Received: 24 September 2024 | Revised: 18 March 2025 | Accepted: 21 March 2025

DOI: 10.1111/sjos.12786

ORIGINAL ARTICLE

Scandinavian Journal of Statistics

Enhanced branching Latin hypercube design and its application in automatic algorithm configuration

Bing Wen[1] | **Sumin Wang**[2] | **Fasheng Sun**[1]

[1]KLAS and School of Mathematics and Statistics, Northeast Normal University, Changchun, China

[2]Department of Statistics, School of Sciences, Hebei University of Technology, Tianjin, China

Correspondence
Fasheng Sun, KLAS and School of Mathematics and Statistics, Northeast Normal University, Changchun 130022, China.
Email: sunfs359@nenu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 12371259, 12401324; Fundamental Research Funds for the Central Universities, Grant/Award Number: 2412023YQ003

Abstract

Designing experiments that involve branching and nested factors is challenging due to the complex relationships between these factors. Identification of optimal settings requires designs with good stratification properties for both nested and shared factors. To meet this requirement, we defined a type of enhanced branching Latin hypercube designs and developed several novel construction methods by integrating orthogonal arrays and sliced Latin hypercube designs. These designs exhibit attractive low-dimensional stratification properties and perform well in terms of column correlation. Additionally, the size of each design can be flexibly chosen based on the trade-off between the experimental budget and estimation accuracy. The simulation results demonstrate that the proposed design method exhibits significant superiority in terms of design metrics and estimation accuracy. Furthermore, we showcase the application of these designs in initializing automatic algorithm configuration. The proofs and additional design tables are provided in the Appendix.

KEYWORDS

branching factor, computer experiment, nested factor, orthogonal array, sliced Latin hypercube design, space-filling design

1 | INTRODUCTION

Computer experiments are extensively used in investigating complex phenomena across various fields. Latin hypercube designs are commonly used in computer experiments. However, these methods are not suitable for experiments involving branching and nested factors (Hung et al., 2009). In these experiments, some factors exist only within one level of another factor, such factors are called nested factors. Factors within which other factors are nested are called branching factors. The remaining factors that are unrelated to the level of branching factors are called shared factors. For example, two surface preparation methods are used in printed circuit board manufacturing: mechanical scrubbing and chemical treatment. Mechanical scrubbing can be optimized by changing the pressure of the scrub, and chemical treatment can be optimized by changing the micro-etch rate. The surface preparation method here is a branching factor, and the pressure and micro-etch rate are nested factors (Hung et al., 2009).

Because the nested factors differ at different levels of a branching factor, their effects will also be different. It is challenging to construct designs involving both branching and nested factors. Hung et al. (2009) proposed a novel design approach for simultaneously identifying the optimal settings of branching, nested, and shared factors. They defined a class of designs called branching Latin hypercube designs (BLHDs), $D = (D_1, D_2, D_3)$, where D_1, D_2, and D_3 represent the designs for branching factors, nested factors, and shared factors, respectively. A BLHD is also abbreviated as $BLHD(N, q + m + r)$, assuming that the experiment incorporates q branching factors, m nested factors, and r shared factors. Furthermore, it is assumed that the ith branching factor encompasses m_i nested factors, leading to the total number of nested factors being $m = m_1 + \cdots + m_q$. Hung et al. (2009) suggested that design D should satisfy the following properties: (1) the design D_1 for branching factors is an orthogonal array (OA), (2) the design for shared factors D_3 is a Latin hypercube design (LHD), and (3) the design for the corresponding nested factors after level collapse under each level of any branching factor is an LHD. For example, the design shown in Table 1 is a BLHD with 16 runs and 5 factors, where z_1 and z_2 represent two branching factors; $v_1^{z_1}$ and $v_1^{z_2}$ are nested factors within each level of z_1 and z_2, respectively; and x_1 represents a shared factor. In addition, Goos and Jones (2019) presented a general method for constructing appropriate models and showed how to generate optimal designs given these models. Wei et al. (2022) considered the orthogonality within BLHDs and constructed a type of design called branching orthogonal Latin hypercube design. In addition to computer experiments, the BLHDs can also be used in the initialization of automatic algorithm configuration, which can be referred to Wessing and López-Ibáñez (2019). Besides, potential applications of BLHDs include Bayesian optimization, which can be used as a space-filling design to generate the initial samples and evaluate to obtain the training set (Guo et al., 2023; Wang & Dowling, 2022; Zhang et al., 2020).

To the best of our knowledge, existing research has not thoroughly considered the space-filling properties of all nested and shared factors across specific combinations of branching factor levels. This aspect is pivotal for enhancing the accuracy of estimating interaction effects. Furthermore, current methodologies often lack flexibility and struggle to adjust the number of experiments freely. In response to these challenges, we define a type of enhanced branching Latin hypercube designs, abbreviated as EBLHDs. And we develop a novel construction approach for EBLHDs by integrating OAs and SLHDs. The proposed method ingeniously uses OAs as the basic framework and combines them with SLHDs to jointly construct efficient designs. For a given OA, we can flexibly choose SLHDs of different scales to meet practical needs based on the trade-off between the experimental budget and estimation accuracy. This flexibility serves as one of the driving forces

TABLE 1 A design $BLHD(16, 2+2+1)$.

	D_1		D_2		D_3
Run	z_1	z_2	$v_1^{z_1}$	$v_1^{z_2}$	x_1
1	0	0	0	3	0
2	0	0	2	5	12
3	0	0	4	1	8
4	0	0	6	7	4
5	0	1	3	0	10
6	0	1	1	6	2
7	0	1	5	2	6
8	0	1	7	4	14
9	1	0	3	0	11
10	1	0	1	6	3
11	1	0	5	2	7
12	1	0	7	4	15
13	1	1	0	3	1
14	1	1	2	5	13
15	1	1	4	1	9
16	1	1	6	7	5

behind the work presented in this article. Additionally, we provide a method to construct the required SLHDs with excellent stratification properties. Through a carefully designed hierarchical structure, we ensure exceptional space-filling, making the selected design points more representative and comprehensive. This design strategy not only enhances the accuracy and reliability of parameter estimation but also improves the precision and robustness of the estimated models. Beyond its application in computer experiments, this design can also be used for the initialization of automatic algorithm configuration.

To enhance the space-filling properties of all nested and shared factors across specific level combinations of branching factors, we define EBLHD as presented in Definition 1. Note that the BLHD defined in Hung et al. (2009) does not satisfy condition (4). Additionally, marginally coupled designs (Deng et al., 2015) are suitable for computer experiments with both qualitative and quantitative factors, which is different from the scenario in which EBLHDs are applied. Specifically, marginal coupled design is a special case of EBLHD, where there are no nested factors in the experiment, and the branching factors are qualitative while the shared factors are quantitative. As an example, we can refer to the $BLHD(16, 2+2+1)$ in Table 1, which is essentially an EBLHD.

Definition 1. $D = (D_1, D_2, D_3)$ is called an enhanced branching Latin hypercube design (EBLHD) if it satisfies the following:

1. The design for branching factors D_1 is an OA.
2. The design for shared factors D_3 is an LHD.

3. Under each level of any branching factor, the design for the corresponding nested factors after level collapse is an LHD.
4. Under each level combination of all branching factors, the design for the corresponding nested factors and shared factors (D_2, D_3) after level collapse is an LHD.

The remainder of this article is organized as follows: Section 2 introduces basic definitions and notations. Section 3 presents two frameworks for constructing EBLHDs with equal-level branching factors. In addition, to enhance the space-filling performance of EBLHDs, we introduce several construction methods for SLHDs with low-dimensional stratification. Section 4 discusses the generation of EBLHDs with mixed-level branching factors. Section 5 presents the performance of our proposed designs in simulation studies. Then, Section 6 demonstrates the application of EBLHDs in the initialization of automatic algorithm configuration. Finally, Section 7 provides a summary. All proofs and some construction results are included in the Appendix A–C.

2 | NOTATION AND PRELIMINARIES

A mixed orthogonal array $MOA(n, s_1^{k_1} s_2^{k_2} \dots s_v^{k_v}, t)$ is an array of $n \times m$, where $m = k_1 + \cdots + k_v$ is the total number of factors, in which the first k_1 columns have entries from $\mathbb{Z}_{s_1}$, the next k_2 columns have entries from $\mathbb{Z}_{s_2}$, and so on, with the property that each possible level combination in any $n \times t$ sub-array appears with the same frequency, where $\mathbb{Z}_s = \{0, 1, \dots, s-1\}$ (Hedayat et al., 1999). If all s_j's are equal to s, the array is called a symmetric orthogonal array, denoted by $OA(n, m, s, t)$. The array is called completely resolvable, denoted by $CROA(n, m, s, 2)$, if it can be expressed as $A = (A_1^T, \dots, A_{n/s}^T)^T$ such that each A_i is an $OA(s, m, s, 1)$. For simplicity, we use the symbol $\cdots \in \varepsilon$ to indicate that a certain matrix or column belongs to a specified design class. For instance, $A \in OA(n, m, s, 2)$ implies that A is an orthogonal array of type $OA(n, m, s, 2)$. The following conclusion can be directly derived from He et al. (2018).

Proposition 1. *Suppose that* $(a_1, a_2) \in OA(n, 2, s, 2)$ *and* $(b_1, b_2) \in MOA(n, \alpha^1\beta^1, 2)$. *Then,* $(\alpha a_1 + b_1, \beta a_2 + b_2)$ *can achieve* $\alpha s \times s$ *and* $s \times \beta s$ *stratification in two dimensions if* (a_1, a_2, b_1) *and* (a_1, a_2, b_2) *constitute* $MOA(n, s^2\alpha^1, 3)$ *and* $MOA(n, s^2\beta^1, 3)$, *respectively.*

An orthogonal array (OA) of type $OA(n, m, n, 1)$ is referred to as a Latin hypercube design (LHD), proposed by McKay et al. (1979), denoted by $LHD(n, m)$. An $LHD(n, m)$ with $n = sq$ is termed a sliced Latin hypercube design represented as $SLHD(n, m, s)$, proposed by Qian (2012), if it can be partitioned into s slices $L = (L_1^T, \dots, L_s^T)^T$ such that each $\left\lfloor \frac{L_i}{s} \right\rfloor$ is an $LHD(q, m)$ for $i = 1, \dots, s$. To guarantee higher-dimensional stratification for LHDs, Tang (1993) proposed orthogonal array-based Latin hypercube designs (OA-based LHDs). They demonstrated that, when used for integration, a sampling scheme based on OA-based LHDs with higher-dimensional stratification offers a significant improvement over traditional Latin hypercube sampling. In this article, we use the following methods to generate OA-based LHDs. For $A \in OA(n, m, s, t)$, we replace n/s occurrences of j in each column with a permutation of $jn/s + \mathbb{Z}_{n/s}$ for $0 \le j \le s-1$.

Additionally, there are several symbols used in the construction. For $n_1 \times m$ matrix $A = (a_1, \dots, a_m)$ and $n_2 \times m$ matrix $B = (b_1, \dots, b_m)$, we define $A \oplus_c B = (a_1 \oplus b_1, \dots, a_m \oplus b_m)$, an $n_1n_2 \times m$ matrix, where $\oplus$ is the Kronecker sum. $A \setminus a_i$ includes all the remaining columns in A except a_i. For any two n-run designs D and D', $\rho_{\max}(D)$ denotes the maximum correlation coefficient of D, and $\rho_{\max}(D, D')$ represents the maximum correlation coefficient of a column from D and a column from D'.

This article assumes that there are q branching factors, denoted by $\mathbf{z} = (z_1, \ldots, z_q)$, and m_u factors nested under each level of branching factor z_u, expressed as $\mathbf{v}^{z_u} = (v_1^{z_u}, \ldots, v_{m_u}^{z_u})$ for $u = 1, \ldots, q$. Additionally, there are r shared factors, denoted by $\mathbf{x} = (x_1, \ldots, x_r)$. The notation $EBLHD(N, q + m + r)$ represents an EBLHD with N-run, q branching factors, $m = \Sigma_{u=1}^{q} m_u$ nested factors and r shared factors. This design can be expressed as $D = (D_1, D_2, D_3)$, where D_1, D_2, and D_3 represent the designs for the branching factors, nested factors, and shared factors, respectively. In this article, the branching factors are assumed to be qualitative factors, and D_1 is chosen to be an OA. Consequently, our primary focus is on the construction of D_2 and D_3.

3 | CONSTRUCTION OF EBLHDS WITH EQUAL-LEVEL BRANCHING FACTORS

This section describes the scenario in which all branching factors are equal-level, and the scenarios involving mixed-level branching factors are described in the next section.

3.1 | Construction of EBLHDs

We construct a type of $EBLHD(N = s^2 n_2, q + m + r)$ in the presence of $OA(n_1 = s^2, q + 1, s, 2)$ and $SLHD(sn_2, k, s)$, where $m = m_1 + \cdots + m_q$, $k = m + r$, s represents the level of branching factor. First, an illustrative example is provided in Example 1, followed by the general construction process in Algorithm 1.

> **Example 1.** We assume that there are two branching factors z_1 and z_2 with levels from $\mathbb{Z}_3$. Each branching factor corresponds to a nested factor, denoted as $v_1^{z_1}$ and $v_1^{z_2}$, respectively. There is also one shared factor. In this case, $s = 3$, $q = 2$, $m_1 = m_2 = 1$, $m = 2$, $r = 1$, and $k = 3$.

Algorithm 1. $EBLHD(N = s^2 n_2, q + m + r)$

Input $A \in OA(n_1 = s^2, q + 1, s, 2)$, $L \in SLHD(sn_2, k = m + r, s)$, $B \in OA(n_1, r, s, 1)$.

Step 1. Denote the columns of A by $A = (a_1, \ldots, a_{q+1})$. For branching factors, we define $D_1 = (A \setminus a_{q+1}) \otimes \mathbf{1}_{n_2}$.

Step 2. Express the slices of L as $L = (L_0^T, \ldots, L_{s-1}^T)^T$ and divide each $L_i = (P_i, Q_i)$, where P_i and Q_i contain m and r columns, respectively.

Step 3. For $i = 0, \ldots, s - 1$, replace the level i in a_{q+1} with slice L_i to obtain the design $(D_2, \tilde{D}_3)$, where D_2 and $\tilde{D}_3$ consist of m and r columns, respectively.

Step 4. Define $D_3 = s\tilde{D}_3 + B \otimes \mathbf{1}_{n_2}$, where $B = (b_1, \ldots, b_r)$, each pair column (b_i, a_{q+1}) forms an OA$(n_1, 2, s, 2)$, $i = 1, \ldots, r$. Typically, we can directly select b_i from $A \setminus a_{q+1}$.

Output Design $D = (D_1, D_2, D_3)$.

TABLE 2 $OA(9,3,3,2)$ and design $D = (D_1, D_2, D_3)$ in Example 1.

$\boldsymbol{A}$: $\boldsymbol{OA(9,3,3,2)}$							
$\boldsymbol{a_1}$	$\boldsymbol{a_2}$	$\boldsymbol{a_3}$	$\boldsymbol{D_1}$		$\boldsymbol{D_2}$	$\tilde{\boldsymbol{D}}_3$	$\boldsymbol{D_3}$
0	0	0	$\mathbf{0}_9$	$\mathbf{0}_9$	P_0	Q_0	$0 + 3Q_0$
0	1	1	$\mathbf{0}_9$	$\mathbf{1}_9$	P_1	Q_1	$1 + 3Q_1$
0	2	2	$\mathbf{0}_9$	$\mathbf{2}_9$	P_2	Q_2	$2 + 3Q_2$
1	0	1	$\mathbf{1}_9$	$\mathbf{0}_9$	P_1	Q_1	$0 + 3Q_1$
1	1	2	$\mathbf{1}_9$	$\mathbf{1}_9$	P_2	Q_2	$1 + 3Q_2$
1	2	0	$\mathbf{1}_9$	$\mathbf{2}_9$	P_0	Q_0	$2 + 3Q_0$
2	0	2	$\mathbf{2}_9$	$\mathbf{0}_9$	P_2	Q_2	$0 + 3Q_2$
2	1	0	$\mathbf{2}_9$	$\mathbf{1}_9$	P_0	Q_0	$1 + 3Q_0$
2	2	1	$\mathbf{2}_9$	$\mathbf{2}_9$	P_1	Q_1	$2 + 3Q_1$

Considering $A \in OA(9,3,3,2)$ in Table 2 and $L \in SLHD(27,3,3)$ in Equation (1) (constructed by Example 4), respectively, our construction is as follows:

$$L^T = \left(\begin{array}{ccccccccc|ccccccccc|ccccccccc} 0 & 9 & 18 & 6 & 15 & 24 & 3 & 12 & 21 & 2 & 11 & 20 & 8 & 17 & 26 & 5 & 14 & 23 & 1 & 10 & 19 & 7 & 16 & 25 & 4 & 13 & 22 \\ 4 & 13 & 22 & 10 & 19 & 1 & 25 & 7 & 16 & 14 & 23 & 5 & 20 & 2 & 11 & 8 & 17 & 26 & 21 & 3 & 12 & 0 & 9 & 18 & 15 & 24 & 6 \\ 0 & 18 & 9 & 12 & 3 & 21 & 24 & 15 & 6 & 10 & 1 & 19 & 22 & 13 & 4 & 7 & 25 & 16 & 20 & 11 & 2 & 5 & 23 & 14 & 17 & 8 & 26 \end{array}\right) \tag{1}$$

Step 1. We consider $A \in OA(9,3,3,2)$ with columns $A = (a_1, a_2, a_3)$ as in Table 2. For branching factors, we define $D_1 = (a_1, a_2) \otimes \mathbf{1}_9$, where $\boldsymbol{i}_j$ represents a j-dimensional column vector with all elements equal to i.

Step 2. We suppose $L = (L_0^T, L_1^T, L_2^T)^T$ in Equation (1) and divide each $L_i = (P_i, Q_i)$, where P_i and Q_i contain 2 and 1 columns, respectively. P_i and Q_i are the building materials for the nested factors and shared factors, respectively.

Step 3. For $i = 0, 1, 2$, level i in a_3 is substituted by slice L_i to obtain the design $(D_2, \tilde{D}_3)$, where D_2 and $\tilde{D}_3$ consist of 2 and 1 columns, respectively.

