A Construction Method for Uniform Projection Latin Hypercube Designs*

LIU Lifeng · SUN Fasheng

DOI: 10.1007/s11424-026-5409-z
Received: 25 September 2025 / Revised: 5 November 2025

Abstract Space-filling designs are widely used in computer experiments to build effective metamodels with limited prior information, as they enable thorough exploration of the design space by uniformly distributing points. However, many existing designs perform poorly in low-dimensional projections, particularly when only a few factors are active. Uniform projection designs address this limitation by optimizing point distribution across low-dimensional subspaces, ensuring uniformity in all dimensions while maintaining desirable distance and column-orthogonality properties. Existing methods for constructing such designs often rely on complex algorithms or can only generate designs with large factor-to-run ratios. In this work, the authors propose a simple approach for constructing uniform projection Latin hypercube designs by employing orthogonal arrays. The proposed method is particularly effective when the number of factors is much smaller than the number of runs. Both theoretical and numerical results demonstrate that the designs produced by the proposed method perform well with respect to the uniform projection, low-dimensional stratification, maximin distance, and column-orthogonality criteria.

Keywords Computer experiment, Latin hypercube designs, space-filling designs, uniform projection designs.

1 Introduction

With the rapid advancement of science and technology, traditional physical experiments are increasingly inadequate for studying complex systems, while computer experiments provide a precise and cost-effective alternative for simulating these systems[1–3]. Space-filling designs, which evenly distribute points across the design space, are commonly used in computer experiments for metamodeling. Commonly used space-filling designs include Latin hypercube designs (LHDs)[4], maximin distance designs[5, 6], uniform designs[7, 8], and strong orthogonal arrays[9].

LIU Lifeng · SUN Fasheng (Corresponding author)
KLAS and School of Mathematics and Statistics, Northeast Normal University, Changchun 130024*, China.*
Email: liulf019@nenu.edu.cn; sunfs359@nenu.edu.cn.
*This research was supported by the National Natural Science Foundation of China under Grant No. 12371259.
◇*This paper was recommended for publication by Editor HE Xu.*

LHDs are popular space-filling designs due to their ability to achieve maximum one-dimensional stratification. However, a randomly generated LHD may perform poorly in two and higher-dimensional projections. To address this issue, various optimal LHDs have been proposed by researchers, including column-orthogonal LHDs[10, 11], maximin distance LHDs[12, 13], and orthogonal array-based LHDs[9, 14].

Uniform designs, which distribute points uniformly across the space by minimizing a metric called discrepancy, are also widely used in physical and computer experiments for their robustness, particularly when the statistical model is unknown[15–17]. Several discrepancies have been proposed to measure the uniformity of a design, with the centered L_2-discrepancy being one of the most commonly used[17]. Uniform designs are considered robust because they prevent inaccurate estimation caused by model irregularities[17].

In many early-stage computer experiments the factor importance is uncertain, and it is common to include many factors in the design, although only a few of them have significant effects[18, 19]. In such cases, space-filling designs with good projection properties are ideal for factor screening. Moon, et al.[20] introduced a two-dimensional distance metric and developed algorithms for constructing designs with favorable projection properties. Similarly, Joseph, et al.[21] proposed another distance metric and constructed maximum projection designs.

In practice, the interaction between two factors is typically more important than that of three or more factors. Based on this effect hierarchy principle, Sun, et al.[22] introduced the uniform projection criterion by focusing on the average two-dimensional centered L_2-discrepancies of a design. They also showed that uniform projection designs ensure not only uniformity in two-dimensional projections but also evenly distributed points across all dimensions, exhibiting strong space-filling properties in terms of distance, uniformity, and column-orthogonality. Additionally, the relationship between the uniform projection criterion and the maximin L_1-distance was established in [22]. Wang, et al.[23] and Liu, et al.[24] further explored the connections between the uniform projection criterion and other space-filling criteria, providing insights into constructing designs that perform well under multiple criteria.

Existing methods for constructing uniform projection designs often involve computationally expensive algorithms or complex algebraic constructions that typically generate designs with many factors (factor-to-run ratios near one)[22–24]. Although maximin L_1-equidistant designs are also uniform projection designs as shown by [22], achieving projection uniformity becomes challenging when the factor-to-run ratio is small. To address these issues, in this paper we propose a method for constructing uniform projection LHDs based on the level expansion of specific orthogonal arrays. Our method is simple and ensures both maximum one-dimensional stratification and excellent space-filling properties in multi-dimensional projections. It is particularly effective when the number of factors is much smaller than the number of runs. Theoretical and numerical results show the excellent performance of the obtained LHDs under the uniform projection, low-dimensional stratification, maximin distance, and column-orthogonality criteria.

The remainder of this paper is organized as follows. Section 2 introduces the notation and definitions. Section 3 presents the construction method and the related theoretical results. Section 4 provides numerical comparisons and simulation results to demonstrate the performance of the proposed design. Section 5 concludes with a discussion of future directions. All proofs are provided in the Supplementary Material†.

