1 is for z_1 and the second column is for z_2. This section mainly gives the construction methods for D_2 and D_3. For details, Section 3.1 gives the construction algorithm for BOLHDs with no shared factors, while Section 3.2 considers the case with shared factors, which is more general and the corresponding algorithm is more complex.

3.1 | BOLHDs with two branching factors and no shared factors

This is the simplest case, but it gives an intuitive explanation of our construction methods. Because there is no shared factor and D_1 has been given, this section mainly discusses how to construct D_2. See Algorithm 1 for details.

Algorithm 1.
Step 1 Let $L = (L_1^T, \ldots, L_s^T)^T$ be an SOLHD($N = ns, m_1 + m_2, s$), where L_i is the i th slice of L for $i = 1,\ldots,s$. For $i = 1,\ldots,s$, let a_i and b_i be the first m_1 and last m_2 columns of L_i, respectively.
Step 2 For the i th level of z_1, let $A_i = \left(a^T_{0\dot{+}(i-1)+1}, \ldots, a^T_{(s-1)\dot{+}(i-1)+1}\right)^T$ and $B_i = \left(b^T_{0\dot{+}(i-1)+1}, \ldots, b^T_{(s-1)\dot{+}(i-1)+1}\right)^T$, $i = 1,\ldots,s$, where $\dot{+}$ represents the modulo s addition operation, that is, for two integers x and y, $x \dot{+} y = (x + y) \bmod s$.
Step 3 Stack the A_i and B_i obtained in Step 3 row by row to obtain $A = (A_1^T, \ldots, A_s^T)^T$ and $B = (B_1^T, \ldots, B_s^T)^T$, where A corresponds to the design for $v_1^{z_1}, \ldots, v_{m_1}^{z_1}$ and B corresponds to the design for $v_1^{z_2}, \ldots, v_{m_2}^{z_2}$.
Step 4 Let $D_2 = (A, B)$. Then, combine $D_1 \otimes \mathbf{1}_n$ and D_2 column by column to get the final design $S = (D_1 \otimes \mathbf{1}_n, D_2)$, where $\otimes$ represents the Kronecker product and $\mathbf{1}_n$ is an $n \times 1$ vector with all elements unity.

Note that at different levels of z_1 or z_2, the designs for the corresponding nested factors are just different in the order of slices and they are essentially the same SOLHD. This is feasible since we assume that the nested factors represent different meanings at different levels of a branching factor.

Theorem 1. *The design S constructed by Algorithm 1 is a BOLHD*(N = ns^2, *2*, (m$_1$, m$_2$)).

Proof. We need to prove that design S satisfies (1) and (2) in Definition 1. For (1), we note that, for any fixed level of each branching factor, the design of nested factors is just obtained by simply changing the order of slices of L, which preserves the orthogonality and one-dimensional space-filling properties of L. Next, we prove that the design S satisfies (2). According to Steps 2 and 3 of Algorithm 1, for any fixed level combination of z_1 and z_2, the design of nested factors corresponds to a certain (a_{i_0}, b_{i_0}) with $i_0 \in \{1, \ldots, s\}$. So the result follows by noting that a_{i_0} and b_{i_0} are the first m_1 and last m_2 columns of L_{i_0}, respectively.

Example 1. Suppose that $s=2$, $t=0$, $m_1=m_2=2$. We consider getting D_2 using the following SOLHD(16, 4, 2) constructed by Algorithm 1 in Yang et al. (2016),

$$L=\left(L_1^T, L_2^T\right)^T=\begin{pmatrix} 0.5 & 2.5 & 4.5 & 6.5 & -0.5 & -2.5 & -4.5 & -6.5 & 1.5 & 3.5 & 5.5 & 7.5 & -1.5 & -3.5 & -5.5 & -7.5 \\ 2.5 & -0.5 & 6.5 & -4.5 & -2.5 & 0.5 & -6.5 & 4.5 & 3.5 & -1.5 & 7.5 & -5.5 & -3.5 & 1.5 & -7.5 & 5.5 \\ -6.5 & -4.5 & 2.5 & 0.5 & 6.5 & 4.5 & -2.5 & -0.5 & -7.5 & -5.5 & 3.5 & 1.5 & 7.5 & 5.5 & -3.5 & -1.5 \\ 4.5 & -6.5 & -0.5 & 2.5 & -4.5 & 6.5 & 0.5 & -2.5 & 5.5 & -7.5 & -1.5 & 3.5 & -5.5 & 7.5 & 1.5 & -3.5 \end{pmatrix}^T.$$