Step 4. We generate $D_3 = 3\tilde{D}_3 + a_2 \otimes \mathbf{1}_9$.

The generating design $D = (D_1, D_2, D_3)$ is an $EBLHD(81, 2+2+1)$. Upon verification, it can be confirmed that for any specified level combination of all branching factors, the corresponding design points of both nested and shared factors can achieve 3×3 stratification in any two dimensions. Furthermore, under each level of any branching factor, the corresponding design points of (D_2, D_3) can attain 9×3 and 3×9 stratification in two dimensions. The stratification properties will be described in detail in the subsequent sections.

Remark 1. In practical scenarios, a suitable choice for L can be selected based on the experimental budget. This underlines the high flexibility and broad applicability of our construction method. For example,

(i) if we replace L in Example 1 with $SLHD(12,3,3)$ in Equation (4), we can obtain $EBLHD(36, 2+2+1)$. The design satisfies the following: for any given level combination of branching factors, the corresponding design points involving nested factors and shared factors can achieve 2×2 stratification in any two dimensions.

TABLE 3 $MOA(18, 3^3 6^1, 2)$.

a_1	a_2	a_3	a_4
0	0	0	0
0	1	2	1
0	2	1	2
0	1	1	3
0	2	0	4
0	0	2	5
1	1	1	0
1	2	0	1
1	0	2	2
1	2	2	3
1	0	1	4
1	1	0	5
2	2	2	0
2	0	1	1
2	1	0	2
2	0	0	3
2	1	2	4
2	2	1	5

(ii) if we replace L in Example 1 with any three columns of $SLHD(24, 4, 3)$ in Equation (5), we can obtain $EBLHD(72, 2 + 2 + 1)$. The design satisfies the following: for any given level combination of branching factors, the corresponding design points involving nested factors and shared factors can achieve $2 \times 2 \times 2$ stratification in three dimensions.

Now, we show the general construction method with q s-level branching factors in Algorithm 1. With this algorithm, a class of design $EBLHD(N = s^2 n_2, q + m + r)$ can be obtained, as in Theorem 1. Its proof is found in Appendix A.1.

Theorem 1. *Given $A \in OA(n_1 = s^2, q + 1, s, 2)$ and $L \in SLHD(sn_2, k = m + r, s)$, the design D obtained by Algorithm 1 is an $EBLHD(N = s^2 n_2, q + m + r)$.*

Next, we discuss the generalized form of Theorem 1, which involves constructing an EBLHD using an MOA with λs^2 runs. Let us first illustrate the construction steps through an example, followed by presenting the general construction algorithm.

Example 2. We assume there are three branching factors with levels from $\mathbb{Z}_3$ that include 2, 1, and 1 nested factors. Additionally, the experiment involves two shared factors. Here, $s = 3$, $q = 3$, $m_1 = 2$, $m_2 = 1$, $m_3 = 1$, $r = 2$, $m = 4$, and $k = 6$.

We use $MOA(18, 3^3 6^1)$ in Table 3 and $SLHD(108, 6, 6)$ in equation B1, generated by Example 4, to construct an $EBLHD(324, 3 + 4 + 2)$, as shown in Table 4.

TABLE 4 Design $D = (D_1, D_2, D_3)$ in Example 2.

D_1			D_2	$\tilde{D}_3$	D_3	
0	**0**	**0**	P_0	Q_0	$3Q_{01}+0$	$3Q_{02}+0$
0	**1**	**2**	P_1	Q_1	$3Q_{11}+0$	$3Q_{12}+1$
0	**2**	**1**	P_2	Q_2	$3Q_{21}+0$	$3Q_{22}+2$
0	**1**	**1**	P_3	Q_3	$3Q_{31}+0$	$3Q_{32}+1$
0	**2**	**0**	P_4	Q_4	$3Q_{41}+0$	$3Q_{42}+2$
0	**0**	**2**	P_5	Q_5	$3Q_{51}+0$	$3Q_{52}+0$
1	**1**	**1**	P_0	Q_0	$3Q_{01}+1$	$3Q_{02}+1$
1	**2**	**0**	P_1	Q_1	$3Q_{11}+1$	$3Q_{12}+2$
1	**0**	**2**	P_2	Q_2	$3Q_{21}+1$	$3Q_{22}+0$
1	**2**	**2**	P_3	Q_3	$3Q_{31}+1$	$3Q_{32}+2$
1	**0**	**1**	P_4	Q_4	$3Q_{41}+1$	$3Q_{42}+0$
1	**1**	**0**	P_5	Q_5	$3Q_{51}+1$	$3Q_{52}+1$
2	**2**	**2**	P_0	Q_0	$3Q_{01}+2$	$3Q_{02}+2$
2	**0**	**1**	P_1	Q_1	$3Q_{11}+2$	$3Q_{12}+0$
2	**1**	**0**	P_2	Q_2	$3Q_{21}+2$	$3Q_{22}+1$
2	**0**	**0**	P_3	Q_3	$3Q_{31}+2$	$3Q_{32}+0$
2	**1**	**2**	P_4	Q_4	$3Q_{41}+2$	$3Q_{41}+1$
2	**2**	**1**	P_5	Q_5	$3Q_{51}+2$	$3Q_{52}+2$

Note: The bold **i** represents an 18-dimensional column vector.

Step 1. We suppose that $A = (a_1, a_2, a_3, a_4) \in MOA(18, 3^3 6^1, 2)$ as shown in Table 3. For the branching factors, we define D_1 as $(A \setminus a_4) \otimes \mathbf{1}_{18}$.

Step 2. The slices of L in equation B1 are expressed as $L = (L_0^T, \ldots, L_5^T)^T$. Each slice is divided into $L_i = (P_i, Q_i)$, where P_i and Q_i each contain 4 and 2 columns, respectively.

Step 3. For $i = 0, \ldots, 5$, level i in a_4 is replaced by slice L_i to obtain the design $(D_2, \tilde{D}_3)$, where D_2 and $\tilde{D}_3$ contain 4 and 2 columns, respectively.

Step 4. We generate $D_3 = 3\tilde{D}_3 + B \otimes \mathbf{1}_{18}$, where $B = (a_1, a_2)$.

The final design $D = (D_1, D_2, D_3)$ is shown in Table 4. Here, $Q_i = (Q_{i1}, Q_{i2})$, and each Q_{ij} consists of one column. It can be verified that (1) under each level combination of all branching factors, the corresponding design points of (D_2, D_3) can achieve stratification on 3×3 grids in any two dimensions; and (2) under each level of any branching factor, the corresponding design points of (D_2, D_3) can achieve stratification on 18×3 and 3×18 grids in any two dimensions. The stratification properties will be discussed later.

Next, Algorithm 2 presents the general construction using an MOA with λs^2 runs. Using Algorithm 2, a type of design $EBLHD(N = \lambda s^2 n_2, q + m + r)$ can be obtained, as shown in Theorem 2.

Theorem 2. *Given $A \in MOA(n_1 = \lambda s^2, s^q (\lambda s)^1, 2)$ and $L \in SLHD(\lambda s n_2, \ k = m + r, \lambda s)$, the design D derived from Algorithm 2 is an $EBLHD(N = \lambda s^2 n_2, q + m + r)$.*

Its proof is similar to Theorem 1, so we omit it here. Based on the structure of the design, it is not difficult to obtain Proposition 2, which evidently holds.

Proposition 2. *Suppose that L in Theorems 1 and 2 can achieve stratification on $\alpha \times \beta$ and $\beta \times \alpha$ grids in two dimensions. Then, the corresponding design points in (D_2, D_3)*

Algorithm 2. *EBLHD*$(N = \lambda s^2 n_2, q + m + r)$

Input $A \in MOA(n_1 = \lambda s^2, s^q(\lambda s), 2)$, $L \in SLHD(\lambda s n_2, k = m + r, \lambda s)$, $B \in OA(n_1, r, s, 1)$.

Step 1. Denote the columns of A by $A = (a_1, \dots, a_{q+1})$, where a_{q+1} is a λs-level column. For branching factors, we define $D_1 = (A \setminus a_{q+1}) \otimes \mathbf{1}_{n_2}$.

Step 2. Express the slices of L as $L = (L_0^T, \dots, L_{\lambda s-1}^T)^T$ and divide each $L_i = (P_i, Q_i)$, where P_i and Q_i contain m and r columns, respectively.

Step 3. For $i = 0, \dots, \lambda s - 1$, replace the level i in a_{q+1} with slice L_i to obtain $(D_2, \tilde{D}_3)$, where D_2 and $\tilde{D}_3$ consist of m and r columns, respectively.

Step 4. Construct $D_3 = s\tilde{D}_3 + B \otimes \mathbf{1}_{n_2}$, where $B = (b_1, \dots, b_r)$ and each pair column (b_i, a_{q+1}) forms an $OA(n_1, s^1(\lambda s)^1, 2)$, $i = 1, \dots, r$. For convenience, we can directly select b_i from $A \setminus a_{q+1}$.

Output Design $D = (D_1, D_2, D_3)$.

under each level of each branching factor can achieve $\alpha \times \beta$ and $\beta \times \alpha$ stratification in two dimensions.

The following Proposition 3 presents the properties of (D_2, D_3) in Theorems 1 and 2.

Proposition 3. *Let (D_2, D_3) represent the design constructed in accordance with Theorems 1 and 2. It follows that*

$$\rho_{\max}(D_2) \le \rho_{\max}(P),$$

$$\rho_{\max}(D_3) \le \frac{s^2(\lambda^2 s^2 n_2^2 - 1)\rho_{\max}(Q) + (s^2 - 1)\rho_{\max}(B)}{\lambda^2 s^4 n_2^2 - 1},$$

$$\rho_{\max}(D_2, D_3) \le s\sqrt{\frac{\lambda^2 s^2 n_2^2 - 1}{\lambda^2 s^4 n_2^2 - 1}}\rho_{\max}(P, Q), \tag{2}$$

where P and Q represent the first m columns and the last r columns of L, respectively. $\lambda = 1$ corresponds to the case in Theorem 1.

Wang et al. (2022) highlighted the connections among three criteria: column orthogonality, projection uniformity, and maximin distance. A design that exhibits excellent column orthogonality generally tends to have superior overall space-filling properties.

Additionally, the construction methods proposed in this section rely on the existence of OAs and MOAs. According to Hedayat et al. (1999),

i. $OA(s^2, s + 1, s, 2)$ exists for any prime power s (Theorem 3.20);
ii. $OA(2s^2, 2s + 1, s, 2)$ exists for any prime power s (Theorem 6.40);
iii. $OA(\lambda s^2, \lambda \frac{s^{d+1}-1}{s^d - s^{d-1}} + 1, s, 2)$ exists for any prime power p, $s = p^v$, $\lambda = p^u$, and $d = \lfloor u/v \rfloor$, where $u \ge 0$, $v \ge 1$ are all integers (Theorem 6.28).

Building on previous work, Hedayat et al. (1999) reported extensive findings for non-prime power scenarios, whereas He et al. (2017) synthesized a comprehensive overview of existing orthogonal arrays.

3.2 | Construction of stratification-enhanced SLHDs

The stratification properties of the proposed EBLHDs are based primarily on the stratification properties of SLHDs. In this section, we study how to construct SLHDs with excellent space-filling properties.

Theorem 3. *Let G and H be $LHD(n, m)$ and $LHD(r, m)$, respectively. Then,*

$$L = H \oplus_c rG = (L_0^T, \cdots, L_{r-1}^T) \tag{3}$$

is an $SLHD(rn, m, r)$, and

$$\rho_{\max}(L) \leq \frac{r^2(n^2-1)\rho_{\max}(G) + (r^2-1)\rho_{\max}(H)}{r^2n^2-1},$$
$$\rho_{\max}(L_i) \leq \rho_{\max}(G),\ i = 0, \ldots, r-1.$$

Remark 2.

(i) For the selection of G and H, we consider several excellent design classes, such as maximin distance designs (Johnson et al., 1990; Wang et al., 2018), (strong) OA-based LHDs (He & Tang, 2013; Tang, 1993), orthogonal-maximin LHDs (Joseph & Hung, 2008), maximum projection designs (Joseph et al., 2015), and uniform projection designs (Sun et al., 2019).

(ii) To enhance the space-filling properties of L in Equation (3), one can generate $L_{i-1} = h_i \oplus_c rG_i$ by r different $G_i \in LHD(n, m)$ for $1 \leq i \leq r$, where h_i is the ith row of H.

Example 3. We now generate the $SLHD(12, 3, 3)$ and $SLHD(24, 4, 3)$ mentioned in Remark 2.

(i) Taking G_1, G_2, and G_3 as three $OA(4, 3, 2, 2)$-based $LHD(4, 3)$s and H as an $LHD(3, 3)$ in Equation (4), Theorem 3 produces an $SLHD(12, 3, 3)$ as in Equation (4), where $OA(4, 3, 2, 2)$ is taken from http://neilsloane.com/oadir/oa.4.3.2.2.txt.

$$H = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix}, L = \left(\begin{array}{cccc|cccc|cccc} 3 & 0 & 6 & 9 & 2 & 5 & 8 & 11 & 1 & 4 & 10 & 7 \\ 1 & 10 & 4 & 7 & 0 & 9 & 3 & 6 & 2 & 11 & 5 & 8 \\ 5 & 8 & 11 & 2 & 1 & 7 & 10 & 4 & 3 & 6 & 9 & 0 \end{array}\right)^T. \tag{4}$$

(ii) Similarly, taking three $LHD(8, 4)$s based on $OA(8, 4, 2, 3)$ as $G_i (1 \leq i \leq 3)$ and H as $LHD(3, 4)$ in Equation (5), Theorem 3 generates an $SLHD(24, 4, 3)$ as Equation (5), where $OA(8, 4, 2, 3)$ is taken from http://neilsloane.com/oadir/oa.8.4.2.3.txt.

$$H = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 2 \\ 2 & 2 & 0 & 1 \end{pmatrix},$$

Algorithm 3. *SLHD*$(\lambda s n_2, k, \lambda s)$

Input $A = (A_0^T, \dots, A_{\lambda s-1}^T)^T \in CROA(n_2 = \lambda s^2, k, s, 2)$, $A_i \in OA(s, k, s, 1)$, $C, G \in LHD(\lambda s, k)$.

Step 1. Define $E = (E_0^T, \dots, E_{\lambda s-1}^T)^T = C \oplus_c \lambda s A$, where each E_i contains n_2 rows.

Step 2. Rearrange the rows of E to form $\tilde{L} = (\tilde{L}_0^T, \dots, \tilde{L}_{\lambda s-1}^T)^T$. Specially, select the 1-st s rows from E_0, the 2nd s rows from E_1, … , the λsth s rows from $E_{\lambda s-1}$;the 2nd s rows from E_0, the 3-rd s rows from E_1, … , the 1st s rows from $E_{\lambda s-1}$, and so on, until all rows of E are taken through.

Step 3. Generate $L = (L_0^T, \dots, L_{\lambda s-1}^T)^T = \lambda s\tilde{L} + G \otimes \mathbf{1}_{n_2}$.

Output Design $L = (L_0^T, \dots, L_{\lambda s-1}^T)^T$.

$$L = \left(\begin{array}{cccccccc|cccccccc|cccccccc} 1 & 10 & 4 & 7 & 13 & 19 & 22 & 16 & 3 & 9 & 0 & 6 & 12 & 21 & 15 & 18 & 5 & 11 & 8 & 2 & 20 & 17 & 23 & 14 \\ 6 & 9 & 18 & 12 & 3 & 0 & 21 & 15 & 4 & 1 & 13 & 19 & 7 & 10 & 22 & 16 & 2 & 8 & 20 & 23 & 5 & 11 & 17 & 14 \\ 5 & 20 & 2 & 23 & 11 & 14 & 8 & 17 & 1 & 16 & 10 & 13 & 4 & 22 & 7 & 19 & 6 & 12 & 0 & 18 & 9 & 21 & 3 & 15 \\ 0 & 21 & 18 & 6 & 15 & 3 & 9 & 12 & 2 & 23 & 20 & 8 & 14 & 5 & 11 & 17 & 1 & 13 & 22 & 10 & 16 & 7 & 4 & 19 \end{array}\right)^T. \tag{5}$$

The SLHDs obtained from Algorithm 3 exhibit excellent low-dimensional stratification properties, which we summarize in Theorem 4. Its proof is found in Appendix A.4.

Theorem 4. *Given* $A \in CROA(n_2 = \lambda s^2, k, s, 2)$, *the design L from Algorithm 3 satisfies*

(*i*) *L is an SLHD*$(\lambda^2 s^3, k, \lambda s)$.

(*ii*) *In any two dimensions, each slice of L can achieve stratification on* $s \times s$ *grids, and the whole L can achieve stratification on* $s \times \lambda s^2$ *and* $\lambda s^2 \times s$ *grids.*

(*iii*)

$$\rho_{\max}(L) \leq \frac{\lambda^2 s^2 - 1}{\lambda^4 s^6 - 1}\left(\lambda^2 s^2 \rho_{\max}(C) + \rho_{\max}(G)\right).$$

Remark 3.

(i) Theorem 3 in He et al. (2017) provided a summary of the existence of CROAs, which can be used as A in Algorithm 3.

(ii) The LHDs C and G can be selected from the design classes mentioned in Remark 2. Additionally, they can be chosen from $LHD(\lambda s, \lambda s)$ provided in Wang et al. (2018) and tab. 2 in Yuan et al. (2025). As an alternative, the R package SLHD in Ba et al. (2015) can also be utilized.

(iii) Using the SLHDs from Algorithm 3 for Algorithms 1 and 2, the resulting EBLHDs have excellent stratification properties. Some of the construction results are listed in Table C1–C3.

Example 4. Here, we generate the *SLHD*$(27, 3, 3)$ used in Example 1 and *SLHD*$(108, 6, 6)$ used in Example 2, respectively.