2 Notation and Definitions

For a positive integer s, let $\mathbb{Z}_s$ denote the set $\{0, 1, \cdots, s-1\}$. A design consists of n runs and k factors can be represented by an $n \times k$ matrix. Let (n, s^k) denote an n-run design with k factors, each taking s levels from $\mathbb{Z}_s = \{0, 1, \cdots, s-1\}$. An (n, s^k) design is called an orthogonal array (OA) of strength t $(t \geq 1)$, denoted by $OA(n, k, s, t)$, if for its every $n \times t$ subarray, each t-tuple appears the same number of times. An $OA(n, k, n, 1)$ is called a Latin hypercube design (LHD) with n runs and k factors, and denoted by $LHD(n, k)$. By definition, each column of an $LHD(n, k)$ is a permutation of the n elements of $\mathbb{Z}_n = \{0, 1, \cdots, n-1\}$.

Let $D = (x_{ij})_{n\times k}$ be an $LHD(n, k)$. The L_p-distance between two rows $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ik})$ and $\boldsymbol{x}_l = (x_{l1}, \cdots, x_{lk})$ in D $(1 \leq i \neq l \leq n)$, is defined as $d_p(\boldsymbol{x}_i, \boldsymbol{x}_l) = \sum_{j=1}^{k} |x_{ij} - x_{lj}|^p$, where $p \geq 1$ and $j = 1, \cdots, k$. The L_p-distance of D is defined as

$$d_p(D) = \min\{d_p(\boldsymbol{x}_i, \boldsymbol{x}_l) : 1 \leq i \neq l \leq n, \boldsymbol{x}_i, \boldsymbol{x}_l \in D\}.$$

An $LHD(n, k)$ D is called a maximin L_p-distance LHD if it has the maximum L_p-distance among all possible $LHD(n, k)$s[5]. Zhou and Xu[25] established two upper bounds for the L_1- and L_2-distances, respectively, of an $LHD(n, k)$ D as follows:

$$d_1(D) \leq \left\lfloor \frac{(n+1)k}{3} \right\rfloor \quad \text{and} \quad d_2(D) \leq \left\lfloor \frac{n(n+1)k}{6} \right\rfloor.$$

Further, the equality holds if and only if D is an L_1- or L_2-equidistant LHD. However, when the factor-to-run ratio is small (e.g., $k/n < 0.5$), an L_1- or L_2-equidistant LHD usually does not exist[23]. In such cases, to measure the fluctuation of pairwise distances of runs in a design, we use the variation of all pairwise L_p-distances:

$$V_p(D) = \sum_{1\leq i<l\leq n} (d_p(\boldsymbol{x}_i, \boldsymbol{x}_l) - \overline{d}_p(D))^2,$$

where $\overline{d}_p(D)$ is the average pairwise L_p-distance of D, which is equal to $(n+1)k/3$ when $p = 1$ or $n(n+1)k/6$ when $p = 2$. A good space-filling design should minimize $V_p(D)$ to maximize the separation of design points in the space.

The column-orthogonality criterion aims to optimize designs by minimizing correlations between factors. A commonly used measure of column-orthogonality is the average value of correlation, with smaller values indicating better orthogonality[23]. For an $LHD(n, k)$ design D, the average value of correlation is defined as

$$\rho_{\text{ave}}(D) = \frac{1}{\binom{k}{2}} \sum_{1\leq j<j'\leq k} |\rho_{jj'}(D)|,$$

†The Supplementary Material of this paper is available at the CSTR and DOI addresses designated by ScienceDB (CSTR: 31253.11.sciencedb.j00207.00034, DOI: 10.57760/sciencedb.j00207.00034).

where $\rho_{jj'}(D)$ is the sample correlation between the jth and j'th columns of the design D.

For an $LHD(n,k)$ $D=(x_{ij})$, its squared centered L_2-discrepancy (CD) is defined as

$$CD(D)=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{l=1}^{n}\prod_{j=1}^{k}\left(1+\frac{1}{2}|z_{ij}|+\frac{1}{2}|z_{lj}|-\frac{1}{2}|z_{ij}-z_{lj}|\right)$$
$$-\frac{2}{n}\sum_{i=1}^{n}\prod_{j=1}^{k}\left(1+\frac{1}{2}|z_{ij}|-\frac{1}{2}|z_{ij}|^2\right)+\left(\frac{13}{12}\right)^k,$$

where $z_{ij}=(2x_{ij}-n+1)/(2n)$ (see, e.g., Fang, et al.[2]). The CD focuses on the uniformity over the whole dimensional design space. Based on the CD, Sun, et al.[22] introduced the uniform projection criterion, focusing on all two-dimensional projections of the design D:

$$\phi(D)=\frac{2}{k(k-1)}\sum_{|u|=2}CD(D_u), \tag{1}$$

where u is a subset of $\{1,2,\cdots,k\}$, $|u|$ denotes the cardinality of u, and D_u represents the projected design of D onto the dimensions indexed by the elements of u. An $LHD(n,k)$ that minimizes $\phi(D)$ among all $LHD(n,k)$s is called a uniform projection LHD.

To evaluate an LHD's performance under the uniform projection criterion, the following lower and upper bounds can be used, which were obtained by [22] and [23].