According to Algorithm 1, we choose the first two columns and last two columns of L_i as a_i and b_i, $i=1,2$. Then, we can obtain $A_1=\left(a_1^T, a_2^T\right)^T$, $B_1=\left(b_1^T, b_2^T\right)^T$, $A_2=\left(a_2^T, a_1^T\right)^T$, and $B_2=\left(b_2^T, b_1^T\right)^T$. Further, we have

$$D_2=\begin{pmatrix} A_1 & B_1 \\ A_2 & B_2 \end{pmatrix}=\begin{pmatrix} a_1 & a_2 & a_2 & a_1 \\ b_1 & b_2 & b_2 & b_1 \end{pmatrix}^T.$$

Combining $D_1 \otimes \mathbf{1}_8$ and D_2 column by column, we can get the BOLHD(32, 2, (2, 2)) as $S=(D_1 \otimes \mathbf{1}_8, D_2)$, where

$$D_1=\begin{pmatrix} 1 & 1 & 2 & 2 \\ 1 & 2 & 1 & 2 \end{pmatrix}^T.$$

We briefly verify the orthogonality of the design S. It is easy to verify that for the i th level of z_1 (the same is true for z_2), the corresponding design of nested factors A_i is an OLHD, $i=1,2$. Simple calculation shows that $(a_1,b_1)^T \cdot (a_1,b_1)=(a_2,b_2)^T \cdot (a_2,b_2)=I_{4\times 4}$, where $I_{n\times m}$ denotes an $n\times m$ identity matrix, which means that for each level combination of branching factors, the corresponding design of nested factors is orthogonal.

3.2 | BOLHDs with two branching factors and t shared factors

This section considers the construction of BOLHDs with two branching factors and t shared factors. The construction of the design for nested factors is the same as in Algorithm 1. So we mainly focus on the construction of the design for shared factors. Algorithm 2 gives the construction details.

Algorithm 2.

Step 1 Let $P=\left(P_1^T,\ldots,P_s^T\right)^T$ and E be an SOLHD($N=ns,k,s$) and an OLHD(s,t) respectively, where $k=m_1+m_2+t$ and $t \le s$.

Step 2 Let $L=\left(L_1^T,\ldots,L_s^T\right)^T$ and $Q=\left(Q_1^T,\ldots,Q_s^T\right)^T$ be the first m_1+m_2 and last t columns of P, respectively, then L is an SOLHD($N=ns$, m_1+m_2, s) and Q is an SOLHD($N=ns,t,s$).

Step 3 Construct design D_2 for nested factors using L by Algorithm 1.

Step 4 On the basis of Q, construct s matrices denoted by $C_i=\left(c_{i,1}^T,\ldots,c_{i,s}^T\right)^T$, $i=1,\ldots,s$. The j th column of C_i is obtained by $C_i(:,j)=E(i,j)\oplus(s*Q(:,j))$, $j=1,\ldots,t$.

Step 5 For the i th level of z_1, let $T_i=\left(c_{i,0\dot{+}(i-1)+1}^T,\ldots,c_{i,(s-1)\dot{+}(i-1)+1}^T\right)^T$, $i=1,\ldots,s$.

Step 6 Stack the s arrays row by row, and obtain $D_3=\left(T_1^T,\ldots,T_s^T\right)^T$.

Step 7 Combine $D_1\otimes \mathbf{1}_n$, D_2 and D_3 column by column to get the final design $S=(D_1\otimes\mathbf{1}_n, D_2, D_3)$.

Theorem 2. *The design S constructed by Algorithm 2 is a BOLHD*($N=ns^2$, s, (m_1, m_2), t).

The proof of Theorem 2 is shown in the appendix. The constraint $t \le s$ in Step 1 of Algorithm 2 is necessary, because there is no OLHD(s, t) with $s<t$. In fact, the design D_3 constructed in Algorithm 2 is a two-layer SOLHD proposed by Chen et al. (2019).

Example 2. Suppose that $s=2$, $t=1$, $m_1=2$ and $m_2=1$. Let P be the SOLHD(16, 4, 2) shown in Example 1. According to Algorithm 1, we choose the first three columns of P as L and the last column as Q. We choose the first two columns and last column of L_i as a_i and b_i, $i=1,2$. Then, we can get D_2 as

$$D_2=\begin{pmatrix}A_1 & B_1\\ A_2 & B_2\end{pmatrix}=\begin{pmatrix}a_1 & a_2 & a_2 & a_1\\ b_1 & b_2 & b_2 & b_1\end{pmatrix}^T,$$

where

$$A_1=\begin{pmatrix}0.5 & 2.5 & 4.5 & 6.5 & -0.5 & -2.5 & -4.5 & -6.5\\ 2.5 & -0.5 & 6.5 & -4.5 & -2.5 & 0.5 & -6.5 & 4.5\end{pmatrix}^T,\quad A_2=\begin{pmatrix}1.5 & 3.5 & 5.5 & 7.5 & -1.5 & -3.5 & -5.5 & -7.5\\ 3.5 & -1.5 & 7.5 & -5.5 & -3.5 & 1.5 & -7.5 & 5.5\end{pmatrix}^T,$$
$$B_1=(-6.5\ \ -4.5\ \ 2.5\ \ 0.5\ \ 6.5\ \ 4.5\ \ -2.5\ \ -0.5)^T,\qquad B_2=(-7.5\ \ -5.5\ \ 3.5\ \ 1.5\ \ 7.5\ \ 5.5\ \ -3.5\ \ -1.5)^T.$$