(i) Considering $A \in CROA(9, 3, 3, 2)$ and $C, G \in LHD(3, 3)$ below, Algorithm 3 can generate *SLHD*$(27, 3, 3)$ in Equation (1).

$$A = \left(\begin{array}{ccc|ccc|ccc} 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 2 & 1 & 1 & 0 & 2 & 2 & 1 & 0 \end{array}\right)^T, C = \begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 2 & 2 \end{pmatrix}, G = \begin{pmatrix} 0 & 1 & 0 \\ 2 & 2 & 1 \\ 1 & 0 & 2 \end{pmatrix}. \tag{6}$$

TABLE 5 Properties of different SLHDs.

Designs	Generation	$\rho_{ave}(\downarrow)$	$\rho_{\max}(\downarrow)$	$d(\uparrow)$	$\phi_p(\downarrow)$	Stratification
$SLHD(27, 3, 3)$	Theorem 4	**0.0549**	**0.0549**	**8**	**0.1451**	$\mathbf{3 \times 9, 9 \times 3}$
	MaxAbsCor	0.0914	0.1477	3	0.3336	–
	phi_p	0.2867	0.4853	4	0.2618	–
	AvgAbsCor	0.0840	0.1843	3	0.3491	–
$SLHD(64, 4, 4)$	Theorem 4	**0.0192**	**0.0491**	**16**	**0.0691**	$\mathbf{4 \times 16, 16 \times 4}$
	MaxAbsCor	0.1236	0.1821	4	0.2500	–
	phi_p	0.1741	0.4063	9	0.1111	–
	AvgAbsCor	0.0930	0.3336	7	0.1496	–
$SLHD(108, 6, 6)$	Theorem 4	**0.0383**	**0.1018**	**52**	**0.0214**	$\mathbf{3 \times 18, 18 \times 3}$
	MaxAbsCor	0.1057	0.1979	19	0.0526	–
	phi_p	0.1110	0.3469	36	0.0278	–
	AvgAbsCor	0.0890	0.2883	21	0.0477	–
$SLHD(125, 5, 5)$	Theorem 4	**0.0160**	**0.0353**	**31**	**0.0368**	$\mathbf{5 \times 25, 25 \times 5}$
	MaxAbsCor	0.0928	0.1717	15	0.0667	–
	phi_p	0.1082	0.2615	21	0.0476	–
	AvgAbsCor	0.0873	0.2521	11	0.0909	–

Note: The bolded values represent the optimal values in four kinds of SLHDs. The "↓" means lower is better, and the "↑" means higher is better.

(ii) Selecting $A \in CROA(18, 6, 3, 2)$ and $C, G \in LHD(6, 6)$ as follows, Algorithm 3 generates an $SLHD(108, 6, 6)$ used in Example 2. The generated $SLHD(108, 6, 6)$ is listed in Equation (B1).

$$A = \left(\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc} 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 1 & 2 & 1 & 2 & 0 & 0 & 1 & 2 & 2 & 0 & 1 & 2 & 0 & 1 & 1 & 2 & 0 \\ 0 & 1 & 2 & 2 & 0 & 1 & 2 & 0 & 1 & 0 & 1 & 2 & 1 & 2 & 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 & 1 & 2 & 0 & 0 & 1 & 2 & 2 & 0 & 1 \\ 0 & 1 & 2 & 2 & 0 & 1 & 1 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 0 & 1 & 2 \end{array}\right)^T,$$

$$C = G = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 5 & 4 & 2 & 0 \\ 2 & 5 & 3 & 0 & 1 & 4 \\ 3 & 4 & 0 & 2 & 5 & 1 \\ 4 & 2 & 1 & 5 & 0 & 3 \\ 5 & 0 & 4 & 1 & 3 & 2 \end{pmatrix}. \tag{7}$$

To display the robustness of the space-filling properties of these SLHDs, we generated 100 SLHDs in different ways and show the properties of the worst SLHDs in Table 5. These methods included "MaxAbsCor", "phi_p", and "AvgAbsCor" in the R package SLHD and the method in

Algorithm 4. *EBLHD*$(N = n_1 n_2, q + m + r)$

Input $A \in MOA(n_1, s_1 \cdots s_q, 2)$, $L = (L_1, \dots, L_q, L_{q+1}) \in LHD(n_2, k = m + r)$, $G \in LHD(n_1, r)$, $B = (B_1, \dots, B_q)$, where L_i and L_{q+1} have m_i and r columns, respectively, $B_i \in OA(n_1, m_i, \frac{n_1}{s_i}, 1)$, each pair column $(a_i, b) \in MOA\left(n_1, s_i \times \frac{n_1}{s_i}, 2\right)$ for $a_i \in A$ and $\forall b \in B_i$, $1 \le i \le q$.

Step 1. Define $D_1 = A \otimes \mathbf{1}_{n_2}$ for branching factors.

Step 2. Define $D_2 = (D_{21}, \dots, D_{2q})$, where $D_{2i} = B_i \oplus_c \frac{n_1}{s_i} L_i$ for $1 \le i \le q$.

Step 3. Generate $D_3 = G \oplus_c n_1 L_{q+1}$.

Output Design $D = (D_1, D_2, D_3)$.

Algorithm 3 (see Appendix B.2 for details and involved designs). The space-filling criteria used for comparison include the average absolute correlation (ρ_{ave}), maximum absolute correlation ($\rho_{\max}$), L_1 distance (d), the maximin distance criterion in Morris and Mitchell (1995) ($\phi_p, p = 15$) and stratification properties. Table 5 shows that the proposed method is the most robust under all criteria.

In addition, the literature provides several construction methods for SLHDs. Based on the maximin criterion, Ba et al. (2015) presented an effective search algorithm for SLHDs that utilizes simulated annealing. Sun et al. (2014) generated SLHDs from sliced OAs. All of these SLHDs can be effectively used to construct EBLHDs that exhibit superior space-filling properties.

4 | CONSTRUCTION OF EBLHDS WITH MIXED-LEVEL BRANCHING FACTORS

Algorithm 4 provides a method for constructing EBLHDs under mixed-level branching factors. The properties of the resulting design are summarized in Theorem 5.

Theorem 5. *The design D constructed by Algorithm 4 is an EBLHD*$(n_1 n_2, q + m + r)$ *satisfying*

(i) For $i \neq j \in 1, \dots, q$,

$$\rho_{\max}(D_3) \le \frac{n_1^2(n_2^2 - 1)\rho_{\max}(L_{q+1}) + (n_1^2 - 1)\rho_{\max}(G)}{n_1^2 n_2^2 - 1},$$

$$\rho_{\max}(D_{2i}) \le \frac{n_1^2(n_2^2 - 1)\rho_{\max}(L_i) + (n_1^2 - s_i^2)\rho_{\max}(B_i)}{n_1^2 n_2^2 - s_i^2},$$

$$\rho_{\max}(D_{2i}, D_{2j}) \le \frac{n_1^2(n_2^2 - 1)\rho_{\max}(L_i, L_j) + \sqrt{(n_1^2 - s_i^2)(n_1^2 - s_j^2)}\rho_{\max}(B_i, B_j)}{\sqrt{(n_1^2 n_2^2 - s_i^2)(n_1^2 n_2^2 - s_j^2)}},$$

$$\rho_{\max}(D_{2i}, D_3) \le \frac{n_1^2(n_2^2 - 1)\rho_{\max}(L_i, L_{q+1}) + \sqrt{(n_1^2 - s_i^2)(n_1^2 - 1)}\rho_{\max}(B_i, G)}{\sqrt{(n_1^2 n_2^2 - s_i^2)(n_1^2 n_2^2 - 1)}}.$$

(*ii*) *For each level combination of all branching factors, the corresponding design of* (D_2, D_3) *has the same low-dimensional stratification properties as* L.

To improve the uniformity of the final EBLHD, L can be generated in the design class proposed in Remark 2 and Remark 3. In addition, the method in Algorithm3 relies on the MOA. Regarding to its existence, we have summarized as follows.

(i) For orthogonal arrays with run size fewer than 100, some existing designs can be found in http://neilsloane.com/oadir/, as proposed by Sloane (2007); Some new orthogonal arrays with run sizes 72 and 96 can refer to Zhang (2007).
(ii) Schoen et al. (2010) specified an algorithm and obtained some MOAs for strength $t = 2$, run size $n \le 28$, $t = 3$, $n \le 64$, and $t = 4$, $n \le 168$. Complete series of non-isomorphic OAs can be found directly at http://www.pietereendebak.nl/oapackage/series.html.
(iii) For algebraic construction methods, Chen and Lei (2017) constructed some MOAs of $t = 3$ by using difference matrices and Hadamard matrices; Jiang and Yin (2013) established a general "expansive replacement method" and produced some series of MOAs of $t = 3$; Pang et al. (2015) constructed many new MOAs of $n = 108$ and $n = 144$; Pang et al. (2021) proposed new methods for MOAs with high strength by using lower strength orthogonal partitions of spaces and OAs.

Example 5. We suppose there are three branching factors z_1, z_2, and z_3 with levels from $\mathbb{Z}_2$, $\mathbb{Z}_2$, and $\mathbb{Z}_3$, respectively. One factor is nested within each level of z_1 and z_2, and two factors are nested within each level of z_3, which are denoted by $v_1^{z_1}$, $v_1^{z_2}$, $v_1^{z_3}$, and $v_2^{z_3}$, respectively. Additionally, there is one shared factor, x_1. In this case, $q = 3$, $s_1 = s_2 = 2$, $s_3 = 3$, $n_1 = 12$, $m_1 = m_2 = 1$, $m_3 = 2$, $m = 4$, $r = 1$, and $k = 5$.

We consider $A \in MOA(12, 2^2 3, 2)$ listed in Table 6, $L \in LHD(8, 5)$, B, and G as follows. L_1, L_2 and L_3 contain 1, 1, and 2 columns, respectively, similarly for B_1, B_2, and B_3. L_4 has 1 column. Next, we construct an $EBLHD(96, 3 + 4 + 1)$ as shown in Table 6.

$$L = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 2 & 6 & 7 & 0 & 1 & 5 & 4 \\ 1 & 7 & 3 & 5 & 2 & 4 & 0 & 6 \\ 2 & 4 & 7 & 1 & 6 & 0 & 3 & 5 \\ 2 & 5 & 0 & 7 & 4 & 3 & 6 & 1 \end{pmatrix}^T, B = \begin{pmatrix} 0 & 2 & 4 & 1 & 3 & 5 & 1 & 3 & 5 & 0 & 2 & 4 \\ 0 & 2 & 4 & 1 & 3 & 5 & 3 & 5 & 1 & 4 & 0 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 0 & 3 & 0 & 1 \\ 3 & 2 & 1 & 2 & 1 & 0 & 1 & 0 & 3 & 0 & 3 & 2 \end{pmatrix}^T,$$

$$G = \begin{pmatrix} 0 & 4 & 8 & 3 & 7 & 11 & 6 & 10 & 2 & 9 & 1 & 5 \end{pmatrix}^T.$$

Step 1. We define $D_1 = A \otimes \mathbf{1}_8$.
Step 2. We define $D_2 = (D_{21}, D_{22}, D_{23})$, where $D_{2i} = B_i \oplus_c \frac{12}{s_i} L_i$, $1 \le i \le 3$.
Step 3. $D_3 = G \oplus_c 12L_4$ is generated.

The design D in Table 6 is an $EBLHD(96, 3 + 4 + 1)$. It is easy to find that D_2 and D_3 corresponding to each level combination in D_1 can achieve 2×2 stratification in two dimensions. Because the first four columns of L after the level collapse form an $OA(8, 4, 2, 3)$, the four columns in D_2 corresponding to each level combination in D_1 can also achieve $2 \times 2 \times 2$ stratification in any three dimensions. In addition, most of

TABLE 6 The structure of design $D = (D_1, D_2, D_3)$, $D_1 = A \otimes 1_8$.

A			**D_2**				**D_3**
a_1	a_2	a_3	$v_1^{z_1}$	$v_1^{z_2}$	$v_1^{z_3}$	$v_2^{z_3}$	x_1
0	0	0	$0 \oplus 6L_1$	$0 \oplus 6L_2$	$0 \oplus 4L_{31}$	$3 \oplus 4L_{32}$	$0 \oplus 12L_4$
0	0	1	$2 \oplus 6L_1$	$2 \oplus 6L_2$	$1 \oplus 4L_{31}$	$2 \oplus 4L_{32}$	$4 \oplus 12L_4$
0	0	2	$4 \oplus 6L_1$	$4 \oplus 6L_2$	$2 \oplus 4L_{31}$	$1 \oplus 4L_{32}$	$8 \oplus 12L_4$
0	1	0	$1 \oplus 6L_1$	$1 \oplus 6L_2$	$1 \oplus 4L_{31}$	$2 \oplus 4L_{32}$	$3 \oplus 12L_4$
0	1	1	$3 \oplus 6L_1$	$3 \oplus 6L_2$	$2 \oplus 4L_{31}$	$1 \oplus 4L_{32}$	$7 \oplus 12L_4$
0	1	2	$5 \oplus 6L_1$	$5 \oplus 6L_2$	$3 \oplus 4L_{31}$	$0 \oplus 4L_{32}$	$11 \oplus 12L_4$
1	0	0	$3 \oplus 6L_1$	$3 \oplus 6L_2$	$2 \oplus 4L_{31}$	$1 \oplus 4L_{32}$	$6 \oplus 12L_4$
1	0	1	$5 \oplus 6L_1$	$5 \oplus 6L_2$	$3 \oplus 4L_{31}$	$0 \oplus 4L_{32}$	$10 \oplus 12L_4$
1	0	2	$1 \oplus 6L_1$	$1 \oplus 6L_2$	$0 \oplus 4L_{31}$	$3 \oplus 4L_{32}$	$2 \oplus 12L_4$
1	1	0	$4 \oplus 6L_1$	$4 \oplus 6L_2$	$3 \oplus 4L_{31}$	$0 \oplus 4L_{32}$	$9 \oplus 12L_4$
1	1	1	$0 \oplus 6L_1$	$0 \oplus 6L_2$	$0 \oplus 4L_{31}$	$3 \oplus 4L_{32}$	$1 \oplus 12L_4$
1	1	2	$2 \oplus 6L_1$	$2 \oplus 6L_2$	$1 \oplus 4L_{31}$	$2 \oplus 4L_{32}$	$5 \oplus 12L_4$

the column correlation coefficients in L are zero. Therefore, the design D performs well in terms of the correlation property.

Remark 4. In fact, the number of nested factors may not be the same involved at different levels of different branching factors, or even at different levels of the same branching factor. In this scenario, we recommend constructing EBLHDs based on the maximum number of nested factors, and then deleting the redundant parts of the design to meet the actual requirements. This strategy is exactly what we adopted in the practical application of Section 6.

5 | SIMULATION

In this section, we conducted two simulations to study the space-filling properties and the model estimation performance of our proposed designs.

The first simulation was performed to explore their space-filling properties. To evaluate their distance properties, we calculated the minimum absolute L_1-distance under each level combination of branching factors and determined the minimum value among all level combinations of branching factors, denoted by d. The larger the distance is, the better the design. To clarify the column correlations of designs, we selected criterion ρ^2 in Hung et al. (2009) for judgment, for which a lower value is better. We generated 100 EBLHDs and calculated the average and the worst values of these criteria, denoted by d_{ave}, $d_{\min}$, ρ^2_{ave}, and $\rho^2_{\max}$, respectively. For comparison, we generated 100 random BLHDs. The simulation results are shown in Table 7 (details for generating these designs are provided in Appendix B.3). The EBLHDs performed significantly better than random BLHDs under both space-filling criteria.

The second simulation studied the model estimation performance under EBLHDs based on the mean square error (MSE) of the model parameter estimation. The models considered here are

TABLE 7 Simulation results on the space-filling properties of EBLHDs.

$(N, q+m+r)$	Type of design	$d(\uparrow)$ d_{ave}	d_{min}	$\rho^2(\downarrow)$ ρ^2_{ave}	ρ^2_{max}
$(40, 1+2+2)$	BLHD	0.9550	0.6000	0.0256	0.0768
	EBLHD	1.3500(42%)	1.0000(67%)	0.0076(70%)	0.0181(76%)
$(64, 1+4+4)$	BLHD	2.1022	1.1875	0.0147	0.0248
	EBLHD	2.9050(38%)	2.1250(79%)	0.0065(56%)	0.0095(62%)
$(90, 1+3+4)$	BLHD	1.5473	0.9111	0.0109	0.0196
	EBLHD	2.2200(43%)	1.5556(71%)	0.0023(79%)	0.0042(79%)
$(324, 3+4+2)$	BLHD	0.8182	0.3272	0.0027	0.0054
	EBLHD	1.2800(56%)	0.8889(172%)	0.0012(56%)	0.0024(56%)

Note: The values in parentheses are the proportion of the improvement of criterion value. The "↓" means lower is better, and the "↑" means higher is better.

shown in Equation (8) (see Goos and Jones (2019) for the complete model):

$$
\begin{aligned}
Y = \beta_0 + \sum_{k=1}^{\kappa} \beta_k \boldsymbol{z}_k + \sum_{j=1}^{r} \beta_{(s)j} x_{(s)j} + \sum_{k=1}^{\kappa} \boldsymbol{z}_k \left(\sum_{l=1}^{m} \beta_{(n)kl} x_{(n)kl} \right) + \sum_{j=1}^{r-1} \sum_{j'=j+1}^{r} \beta_{(s)jj'} x_{(s)j} x_{(s)j'} \\
+ \sum_{k=1}^{\kappa} \boldsymbol{z}_k \left(\sum_{l=1}^{m-1} \sum_{l'=l+1}^{m} \beta_{(n)kll'} x_{(n)kl} x_{(n)kl'} \right) + \epsilon,
\end{aligned}
\tag{8}
$$

where (i) β_0 is an intercept term. (ii) $\boldsymbol{z}_k$ is the indicator variable corresponding to the kth nesting relationship. If this indicator variable takes the value 1, then the corresponding nested factors, $x_{(n)k1}, \ldots, x_{(n)km}$, appear in the model. If it takes the value 0, then the corresponding nested factors do not appear in the model. To avoid collinearity, we drop one of the κ associated linear indicator variable terms from the model. (iii) $\beta_{(n)kl}$ and $\beta_{(n)kll'}$ represent the main effects and two-factor interaction effects of the nested factors, respectively. (iv) $x_{(s)j}$ denotes the level of the jth shared factor. $\beta_{(s)j}$ and $\beta_{(s)jj'}$ represent the main effects and two-factor interaction effects of the shared factors, respectively. (v) $\epsilon \sim N(0, 0.01)$ is the error term. The true regression coefficients are randomly generated from $(-1, 1)$ and all the designs are scaled to $(-1, 1)$. We generated 100 EBLHDs and random BLHDs, and calculated the MSEs of regression coefficient estimation. Table 8 shows the mean and maximum values of these 100 MSEs (details for generating these designs are provided in Appendix B.3). The MSEs under the EBLHDs are much smaller than those under the random BLHDs.