Lemma 2.1 *For an $LHD(n,k)$ D, we have $\phi_{LB}\le\phi(D)\le\phi_{UB}$, where $\phi_{LB}=\max\{\phi_{LB1},\phi_{LB2}\}$,*

$$\phi_{LB1}=\frac{5\left(4n^3+30n^2-4n-5\right)k-8n^4-150n^2+33}{720n^4(k-1)}+\frac{1+(-1)^n}{64n^4},$$
$$\phi_{LB2}=\frac{26n^2-1}{144n^4}+\frac{1+(-1)^n}{64n^4},$$

and

$$\phi_{UB}=\frac{(10k-8)n^4+(140k-150)n^2-25k+33}{720n^4(k-1)}+\frac{1+(-1)^n}{64n^4}.$$

Furthermore, the lower bound $\phi_{LB2}<\phi_{LB1}$ if and only if $k<(2n^2+7)/(5n+5)$.

Based on the above bounds, Wang, et al.[23] defined the ϕ-efficiency of a design D as

$$\phi_{\text{eff}}(D)=\frac{\phi_{UB}-\phi(D)}{\phi_{UB}-\phi_{LB}}\times 100\%. \tag{2}$$

The value of $\phi_{\text{eff}}(D)$ ranges from 0 to 1, with higher values indicating better projection uniformity. Specifically, $\phi_{\text{eff}}(D)=1$ if and only if $\phi(D)=\phi_{LB}$.

One popular method for generating LHDs is through orthogonal arrays or strong orthogonal arrays, which offer guaranteed stratification properties in low-dimensional projections[9, 14]. To evaluate a design's stratification properties on various low-dimensional grids, Tian and Xu[26] introduced the space-filling pattern and an associated minimum aberration type space-filling criterion. Let $D=(x_{ij})$ be an $(n,(s^p)^k)$ design with $x_{ij}\in\mathbb{Z}_{s^p}$, where p is a positive integer.

For $x \in \mathbb{Z}_{s^p}$, let $f_i(x) = \lfloor x/s^{p-i} \rfloor \bmod s$ and $r(x) = p + 1 - \min\{i : f_i(x) \neq 0, i = 1, \cdots, p\}$ if $x \neq 0$ and $r(0) = 0$. For $u, x \in \mathbb{Z}_{s^p}$, define character $\chi_u(x) = \xi^{\sum_{i=1}^{p} f_{p+1-i}(u) f_i(x)}$, where $\xi = \mathrm{e}^{2\pi \mathrm{i}/s}$ is the primitive sth root of unity and $\mathrm{i} = (-1)^{1/2}$. For $\boldsymbol{u} = (u_1, \cdots, u_k), \boldsymbol{x} = (x_1, \cdots, x_k) \in \mathbb{Z}_{s^p}^k$, let $\chi_{\boldsymbol{u}}(\boldsymbol{x}) = \prod_{i=1}^{k} \chi_{u_i}(x_i)$ and $r(\boldsymbol{u}) = \sum_{i=1}^{k} r(u_i)$. For the design D, define $\chi_{\boldsymbol{u}}(D) = \sum_{\boldsymbol{x} \in D} \chi_{\boldsymbol{u}}(\boldsymbol{x})$, where $\boldsymbol{x} \in D$ means $\boldsymbol{x}$ is a row of D and the summation $\sum_{\boldsymbol{x} \in D}$ is over all rows of D. For $j = 1, \cdots, kp$, let

$$S_j(D) = n^{-2} \sum_{r(\boldsymbol{u})=j} |\chi_{\boldsymbol{u}}(D)|^2 = n^{-2} \sum_{r(\boldsymbol{u})=j} \chi_{\boldsymbol{u}}(D) \overline{\chi_{\boldsymbol{u}}(D)}, \tag{3}$$

where the summation is over all $\boldsymbol{u} \in \mathbb{Z}_{s^p}^k$ with $r(\boldsymbol{u}) = j$ and $\overline{\chi_{\boldsymbol{u}}(D)}$ is the complex conjugate of $\chi_{\boldsymbol{u}}(D)$. The vector $(S_1(D), \cdots, S_{kp}(D))$ is called the space-filling pattern of D. The minimum aberration type space-filling criterion of Tian and Xu[26] is to sequentially minimize $S_j(D)$ for $j = 1, \cdots, kp$.

3 Construction Method and Theoretical Results

An effective approach to constructing space-filling LHDs is through the level expansion of orthogonal arrays. In this section, we introduce a straightforward method for generating LHDs based on orthogonal arrays. The resulting design not only maximizes the one-dimensional space-filling property but also exhibits good projection uniformity and stratification properties in two or higher dimensions. We will present the construction method and theoretical results for two different design sizes, $n = s^2$ and $n = s^3$, in separate subsections, where s is a prime number.

3.1 Construction and Theoretical for $n = s^2$

Bush[27] established that for any prime power $s \geq 2$, an orthogonal array $OA(s^t, s + 1, s, t)$ of index unity exists whenever $s \geq t - 1 \geq 0$. In particular, when $s \geq 3$ is an odd prime, an orthogonal array $OA(s^2, s + 1, s, 2)$ exists[28]. Let S denote the set of odd prime numbers s, that is, $S = \{3, 5, 7, 11, \cdots\}$. The position of each prime s in S is indexed by the parameter f_s. That is, $\{f_3, f_5, f_7, f_{11}, \cdots\} = \{1, 2, 3, 4, \cdots\}$. The well-known Rao-Hamming construction of an $OA(s^2, s + 1, s, 2)$ proceeds as follows. Take two independent columns $\boldsymbol{a}$ and $\boldsymbol{b}$ as the first two columns of the OA. The remaining $s - 1$ columns are generated as:

$$\{\boldsymbol{a} + \boldsymbol{b},\ \boldsymbol{a} + 2\boldsymbol{b},\ \cdots,\ \boldsymbol{a} + (s - 1)\boldsymbol{b}\} \pmod{s}.$$

Thus, all columns of the OA are linear combinations of $\boldsymbol{a}$ and $\boldsymbol{b}$.