Further, by letting $E=(-0.5,0.5)^T$, we can obtain D_3 using Algorithm 2 as $D_3=(T_1,T_2)^T=(c_{1,1},c_{1,2},c_{2,2},c_{2,1})^T$, where

$$c_{1,1}=(8.5\ \ -13.5\ \ -1.5\ \ 4.5\ \ -9.5\ \ 12.5\ \ 0.5\ \ -5.5)^T,\quad c_{1,2}=(10.5\ \ -15.5\ \ -3.5\ \ 6.5\ \ -11.5\ \ 14.5\ \ 2.5\ \ -7.5)^T,$$
$$c_{2,2}=(11.5\ \ -14.5\ \ -2.5\ \ 7.5\ \ -10.5\ \ 15.5\ \ 3.5\ \ -6.5)^T,\quad c_{2,1}=(9.5\ \ -12.5\ \ -0.5\ \ 5.5\ \ -8.5\ \ 13.5\ \ 1.5\ \ -4.5)^T.$$

Combining $D_1\otimes\mathbf{1}_8$ (shown in Example 1), D_2 and D_3 column by column, we can get the BOLHD (32, 2, (2, 1), 1) as $S=(D_1\otimes\mathbf{1}_8,D_2,D_3)$.

Just the same as in Example 1, we can verify the orthogonality of design S. It is easy to verify that for the i th level of z_1 (the same is true for z_2), the corresponding design of nested factors A_i is an OLHD, $i=1,2$. Through simple calculations, we can get that $(a_1,b_1)^T\cdot(a_1,b_1)=(a_2,b_2)^T\cdot(a_2,b_2)=I_{3\times3}$, meaning that for each level combination of branching factors, the corresponding design of nested factors is orthogonal, and $(a_1,b_1)^T\cdot c_{1,1}=(a_2,b_2)^T\cdot c_{1,2}=(a_2,b_2)^T\cdot c_{2,2}=(a_1,b_1)^T\cdot c_{2,1}=(0,0,0)^T$, meaning that for each level combination of branching factors, the corresponding design points of D_3 is orthogonal to the design of nested factors.

4 | BOLHDS WITH MULTIPLE BRANCHING FACTORS

From the construction results in Section 3, we know that, in the case of two branching factors, we can get a BOLHD by rearranging the slices of an SOLHD and stacking them together row by row. This section will extend this idea further to the case of q branching factors each of s levels, $q>2$. Still the same as in Section 3, we arrange a full factorial experiment for branching factors, that is,

$$D_1=\begin{pmatrix}1 & 1 & \dots & 1 & \dots & s\\ 1 & 1 & \dots & 1 & \dots & s\\ \vdots & \vdots & \dots & \vdots & \dots & \vdots\\ 1 & 2 & \dots & s & \dots & s\end{pmatrix}^T,$$

where the ith column of D_1 corresponds to z_i, $i=1,\dots,q$. This section mainly focuses on the construction of D_2 and D_3. Section 4.1 considers the construction of BOLHDs without shared factors, and Section 4.2 considers the case with shared factors, which is slightly more complex and more general.

4.1 | BOLHDs with q branching factors and no shared factors

This section considers the construction of BOLHDs with q branching factors and no shared factors. For convenience, we first make a top-down order of s^{q-1} slices of an SOLHD, denoted as $\mathcal{A}=(1,2,\dots,s^{q-1})^T$. Then, we focus on the index vector $\mathcal{A}$. The basic idea of constructing D_2 is to rearrange the slices of the SOLHD according to the index vector $\mathcal{B}$ which is a sequential rearrangement of the indices in $\mathcal{A}$. Finally, we can get a BOLHD by combining D_1 and D_2. Algorithm 3 shows the details. To facilitate the description of the algorithm, we first define a mapping function ϕ as shown in Definition 2.

Definition 2. Suppose that $X=\text{diag}\{Z,Z,\dots,Z\}$ is an $s^k\times s^k$ block diagonal matrix, where Z is an $s^{k-1}\times s^{k-1}$ matrix. We define the mapping function ϕ: $X\to Y$ as

$$X=\begin{pmatrix} Z & 0 & \dots & 0\\ 0 & Z & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & Z\end{pmatrix}_{s^k\times s^k} \quad \underset{\rightarrow}{\phi} \quad Y=\begin{pmatrix} 0 & Z & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & Z\\ Z & 0 & \dots & 0\end{pmatrix}_{s^k\times s^k}.$$

Algorithm 3.