Based on the two simulation studies, we conclude that the EBLHDs not only possess better space-filling properties but also perform well in model parameter estimation.

6 | APPLYING EBLHDS FOR THE ALGORITHM CONFIGURATION PROBLEM

With the continuous increase in data volume and growing computational demands, the performance of algorithms has become increasingly crucial. Many high-performance algorithms have

TABLE 8 Simulations results on the estimation of model parameters of EBLHDs.

		MSE	
$(N, q+m+r)$	Type of design	MSE_{ave}	$\mathrm{MSE}_{\max}$
$(40, 1+2+2)$	BLHD	0.0093	0.5904
	EBLHD	0.0071(24%)	0.2890(51%)
$(64, 1+4+4)$	BLHD	0.0226	1.3504
	EBLHD	0.0130(42%)	0.4349(68%)
$(90, 1+3+4)$	BLHD	0.0053	0.2786
	EBLHD	0.0037(30%)	0.1221(56%)
$(324, 3+4+2)$	BLHD	0.0167	1.0695
	EBLHD	0.0092(45%)	0.5309(50%)

Note: The values in parentheses are the proportion of the improvement of MSE value.

parameters whose settings control important aspects of their behavior. The process of determining the optimal parameter settings to achieve the best performance is known as the algorithm configuration problem. Traditionally, algorithm configuration has been a laborious and manual task that requires a significant amount of effort (Adenso-Diaz & Laguna, 2006). To simplify this process, leveraging automated configuration methods is recommended (Bartz-Beielstein & Preuss, 2006; Bezerra et al., 2016; Birattari & Kacprzyk, 2009; Hoos, 2012). Notable among these methods is the Irace approach, which initializes configurations through uniform sampling within parameter domains, facilitated by the R package **irace**. Uniform sampling is efficient, but it may lead to suboptimal exploration, resulting in increased computational burdens and local optimization. Grid search is another widely used strategy for hyper-parameter optimization (Bergstra & Bengio, 2012). It forms a grid of hyper-parameter settings and assesses the performance of each combination using cross-validation, ultimately selecting the best-performing set of hyper-parameters. Nonetheless, grid search is encumbered by the curse of dimensionality. In high-dimensional spaces, this drawback renders the method notably inefficient and computationally costly, as the number of hyper-parameter combinations to be explored grows exponentially with the increase in the dimensionality of the hyperparameter space. Advanced techniques such as fractional factorial designs and LHDs offer more balanced parameter space exploration but may be challenging in scenarios involving mixed-variable and conditional parameters.

In algorithm configuration, the parameters that depend on specific values of other parameters are regarded as nested factors, and the parameters within which other parameters are nested are branching factors. All remaining parameters are categorized as shared factors. For example, Tables 9 and 10 provide the parameter types and relationships for two optimization algorithms. To address the challenge of finding the optimal algorithm configuration within a complex parameter space, Wessing and López-Ibáñez (2019) introduced an enhanced initialization method that utilizes BLHDs for computer experiments involving branching and nested factors. This method optimizes a weighted criterion by combining a general energy criterion with a correlation criterion using an algorithm to achieve an optimal BLHD. However, this process increases the computational load and may result in convergence to local optima in practical scenarios.

The EBLHDs proposed in this article are constructed through algebraic methods and have good space-filling properties, which can significantly reduce the computational burden when generating EBLHDs. In this section, we apply EBLHDs to two specific algorithm configuration

TABLE 9 Parameters in optim algorithm configuration scenarios.

Branching factor		Nested factors	Shared factors
Method	Simulated Annealing Method	tmax **i** $(1, 5000)$ temp **r** $(0, 100)$	reltol **r** $(-12, -3)^{*}$ maxit **r** $(1, 3.024)^{*}$
	Nelder-Mead Method	alpha **r** $(0.5, 1.5)$ beta **r** $(0.1, 0.9)$ gamma **r** $(1.1, 3.0)$	

Note: The "**r**" and "**i**" are parameter type: real and integer. Rows with a $*$ indicate that a log 10-transformation is applied.

TABLE 10 Nested relationship in ACOQAP algorithm.

Branching factors		Nested factors
Algorithm	MMAS	$m, \alpha, \beta, \rho, q_0, slen$
	AS	$m, \alpha, \beta, \rho, q_0, slen$
	EAS	$m, \alpha, \rho, q_0, m_{elite}$
	RAS	$m, \alpha, \rho, q_0, rasrank$
	BWAS	m, α, ρ, q_0
Restart	Never	None
	Branch-factor	$res_{\mathrm{bf}}, res_{\mathrm{it}}$
	Distance	$res_{\mathrm{dist}}, res_{\mathrm{it}}$
	Always	res_{it}
PH-limits	Yes	p_{dec}
	No	None

problems. The first scenario is an optimization algorithm in the R standard library (**optim**). The second scenario is the ant colony optimization for the quadratic assignment problem (**ACOQAP**) (López-Ibáñez et al., 2018). We compare the optimal configurations obtained using EBLHDs with those derived using the BLHD method proposed in López-Ibáñez et al. (2018), Irace, grid search and the default method. Here, we use the package **iracelhs** (https://github.com/search?q=iracelhs/type=repositories) to implement the BLHD method. The default method refers to solving the optimization problem using the default parameter settings of the algorithm. Notably, in these two algorithms, the numbers of nested factors corresponding to different levels of branching factors are not the same. Therefore, when using EBLHDs for algorithm configuration, we construct nested factor designs based on the maximum number of nested factors at each level of branching factor, as stated in Remark 4.

6.1 | Applying EBLHDs for Optim

We first examine a simpler scenario, referred to as "optim", in which optimization algorithms from the R standard library, specifically the Nelder–Mead and simulated annealing methods, are used to minimize any function. In this case, there is $q = 1$ branching factor with 2 levels. Because the number of nested factors corresponding to different levels of branching factors is either 2 or 3, we choose $m = 3$ when constructing the design. There are also $r = 2$ shared factors. The parameter space associated with the optim algorithm is shown in Table 9.

By utilizing an OA $A \in OA(4, 2, 2, 2)$, an SLHD $L = (L_0^T, L_1^T)^T \in SLHD(50, 5, 2)$ with each slice having an equal number of runs, and a necessary matrix $B \in OA(4, 2, 2, 1)$, we construct an $EBLHD(100, 1 + 3 + 2)$ by Algorithm 1 and use it to generate a series of parameter configuration sets. The A, L, and B used in this subsection can be found in Table B4 of Appendix B.4. For the Irace method, we randomly generate 100 configurations within the parameter space. And the default method of optim is Nelder-Mead Method. We conduct a comparative analysis of the best configuration identified by the EBLHD method against BLHDs, Irace, grid search and the default method across the following family of 10 functions:

$$f(x) = \lambda f_1(x) + (1 - \lambda) f_2(x),$$

where λ follows a normal distribution with a mean of 0.9 and a standard deviation of 0.02. f_1 and f_2 are the well-known Rastrigin and Rosenbrock benchmark functions, respectively (taken from the cmaes package in R). In this scenario, different functions are given by different values of λ. This comparison focuses on optimal configurations that seek to minimize the values of these 10 functions, all of which have theoretical minimum values of 0. Figure 1 presents a boxplot of the minimum values of these 10 functions obtained using the parameter

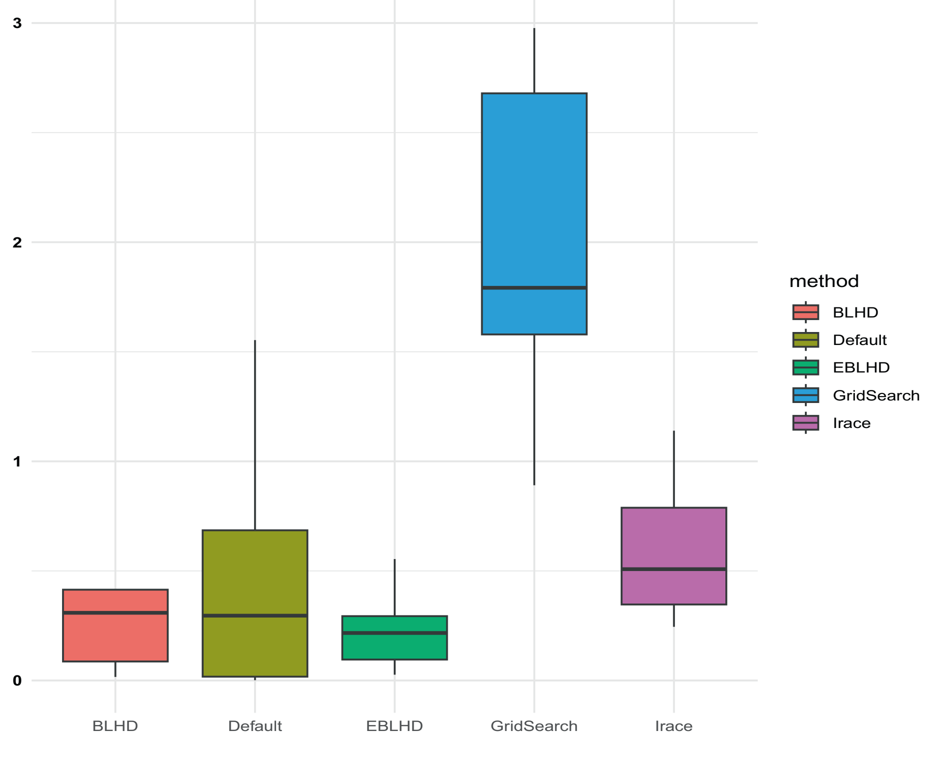

FIGURE 1 Boxplot for measuring the quality of the best configuration obtained with EBLHDs, BLHDs, Irace, and grid search, along with the default parametrization of optim function (a lower value is better).

configurations discovered by the five methods. The results clearly demonstrate that EBLHDs yield an outstanding configuration.

6.2 | Applying EBLHDs for ACOQAP

In this subsection, we apply EBLHDs to find the best configuration for a complex algorithm, namely, "ACOQAP". This scenario uses a component-wise framework of diverse ant colony optimization (ACO) algorithms (López-Ibáñez et al., 2018) for instances of the quadratic assignment problem (QAP). The QAP can be outlined as follows: given n locations and n facilities, the distances between pairs of locations are denoted by an $n \times n$ distance matrix $D = \left(d_{ij}\right)_{n\times n}$, and the transportation flows between pairs of facilities are also represented by an $n \times n$ flow matrix $F = (f_{ij})_{n\times n}$. The aim is to assign the n facilities to the n locations in a manner that minimizes the overall cost, which can be formulated as

$$\min_{\pi\in S} cost = \min_{\pi\in S} \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \cdot f_{\pi_i\pi_j},$$

where $\pi_i \in \pi$, $i = 1,\dots,n$, and π is a uniform arrangement on the set $\{1,\dots,n\}$.

We focus on a specific quadratic assignment problem provided in Heris (2015). We choose $n = 7$ positions, denoted by

$$\{(69, 9), (80, 81), (63, 43), (34, 89), (54, 30), (36, 95), (53, 97)\},$$

along with the corresponding 7×7 flow matrix F:

$$F = \begin{pmatrix} 0 & 6 & 6 & 3 & 5 & 5 & 5 \\ 6 & 0 & 6 & 4 & -10 & 3 & 6 \\ 6 & 6 & 0 & 4 & 5 & 8 & 6 \\ 3 & 4 & 4 & 0 & 4 & 4 & 100 \\ 5 & -10 & 5 & 4 & 0 & 3 & 4 \\ 5 & 3 & 8 & 4 & 3 & 0 & 4 \\ 5 & 6 & 6 & 100 & 4 & 4 & 0 \end{pmatrix}.$$

The distance matrix D is calculated using the Euclidean distance.

When using the ACO algorithm to address this QAP, the parameter space encompasses 3 branching parameters and 11 nested parameters. A comprehensive list of these parameters is available in Table B5 of Appendix B.4, and the specific nesting relationships among these parameters are delineated in Table 10.

In this scenario, the number of branching parameters is $q = 3$, with $s_1 = 5$, $s_2 = 4$, and $s_3 = 2$ levels each. Although the number of nested factors varies across branching factor levels, we adopt $m_1 = 6$, $m_2 = 2$, and $m_3 = 1$ nested parameters under these three branching parameters, yielding a total of $m = 9$ nested parameters. To use the EBLHDs proposed in this article, we sample 160 configurations (equal to the number of design points). The following inputs are needed: a mixed orthogonal array $A \in MOA(40, 5^1 4^1 2^1, 2)$; an LHD $L = (L_1, L_2, L_3)$, where $L_1 \in LHD(4, 6)$, $L_2 \in$

TABLE 11 Default and optimal ACO configurations identified by EBLHDs, BLHDs, Irace and grid search.

	Algorithm	m	α	β	ρ	q_0	$slen$	Restart	res_{it}
EBLHDs	MMAS	311	1.149	6.971	0.904	0.543	170	Distance $res_{\mathrm{dist}} = 1.822$	40
BLHDs	AS	10	1.0	4.662	0.95	0.0	97	Branch-factor $res_{\mathrm{bf}} = 1.412$	60
Irace	MMAS	6	0.324	3.156	0.29	0.062	153	Distance $res_{\mathrm{dist}} = 0.051$	22
Grid search	AS	100	0.1495	2.135	0.5412	0.0	210	Distance $res_{\mathrm{dist}} = 1.822$	38
Default	MMAS	25	1.0	2.0	0.2	0.0	250	Branch-factor $res_{\mathrm{bf}} = 1.0$	250

$LHD(4,2)$, and $L_3 \in LHD(4,1)$; and a balanced design $B = (B_1, B_2, B_3)$, with $B_1 \in OA(40, 8^6, 1)$, $B_2 \in OA(40, 10^2, 1)$, and $B_3 \in OA(40, 20^1, 1)$. Based on A, L, and B, we use Algorithm 4 to generate an $EBLHD(160, 3+9+0)$, which represents a set of parameter configurations within the specified parameter space. The A, B, and L used in this case are listed in Table B6 of Appendix B.4. In addition, for the BLHD method, we implement an energy criterion to search for the optimal BLHD containing 160 runs and then use this design to find the optimal configuration of parameters. To ensure a fair comparison, we generate 160 configurations in the parameter space for the Irace method and grid search method, and this procedure is repeated 50 times each to determine the optimal configuration. The final optimal configurations obtained through the EBLHDs, BLHDs, Irace, grid search, and default method are outlined in Table 11.

Figure 2 compares the results from the four tuned configurations (obtained using EBLHDs, BLHDs, grid search, and Irace) with those from the default configuration on the aforementioned QAP problem. Each optimal configuration is run once on this QAP problem for up to 100 iterations, with the cost computed at each step. From the figure, the following conclusions can be drawn: (i) The convergence of the EBLHDs method is the fastest among the five methods. (ii) The final cost achieved by the configuration found by EBLHDs and grid search method is significantly lower than those found by BLHDs, Irace, and the default approach.

7 | CONCLUSIONS

This article focuses on experiments with branching factors and nested factors. Several methods are proposed for constructing EBLHDs under equal-level branching factors and mixed-level branching factors. For equal-level branching factors, two frameworks are presented, along with the constructions of the corresponding SLHDs. Considering that branching factors may have different number of levels in practice, we also constructed EBLHDs with mixed-level branching factors. The resulting EBLHDs exhibited good stratification properties and satisfactory correlation performance. A significant advantage of our method is that it divides the EBLHD into two core components: the OA and the SLHD. The OA serves as the overall framework, arranging the branching factors, whereas the SLHD is specifically used to adjust the stratification to meet various specific needs. The proposed method enables experimenters to easily obtain EBLHDs with diverse stratification structures, thereby enabling flexible adaptation to various experimental and

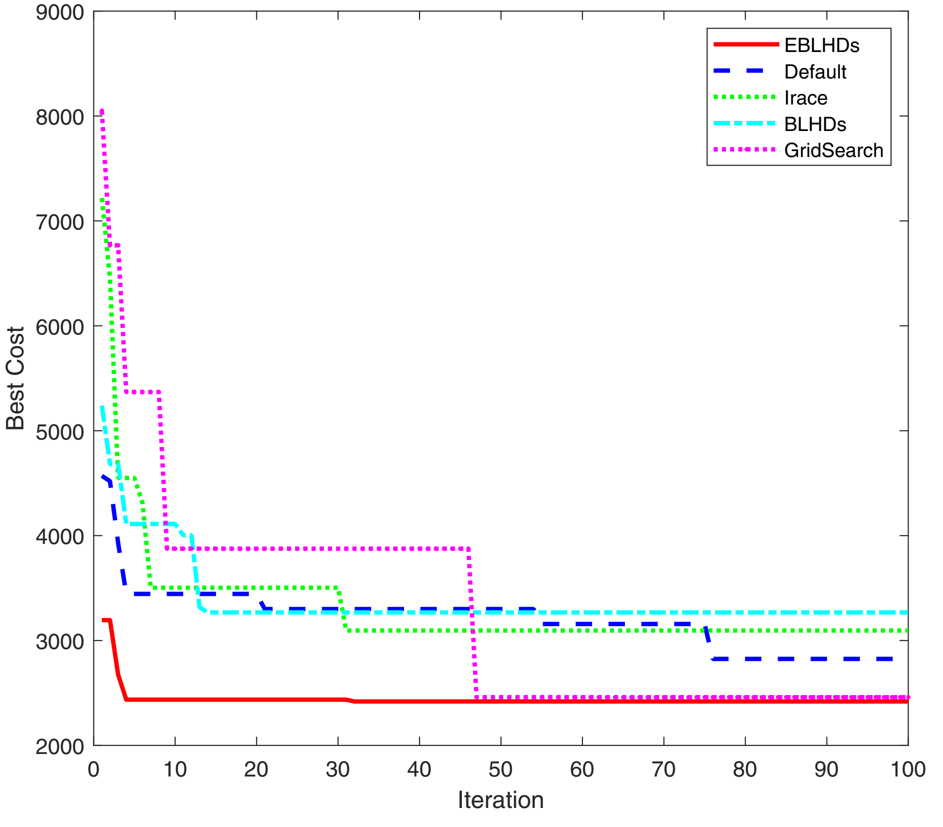

FIGURE 2 The cost as a function of the number of iterations across four configurations discovered by EBLHDs, BLHDs, Irace, and grid search, along with the default configuration (a lower value is better).

research requirements. Additionally, we propose applying these designs to the initialization phase of automatic algorithm configuration, thereby enhancing both efficiency and effectiveness.