Now we present the construction of the uniform projection $LHD(s^2, k)$ in Algorithm 1, where s is an odd prime, $k = 2, 3, 4$ for $s = 3$ and $k = 2, 3, 4, 5$ for $s \geq 5$.

The following properties hold for D_k.

Proposition 3.1 *The $s^2 \times k$ matrix D_k constructed by Algorithm 1 is an $LHD(s^2, k)$. Furthermore, for this design, we have $S_1(D_k) = S_2(D_k) = 0$, where $S_j(D_k)$ is the space-filling pattern of D_k as defined in* (3).

Algorithm 1 Construction of $LHD(s^2, k)$ with s an odd prime

Step 1 For an odd prime s, construct an $OA(s^2, s+1, s, 2)$ named A, where the $s+1$ columns are given by $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a}+\boldsymbol{b}, \boldsymbol{a}+2\boldsymbol{b}, \cdots, \boldsymbol{a}+(s-1)\boldsymbol{b}\} \pmod s$.

Step 2 Let $S = \{3, 5, 7, 11, \cdots\}$ denote the set of odd primes, and define f_s as the position of s in S.

Step 3 When $s = 3$, define $\boldsymbol{l}_1 = s\boldsymbol{a} + \boldsymbol{b}$ and $\boldsymbol{l}_2 = s[\boldsymbol{a} + f_s\boldsymbol{b} \pmod s)] + \boldsymbol{b})$. Output:
$D_2 = (\boldsymbol{l}_1, \boldsymbol{l}_2)$, $D_3 = (\boldsymbol{l}_1, \boldsymbol{l}_2, s[\boldsymbol{a}+2\boldsymbol{b} \pmod s)] + \boldsymbol{b})$ and $D_4 = (s\boldsymbol{a} + [\boldsymbol{a}+\boldsymbol{b} \pmod s)], s\boldsymbol{b}+\boldsymbol{a}$, $s[\boldsymbol{a}+\boldsymbol{b} \pmod s)] + [\boldsymbol{a}+2\boldsymbol{b} \pmod s)], s[\boldsymbol{a}+2\boldsymbol{b} \pmod s)] + \boldsymbol{a})$.
When $s \geq 5$, define the following:

$$\begin{cases} \boldsymbol{l}_1 = s\boldsymbol{a} + \boldsymbol{b}, \\ \boldsymbol{l}_2 = s[\boldsymbol{a} + f_s\boldsymbol{b} \pmod s)] + \boldsymbol{b}, \\ \boldsymbol{l}_3 = s\left[\boldsymbol{a} + \dfrac{s+3}{2}\boldsymbol{b} \pmod s)\right] + \boldsymbol{b}, \\ \boldsymbol{l}_4 = s[\boldsymbol{a} + (f_s - 1)\boldsymbol{b} \pmod s)] + \boldsymbol{b}, \\ \boldsymbol{l}_5 = s\left[\boldsymbol{a} + \dfrac{s+1}{2}\boldsymbol{b} \pmod s)\right] + \boldsymbol{b}. \end{cases}$$

Output $D_k = (\boldsymbol{l}_1, \cdots, \boldsymbol{l}_k)$ for $k = 2, 3, 4, 5$.

Proposition 3.1 implies that the LHD D_k can achieve stratifications on $s \times s$ grids in any two dimensions. This property inherits from the fact that D_k is a strength two OA-based LHD.

Let $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ be two distinct columns in the $OA(s^2, s+1, s, 2)$. Then $\boldsymbol{l} = s\boldsymbol{y}_1 + \boldsymbol{y}_2$ is a permutation of $\mathbb{Z}_{s^2} = \{0, 1, \cdots, s^2-1\}$. Therefore, given an $OA(s^2, s+1, s, 2)$, a total of $s(s+1)$ different column vectors $\boldsymbol{l}$ can be generated, yielding $\binom{s(s+1)}{k}$ $LHD(n, k)$s without repeated columns. Among them, the D_k constructed from Algorithm 1 not only has the stratification properties given in Proposition 3.1, but also performs extremely well under the uniform projection criterion. Below, we provide an example for $s = 3$. Additional numerical results and comparisons will be given in Section 4.

Example 3.2 Let $s = 3$, then $f_3 = 1$ and there exists an $OA(9, 4, 3, 2)$ whose four columns are obtained by $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a}+\boldsymbol{b}, \boldsymbol{a}+2\boldsymbol{b}\} \pmod s$. We can represent the OA as

$$\begin{matrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{a}+\boldsymbol{b} \\ \boldsymbol{a}+2\boldsymbol{b} \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 2 & 1 & 1 & 0 & 2 & 2 & 1 & 0 \end{pmatrix}^{\mathrm{T}}.$$