Step 1 Let $L=(L_1^T,\dots,L_{s^{q-1}}^T)^T$ be an SOLHD($N=s^{q-1}n,r,s^{q-1}$), $r=\Sigma_{i=1}^q m_i$. The index vector of these slices is denoted by $\mathcal{A}=(1,2,\dots,s^{q-1})^T$.

Step 2 For $c=1$, let E_1 be an $s\times s$ identity matrix and define $F_1=\phi(E_1)$. For $1<c\leq q-1$, define E_c as

$$E_c=\begin{pmatrix} F_{c-1} & 0 & \dots & 0\\ 0 & F_{c-1} & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & F_{c-1}\end{pmatrix}_{s^c\times s^c}$$

and $F_c=\phi(E_c)$.

Step 3 For $l=1$, let $\mathcal{A}_1=\mathcal{A}=(1,2,\dots,s^{q-1})^T$. For $1<l\leq s$, define $\mathcal{A}_l$ as $\mathcal{A}_l=F_{q-1}\cdot\mathcal{A}_{l-1}$.

Step 4 For $l=1,\dots,s$, generate design G_l by arranging slices of L according to the index vector $\mathcal{A}_l$ and get $G=(G_1^T,\dots,G_s^T)^T$.

Step 5 For $i=1,\dots,q$, select m_i columns from design G without repetition denoted by K_i, where K_i corresponds to the design for $v_1^{z_i},\dots,v_{m_i}^{z_i}$.

Step 6 Let $D_2=(K_1,\dots,K_q)$. Then, combine $D_1\otimes\mathbf{1}_n$ and D_2 column by column to get the final design $S=(D_1\otimes\mathbf{1}_n,D_2)$.

Theorem 3. *The design S constructed by Algorithm 3 is a BOLHD*($N = ns^q$, q, (m_1, m_2, … , m_q)).

4.2 | BOLHDs with *q* branching factors and *t* shared factors

This section considers the construction of BOLHDs with q branching factors and t shared factors. As in Section 3.2, we refer to the construction method of two-layer SOLHDs in Chen et al. (2019) to give the construction of the part of design for the shared factors. See Algorithm 4 for details.

Algorithm 4.

Step 1 Let $P=(P_1^T,\dots,P_{s^{q-1}}^T)^T$ and E be an SOLHD($N=s^{q-1}n,r,s^{q-1}$) and an OLHD(s,t), respectively, where $r=\Sigma_{i=1}^q m_i+t$ and $t\leq s$. The index vector of these slices is denoted as $\mathcal{A}=(1,2,\dots,s^{q-1})^T$.

Step 2 Do the same thing as Steps 2 and 3 in Algorithm 3, and we get a series of index vectors $\mathcal{A}_i$, $i=1,\dots,s$.

Step 3 Let $L=(L_1^T,\dots,L_{s^{q-1}}^T)^T$ and $Q=(Q_1^T,\dots,Q_{s^{q-1}}^T)^T$ be the first $\Sigma_{i=1}^q m_i$ and last t columns of P, respectively, then, L is an SOLHD($N=s^{q-1}n,\Sigma_{i=1}^q m_i,s^{q-1}$), and Q is an SOLHD($N=s^{q-1}n,t,s^{q-1}$).

Step 4 Construct design D_2 for nested factors using L following Steps 4–6 of Algorithm 3.

Step 5 For $i=1,\dots,s$, generate the array A_i by arranging slices of Q according to the index vector $\mathcal{A}_i$.

Step 6 For $i=1,\dots,s$, obtain T_i by $T_i(:,j)=E(i,j)\oplus(s*A_i(:,j))$, $j=1,\dots,t$.

Step 7 Stack the s arrays obtained in Step 7 row by row, and obtain $D_3=(T_1^T,\dots,T_s^T)^T$.

Step 8 Combine $D_1\otimes\mathbf{1}_n$, D_2 and D_3 column by column to get the final design $S=(D_1\otimes\mathbf{1}_n,D_2,D_3)$.

Theorem 4. *The design S constructed by Algorithm 4 is a BOLHD*($N = ns^q$, q, (m_1, m_2, … , m_q), t).