There are several directions worthy of further study. One such direction involves the construction of EBLHDs with additional desirable properties, encompassing maximin distance EBLHDs, uniform projection EBLHDs, etc. Moreover, when the experimental budget varies across different levels of branching factors, leading to disparities in run size, the following question arises: How can we devise such designs that possess optimal properties? This inquiry holds significant research value. In addition, the application of EBLHDs to Bayesian optimization is also a valuable research topic.

FUNDING INFORMATION

This research was supported by the National Natural Science Foundation of China (12371259 and 12401324) and the Fundamental Research Funds for the Central Universities, China (2412023YQ003). The first two authors contributed equally to this work.

ORCID

Fasheng Sun https://orcid.org/0000-0003-2410-4018

REFERENCES

Adenso-Diaz, B., & Laguna, M. (2006). Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, *54*(1), 99–114.

Ba, S., Myers, W. R., & Brenneman, W. A. (2015). Optimal sliced latin hypercube designs. *Technometrics*, *57*(4), 479–487.

Bartz-Beielstein, T., & Preuss, M. (2006). *Considerations of budget allocation for sequential parameter optimization (spo)*. In *Workshop on empirical methods for the analysis of algorithms, proceedings* (pp. 35–40). Citeseer.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(10), 281–305.

Bezerra, L. C. T., López-Ibánez, M., & Stützle, T. (2016). Automatic component-wise design of multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *20*(3), 403–417.

Birattari, M., & Kacprzyk, J. (2009). *Tuning metaheuristics: A machine learning perspective*. Springer.

Chen, G., & Lei, J. (2017). Constructions of mixed orthogonal arrays of strength three. *Science China Mathematics*, *47*(4), 545–564.

Deng, X., Hung, Y., & Lin, C. D. (2015). Design for computer experiments with qualitative and quantitative factors. *Statistica Sinica*, *25*(4), 1567–1581.

Goos, P., & Jones, B. (2019). Optimal experimental design in the presence of nested factors. *Technometrics*, *61*(4), 533–544.

Guo, Z., Liu, H., Ong, Y.-S., Qu, X., Zhang, Y., & Zheng, J. (2023). Generative multiform bayesian optimization. *IEEE Transactions on Cybernetics*, *53*(7), 4347–4360.

He, Y., Cheng, C.-S., & Tang, B. (2018). Strong orthogonal arrays of strength two plus. *Annals of Statistics*, *46*(2), 457–468.

He, Y., Lin, C. D., & Sun, F. (2017). On construction of marginality coupled designs. *Statistica Sinica*, *27*(2), 665–683.

He, Y., & Tang, B. (2013). Strong orthogonal arrays and associated latin hypercubes for computer experiments. *Biometrika*, *100*(1), 254–260.

Hedayat, A. S., Sloane, N. J. A., & Stufken, J. (1999). *Orthogonal arrays: Theory and applications*. Springer.

Heris, M. K. (2015). *Quadratic assignment problem (qap) using ga, pso and fa*. https://yarpiz.com/359/ypap104-quadratic-assignment-problem

Hoos, H. H. (2012). Programming by optimization. *Communications of the ACM*, *55*(2), 70–80.

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.

Jiang, L., & Yin, J. (2013). An approach of constructing mixed-level orthogonal arrays of strength ≥ 3. *Science China Mathematics*, *56*(6), 1109–1115.

Johnson, M. E., Moore, L. M., & Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, *26*(2), 131–148.

Joseph, V. R., Gul, E., & Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika*, *102*(2), 371–380.

Joseph, V. R., & Hung, Y. (2008). Orthogonal-maximin latin hypercube designs. *Statistica Sinica*, *18*(1), 171–186.

López-Ibáñez, M., Stützle, T., & Dorigo, M. (2018). *Ant colony optimization: A component-wise overview*. In R. Martí, P. M. Pardalos, & M. G. C. Resende (Eds.), *Handbook of heuristics* (pp. 371–407). Springer.

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*(2), 239–245.

Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, *43*(3), 381–402.

Pang, S., Wang, J., Lin, D. K. J., & Liu, M.-Q. (2021). Construction of mixed orthogonal arrays with high strength. *Annals of Statistics*, *49*(5), 2870–2884.

Pang, S., Zhu, Y., & Wang, Y. (2015). A class of mixed orthogonal arrays obtained from projection matrix inequalities. *Journal of Inequalities and Applications*, *2015*(241), 1–9.

Qian, P. Z. G. (2012). Sliced latin hypercube designs. *Journal of the American Statistical Association*, *107*(497), 393–399.

Schoen, E. D., Eendebak, P. T., & Nguyen, M. V. M. (2010). Complete enumeration of pure-level and mixed-level orthogonal arrays. *Journal of Combinatorial Designs*, *18*(2), 123–140.

Sloane, N. J. A. (2007). *A library of orthogonal arrays*. http://neilsloane.com/oadir/

Sun, F., Liu, M.-Q., & Qian, P. Z. G. (2014). On the construction of nested space-filling designs. *Annals of Statistics*, *42*(4), 1394–1425.

Sun, F., Wang, Y., & Xu, H. (2019). Uniform projection designs. *Annals of Statistics*, *47*(1), 641–661.

Tang, B. (1993). Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, *88*(424), 1392–1397. https://doi.org/10.1080/01621459.1993.10476423

Wang, K., & Dowling, A. W. (2022). Bayesian optimization for chemical products and functional materials. *Current Opinion in Chemical Engineering*, *36*, 100728.

Wang, L., Xiao, Q., & Xu, H. (2018). Optimal maximin l_1-distance latin hypercube designs based on good lattice point designs. *Annals of Statistics*, *46*(6B), 3741–3766.

Wang, Y., Sun, F., & Xu, H. (2022). On design orthogonality, maximin distance, and projection uniformity for computer experiments. *Journal of the American Statistical Association*, *117*(537), 375–385.

Wei, Q., Liu, M.-Q., & Yang, J.-F. (2022). Orthogonal designs with branching and nested factors. *Stat*, *11*(1), e447.

Wessing, S., & López-Ibáñez, M. (2019). Latin hypercube designs with branching and nested factors for initialization of automatic algorithm configuration. *Evolutionary Computation*, *27*(1), 129–145.

Yuan, R., Yin, Y., Xu, H., & Liu, M.-Q. (2025). A construction method for maximin l_1-distance latin hypercube designs. *Statistica Sinica*, *35*(1), 249–272.

Zhang, Y. (2007). Orthogonal arrays obtained by repeating-column difference matrices. *Discrete Mathematics*, *307*(2), 246–261.

Zhang, Y., Apley, D. W., & Chen, W. (2020). Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific Reports*, *10*(1), 4924.

How to cite this article: Wen, B., Wang, S., & Sun, F. (2025). Enhanced branching Latin hypercube design and its application in automatic algorithm configuration. *Scandinavian Journal of Statistics*, 1–42. https://doi.org/10.1111/sjos.12786

APPENDIX

The Appendix consists of three parts. In Appendix A, we present all proofs related to the theorems discussed in this article. In Appendix B, we provide some notes on involved designs and simulations. Appendix C includes some construction results from Algorithm presented in this article.

A PROOFS OF THE MAIN THEORETICAL RESULTS

A.1 Proof of Theorem 1

We establish the theorem based on the four conditions outlined in Definition 1. The design for branching factors is evidently an OA. Next, we prove condition (3). For each level of z_u, the design for $\mathbf{v}^{z_u}$ is $L\left[, (1+\sum_{i=1}^{u-1} m_i) : \sum_{i=1}^{u} m_i\right]$ with a different order of slices. It can be classified as a $SLHD(sn_2,\ m_u, s)$, thereby confirming the condition (3). Now we consider conditions (2) and (4). For each level combination of all branching factors, the design for nested and shared factors is generated by slice L_{i^*} of L, where $i^* \in \{0, \ldots, s-1\}$. Therefore, when this part of the design is collapsed into n_2 levels, it is an $LHD(n_2, k)$. Additionally, based on the structure of D_3 and the process of level expansion, D_3 can be classified as an $LHD(s^2 n_2, r)$. Hence, conditions (2) and (4) are both satisfied. Therefore, design D is an EBLHD with $N = s^2 n_2$ runs and $q+m+r$ columns.

A.2 Proof of Proposition 3

Algorithm 1 can be regarded as an example of Algorithm 2, corresponding to $\lambda = 1$. Therefore, the following calculations are performed for Algorithm 2. To calculate the maximum correlation coefficient of a design, we need to compute the correlation coefficient of any two columns and find the maximum value. Now, we consider the ith and jth columns of D_2, denoted by $d_i^{(n)}$ and $d_j^{(n)}$, and the uth and vth columns of D_3, denoted by $d_u^{(s)}$ and $d_v^{(s)}$, respectively. Next, we compute $\rho(d_i^{(n)}, d_j^{(n)})$, $\rho(d_u^{(s)}, d_v^{(s)})$, and $\rho(d_i^{(n)}, d_u^{(s)})$, respectively.

Based on the construction, D_2 is essentially s L's stacked on top of each other. We let l_i and l_j represent the ith and jth columns of L, respectively. Then, we have

$$\begin{aligned}
\rho(d_i^{(n)}, d_j^{(n)}) &= \frac{\left(d_i^{(n)} - \overline{d}_i^{(n)}\right)\left(d_j^{(n)} - \overline{d}_j^{(n)}\right)}{\sqrt{\left(d_i^{(n)} - \overline{d}_i^{(n)}\right)^2} \cdot \sqrt{\left(d_j^{(n)} - \overline{d}_j^{(n)}\right)^2}} \\
&= \frac{s\left(l_i - \bar{l}_i\right)\left(l_j - \bar{l}_j\right)}{s\sqrt{\left(l_i - \bar{l}_i\right)^2} \cdot \sqrt{\left(l_j - \bar{l}_j\right)^2}} = \rho(l_i, l_j),
\end{aligned}$$

where $\overline{x}$ is defined as the average value of column vector x. Therefore,

$$\rho_{\max}(D_2) \leq \rho_{\max}(P)$$

can be obtained immediately, where P is the first m columns of L.

For design D_3, we know that (1) $d_u^{(s)}$ is generated by $\tilde{d}_u$ and b_u, where $\tilde{d}_u$ is the uth column of $\tilde{D}_3$ and b_u is the uth column of B; (2) $d_v^{(s)}$ is generated by $\tilde{d}_v$ and b_v, where $\tilde{d}_v$ is the vth column of $\tilde{D}_3$ and b_v is the vth column of B. Then, the correlation between $d_u^{(s)}$ and $d_v^{(s)}$ is computed as

$$\begin{aligned}
\rho(d_u^{(s)}, d_v^{(s)}) &= \frac{\left(d_u^{(s)} - \overline{d}_u^{(s)}\right) \cdot \left(d_v^{(s)} - \overline{d}_v^{(s)}\right)}{\sqrt{\left(d_u^{(s)} - \overline{d}_u^{(s)}\right)^2} \cdot \sqrt{\left(d_v^{(s)} - \overline{d}_v^{(s)}\right)^2}} \\
&= \frac{\left(s\tilde{d}_u + b_u \otimes 1_{n_2} - (s\overline{\tilde{d}}_u + \overline{b}_u)\right)\left(s\tilde{d}_v + b_v \otimes 1_{n_2} - (s\overline{\tilde{d}}_v + \overline{b}_v)\right)}{\sqrt{\left(d_u^{(s)} - \overline{d}_u^{(s)}\right)^2} \cdot \sqrt{\left(d_v^{(s)} - \overline{d}_v^{(s)}\right)^2}} \\
&= \Big[s^2\left(\tilde{d}_u - \overline{\tilde{d}}_u\right)\left(\tilde{d}_v - \overline{\tilde{d}}_v\right) + n_2\left(b_u \otimes 1_{n_2} - \overline{b}_u\right)\left(b_v \otimes 1_{n_2} - \overline{b}_v\right) \\
&\quad + s\left(\tilde{d}_u - \overline{\tilde{d}}_u\right)\left(b_v \otimes 1_{n_2} - \overline{b}_v\right) + s\left(\tilde{d}_v - \overline{\tilde{d}}_v\right)\left(b_u \otimes 1_{n_2} - \overline{b}_u\right)\Big] \\
&\quad / \left[\sqrt{\left(d_u^{(s)} - \overline{d}_u^{(s)}\right)^2} \cdot \sqrt{\left(d_v^{(s)} - \overline{d}_v^{(s)}\right)^2}\right] \\
&\hat{=} \frac{I_{11} + I_{22} + I_{12}}{\sqrt{\left(d_u^{(s)} - \overline{d}_u^{(s)}\right)^2} \cdot \sqrt{\left(d_v^{(s)} - \overline{d}_v^{(s)}\right)^2}},
\end{aligned}$$

where each term in the formula is calculated as follows:

$$
\begin{aligned}
&\sqrt{\left(d_u^{(s)}-\overline{d}_u^{(s)}\right)^2}\cdot\sqrt{\left(d_v^{(s)}-\overline{d}_v^{(s)}\right)^2}=\frac{\lambda s^2 n_2(\lambda^2 s^4 n_2^2-1)}{12},\\
&I_{11}=s^2\left(\tilde{d}_u-\overline{\tilde{d}}_u\right)\left(\tilde{d}_v-\overline{\tilde{d}}_v\right)=s^2\cdot s\left(l_{u+m}-\bar{l}_{u+m}\right)\left(l_{v+m}-\bar{l}_{v+m}\right)\\
&\quad=\frac{s^3\lambda s n_2\left(\lambda^2 s^2 n_2^2-1\right)}{12}\rho_{L,(m+u)(m+v)},\\
&I_{22}=n_2\left(b_u\otimes 1_{n_2}-\overline{b}_u\right)\left(b_v\otimes 1_{n_2}-\overline{b}_v\right)=\frac{n_2\lambda s\cdot s(s^2-1)}{12}\rho_{B,uv},\\
&I_{12}=s\left(\tilde{d}_u-\overline{\tilde{d}}_u\right)\left(b_v\otimes 1_{n_2}-\overline{b}_v\right)\\
&\qquad+s\left(\tilde{d}_v-\overline{\tilde{d}}_v\right)\left(b_u\otimes 1_{n_2}-\overline{b}_u\right)=0.
\end{aligned}
$$

Therefore, we have

$$
\rho(d_u^{(s)},d_v^{(s)})=\frac{I_{11}+I_{22}}{\sqrt{\left(d_u^{(s)}-\overline{d}_u^{(s)}\right)^2}\cdot\sqrt{\left(d_v^{(s)}-\overline{d}_v^{(s)}\right)^2}}=\frac{s^2(\lambda^2 s^2 n_2^2-1)\rho_{L,(m+u)(m+v)}+(s^2-1)\rho_{B,uv}}{\lambda^2 s^4 n_2^2-1},
$$

and

$$
\rho_{\max}(D_3)\le\frac{s^2(\lambda^2 s^2 n_2^2-1)\rho_{\max}(Q)+(s^2-1)\rho_{\max}(B)}{\lambda^2 s^4 n_2^2-1},
$$

where Q is the last r columns of L.

The correlation $\rho(d_i^{(n)},d_u^{(s)})$, it can be computed as

$$
\begin{aligned}
\rho(d_i^{(n)},d_u^{(s)})&=\frac{\left(d_i^{(n)}-\overline{d}_i^{(n)}\right)\left(d_u^{(s)}-\overline{d}_u^{(s)}\right)}{\sqrt{\left(d_i^{(n)}-\overline{d}_i^{(n)}\right)^2}\cdot\sqrt{\left(d_u^{(s)}-\overline{d}_u^{(s)}\right)^2}}\\
&=\frac{\left(d_i^{(n)}-\overline{d}_i^{(n)}\right)\left(s\tilde{d}_u+b_u\otimes 1_{n_2}-(s\overline{\tilde{d}}_u+\overline{b}_u)\right)}{\sqrt{\left(d_i^{(n)}-\overline{d}_i^{(n)}\right)^2}\cdot\sqrt{\left(d_u^{(s)}-\overline{d}_u^{(s)}\right)^2}}\\
&=\frac{s\left(d_i^{(n)}-\overline{d}_i^{(n)}\right)\left(\tilde{d}_u-\overline{\tilde{d}}_u\right)+\left(d_i^{(n)}-\overline{d}_i^{(n)}\right)\left(b_u\otimes 1_{n_2}-\overline{b}_u\right)}{\sqrt{\left(d_i^{(n)}-\overline{d}_i^{(n)}\right)^2}\cdot\sqrt{\left(d_u^{(s)}-\overline{d}_u^{(s)}\right)^2}}\\
&=s\sqrt{\frac{\lambda^2 s^2 n_2^2-1}{\lambda^2 s^4 n_2^2-1}}\rho_{L,i(m+u)},
\end{aligned}
$$

where $\rho_{L,i(m+u)}$ is the correlation of the ith and $m+u$th columns of L. Therefore, $\rho_{\max}(D_2,D_3)\le s\sqrt{\frac{\lambda^2 s^2 n_2^2-1}{\lambda^2 s^4 n_2^2-1}}\rho_{\max}(P,Q)$. The proof of Proposition 3 is complete.

A.3 Proof of Theorem 3

According to the way that L is generated, it is clearly an $SLHD(rn, m, r)$. Next, we calculate $\rho_{\max}(L)$ and $\rho_{\max}(L_i)$, $i = 0, \ldots, r-1$.