From Algorithm 1, $\boldsymbol{l}_1 = 3\boldsymbol{a} + \boldsymbol{b}$, $\boldsymbol{l}_2 = 3[\boldsymbol{a} + \boldsymbol{b} \pmod s)] + \boldsymbol{b}$, thus

$$D_2 = (\boldsymbol{l}_1, \boldsymbol{l}_2) = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 4 & 8 & 3 & 7 & 2 & 6 & 1 & 5 \end{pmatrix}^{\mathrm{T}}$$

is an $LHD(9, 2)$. Under the uniform projection criterion, we have $\phi(D_2) = CD(D_2) = 4.226 \times 10^{-3}$ (It should be noted that when a design D has only two columns, $\phi(D) = CD(D)$). By

Lemma 2.1, $\phi_{UB} = 1.889 \times 10^{-2}$. Since $k = 2 < (2 \times 9^2 + 7)/(5 \times 9 + 5) = 3.38$, the lower bound $\phi_{LB} = \phi_{LB2} = 2.228 \times 10^{-3}$. From Equation (2), the ϕ-efficiency $\phi_{eff}(D_2) = 88.00\%$.

Following Algorithm 1, we can also obtain an $LHD(9,3)$ D_3 and an $LHD(9,4)$ D_4. Their ϕ-efficiencies are $\phi_{\text{eff}}(D_3) = 87.00\%$ and $\phi_{\text{eff}}(D_4) = 89.64\%$.

For an $LHD(s^2, k)$, the original definition of the uniform projection criterion $\phi(D)$ in (1) requires $O(s^4k^2)$ computational complexity. Inspired by [22], we present Theorem 3.3, which reduces this computational complexity to $O(s^4k)$. This theorem provides an efficient method to compute $\phi(D)$ using pairwise L_1- and L_2-distances between rows of D. It also links the variation of all pairwise L_1-distances in an $LHD(s^2, k)$ to its $\phi(D)$.

Theorem 3.3 *Let $D = (x_{ik})$ be an $LHD(s^2, k)$ with s an odd prime. Then,*

$$\phi(D) = \frac{2V_1(D) - \frac{2}{s^2}\sum_{i=1}^{s^2} d_2^2(\boldsymbol{x}_i, s_0)}{4s^8k(k-1)} + C_1(s,k),$$

where $V_1(D)$ is the L_1-distance variance of D, $s_0 = (s^2-1)/2$, and

$$C_1(s,k) = \frac{5k\left(s^8 + 8s^6 + 58s^4 - 8s^2 - 9\right) - 16s^8 - 300s^4 + 66}{1440(k-1)s^8}$$

is a constant dependent only on s and k.

One limitation of Algorithm 1 is that the number of columns is fewer than 6. We now extend this construction by incorporating additional columns. Step 1 of Algorithm 1 constructs the orthogonal array by utilizing vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ as its independent columns, with the remaining $s-1$ columns generated through linear combinations of these basis vectors. By redefining the basis as $\boldsymbol{a}_1 = \boldsymbol{b}$ and $\boldsymbol{b}_1 = \boldsymbol{a} + \boldsymbol{b}$, we can derive a new orthogonal array A_1 following the same construction methodology:

$$\{\boldsymbol{a}_1,\, \boldsymbol{b}_1,\, \boldsymbol{a}_1 + \boldsymbol{b}_1,\, \boldsymbol{a}_1 + 2\boldsymbol{b}_1,\, \cdots,\, \boldsymbol{a}_1 + (s-1)\boldsymbol{b}_1\} \pmod{s}.$$

In fact, A_1 is structurally isomorphic to A, meaning they share the same combinatorial properties. Starting from A_1 in Algorithm 1 yields an $LHD(s^2, 5)$, $D_{5,1}$, that is ϕ_{eff}-equivalent to D_5. Let $D = (D_5, D_{5,1})$, then D is an $LHD(s^2, 10)$. Algorithm 2 constructs an $LHD(s^2, 5m)$, where $m = 1, \cdots, s$, and the numerical results and comparisons will be given in Section 4.

It should be clarified that Algorithm 2 is more accurately defined as an extension and application of Algorithm 1, specifically designed to systematically and repeatedly invoke Algorithm 1 for constructing larger-scale designs. Within this framework, Algorithm 1 serves as a core subroutine. Therefore, for the specific task of generating an LHD with $k \leq 5$ columns, directly calling this core subroutine (Algorithm 1) represents the most straightforward and efficient implementation approach.

Algorithm 2 Construction of $LHD(s^2, 5m)$ for $s \geq 5$

Step 1 For an odd prime $s \geq 5$, construct an $OA(s^2, s+1, s, 2)$ named A, where the $s+1$ columns are given by $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a}+\boldsymbol{b}, \boldsymbol{a}+2\boldsymbol{b}, \cdots, \boldsymbol{a}+(s-1)\boldsymbol{b}\} \pmod{s}$.

Step 2 Define $\boldsymbol{a}_1 = \boldsymbol{b}$ and $\boldsymbol{b}_1 = \boldsymbol{a}+\boldsymbol{b}$. The orthogonal array A_1 derived from $\boldsymbol{a}_1$ and $\boldsymbol{b}_1$ yields an $LHD(s^2, 5)$ $D_{5,1}$ when processed through Algorithm 1.

Step 3 Next, iteratively set:
$\{\boldsymbol{a}_2 = \boldsymbol{a}+\boldsymbol{b},\ \boldsymbol{b}_2 = \boldsymbol{a}+2\boldsymbol{b}\}, \cdots, \{\boldsymbol{a}_{s-1} = \boldsymbol{a}+(s-2)\boldsymbol{b},\ \boldsymbol{b}_{s-1} = \boldsymbol{a}+(s-1)\boldsymbol{b}\}$.
Then $A_2, \cdots, A_{s-1}$ can be constructed in Step 1 and $D_{5,2}, \cdots, D_{5,(s-1)}$ can be constructed in Step 2.