Example 3. Suppose that $s=2$, $t=1$, $m_1=m_2=m_3=1$. Let $P=(P_1^T,P_2^T,P_3^T,P_4^T)^T$ be the SOLHD(32,4,4) (shown in the appendix) constructed by Algorithm 1 in Yang et al. (2016). According to Algorithm 4, we choose the first three columns of P as $L=(L_1^T,L_2^T,L_3^T,L_4^T)^T$ and the last column as $Q=(Q_1^T,Q_2^T,Q_3^T,Q_4^T)^T$. Then, we can get

$$G_1=(L_1^T,L_2^T,L_3^T,L_4^T)^T, G_2=(L_4^T,L_3^T,L_2^T,L_1^T)^T, A_1=(Q_1^T,Q_2^T,Q_3^T,Q_4^T)^T, \text{ and} A_2=(Q_4^T,Q_3^T,Q_2^T,Q_1^T)^T.$$

We choose the i th column of $G=\left(G_1^T,G_2^T\right)^T$ as K_i, $i=1,2,3$. Then, we can get D_2 as $D_2=(K_1,K_2,K_3)$. Further, by letting $E=(-0.5,0.5)^T$, we can obtain D_3 as $D_3=(T_1^T,T_2^T)^T$ (shown in the appendix), where $T_1=2*A_1-0.5$ and $T_2=2*A_2+0.5$.

Combining $D_1 \otimes \mathbf{1}_8$, D_2 and D_3 column by column, we can get the BOLHD(64, 3, (1, 1, 1), 1) as $S = (D_1 \otimes \mathbf{1}_8, D_2, D_3)$, where

$$D_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{pmatrix}^T.$$

In fact, Algorithms 3 and 4 are direct generalizations of Algorithms 1 and 2, respectively. They are all based on the same idea, that is, switching the sequence of slices of an SOLHD to get the BOLHD. Next, we will list some BOLHDs that can be constructed using Algorithms 3 and 4.

4.3 | Some construction results

In the above sections, we have considered how to construct BOLHDs. This section lists some BOLHD (N, q, (m_1, m_2, … , m_q), t)'s that can be constructed by above algorithms. According to the algorithms, to build a BOLHD, we first need an SOLHD and an OLHD. According to the results of Yang et al. (2016), all SOLHD($2^{r+1}s$, 2^r, s)'s are available, where $s \geq 2$, $r \geq 1$. Also, we have a collection of small OLHDs according to Lin et al. (2010) (see Table B4). Based on these results, we can construct many BOLHDs. Some parameters of the constructed BOLHDs are listed in Tables B5 and B6.

Table B5 lists some BOLHD (N, q, (m_1, m_2, … , m_q))'s obtainable by Algorithm 3 when there is no shared factor, where q represents the number of branching factors, s represents the number of levels of each branching factor, $\max\{\Sigma_{i=1}^q m_i\}$ represents the maximum number of nested factors that can be allowed and N represents the number of runs.

When there are shared factors in the BOLHDs to be constructed, the number of experiments will increase relatively. Table B6 lists some small size BOLHD (N, q, (m_1, m_2, … , m_q), t)'s obtainable by Algorithm 4, where q, s, $\max\{\Sigma_{i=1}^q m_i\}$ and N have the same meanings as in Table B5, and t represents the number of shared factors.

5 | CONCLUDING REMARKS

In this paper, we define a new type of design called BOLHD for computer experiments with branching and nested factors. Compared with the BLHDs proposed by Hung et al. (2009), the BOLHDs can guarantee one-dimensional balance and orthogonality for the nested factors no matter at each level of a branching factor or each level combination of branching factors. These good properties can ensure that (1) both the modelling of nested factors at a certain level of a branching factor alone and the modelling of nested factors at a level combination of branching factors can result good parameter estimations; (2) significant factors can be identified. Furthermore, we propose some algorithms for constructing BOLHDs when branching factors have the same number of levels. One advantage of these algorithms is that no computer search is required.

There are some issues worth further discussion. One issue is that this paper only considers the case that branching factors have the same number of levels. When the numbers of levels are not the same, the construction will be more complex. To solve this problem, we can start with the simplest case which assumes that there is a multiple relationship between the numbers of levels of different branching factors. Another issue is that the designs we constructed require more experimental runs due to better properties. How to construct designs with fewer runs and the properties of a BOLHD at the same time deserves further studies.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Grant Nos. 12131001, 11771220 and 11871033), National Ten Thousand Talents Program of China, Fundamental Research Funds for the Central Universities (63211090), Natural Science Foundation of Tianjin (20JCYBJC01050), and the 111 Project (B20016).

DATA AVAILABILITY STATEMENT

Data sharing not applicable - no new data generated

ORCID

Jian-Feng Yang https://orcid.org/0000-0002-2271-4798

REFERENCES

Chen, H., Yang, J. Y., Lin, D. K. J., & Liu, M. Q. (2019). Sliced Latin hypercube designs with both branching and nested factors. *Statistics & Probability Letters*, *146*, 124–131.