We take any two columns from L, denoted as l_u and l_v, where $u, v = 1, \ldots, m$. Based on the way that L is generated, we have

$$l_u = 1_r \otimes rg_u + h_u \otimes 1_n, \; l_v = 1_r \otimes rg_v + h_v \otimes 1_n,$$

where g_u and h_u represent the uth columns of G and H, respectively, and g_v and h_v represent the vth columns of G and H, respectively. Then, we have

$$\begin{aligned}
\rho(l_u, l_v) &= \frac{\left(l_u - \bar{l}_u\right)\left(l_v - \bar{l}_v\right)}{\sqrt{\left(l_u - \bar{l}_u\right)^2} \cdot \sqrt{\left(l_u - \bar{l}_u\right)^2}} \\
&= \frac{\left(1_r \otimes rg_u + h_u \otimes 1_n - (r\bar{g}_u + \bar{h}_u)\right)\left(1_r \otimes rg_v + h_v \otimes 1_n - (r\bar{g}_v + \bar{h}_v)\right)}{\sqrt{\left(l_u - \bar{l}_u\right)^2} \cdot \sqrt{\left(l_u - \bar{l}_u\right)^2}} \\
&= \Big[\left(1_r \otimes rg_u - r\bar{g}_u\right)\left(1_r \otimes rg_v - r\bar{g}_v\right) + \left(h_u \otimes 1_n - \bar{h}_u\right)\left(h_v \otimes 1_n - \bar{h}_v\right) \\
&\quad \left(1_r \otimes rg_u - r\bar{g}_u\right)\left(h_v \otimes 1_n - \bar{h}_v\right) + \left(1_r \otimes rg_v - r\bar{g}_v\right)\left(h_u \otimes 1_n - \bar{h}_u\right)\Big] \\
&\quad / \left[\sqrt{\left(l_u - \bar{l}_u\right)^2} \cdot \sqrt{\left(l_u - \bar{l}_u\right)^2}\right] \\
&\hat{=} \frac{I_{11} + I_{22} + I_{12}}{\sqrt{\left(l_u - \bar{l}_u\right)^2} \cdot \sqrt{\left(l_u - \bar{l}_u\right)^2}},
\end{aligned}$$

where

$$\begin{aligned}
I_{11} &= \left(1_r \otimes rg_u - r\bar{g}_u\right)\left(h_v \otimes 1_n - \bar{h}_v\right) = r^3(g_u - \bar{g}_u)(g_v - \bar{g}_v) = \frac{r^3 n(n^2-1)\rho_{G,uv}}{12}, \\
I_{22} &= \left(h_u \otimes 1_n - \bar{h}_u\right)\left(h_v \otimes 1_n - \bar{h}_v\right) = n(h_u - \bar{h}_u)(h_v - \bar{h}_v) = \frac{nr(r^2-1)\rho_{H,uv}}{12}, \\
I_{12} &= \left(1_r \otimes rg_u - r\bar{g}_u\right)\left(h_v \otimes 1_n - \bar{h}_v\right) + \left(1_r \otimes rg_v - r\bar{g}_v\right)\left(h_u \otimes 1_n - \bar{h}_u\right) \\
&= r\sum_{j=1}^{n}\sum_{l=1}^{r}(g_{ju} - \bar{g}_u)(h_{lv} - \bar{h}_v) + r\sum_{j=1}^{n}\sum_{l=1}^{r}(g_{jv} - \bar{g}_v)(h_{lu} - \bar{h}_u) = 0.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\rho(l_u, l_v) &= \frac{r^2(n^2-1)\rho_{G,uv} + (r^2-1)\rho_{H,uv}}{r^2n^2-1}, \\
\rho_{\max}(L) &\leq \frac{r^2(n^2-1)\rho_{\max}(G) + (r^2-1)\rho_{\max}(H)}{r^2n^2-1}.
\end{aligned}$$

Now, we focus on the correlation of each slice L_i, $i = 0, \ldots, r-1$. We let l_u^i and l_v^i denote the uth and vth columns of L_i, then, we have $l_u^i = rg_u + h_{(i+1)u}$ and $l_v^i = rg_v + h_{(i+1)v}$. $\rho(l_u^i, l_v^i)$ can be calculated as

$$\begin{aligned}
\rho(l_u^i, l_v^i) &= \frac{\left(l_u^i - \bar{l}_u^i\right)\left(l_v^i - \bar{l}_v^i\right)}{\sqrt{\left(l_u^i - \bar{l}_u^i\right)^2} \cdot \sqrt{\left(l_v^i - \bar{l}_v^i\right)^2}} \\
&= \frac{\left(rg_u + h_{(i+1)u} - (r\bar{g}_u + h_{(i+1)u})\right)\left(rg_v + h_{(i+1)v} - (r\bar{g}_v + h_{(i+1)v})\right)}{\sqrt{\left(l_u^i - \bar{l}_u^i\right)^2} \cdot \sqrt{\left(l_v^i - \bar{l}_v^i\right)^2}} \\
&= \frac{r^2\left(g_u - \bar{g}_u\right)\left(g_v - \bar{g}_v\right)}{\sqrt{r^2\left(g_u - \bar{g}_u\right)^2} \cdot \sqrt{r^2\left(g_v - \bar{g}_v\right)^2}} \\
&= \rho_{G,uv}.
\end{aligned}$$

Therefore,

$$\rho_{\max}(L_i) \leq \rho_{\max}(G), \ i = 0, \ldots, r-1.$$

The proof of Theorem 3 is complete.

A.4 Proof of Theorem 4

First, we demonstrate that L is an $SLHD(\lambda^2 s^3, k, \lambda s)$. Because of step 1 and step 2 in Algorithm 3, it is not difficult to see that each $\tilde{L}_i$ is an $LHD(n_2, k)$, $i = 0, \ldots, \lambda s - 1$. After the level expansion of $\tilde{L}$ in step 3, we know that L is an $SLHD(\lambda^2 s^3, k, \lambda s)$.

Now, we prove conclusion (ii). Based on the construction, it is easily determined that each L_i is essentially A after collapsing to s levels; hence, $s \times s$ stratification of each slice L_i can be realized. Additionally, it follows from step 1 in Algorithm 3 and Proposition 1 in He et al. (2018) that design E is an $SOA(\lambda s n_2, k, \lambda s^2, 2+)$, which can achieve stratification on $s \times \lambda s^2$ and $\lambda s^2 \times s$ grids. Because row permutations and level expansion do not affect the stratification properties of a design, L can also achieve stratification on $s \times \lambda s^2$ and $\lambda s^2 \times s$ grids.

Next, we compute the correlation of L. Without loss of generality, we take the uth and vth columns from L, denoted as l_u and l_v, $u, v = 1, \ldots, k$. Based on the construction, they are generated by the uth and vth columns of $\tilde{L}$ and G, respectively, which are called $\tilde{l}_u$, $\tilde{l}_v$, g_u, and g_v. Specifically,

$$l_u = \lambda s \tilde{l}_u + g_u \otimes 1_{n_2}, l_v = \lambda s \tilde{l}_v + g_v \otimes 1_{n_2}.$$

We let $\bar{x}$ denote the mean value of vector x. Then, we have

$$\rho(l_u, l_v) = \frac{\left(l_u - \bar{l}_u\right)\left(l_v - \bar{l}_v\right)}{\sqrt{\left(l_u - \bar{l}_u\right)^2} \cdot \sqrt{\left(l_v - \bar{l}_v\right)^2}}$$

$$
\begin{aligned}
&= \frac{\left(\lambda s\tilde{l}_u + g_u \otimes 1_{n_2} - (\lambda s\bar{\tilde{l}}_u + \bar{g}_u)\right)\left(\lambda s\tilde{l}_v + g_v \otimes 1_{n_2} - (\lambda s\bar{\tilde{l}}_v + \bar{g}_v)\right)}{\sqrt{\left(l_u - \bar{l}_u\right)^2} \cdot \sqrt{\left(l_v - \bar{l}_v\right)^2}} \\
&= \left[\lambda^2 s^2\left(\tilde{l}_u - \bar{\tilde{l}}_u\right)\left(\tilde{l}_v - \bar{\tilde{l}}_v\right) + \left(g_u \otimes 1_{n_2} - \bar{g}_u\right)\left(g_v \otimes 1_{n_2} - \bar{g}_v\right)\right. \\
&\quad \left. + \lambda s\left(\tilde{l}_u - \bar{\tilde{l}}_u\right)\left(g_v \otimes 1_{n_2} - \bar{g}_v\right) + \lambda s\left(\tilde{l}_v - \bar{\tilde{l}}_v\right)\left(g_u \otimes 1_{n_2} - \bar{g}_u\right)\right] \\
&\quad / \left[\sqrt{\left(l_u - \bar{l}_u\right)^2} \cdot \sqrt{\left(l_v - \bar{l}_v\right)^2}\right] \\
&\hat{=} \frac{I_{11} + I_{22} + I_{12}}{\sqrt{(l_u - \bar{l}_u)^2} \cdot \sqrt{\left(l_v - \bar{l}_v\right)^2}},
\end{aligned}
$$

where

$$
\begin{aligned}
I_{11} &= \lambda^2 s^2\left(\tilde{l}_u - \bar{\tilde{l}}_u\right)\left(\tilde{l}_v - \bar{\tilde{l}}_v\right) = \lambda^2 s^2 (e_u - \bar{e}_u)(e_v - \bar{e}_v) \\
&= \lambda^2 s^2 \left(1_{\lambda s} \otimes \lambda s a_u + c_u \otimes 1_{n_2} - (\lambda s \bar{a}_u + \bar{c}_u)\right)\left(1_{\lambda s} \otimes \lambda s a_v + c_v \otimes 1_{n_2} - (\lambda s \bar{a}_v + \bar{c}_v)\right) \\
&= \lambda^2 s^2 \left[(1_{\lambda s} \otimes \lambda s a_u - \lambda s \bar{a}_u)(1_{\lambda s} \otimes \lambda s a_v - \lambda s \bar{a}_v) + \left(c_u \otimes 1_{n_2} - \bar{c}_u\right)\left(c_v \otimes 1_{n_2} - \bar{c}_v\right)\right] \\
&= \lambda^2 s^2 n_2 \left(c_u - \bar{c}_u\right)\left(c_v - \bar{c}_v\right) \\
&= \lambda^2 s^2 n_2 \frac{\lambda s(\lambda^2 s^2 - 1)}{12} \rho_{C,uv}, \\
I_{22} &= n_2 (g_u - \bar{g}_u)(g_v - \bar{g}_v) = \frac{n_2 \lambda s(\lambda^2 s^2 - 1)}{12} \rho_{G,uv}, \\
I_{12} &= \lambda s\left(\tilde{l}_u - \bar{\tilde{l}}_u\right)\left(g_v \otimes 1_{n_2} - \bar{g}_v\right) + \lambda s\left(\tilde{l}_v - \bar{\tilde{l}}_v\right)\left(g_u \otimes 1_{n_2} - \bar{g}_u\right) = 0,
\end{aligned}
$$

e_u and e_v denote the uth and vth columns of E; a_u and a_v denote the uth and vth columns of A; and some of the cross terms in the calculation are clearly 0. Therefore, we have

$$
\begin{aligned}
\rho(l_u, l_v) &= \frac{\lambda^2 s^2 - 1}{\lambda^2 s^2 n_2^2 - 1}(\lambda^2 s^2 \rho_{C,uv} + \rho_{G,uv}), \\
\rho_{\max}(L) &\leq \frac{\lambda^2 s^2 - 1}{\lambda^2 s^2 n_2^2 - 1}(\lambda^2 s^2 \rho_{\max}(C) + \rho_{\max}(G)).
\end{aligned}
$$

The proof of Theorem 4 is complete.

A.5 Proof of Theorem 5

Conclusion (ii) is obviously true; thus, we only prove (i). From Theorem 3, we immediately know that

$$
\rho_{\max}(D_3) \leq \frac{n_1^2(n_2^2 - 1)\rho_{\max}(L_{q+1}) + (n_1^2 - 1)\rho_{\max}(G)}{n_1^2 n_2^2 - 1}.
$$

Next, we calculate $\rho_{\max}(D_{2i})$, $\rho_{\max}(D_{2i}, D_{2j})$ and $\rho_{\max}(D_{2i}, D_3)$, respectively.

For D_{2i}, we take the uth and vth columns from D_{2i}, which are denoted as d_u^i and d_v^i, respectively. They can be expressed as

$$d_u^i = 1_{n_1} \otimes \frac{n_1}{s_i} l_u + b_u \otimes 1_{n_2},\ d_v^i = 1_{n_1} \otimes \frac{n_1}{s_i} l_v + b_v \otimes 1_{n_2},$$

where l_u, l_v, b_u, and b_v are the uth and vth columns of L and B, respectively. Then, we can compute $\rho(d_u^i, d_v^i)$ as follows:

$$\begin{aligned}
\rho(d_u^i, d_v^i) &= \frac{\left(d_u^i - \overline{d}_u^i\right)\left(d_v^i - \overline{d}_v^i\right)}{\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^i - \overline{d}_v^i\right)^2}} \\
&= \frac{\left(1_{n_1} \otimes \frac{n_1}{s_i} l_u + b_u \otimes 1_{n_2} - (\frac{n_1}{s_i}\bar{l}_u + \bar{b}_u)\right)\left(1_{n_1} \otimes \frac{n_1}{s_i} l_v + b_v \otimes 1_{n_2} - (\frac{n_1}{s_i}\bar{l}_v + \bar{b}_v)\right)}{\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^i - \overline{d}_v^i\right)^2}} \\
&= \left[\left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(1_{n_1} \otimes \frac{n_1}{s_i} l_v - \frac{n_1}{s_i}\bar{l}_v\right) + \left(b_u \otimes 1_{n_2} - \bar{b}_u\right)\left(b_v \otimes 1_{n_2} - \bar{b}_v\right)\right. \\
&\quad \left. + \left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(b_v \otimes 1_{n_2} - \bar{b}_v\right) + \left(1_{n_1} \otimes \frac{n_1}{s_i} l_v - \frac{n_1}{s_i}\bar{l}_v\right)\left(b_u \otimes 1_{n_2} - \bar{b}_u\right)\right] \\
&\quad / \left[\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^i - \overline{d}_v^i\right)^2}\right] \\
&\hat{=} \frac{I_{11} + I_{22} + I_{12}}{\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^i - \overline{d}_v^i\right)^2}},
\end{aligned}$$

where

$$\begin{aligned}
&\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^i - \overline{d}_v^i\right)^2} = s_i \cdot \frac{n_1 n_2}{s_i}\left(\frac{n_1^2 n_2^2}{s_i^2} - 1\right)/12, \\
&I_{11} = \left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(1_{n_1} \otimes \frac{n_1}{s_i} l_v - \frac{n_1}{s_i}\bar{l}_v\right) = \frac{n_1^3}{s_i^2}\left(l_u - \bar{l}_u\right)\left(l_v - \bar{l}_v\right) = \frac{n_1^3 n_2 (n_2^2 - 1)}{12 s_i^2}\rho_{L_i,uv}, \\
&I_{22} = \left(b_u \otimes 1_{n_2} - \bar{b}_u\right)\left(b_v \otimes 1_{n_2} - \bar{b}_v\right) = n_2\left(b_u - \bar{b}_u\right)\left(b_v - \bar{b}_v\right) = n_2 s_i \cdot \frac{n_1}{s_i}\left(\frac{n_1^2}{s_i^2} - 1\right)\rho_{B_i,uv}/12, \\
&I_{12} = \left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(b_v \otimes 1_{n_2} - \bar{b}_v\right) + \left(1_{n_1} \otimes \frac{n_1}{s_i} l_v - \frac{n_1}{s_i}\bar{l}_v\right)\left(b_u \otimes 1_{n_2} - \bar{b}_u\right) = 0,
\end{aligned}$$

$\rho_{L_i,uv}$ represents the correlation coefficient of the uth and vth columns in L_i, and $\rho_{B_i,uv}$ is similarly defined. Therefore, after a simple calculation, we have

$$\begin{aligned}
\rho(d_u^i, d_v^i) &= \frac{n_1^2(n_2^2 - 1)\rho_{L_i,uv} + (n_1^2 - s_i^2)\rho_{B_i,uv}}{n_1^2 n_2^2 - s_i^2}, \\
\rho_{\max}(D_{2i}) &\le \frac{n_1^2(n_2^2 - 1)\rho_{\max}(L_i) + (n_1^2 - s_i^2)\rho_{\max}(B_i)}{n_1^2 n_2^2 - s_i^2}.
\end{aligned}$$