Step 4 Let $D = (D_5, D_{5,1}, \cdots, D_{5,s-1})$. By sequentially selecting m component designs from D, we can construct an $LHD(s^2, 5m)$, $m = 1, \cdots, s$.

3.2 Construction and Theoretical Results for $n = s^3$

In this subsection, we explore the construction of uniform projection LHDs for $n = s^3$ where s is an odd prime. According to [27], when s is an odd prime, an $OA(s^3, s+1, s, 3)$ exists. Let $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ be three independent columns, i.e., $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are the three columns of an $OA(s^3, 3, s, 3)$ whose elements are from $\mathbb{Z}_s$. By the Bush's construction[28], the $s+1$ columns of an $OA(s^3, s+1, s, 3)$ can be obtained by the linear combinations of $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$. We present a simplified construction method for an $OA(s^3, s+1, s, 3)$, as shown in Algorithm 3, which will be used later in the construction of uniform projection LHDs.

Algorithm 3 Construction of $OA(s^3, s+1, s, 3)$ for odd prime $s \geq 3$

Step 1 Let $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ be three independent columns of an $OA(s^3, s+1, s, 3)$ and define $\boldsymbol{l}_1 = \boldsymbol{a}$, $\boldsymbol{l}_2 = \boldsymbol{c}$.

Step 2 Let $\boldsymbol{l}_j = \boldsymbol{a} + (j-2)\boldsymbol{b} + (j-2)^2\boldsymbol{c} \pmod{s}$, where $j = 3, \cdots, s+1$.

Step 3 Combine the two columns from Step 1 and the $s-1$ columns from Step 2 to form an $s^3 \times (s+1)$ matrix A.

Lemma 3.4 *The $s^3 \times (s+1)$ matrix A obtained by Algorithm 3 is an $OA(s^3, s+1, s, 3)$.*

The $s+1$ columns of the $OA(s^3, s+1, s, 3)$ obtained by Algorithm 3 for $s = 3, 5, 7$ and 11 are shown in Table 1.

Table 1 The $s+1$ columns of the $OA(s^3, s+1, s, 3)$ obtained by Algorithm 3

s	columns (mod s)					
	1	2	3	4	5	6
3	$\boldsymbol{a}$	$\boldsymbol{c}$	$\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}$	$\boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c}$		
5	$\boldsymbol{a}$	$\boldsymbol{c}$	$\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}$	$\boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c}$	$\boldsymbol{a}+3\boldsymbol{b}+9\boldsymbol{c}$	$\boldsymbol{a}+4\boldsymbol{b}+16\boldsymbol{c}$
7	$\boldsymbol{a}$	$\boldsymbol{c}$	$\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}$	$\boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c}$	$\boldsymbol{a}+3\boldsymbol{b}+9\boldsymbol{c}$	$\boldsymbol{a}+4\boldsymbol{b}+16\boldsymbol{c}$
11	$\boldsymbol{a}$	$\boldsymbol{c}$	$\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}$	$\boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c}$	$\boldsymbol{a}+3\boldsymbol{b}+9\boldsymbol{c}$	$\boldsymbol{a}+4\boldsymbol{b}+16\boldsymbol{c}$
	7	8	9	10	11	12
3						
5						
7	$\boldsymbol{a}+5\boldsymbol{b}+25\boldsymbol{c}$	$\boldsymbol{a}+6\boldsymbol{b}+36\boldsymbol{c}$				
11	$\boldsymbol{a}+5\boldsymbol{b}+25\boldsymbol{c}$	$\boldsymbol{a}+6\boldsymbol{b}+36\boldsymbol{c}$	$\boldsymbol{a}+7\boldsymbol{b}+49\boldsymbol{c}$	$\boldsymbol{a}+8\boldsymbol{b}+64\boldsymbol{c}$	$\boldsymbol{a}+9\boldsymbol{b}+81\boldsymbol{c}$	$\boldsymbol{a}+10\boldsymbol{b}+100\boldsymbol{c}$

Based on Algorithm 3 and Lemma 3.4, we now propose the construction method of the uniform projection $LHD(s^3, k)$ in Algorithm 4, where s is an odd prime, $k = 2, 3, 4$ for $s = 3$ and $k = 2, 3, 4, 5$ for $s \geq 5$.

Algorithm 4 Construction of $LHD(s^3, k)$ with s an odd prime

Step 1 For an odd prime $s \geq 3$, obtain the $OA(s^3, s+1, s, 3)$ named A using Algorithm 3.

Step 2 Label the $s+1$ columns of A as $(\boldsymbol{a}, \boldsymbol{c}, [\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s], \cdots)$.