Chen, H., Yang, J. Y., & Liu, M. Q. (2021). Construction of improved branching Latin hypercube designs. *Acta Mathematica Scientia*, *41B*(4), 1023–1033.
Deng, X., Hung, Y., & Lin, C. D. (2015). Design for computer experiments with qualitative and quantitative factors. *Statistica Sinica*, *25*, 1567–1581.
Fang, K. T., Li, R., & Sudjianto, A. (2006). *Design and modeling for computer experiments*. New York: CRC Press.
Goos, P., & Jones, B. (2019). Optimal experimental design in the presence of nested factors. *Technometrics*, *61*, 533–544.
Hung, Y., Joseph, V. R., & Melkote, S. N. (2009). Design and analysis of computer experiments with branching and nested factors. *Technometrics*, *51*(4), 354–365.
Lin, C. D., Bingham, D., Sitter, R. R., & Tang, B. (2010). A new and flexible method for constructing designs for computer experiments. *The Annals of Statistics*, *38*, 1460–1477.
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.
Qian, P. Z. G. (2012). Sliced Latin hypercube designs. *Journal of the American Statistical Association*, *107*(497), 393–399.
Yang, J. Y., Chen, H., Lin, D. K. J., & Liu, M. Q. (2016). Construction of sliced maximin-orthogonal Latin hypercube designs. *Statistica Sinica*, *26*, 589–603.

How to cite this article: Wei, Q., Liu, M.-Q., & Yang, J.-F. (2022). Orthogonal designs with branching and nested factors. *Stat*, *11*(1), e447. https://doi.org/10.1002/sta4.447

APPENDIX A: PROOF OF THEOREM 2

For the construction of the design for nested factors, Algorithms 2 and 1 are the same, so here, we only need to prove that D satisfies (3) in Definition 1, that is, (a) D_3 is an OLHD, and (b) for each level-combination of branching factors, the corresponding design points of D_3 form an LHD which is orthogonal to the design of nested factors.

First, we prove that D_3 is an OLHD. According to Algorithm 2, T_i is obtained by row permutation of C_i, $i=1,\dots,s$, and row permutation does not change the orthogonality and one-dimensional space-filling property. So for simplicity, we only need to prove that $D_3' = (C_1^T,\dots,C_s^T)^T$ is an OLHD. To prove that D_3' is an LHD, without loss of generality, we only consider the jth column $D_3'(:,j)$ for any $j=1,\dots,t$. It is obvious that $D_3'(:,j)$ is a permutation on $\{C_1(:,j),\dots,C_s(:,j)\} = \{\frac{-(N-1)s}{2}-\frac{s-1}{2},\dots,\frac{-(N-1)s}{2}+\frac{s-1}{2},\dots,\frac{(N-1)s}{2}-\frac{s-1}{2},\dots,\frac{(N-1)s}{2}+\frac{s-1}{2}\} = \{-\frac{Ns-1}{2},\dots,\frac{Ns-1}{2}\}$. This shows that D_3' is an LHD. To prove that D_3' is orthogonal, without loss of generality, we need to consider the orthogonality of any two columns of D_3', 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 q_1 and q_2 from Q, respectively, where $e_i=(e_{1i},\dots,e_{si})^T$, and $q_i=(q_{1i},\dots,q_{Ni})^T$ for $i=1,2$. Then,

$$d_1 = (e_{11}+sq_{11},\dots,e_{11}+sq_{N1},\dots,e_{s1}+sq_{11},\dots,e_{s1}+sq_{N1})^T,$$
$$d_2 = (e_{12}+sq_{12},\dots,e_{12}+sq_{N2},\dots,e_{s2}+sq_{12},\dots,e_{s2}+sq_{N2})^T.$$

Since $\mathrm{corr}(e_1,e_2)=0$ and $\mathrm{corr}(q_1,q_2)=0$, that is,

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

where $\bar{e}_1=\frac{1}{s}\sum_{i=1}^{s}e_{i1}$ and $\bar{e}_2$, $\bar{q}_1$, $\bar{q}_2$ are similarly defined, then the numerator of $\mathrm{corr}(d_1,d_2)$ equals

$$\begin{aligned}
&\sum_{i=1}^{s}\sum_{j=1}^{N}\left((e_{i1}+sq_{j1})-(\bar{e}_1+s\bar{q}_1)\right)\left((e_{i2}+sq_{j2})-(\bar{e}_2+s\bar{q}_2)\right)\\
=&\sum_{i=1}^{s}\sum_{j=1}^{N}\left((e_{i1}-\bar{e}_1)+s(q_{j1}-\bar{q}_1)\right)\left((e_{i2}-\bar{e}_2)+s(q_{j2}-\bar{q}_2)\right)\\
=&\sum_{i=1}^{s}\sum_{j=1}^{N}(e_{i1}-\bar{e}_1)(e_{i2}-\bar{e}_2)+s\sum_{i=1}^{s}\sum_{j=1}^{N}(e_{i1}-\bar{e}_1)(q_{j2}-\bar{q}_2)\\
+&\ s\sum_{i=1}^{s}\sum_{j=1}^{N}(q_{j1}-\bar{q}_1)(e_{i2}-\bar{e}_2)+s^2\sum_{i=1}^{s}\sum_{j=1}^{N}(q_{j1}-\bar{q}_1)(q_{j2}-\bar{q}_2)\\
=&\ 0.
\end{aligned}$$

Thus, $\mathrm{corr}(d_1,d_2)=0$. This shows that D_3' is orthogonal.