From D_{2i} and D_{2j}, we take the uth and vth columns, respectively, and denote them as d_u^i and d_v^j, respectively. Based on the algorithm, they can be expressed as

$$d_u^i = 1_{n_1} \otimes \frac{n_1}{s_i} l_u + b_u \otimes 1_{n_2}, \quad d_v^j = 1_{n_1} \otimes \frac{n_1}{s_j} l_v + b_v \otimes 1_{n_2},$$

where l_u and b_u are the uth columns of B_i and L_i, respectively, and l_v and b_v are the vth columns of B_j and L_j, respectively. Then, we have

$$\begin{aligned}
\rho(d_u^i, d_v^j) &= \frac{\left(d_u^i - \overline{d}_u^i\right)\left(d_v^j - \overline{d}_v^j\right)}{\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^j - \overline{d}_v^j\right)^2}} \\
&= \frac{\left(1_{n_1} \otimes \frac{n_1}{s_i} l_u + b_u \otimes 1_{n_2} - \left(\frac{n_1}{s_i}\bar{l}_u + \overline{b}_u\right)\right)\left(1_{n_1} \otimes \frac{n_1}{s_j} l_v + b_v \otimes 1_{n_2} - \left(\frac{n_1}{s_j}\bar{l}_v + \overline{b}_v\right)\right)}{\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^j - \overline{d}_v^j\right)^2}} \\
&= \left[\left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(1_{n_1} \otimes \frac{n_1}{s_j} l_v - \frac{n_1}{s_j}\bar{l}_v\right) + \left(b_u \otimes 1_{n_2} - \overline{b}_u\right)\left(b_v \otimes 1_{n_2} - \overline{b}_v\right)\right. \\
&\quad \left. + \left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(b_v \otimes 1_{n_2} - \overline{b}_v\right) + \left(1_{n_1} \otimes \frac{n_1}{s_j} l_v - \frac{n_1}{s_j}\bar{l}_v\right)\left(b_u \otimes 1_{n_2} - \overline{b}_u\right)\right] \\
&\quad / \left[\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^j - \overline{d}_v^j\right)^2}\right] \\
&\triangleq \frac{I_{11} + I_{22} + I_{12}}{\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^j - \overline{d}_v^j\right)^2}},
\end{aligned}$$

where

$$\begin{aligned}
\sqrt{\left(d_u^i - \overline{d}_u^i\right)^2} \cdot \sqrt{\left(d_v^i - \overline{d}_v^i\right)^2} &= \sqrt{\frac{s_i n_1 n_2}{s_i}\left(\frac{n_1^2 n_2^2}{s_i^2} - 1\right)/12} \cdot \sqrt{\frac{s_j n_1 n_2}{s_j}\left(\frac{n_1^2 n_2^2}{s_j^2} - 1\right)/12}, \\
I_{11} &= \left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(1_{n_1} \otimes \frac{n_1}{s_j} l_v - \frac{n_1}{s_j}\bar{l}_v\right) = \frac{n_1^3}{s_i s_j}\left(l_u - \bar{l}_u\right)\left(l_v - \bar{l}_v\right) \\
&= \frac{n_1^3 n_2 (n_2^2 - 1)}{12 s_i s_j} \rho(L_{iu}, L_{jv}), \\
I_{22} &= \left(b_u \otimes 1_{n_2} - \overline{b}_u\right)\left(b_v \otimes 1_{n_2} - \overline{b}_v\right) = n_2\left(b_u - \overline{b}_u\right)\left(b_v - \overline{b}_v\right) \\
&= n_2 \cdot \sqrt{s_i \frac{n_1}{s_i}\left(\frac{n_1^2}{s_i^2} - 1\right)/12} \cdot \sqrt{s_j \frac{n_1}{s_j}\left(\frac{n_1^2}{s_j^2} - 1\right)/12} \cdot \rho(B_{iu}, B_{jv}), \\
I_{12} &= \left(1_{n_1} \otimes \frac{n_1}{s_i} l_u - \frac{n_1}{s_i}\bar{l}_u\right)\left(b_v \otimes 1_{n_2} - \overline{b}_v\right) + \left(1_{n_1} \otimes \frac{n_1}{s_j} l_v - \frac{n_1}{s_j}\bar{l}_v\right)\left(b_u \otimes 1_{n_2} - \overline{b}_u\right) = 0,
\end{aligned}$$

$\rho(L_{iu}, L_{jv})$ represents the correlation coefficient between the uth column of L_i and the vth column of L_j, and $\rho(B_{iu}, B_{jv})$ is defined similarly. After simplification, we obtain

$$\rho(d_u^i, d_v^j) = \frac{n_1^2(n_2^2-1)\rho(L_{iu}, L_{jv}) + \sqrt{(n_1^2 - s_i^2)(n_1^2 - s_j^2)}\rho(B_{iu}, B_{jv})}{\sqrt{(n_1^2 n_2^2 - s_i^2)(n_1^2 n_2^2 - s_j^2)}},$$

$$\rho_{\max}(D_{2i}, D_{2j}) \leq \frac{n_1^2(n_2^2-1)\rho_{\max}(L_i, L_j) + \sqrt{(n_1^2 - s_i^2)(n_1^2 - s_j^2)}\rho_{\max}(B_i, B_j)}{\sqrt{(n_1^2 n_2^2 - s_i^2)(n_1^2 n_2^2 - s_j^2)}}.$$

For D_{2i} and D_3, the calculation process for $\rho(d_u^i, d_{u'}^{(s)})$ is similar to that described above, where $d_{u'}^{(s)}$ represents the u'th column of D_3; hence, we omit it here. The following results can be obtained:

$$\rho(d_u^i, d_{u'}^{(s)}) = \frac{n_1^2(n_2^2-1)\rho(L_{iu}, L_{(q+1)u'}) + \sqrt{(n_1^2 - s_i^2)(n_1^2 - 1)}\rho(B_{iu}, G_{u'})}{\sqrt{(n_1^2 n_2^2 - s_i^2)(n_1^2 n_2^2 - 1)}},$$

$$\rho_{\max}(D_{2i}, D_3) \leq \frac{n_1^2(n_2^2-1)\rho_{\max}(L_i, L_{q+1}) + \sqrt{(n_1^2 - s_i^2)(n_1^2 - 1)}\rho_{\max}(B_i, G)}{\sqrt{(n_1^2 n_2^2 - s_i^2)(n_1^2 n_2^2 - 1)}},$$

where $L_{(q+1)u'}$ is the u'th column of L_{q+1} and $G_{u'}$ is the u'th column of G. The proof of Theorem 5 is complete.

B NOTES ON THE DESIGNS AND SIMULATIONS

B.1 The SLHD in Example 4

The generated L: *SLHD* (108, 6, 6) is

$$L^T = \left(\begin{array}{cccccccccccccccccc}
0 & 36 & 72 & 6 & 42 & 78 & 12 & 48 & 84 & 18 & 54 & 90 & 24 & 60 & 96 & 30 & 66 & 102 \\
7 & 43 & 79 & 19 & 55 & 91 & 67 & 103 & 31 & 97 & 25 & 61 & 49 & 85 & 13 & 7 & 1 & 37 \\
14 & 50 & 86 & 68 & 104 & 32 & 20 & 56 & 92 & 74 & 2 & 38 & 80 & 8 & 44 & 62 & 98 & 26 \\
21 & 57 & 93 & 99 & 27 & 63 & 75 & 3 & 39 & 15 & 51 & 87 & 69 & 105 & 33 & 45 & 81 & 9 \\
28 & 64 & 100 & 52 & 88 & 16 & 82 & 10 & 46 & 70 & 106 & 34 & 4 & 40 & 76 & 94 & 22 & 58 \\
35 & 71 & 107 & 77 & 5 & 41 & 65 & 101 & 29 & 47 & 83 & 11 & 95 & 23 & 59 & 17 & 53 & 89
\end{array} \right.$$

$$\begin{array}{cccccccccccccccccc}
1 & 37 & 73 & 7 & 43 & 79 & 13 & 49 & 85 & 19 & 55 & 91 & 25 & 61 & 97 & 31 & 67 & 103 \\
9 & 45 & 81 & 57 & 93 & 21 & 105 & 33 & 69 & 63 & 99 & 27 & 87 & 15 & 51 & 3 & 39 & 75 \\
53 & 89 & 17 & 35 & 71 & 107 & 95 & 23 & 59 & 77 & 5 & 41 & 47 & 83 & 11 & 29 & 65 & 101 \\
94 & 22 & 58 & 100 & 28 & 64 & 4 & 40 & 76 & 52 & 88 & 16 & 70 & 106 & 34 & 10 & 46 & 82 \\
62 & 98 & 26 & 86 & 14 & 50 & 44 & 80 & 8 & 32 & 68 & 104 & 74 & 2 & 38 & 20 & 56 & 92 \\
102 & 30 & 66 & 36 & 72 & 0 & 60 & 96 & 24 & 78 & 6 & 42 & 18 & 54 & 90 & 12 & 48 & 84
\end{array}$$

$$
\left.\begin{array}{cccccccccccccccccc}
2 & 38 & 74 & 8 & 44 & 80 & 14 & 50 & 86 & 20 & 56 & 92 & 26 & 62 & 98 & 32 & 68 & 104 \\
47 & 83 & 11 & 95 & 23 & 59 & 71 & 107 & 35 & 101 & 29 & 65 & 17 & 53 & 89 & 5 & 41 & 77 \\
15 & 51 & 87 & 105 & 33 & 69 & 93 & 21 & 57 & 39 & 75 & 3 & 9 & 45 & 81 & 63 & 99 & 27 \\
90 & 18 & 54 & 24 & 60 & 96 & 36 & 72 & 0 & 48 & 84 & 12 & 30 & 66 & 102 & 78 & 6 & 42 \\
97 & 25 & 61 & 49 & 85 & 13 & 7 & 43 & 79 & 103 & 31 & 67 & 1 & 37 & 73 & 55 & 91 & 19 \\
70 & 106 & 34 & 40 & 76 & 4 & 100 & 28 & 64 & 10 & 46 & 82 & 22 & 58 & 94 & 88 & 16 & 52 \\[6pt]
3 & 39 & 75 & 9 & 45 & 81 & 15 & 51 & 87 & 21 & 57 & 93 & 27 & 63 & 99 & 33 & 69 & 105 \\
82 & 10 & 46 & 58 & 94 & 22 & 106 & 34 & 70 & 28 & 64 & 100 & 16 & 52 & 88 & 40 & 76 & 4 \\
84 & 12 & 48 & 102 & 30 & 66 & 54 & 90 & 18 & 0 & 36 & 72 & 42 & 78 & 6 & 24 & 60 & 96 \\
20 & 56 & 92 & 62 & 98 & 26 & 38 & 74 & 2 & 14 & 50 & 86 & 104 & 32 & 68 & 80 & 8 & 44 \\
65 & 101 & 29 & 17 & 53 & 89 & 83 & 11 & 47 & 35 & 71 & 107 & 41 & 77 & 5 & 95 & 23 & 59 \\
67 & 103 & 31 & 73 & 1 & 37 & 25 & 61 & 97 & 7 & 43 & 79 & 91 & 19 & 55 & 49 & 85 & 13 \\[6pt]
4 & 40 & 76 & 10 & 46 & 82 & 16 & 52 & 88 & 22 & 58 & 94 & 28 & 64 & 100 & 34 & 70 & 106 \\
44 & 80 & 8 & 92 & 20 & 56 & 32 & 68 & 104 & 26 & 62 & 98 & 50 & 86 & 14 & 74 & 2 & 38 \\
85 & 13 & 49 & 67 & 103 & 31 & 19 & 55 & 91 & 37 & 73 & 1 & 7 & 43 & 79 & 97 & 25 & 61 \\
59 & 95 & 23 & 65 & 101 & 29 & 5 & 41 & 77 & 89 & 17 & 53 & 107 & 35 & 71 & 11 & 47 & 83 \\
24 & 60 & 96 & 84 & 12 & 48 & 6 & 42 & 78 & 66 & 102 & 30 & 72 & 0 & 36 & 54 & 90 & 18 \\
105 & 33 & 69 & 3 & 39 & 75 & 27 & 63 & 99 & 81 & 9 & 45 & 57 & 93 & 21 & 51 & 87 & 15 \\[6pt]
5 & 41 & 77 & 11 & 47 & 83 & 17 & 53 & 89 & 23 & 59 & 95 & 29 & 65 & 101 & 35 & 71 & 107 \\
78 & 6 & 42 & 18 & 54 & 90 & 30 & 66 & 102 & 60 & 96 & 24 & 84 & 12 & 48 & 36 & 72 & 0 \\
52 & 88 & 16 & 34 & 70 & 106 & 58 & 94 & 22 & 4 & 40 & 76 & 82 & 10 & 46 & 100 & 28 & 64 \\
55 & 91 & 19 & 25 & 61 & 97 & 73 & 1 & 37 & 85 & 13 & 49 & 31 & 67 & 103 & 43 & 79 & 7 \\
99 & 27 & 63 & 15 & 51 & 87 & 45 & 81 & 9 & 105 & 33 & 69 & 39 & 75 & 3 & 21 & 57 & 93 \\
32 & 68 & 104 & 2 & 38 & 74 & 98 & 26 & 62 & 44 & 80 & 8 & 56 & 92 & 20 & 86 & 14 & 50
\end{array}\right). \tag{B1}
$$

B.2 The simulations in Table 5

The CROAs involved in the construction of the SLHDs in Table 5 are listed below, whereas C and G were generated by the R function SLHD.

- $SLHD(27, 3, 3)$ in Table 5: $CROA(9, 3, 3, 2)$ is as follows, and $C, G \in LHD(3, 3)$.

$$
CROA(9, 3, 3, 2): \left(\begin{array}{ccc|ccc|ccc}
0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\
0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\
0 & 2 & 1 & 1 & 0 & 2 & 2 & 1 & 0
\end{array}\right)^T \tag{B2}
$$

- $SLHD(64, 4, 4)$ in Table 5: $CROA(16, 4, 4, 2)$ is as follows, and $C, G \in LHD(4, 4)$.

$$CROA(16,4,4,2) : \left(\begin{array}{cccc|cccc|cccc|cccc} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 0 & 3 & 2 & 2 & 3 & 0 & 1 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 2 & 3 & 0 & 1 & 3 & 2 & 1 & 0 & 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 & 3 & 2 & 1 & 0 & 1 & 0 & 3 & 2 & 2 & 3 & 0 & 1 \end{array}\right)^T . \tag{B3}$$

- $SLHD(125, 5, 5)$ in Table 5: $CROA(25, 5, 5, 2)$ is as follows, and $C, G \in LHD(5, 5)$.

$$CROA(25,5,5,2) : \left(\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc} 0&1&2&3&4 & 0&1&2&3&4 & 0&1&2&3&4 & 0&1&2&3&4 & 0&1&2&3&4 \\ 0&1&2&3&4 & 1&2&4&0&3 & 2&4&3&1&0 & 3&0&1&4&2 & 4&3&0&2&1 \\ 0&2&3&4&1 & 1&4&0&3&2 & 2&3&1&0&4 & 3&1&4&2&0 & 4&0&2&1&3 \\ 0&3&4&1&2 & 1&0&3&2&4 & 2&1&0&4&3 & 3&4&2&0&1 & 4&2&1&3&0 \\ 0&4&1&2&3 & 1&3&2&4&0 & 2&0&4&3&1 & 3&2&0&1&4 & 4&1&3&0&2 \end{array}\right)^T . \tag{B4}$$

- $SLHD(108, 6, 6)$ in Table 5: The required $CROA(18, 6, 3, 2)$ is the same as A in Equation (7), and $C, G \in LHD(6, 6)$.

B.3 On the simulation studies

The first three designs are $EBLHD(40, 1 + 2 + 2)$ with a 5-level branching factor, $EBLHD(64, 1 + 4 + 4)$ with a 4-level branching factor, and $EBLHD(90, 1 + 3 + 4)$ with a 5-level branching factor. Their structures are listed in Tables B1–B3.

1. For the EBLHDs:
   a. For L in the three designs, $OA(8, 4, 2, 3)$-based $LHD(8, 4)$, $OA(16, 8, 2, 3)$-based $LHD(16, 8)$, and $OA(18, 7, 3, 2)$-based $LHD(18, 7)$ are used, respectively.
   (All the OAs involved here can be found at http://neilsloane.com/oadir/#2_3.)

TABLE B1 $EBLHD(40, 1 + 2 + 2)$.

$\boldsymbol{D_1}$	$(\boldsymbol{D_2}, \tilde{\boldsymbol{D}}_3)$
$\mathbf{0}_8$	L
$\mathbf{1}_8$	L
$\mathbf{2}_8$	L
$\mathbf{3}_8$	L
$\mathbf{4}_8$	L

TABLE B2 $EBLHD(64, 1 + 4 + 4)$.

$\boldsymbol{D_1}$	$(\boldsymbol{D_2}, \tilde{\boldsymbol{D}}_3)$
$\mathbf{0}_{16}$	L
$\mathbf{1}_{16}$	L
$\mathbf{2}_{16}$	L
$\mathbf{3}_{16}$	L

TABLE B3 $EBLHD(90, 1+3+4)$.

$\boldsymbol{D_1}$	$(\boldsymbol{D_2}, \tilde{\boldsymbol{D}}_3)$
$\mathbf{0}_{18}$	L
$\mathbf{1}_{18}$	L
$\mathbf{2}_{18}$	L
$\mathbf{3}_{18}$	L
$\mathbf{4}_{18}$	L

b. For B in expansion $D_3 = \tilde{D}_3 + B \otimes 1_n$ for the three designs, $LHD(5, 2)$, $LHD(4, 4)$, and $LHD(5, 4)$ are used, respectively, which are randomly generated by the SLHD function.

2. For random BLHDs, the D_3 is generated by rLHD function, and the D_2 part is obtained by juxtaposing rows of s random LHDs, where s represents the level number of branching factor.

EBLHD(324, 2+4+2) with the same experimental settings as in Example 2:

1. The EBLHDs are produced based on Theorems 2 and 4. We ran the algorithm 100 times with different values of C, G, and B, which were generated randomly by the R function SLHD. The remaining $MOA(18, 3^3 6^1, 2)$ and $CROA(18, 6, 3, 2)$ are the same as in Examples 2 and 4.
2. For random BLHDs, all the required LHDs are generated by the rLHD function.

B.4 On the applications in Section 6

TABLE B4 Designs used in optim.

				L											
A		***B***		**Run**						**Run**					
0	0	0	1	1	3	5	9	4	4	26	2	4	6	7	1
0	1	0	1	2	17	19	27	38	40	27	12	18	22	31	49
1	0	1	0	3	29	29	33	46	14	28	24	22	32	49	19
1	1	1	0	4	35	35	41	18	26	29	32	34	42	11	27
				5	41	45	19	28	38	30	42	46	14	25	31
				6	5	15	11	14	12	31	4	10	16	15	17
				7	15	21	43	6	32	32	18	26	46	1	33
				8	27	43	7	32	20	33	26	42	0	33	23
				9	39	3	39	26	46	34	36	8	36	29	43
				10	49	31	29	40	0	35	48	36	20	41	9
				11	1	25	23	20	28	36	6	28	24	27	21
				12	13	47	37	10	2	37	14	48	34	17	5
				13	23	37	13	8	48	38	20	30	18	5	47
				14	37	17	1	42	30	39	34	16	2	43	35
				15	43	7	47	36	16	40	40	0	48	35	15

TABLE B4 (Continued)

		L											
A	*B*	Run						Run					
		16	7	39	31	30	34	41	0	32	30	37	37
		17	19	1	17	48	24	42	16	2	10	47	29
		18	25	11	45	22	6	43	28	12	44	21	3
		19	31	49	21	2	18	44	30	44	26	9	11
		20	45	27	3	16	44	45	46	20	8	13	45
		21	9	41	49	44	42	46	8	40	40	45	41
		22	11	33	5	24	10	47	10	38	4	23	13
		23	21	9	25	12	36	48	22	6	28	19	39
		24	33	23	15	34	8	49	38	24	12	39	7
		25	47	13	35	0	22	50	44	14	38	3	25

TABLE B5 Factors in the ACOQAP algorithm.