Step 3 When $s = 3$, define $\boldsymbol{l}_1 = s^2\boldsymbol{a} + s\boldsymbol{c} + \boldsymbol{y}$, $\boldsymbol{l}_2 = s^2[\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s] + s\boldsymbol{y} + \boldsymbol{c}$ and $\boldsymbol{l}_3 = s^2\boldsymbol{c} + s\boldsymbol{y} + [\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s]$, where $\boldsymbol{y} = \boldsymbol{a} + 2\boldsymbol{b} + 4\boldsymbol{c} \pmod s$. Output: $D_2 = (\boldsymbol{l}_1, \boldsymbol{l}_2)$, $D_3 = (\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3)$ and $D_4 = (s^2\boldsymbol{a} + s\boldsymbol{c} + \boldsymbol{y}, s^2[\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s] + s\boldsymbol{y} + \boldsymbol{c}, s^2\boldsymbol{c} + s\boldsymbol{y} + [\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s], s^2\boldsymbol{y} + s\boldsymbol{a} + \boldsymbol{b})$.

When $s \geq 5$, define the following:

$$\begin{cases} \boldsymbol{l}_1 = s^2\boldsymbol{a} + s\boldsymbol{c} + \boldsymbol{y}, \\ \boldsymbol{l}_2 = s^2[\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s] + s\boldsymbol{y} + \boldsymbol{c}, \\ \boldsymbol{l}_3 = s^2[\boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c} \pmod s] + s\boldsymbol{c} + [\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s], \\ \boldsymbol{l}_4 = s^2[\boldsymbol{a}+3\boldsymbol{b}+9\boldsymbol{c} \pmod s] + s\boldsymbol{c} + [\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod s], \\ \boldsymbol{l}_5 = s^2[\boldsymbol{a}+4\boldsymbol{b}+16\boldsymbol{c} \pmod s] + s\boldsymbol{c} + [\boldsymbol{a}+3\boldsymbol{b}+9\boldsymbol{c} \pmod s], \end{cases}$$

where $\boldsymbol{y} = \boldsymbol{a} + (s-1)\boldsymbol{b} + (s-1)^2\boldsymbol{c} \pmod s$.

Output $D_k = (\boldsymbol{l}_1, \cdots, \boldsymbol{l}_k)$ for $k = 2, 3, 4, 5$.

The following properties hold for D_k.

Proposition 3.5 *The $s^3 \times k$ matrix D_k constructed by Algorithm 4 is an $LHD(s^3, k)$. Furthermore, for this design, we have $S_1(D_k) = S_2(D_k) = S_3(D_k) = 0$, where $S_j(D_k)$ is the space-filling pattern of D_k as defined in* (3).

Proposition 3.5 implies that the D_k can achieve stratifications on $s^2 \times s$, $s \times s^2$ and $s \times s$ grids in any two dimensions, and $s \times s \times s$ in any three dimensions if $k \geq 3$. This property inherits from the fact that D_k is a strength three OA-based LHD.

The $LHD(s^3, k)$ D_k constructed by Algorithm 4 performs extremely well under the uniform projection criterion. Below, we provide an example for $s = 3$. Additional numerical results and comparisons will be given in Section 4.

Example 3.6 When $s = 3$, Algorithm 3 generates an $OA(27, 4, 3, 3)$ A. Its four columns are $\{\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}, \boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c}\} \pmod 3$, as shown below:

$$\begin{matrix} \boldsymbol{a} \\ \boldsymbol{c} \\ \boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \\ \boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c} \end{matrix} \begin{pmatrix} 0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&2&2&2&2&2&2&2&2&2 \\ 0&1&2&0&1&2&0&1&2&0&1&2&0&1&2&0&1&2&0&1&2&0&1&2&0&1&2 \\ 0&1&2&1&2&0&2&0&1&1&2&0&2&0&1&0&1&2&2&0&1&0&1&2&1&2&0 \\ 0&1&2&2&0&1&1&2&0&1&2&0&0&1&2&2&0&1&2&0&1&1&2&0&0&1&2 \end{pmatrix}^{\mathrm{T}}.$$

By Algorithm 4, let $\boldsymbol{y} = \boldsymbol{a} + 2\boldsymbol{b} + 4\boldsymbol{c} \pmod 3$ and

$$\begin{cases} \boldsymbol{l}_1 = 9\boldsymbol{a} + 3\boldsymbol{c} + \boldsymbol{y}, \\ \boldsymbol{l}_2 = 9[\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c} \pmod 3] + 3\boldsymbol{y} + \boldsymbol{c}, \end{cases}$$

then an $LHD(27,2)$

$$D_2=\begin{pmatrix}0&4&8&2&3&7&1&5&6&10&14&15&9&13&17&11&12&16&20&21&25&19&23&24&18&22&26\\0&13&26&15&19&5&21&7&11&12&25&2&18&4&17&6&10&23&24&1&14&3&16&20&9&22&8\end{pmatrix}^{\mathrm{T}}$$

can be obtained, and its ϕ-efficiency is $\phi_{\text{eff}}(D_2)=97.17\%$.

Following Algorithm 4, we can also obtain an $LHD(27,3)$ D_3 and an $LHD(27,4)$ D_4. Their ϕ-efficiencies are $\phi_{\text{eff}}(D_3)=94.05\%$ and $\phi_{\text{eff}}(D_4)=91.80\%$.

For an $LHD(s^3,k)$, the original definition of the uniform projection criterion $\phi(D)$ in (1) requires $O(s^6k^2)$ computational complexity. Theorem 3.7 reduces this complexity to $O(s^6k)$ and establishes a link between the variation of all pairwise L_1-distances of D and $\phi(D)$.