Next, we prove that D_3 satisfies (b). According to Algorithm 2, for any fixed level-combination of z_1 and z_2, the design for nested factors corresponds to a certain (a_{i_0}, b_{i_0}), and the design for shared factors corresponds to a certain c_{k_0,i_0} with $k_0, i_0 \in \{1, \ldots, s\}$. First, we need to prove that c_{k_0,i_0} is an LHD, which is obvious because the structure of c_{k_0,i_0} follows the structure of Q_{i_0} for $i_0 \in \{1, \ldots, s\}$. Then, we need to prove that any column from c_{k_0,i_0} is orthogonal to any column from (a_{i_0}, b_{i_0}). Without loss of generality, let $i_0 = k_0 = 1$, and we prove that the first column of $c_{1,1}$ denoted by $g_1 = (g_{11}, \ldots, g_{n1})^T$ is orthogonal to the first column of $(a_1, b_1) = L_1$ denoted by $l_1 = (l_{11}, \ldots, l_{n1})^T$. According to Step 4, $g_1 = (e_{11} + sq_{11}, \ldots, e_{11} + sq_{n1})^T = e_{11} * \mathbf{1} + s * c_1$, where $c_1 = (q_{11}, \ldots, q_{n1})^T$ is the first column in the first slice Q_1 of Q. Since L and Q are all from the SOLHD P, it is clear that $\mathrm{corr}(c_1, l_1) = 0$, that is,

$$\sum_{i=1}^{n} (q_{i1} - \bar{c}_1)(l_{i1} - \bar{l}_1) = 0,$$

where $\bar{c}_1 = \frac{1}{n}\sum_{i=1}^{n} q_{i1}$ and $\bar{l}_1$ is defined similarly, then, the numerator of $\mathrm{corr}(g_1, l_1)$ equals

$$\begin{aligned} & \sum_{i=1}^{n} (g_{i1} - \bar{g}_1)(l_{i1} - \bar{l}_1) \\ = & \sum_{i=1}^{n} (e_{11} + sq_{i1} - (e_{11} + s\bar{c}_1))(l_{1i} - \bar{l}_1) \\ = & s\sum_{i=1}^{n} (q_{i1} - \bar{c}_1)(l_{i1} - \bar{l}_1) \\ = & 0. \end{aligned}$$

Thus, $\mathrm{corr}(c_1, l_1) = 0$. This completes the proof of (b).

APPENDIX B: THE P AND D_3 IN EXAMPLE 3

$P = (P_1^T, P_2^T, P_3^T, P_4^T)^T$, where

$$P_1 = \begin{pmatrix} 0.5 & 4.5 & 8.5 & 12.5 & -0.5 & -4.5 & -8.5 & -12.5 \\ 4.5 & -0.5 & 12.5 & -8.5 & -4.5 & 0.5 & -12.5 & 8.5 \\ -12.5 & -8.5 & 4.5 & 0.5 & 12.5 & 8.5 & -4.5 & -0.5 \\ 8.5 & -12.5 & -0.5 & 4.5 & -8.5 & 12.5 & 0.5 & -4.5 \end{pmatrix}^T, P_2 = \begin{pmatrix} 1.5 & 5.5 & 9.5 & 13.5 & -1.5 & -5.5 & -9.5 & -13.5 \\ 5.5 & -1.5 & 13.5 & -9.5 & -5.5 & 1.5 & -13.5 & 9.5 \\ -13.5 & -9.5 & 5.5 & 1.5 & 13.5 & 9.5 & -5.5 & -1.5 \\ 9.5 & -13.5 & -1.5 & 5.5 & -9.5 & 13.5 & 1.5 & -5.5 \end{pmatrix}^T,$$

$$P_3 = \begin{pmatrix} 2.5 & 6.5 & 10.5 & 14.5 & -2.5 & -6.5 & -10.5 & -14.5 \\ 6.5 & -2.5 & 14.5 & -10.5 & -6.5 & 2.5 & -14.5 & 10.5 \\ -14.5 & -10.5 & 6.5 & 2.5 & 14.5 & 10.5 & -6.5 & -2.5 \\ 10.5 & -14.5 & -2.5 & 6.5 & -10.5 & 14.5 & 2.5 & -6.5 \end{pmatrix}^T, P_4 = \begin{pmatrix} 3.5 & 7.5 & 11.5 & 15.5 & -3.5 & -7.5 & -11.5 & -15.5 \\ 7.5 & -3.5 & 15.5 & -11.5 & -7.5 & 3.5 & -15.5 & 11.5 \\ -15.5 & -11.5 & 7.5 & 3.5 & 15.5 & 11.5 & -7.5 & -3.5 \\ 11.5 & -15.5 & -3.5 & 7.5 & -11.5 & 15.5 & 3.5 & -7.5 \end{pmatrix}^T.$$