Parameter	Domain	Default
algorithm	{AS, EAS, RAS, ACS, MMAS, BWAS}	MMAS
m	[1, 500]	25
α	(0.0, 5.0)	1.0
β	(0.0, 10.0)	2.0
ρ	(0.01, 1.0)	0.2
q_0	(0.0, 1.0)	0.0
cl	[5, 50]	20
ξ	(0.01, 1.0)	-
rasrank	[1, 100]	-
m_{elite}	[1, 750]	-
p_{dec}	(0.001, 0.5)	-
ph-limits	{Yes, no}	Yes
slen	[20, 500]	250
restart	{Never, branch-factor, distance, always}	Branch-factor
res_{bf}	(1.0, 2.0)	1.00001
res_{dist}	(0.01, 0.1)	-
res_{it}	[1, 500]	250
localsearch	{None, 2-opt, 2.5-opt,3-opt}	3-opt
dlb-bits	{Yes, no}	Yes
nnls	[5,50]	20

TABLE B5 (Continued)

Parameter	Domain	Default
rasrank	When *algo* = RAS	
m_{elite}	When *algo* = EAS	
ξ	When *algo* = ACS	
p_{dec}	When *ph-limits* = yes	
ph-limits	When *algo* $\neq$ ACS	
slen	When *algo* = MMAS	
res_{bf}	When *restart* = branch-factor	
res_{dist}	When *restart* = distance	
res_{it}	When *restart* $\neq$ never	
$dlb - bits, nnls$	When *localsearch* $\neq$ none	

TABLE B6 Designs used in ACOQAP.

Run	A			B									L^T			
1	0	0	0	6	3	6	2	6	3	4	8	13	1	3	0	2
2	0	0	1	4	2	2	0	0	5	2	4	4	3	1	0	2
3	0	1	0	5	5	1	3	7	7	6	6	3	1	0	2	3
4	0	1	1	1	6	4	1	1	0	1	7	18	3	1	0	2
5	0	2	0	2	1	7	6	3	1	3	0	18	0	3	1	2
6	0	2	1	0	4	5	4	2	6	5	7	19	2	3	1	0
7	0	3	0	3	7	3	5	5	4	0	2	1	0	2	1	3
8	0	3	1	7	0	0	7	4	2	7	0	3	2	1	0	3
9	1	0	0	6	5	4	3	5	7	1	7	4	1	0	2	3
10	1	0	1	3	3	7	0	6	3	5	5	12				
11	1	1	0	1	7	6	6	4	6	5	9	12				
12	1	1	1	5	1	3	1	1	0	3	5	13				
13	1	2	0	7	6	2	5	2	1	9	3	6				
14	1	2	1	0	0	5	4	3	4	7	1	6				
15	1	3	0	4	2	1	7	7	2	5	6	15				
16	1	3	1	2	4	0	2	0	5	2	8	10				
17	2	0	0	5	1	4	7	1	2	9	6	9				
18	2	0	1	6	2	7	3	3	5	0	2	16				
19	2	1	0	4	7	0	1	0	3	8	0	8				
20	2	1	1	0	5	3	6	2	7	7	3	0				
21	2	2	0	7	4	2	5	4	0	0	5	2				
22	2	2	1	3	0	5	0	5	1	1	2	17				
23	2	3	0	1	3	1	4	6	4	4	3	14				
24	2	3	1	2	6	6	2	7	6	3	1	2				
25	3	0	0	7	6	3	7	1	1	7	9	16				
26	3	0	1	0	5	2	2	5	7	3	1	9				

TABLE B6 (Continued)

Run	A		B									L^T
27	3	1	0	6	4	0	0	6	3	4	2	17
28	3	1	1	1	3	7	6	4	0	9	8	15
29	3	2	0	5	0	1	4	3	6	4	4	10
30	3	2	1	3	1	5	1	2	2	6	9	14
31	3	3	0	2	7	4	3	0	4	1	5	0
32	3	3	1	4	2	6	5	7	5	9	7	7
33	4	0	0	3	3	7	2	6	7	8	3	11
34	4	0	1	6	6	2	5	7	6	6	0	8
35	4	1	0	1	0	3	0	4	3	0	4	7
36	4	1	1	0	2	4	4	0	1	2	1	5
37	4	2	0	2	7	5	3	5	0	8	6	19
38	4	2	1	5	4	6	6	2	4	2	8	11
39	4	3	0	4	5	0	1	1	5	8	4	5
40	4	3	1	7	1	1	7	3	2	6	9	1

C CONSTRUCTION RESULTS

TABLE C1 Results from Algorithms 1 and 3.

$OA(n_1, q, s, 2)$	$CROA(n_2, k, s, 2)$	$SLHD(n, k, s)$	$EBLHD(N, q + k)$	Stratification
$OA(4, 3, 2, 2)$	$CROA(4, 2, 2, 2)$	$SLHD(8, 2, 2)$	$EBLHD(16, 2 + 2)$	$2 \times 2 *, 2 \times 4, 4 \times 2$
$OA(9, 4, 3, 2)$	$CROA(9, 3, 3, 2)$	$SLHD(27, 3, 3)$	$EBLHD(81, 3 + 3)$	$3 \times 3 *, 3 \times 9, 9 \times 3$
$OA(16, 5, 4, 2)$	$CROA(16, 4, 4, 2)$	$SLHD(64, 4, 4)$	$EBLHD(256, 4 + 4)$	$4 \times 4 *, 4 \times 16, 16 \times 4$
$OA(25, 6, 5, 2)$	$CROA(25, 5, 5, 2)$	$SLHD(125, 5, 5)$	$EBLHD(625, 5 + 5)$	$5 \times 5 *, 5 \times 25, 25 \times 5$
$OA(36, 7, 6, 2)$	$CROA(36, 6, 6, 2)$	$SLHD(216, 6, 6)$	$EBLHD(1296, 6 + 6)$	$6 \times 6 *, 6 \times 36, 36 \times 6$
$OA(49, 8, 7, 2)$	$CROA(49, 7, 7, 2)$	$SLHD(343, 7, 7)$	$EBLHD(2401, 7 + 7)$	$7 \times 7 *, 7 \times 49, 49 \times 7$

Note: In the stratification column, the values with "$*$" indicate the stratification property of each slice of L. In the stratification column, the values without "$*$" represent the stratification property of L.

TABLE C2 Results from Algorithms 2 and 3.

$MOA(n_1, s^q(\frac{n_1}{s}), 2)$	$CROA(n_2, k, s, 2)$	$SLHD(n, k, \frac{n_1}{s})$	$EBLHD(N, q + k)$	Stratification
$MOA(8, 2^4 4^1)$	$CROA(8, 4, 2, 2)$	$SLHD(32, 4, 4)$	$EBLHD(64, 4 + 4)$	$2 \times 2 *, 2 \times 8, 8 \times 2$
$MOA(16, 2^8 8^1)$	$CROA(16, 8, 2, 2)$	$SLHD(128, 8, 8)$	$EBLHD(256, 8 + 8)$	$2 \times 2 *, 2 \times 16, 16 \times 2$
$MOA(18, 3^6 6^1)$	$CROA(18, 6, 3, 2)$	$SLHD(108, 6, 6)$	$EBLHD(324, 6 + 6)$	$3 \times 3 *, 3 \times 18, 18 \times 3$
$MOA(27, 3^9 9^1)$	$CROA(27, 9, 3, 2)$	$SLHD(243, 9, 9)$	$EBLHD(729, 9 + 9)$	$3 \times 3 *, 3 \times 27, 27 \times 3$
$MOA(32, 4^8 8^1)$	$CROA(32, 8, 4, 2)$	$SLHD(256, 8, 8)$	$EBLHD(1024, 8 + 8)$	$4 \times 4 *, 4 \times 32, 32 \times 4$
$MOA(32, 2^{16} 16^1)$	$CROA(32, 16, 2, 2)$	$SLHD(512, 16, 16)$	$EBLHD(1024, 16 + 16)$	$2 \times 2 *, 2 \times 32, 32 \times 2$

Note: In the stratification column, the values with "$*$" indicate the stratification property of each slice of L. In the stratification column, the values without "$*$" represent the stratification property of L.

TABLE C3 Results from Algorithm 4.

$OA(n,k,s,t)$	s_1	s_2	$EBLHD(n\cdot s_1s_2,p)$	Stratification
$OA(4,3,2,2)$	2	3	$EBLHD(24,5)$	2×2
	2	4	$EBLHD(32,5)$	2×2
	2	5	$EBLHD(40,5)$	2×2
	2	6	$EBLHD(48,6)$	2×2
	3	4	$EBLHD(48,5)$	2×2
	3	5	$EBLHD(60,5)$	2×2
	3	6	$EBLHD(72,5)$	2×2
	4	5	$EBLHD(80,5)$	2×2
	4	6	$EBLHD(96,5)$	2×2
	5	6	$EBLHD(120,5)$	2×2
$OA(8,7,2,2)$	2	3	$EBLHD(48,9)$	2×2
	2	4	$EBLHD(64,9)$	2×2
	2	5	$EBLHD(80,9)$	2×2
	2	6	$EBLHD(96,9)$	2×2
	3	4	$EBLHD(96,9)$	2×2
	3	5	$EBLHD(120,9)$	2×2
	3	6	$EBLHD(144,9)$	2×2
	4	5	$EBLHD(160,9)$	2×2
	4	6	$EBLHD(192,9)$	2×2
	5	6	$EBLHD(240,9)$	2×2
$OA(8,4,2,3)$	2	3	$EBLHD(48,6)$	$2\times2\times2$
	2	4	$EBLHD(64,6)$	$2\times2\times2$
	2	5	$EBLHD(80,6)$	$2\times2\times2$
	2	6	$EBLHD(96,6)$	$2\times2\times2$
	3	4	$EBLHD(96,6)$	$2\times2\times2$
	3	5	$EBLHD(120,6)$	$2\times2\times2$
	3	6	$EBLHD(144,6)$	$2\times2\times2$
	4	5	$EBLHD(160,6)$	$2\times2\times2$
	4	6	$EBLHD(192,6)$	$2\times2\times2$
	5	6	$EBLHD(240,6)$	$2\times2\times2$
$OA(12,11,2,2)$	2	3	$EBLHD(72,13)$	2×2
	2	4	$EBLHD(96,13)$	2×2
	2	5	$EBLHD(120,13)$	2×2
	2	6	$EBLHD(144,13)$	2×2
	3	4	$EBLHD(144,13)$	2×2

TABLE C3 (Continued)

$OA(n,k,s,t)$	s_1	s_2	$EBLHD(n \cdot s_1 s_2, p)$	Stratification
	3	5	$EBLHD(180, 13)$	2×2
	3	6	$EBLHD(216, 13)$	2×2
	4	5	$EBLHD(240, 13)$	2×2
	4	6	$EBLHD(288, 13)$	2×2
	5	6	$EBLHD(360, 13)$	2×2
$OA(16, 15, 2, 2)$	2	3	$EBLHD(96, 17)$	2×2
	2	4	$EBLHD(128, 17)$	2×2
	2	5	$EBLHD(160, 17)$	2×2
	2	6	$EBLHD(192, 17)$	2×2
	3	4	$EBLHD(192, 17)$	2×2
	3	5	$EBLHD(240, 17)$	2×2
	3	6	$EBLHD(288, 17)$	2×2
	4	5	$EBLHD(320, 17)$	2×2
	4	6	$EBLHD(384, 17)$	2×2
	5	6	$EBLHD(480, 17)$	2×2
$OA(16, 8, 2, 3)$	2	3	$EBLHD(96, 10)$	$2 \times 2 \times 2$
	2	4	$EBLHD(128, 10)$	$2 \times 2 \times 2$
	2	5	$EBLHD(160, 10)$	$2 \times 2 \times 2$
	2	6	$EBLHD(192, 10)$	$2 \times 2 \times 2$
	3	4	$EBLHD(192, 10)$	$2 \times 2 \times 2$
	3	5	$EBLHD(240, 10)$	$2 \times 2 \times 2$
	3	6	$EBLHD(288, 10)$	$2 \times 2 \times 2$
	4	5	$EBLHD(320, 10)$	$2 \times 2 \times 2$
	4	6	$EBLHD(384, 10)$	$2 \times 2 \times 2$
	5	6	$EBLHD(480, 10)$	$2 \times 2 \times 2$
$OA(20, 19, 2, 2)$	2	3	$EBLHD(120, 21)$	2×2
	2	4	$EBLHD(160, 21)$	2×2
	2	5	$EBLHD(200, 21)$	2×2
	2	6	$EBLHD(240, 21)$	2×2
	3	4	$EBLHD(240, 21)$	2×2
	3	5	$EBLHD(300, 21)$	2×2
	3	6	$EBLHD(360, 21)$	2×2
	4	5	$EBLHD(400, 21)$	2×2
	4	6	$EBLHD(480, 21)$	2×2
	5	6	$EBLHD(600, 21)$	2×2

TABLE C3 (Continued)

$OA(n,k,s,t)$	s_1	s_2	$EBLHD(n \cdot s_1 s_2, p)$	Stratification
$OA(24,12,2,3)$	2	3	$EBLHD(144,14)$	$2\times2\times2$
	2	4	$EBLHD(192,14)$	$2\times2\times2$
	2	5	$EBLHD(240,14)$	$2\times2\times2$
	2	6	$EBLHD(288,14)$	$2\times2\times2$
	3	4	$EBLHD(288,14)$	$2\times2\times2$
	3	5	$EBLHD(360,14)$	$2\times2\times2$
	3	6	$EBLHD(432,14)$	$2\times2\times2$
	4	5	$EBLHD(480,14)$	$2\times2\times2$
	4	6	$EBLHD(576,14)$	$2\times2\times2$
	5	6	$EBLHD(720,14)$	$2\times2\times2$
$OA(32,16,2,3)$	2	3	$EBLHD(192,18)$	$2\times2\times2$
	2	4	$EBLHD(256,18)$	$2\times2\times2$
	2	5	$EBLHD(320,18)$	$2\times2\times2$
	2	6	$EBLHD(384,18)$	$2\times2\times2$
	3	4	$EBLHD(384,18)$	$2\times2\times2$
	3	5	$EBLHD(480,18)$	$2\times2\times2$
	3	6	$EBLHD(576,18)$	$2\times2\times2$
	4	5	$EBLHD(640,18)$	$2\times2\times2$
	4	6	$EBLHD(768,18)$	$2\times2\times2$
	5	6	$EBLHD(960,18)$	$2\times2\times2$
$OA(40,20,2,3)$	2	3	$EBLHD(240,22)$	$2\times2\times2$
	2	4	$EBLHD(320,22)$	$2\times2\times2$
	2	5	$EBLHD(400,22)$	$2\times2\times2$
	2	6	$EBLHD(480,22)$	$2\times2\times2$
	3	4	$EBLHD(480,22)$	$2\times2\times2$
	3	5	$EBLHD(600,22)$	$2\times2\times2$
	3	6	$EBLHD(720,22)$	$2\times2\times2$
	4	5	$EBLHD(800,22)$	$2\times2\times2$
	4	6	$EBLHD(960,22)$	$2\times2\times2$
$OA(9,4,3,2)$	2	3	$EBLHD(54,6)$	3×3
	2	4	$EBLHD(72,6)$	3×3
	2	5	$EBLHD(90,6)$	3×3
	2	6	$EBLHD(108,6)$	3×3
	3	4	$EBLHD(108,6)$	3×3
	3	5	$EBLHD(135,6)$	3×3
	3	6	$EBLHD(162,6)$	3×3

TABLE C3 (Continued)

$OA(n, k, s, t)$	s_1	s_2	$EBLHD(n \cdot s_1 s_2, p)$	Stratification
	4	5	$EBLHD(180, 6)$	3×3
	4	6	$EBLHD(216, 6)$	3×3
	5	6	$EBLHD(270, 6)$	3×3
$OA(18, 7, 3, 2)$	2	3	$EBLHD(108, 9)$	3×3
	2	4	$EBLHD(144, 9)$	3×3
	2	5	$EBLHD(180, 9)$	3×3
	2	6	$EBLHD(216, 9)$	3×3
	3	4	$EBLHD(216, 9)$	3×3
	3	5	$EBLHD(270, 9)$	3×3
	3	6	$EBLHD(324, 9)$	3×3
	4	5	$EBLHD(360, 9)$	3×3
	4	6	$EBLHD(432, 9)$	3×3
	5	6	$EBLHD(540, 9)$	3×3
$OA(54, 5, 3, 3)$	2	3	$EBLHD(324, 7)$	$3 \times 3 \times 3$
	2	4	$EBLHD(432, 7)$	$3 \times 3 \times 3$
	2	5	$EBLHD(540, 7)$	$3 \times 3 \times 3$
	2	6	$EBLHD(648, 7)$	$3 \times 3 \times 3$
	3	4	$EBLHD(648, 7)$	$3 \times 3 \times 3$
	3	5	$EBLHD(810, 7)$	$3 \times 3 \times 3$
	3	6	$EBLHD(972, 7)$	$3 \times 3 \times 3$
$OA(16, 5, 4, 2)$	2	3	$EBLHD(96, 7)$	4×4
	2	4	$EBLHD(128, 7)$	4×4
	2	5	$EBLHD(160, 7)$	4×4
	2	6	$EBLHD(192, 7)$	4×4
	3	4	$EBLHD(192, 7)$	4×4
	3	5	$EBLHD(240, 7)$	4×4
	3	6	$EBLHD(288, 7)$	4×4
	4	5	$EBLHD(320, 7)$	4×4
	4	6	$EBLHD(384, 7)$	4×4
	5	6	$EBLHD(480, 7)$	4×4