$D_3 = (T_1^T, T_2^T)^T$, where

$$\begin{aligned} T_1 = & (16.5 - 25.5 - 1.58.5 - 17.524.50.5 - 9.518.5 - 27.5 - 3.510.5 - 19.526.52.5 - 11.5 \\ & 20.5 - 29.5 - 5.512.5 - 21.528.54.5 - 13.522.5 - 31.5 - 7.514.5 - 23.530.56.5 - 15.5)^T, \\ T_2 = & (23.5 - 30.5 - 6.515.5 - 22.531.57.5 - 14.521.5 - 28.5 - 4.513.5 - 20.529.55.5 - 12.5 \\ & 19.5 - 26.5 - 2.511.5 - 18.527.53.5 - 10.517.5 - 24.5 - 0.59.5 - 16.525.51.5 - 8.5)^T. \end{aligned}$$

TABLE B1 An SLHD-based BLHD with two branching factors each of two levels

Run	z_1	z_2	$v_1^{z_1} \ldots v_{m_1}^{z_1}$	$v_1^{z_2} \ldots v_{m_2}^{z_2}$	$x_1 \ldots x_t$
1	1	1	SLHD(n_1, m_1,2)	SLHD(n_2, m_2,2) first slice	SLHD(N, t,4)
⋮	1	2		SLHD(n_2, m_2,2) first slice	
⋮	2	1	SLHD(n_1, m_1,2)	SLHD(n_2, m_2,2) second slice	
N	2	2		SLHD(n_2, m_2,2) second slice	

TABLE B2 A BLHD with two branching factors each of two levels

Run	z_1	z_2	$v_1^{z_1}...v_{m_1}^{z_1}$	$v_1^{z_2}...v_{m_2}^{z_2}$	$x_1...x_t$
1	1	1	LHD(n_1,m_1)	LHD(n_2,m_2) first half	LHD(N,t)
⋮	1	2		LHD(n_2,m_2) first half	
⋮	2	1	LHD(n_1,m_1)	LHD(n_2,m_2) second half	
N	2	2		LHD(n_2,m_2) second half	

TABLE B3 SLHD-based BLHD with two branching factors each of s levels

Run	z_1	z_2	$v_1^{z_1}...v_{m_1}^{z_1}$	$v_1^{z_2}...v_{m_2}^{z_2}$	$x_1...x_t$
1	1	1	$\text{SLHD}_1(n,m_1,s)$'s slice 1	$\text{SLHD}_1(n,m_2,s)$'s slice 1	SLHD(N,t,s^2)
⋮	⋮	⋮	⋮	⋮	
s	1	s	$\text{SLHD}_1(n,m_1,s)$'s slice s	$\text{SLHD}_s(n,m_2,s)$'s slice 1	
⋮	⋮	⋮	⋮	⋮	
N	s	s	$\text{SLHD}_s(n,m_1,s)$'s slice s	$\text{SLHD}_s(n,m_2,s)$'s slice s	

TABLE B4 The maximum number of columns m in OLHD (n,m) for $1 \le n \le 21$

n	2	3	4	5	7	8	9	11	12	13	15	16	17	19	20	21
m	1	1	2	2	3	4	5	7	6	6	6	12	6	6	6	6

TABLE B5 Some BOLHD $(N,q,(m_1,m_2,...,m_q))$'s obtainable by Algorithm 3

q	2	2	2	3	2	3	4	2	3	4	3
s	2	2	2	2	3	2	2	3	3	2	3
$\max\{\Sigma_{i=1}^{q} m_i\}$	2	4	8	4	4	8	4	8	4	8	8
N	16	32	64	64	72	128	128	144	216	256	432

TABLE B6 Some BOLHD $(N,q,(m_1,m_2,...,m_q),t)$'s obtainable by Algorithm 4

q	2	2	3	2	2	3	2	3	2	3	2	2	3
s	2	2	2	3	4	2	3	3	4	3	7	8	4
$\max\{\Sigma_{i=1}^{q} m_i\}$	3	7	3	3	2	7	7	3	6	7	5	4	6
t	1	1	1	1	2	1	1	1	2	1	3	4	2
N	32	64	64	72	128	128	144	216	256	432	784	1024	1024