Theorem 3.7 *Let $D=(x_{ik})$ be an $LHD(s^3,k)$ with s an odd prime. Then,*

$$\phi(D)=\frac{2V_1(D)-\frac{2}{s^3}\sum_{i=1}^{s^3}d_2^2(\boldsymbol{x}_i,s_0)}{4s^{12}k(k-1)}+C_1(s,k),$$

where $V_1(D)$ is the L_1-distance variance of D, $s_0=(s^3-1)/2$, and

$$C_1(s,k)=\frac{5k\left(s^{12}+8s^9+58s^6-8s^3-9\right)-16s^{12}-300s^6+66}{1440(k-1)s^{12}}$$

is a constant dependent only on s and k.

Building upon the construction in Algorithm 4, we further develop a method to generate $LHD(s^3,5m)$ which can accommodate more factors. Notably, the LHDs obtained through Algorithm 5 also achieve remarkable performance under the uniform projection criterion, with numerical results reported in Section 4. Similarly, Algorithm 5 serves as an extension and application of Algorithm 4, where Algorithm 4 functions as the core computational subroutine. In practical application, the choice of algorithm should be determined by the required number of columns, k: Algorithm 4 should be used to construct the $LHD(s^3,k)$ when $k\le 5$, and Algorithm 5 should be employed when $k>5$.

Algorithm 5 Construction of $LHD(s^3,5m)$ for $s\ge 5$

Step 1 For an odd prime $s\ge 5$, construct an $OA(s^3,s+1,s,3)$ named A, where the $s+1$ columns are given by $(\boldsymbol{a},\boldsymbol{c},\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c},\cdots)\pmod{s}$.

Step 2 Define $\boldsymbol{a}_1=\boldsymbol{c}$, $\boldsymbol{b}_1=\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}\pmod{s}$ and $\boldsymbol{c}_1=\boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c}\pmod{s}$. The orthogonal array A_1 derived from $\boldsymbol{a}_1$, $\boldsymbol{b}_1$ and $\boldsymbol{c}_1$ yields an $LHD(s^3,5)$ $D_{5,1}$ when processed through Algorithm 3.

Step 3 Next, iteratively set:
$\{\boldsymbol{a}_2=\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c},\ \boldsymbol{b}_2=\boldsymbol{a}+2\boldsymbol{b}+4\boldsymbol{c},\boldsymbol{c}_2=\boldsymbol{a}+3\boldsymbol{b}+9\boldsymbol{c}\},\cdots,\{\boldsymbol{a}_{s-2}=\boldsymbol{a}+(s-3)\boldsymbol{b}+(s-3)^2\boldsymbol{c},\ \boldsymbol{b}_{s-2}=\boldsymbol{a}+(s-2)\boldsymbol{b}+(s-2)^2\boldsymbol{c},\ \boldsymbol{c}_{s-2}=\boldsymbol{a}+(s-1)\boldsymbol{b}+(s-1)^2\boldsymbol{c}\}$.
Then $A_2,\cdots,A_{s-2}$ can be constructed in the Step 1 and $D_{5,2},\cdots,D_{5,(s-1)}$ can be constructed in Step 2.

Step 4 Let $D=(D_5,D_{5,1},\cdots,D_{5,s-2})$. By sequentially selecting m component designs from D, we can construct an $LHD(s^3,5m)$, $m=1,\cdots,s-1$.

4 Numerical Comparison and Simulation

In this section, we compare the $LHD(s^2, k)$ and $LHD(s^3, k)$ designs (referred to as UPLHDs) constructed by Algorithms 1 and 4 with LHDs generated by popular existing methods.Evaluation metrics include $\phi_{\rm eff}$-values (the larger the better), average value of correlation $\rho_{\rm ave}$ (the smaller the better) and the L_1-distances d_1 (the larger the better), compared against: (i) RLHD: Randomly generated LHDs, with each size run 1000 times, recording both the average and maximum $\phi_{\rm eff}$-values. The $\rho_{\rm ave}$ and d_1 values are computed for the design achieving the maximum $\phi_{\rm eff}$-value among the 1000 runs. (ii) OBLHD: Orthogonal array-based LHDs from [14], with each size run 1000 times (as expanding the levels from an OA to LHD is random), recording both the average and maximum $\phi_{\rm eff}$ values. The $\rho_{\rm ave}$ and d_1 values are computed for the design achieving the maximum $\phi_{\rm eff}$-value among the 1000 runs. (iii) SLHD: Maximin distance LHDs generated using SLHD Packages[29], with each size run 1000 times (with default setting of the maximinSLHD function), recording both the average and maximum $\phi_{\rm eff}$-values. The $\rho_{\rm ave}$ and d_1 values are computed for the design achieving the maximum $\phi_{\rm eff}$-value among the 1000 runs.

The comparison results under the metrics $\phi_{\rm eff}$, $\rho_{\rm ave}$, and d_1 are shown in Tables 2, 3, and 4, respectively. For small s, UPLHDs exhibit slightly lower $\phi_{\rm eff}$-values compared to RLHDs and OBLHDs, likely due to the limited number of columns in the initial OA. However, for larger s ($s \geq 7$), UPLHDs outperform both RLHDs and OBLHDs, consistently surpassing SLHDs in projection performance and exceeding the average $\phi_{\rm eff}$-values of other methods (The bolded parts represent the designs with a highest $\phi_{\rm eff}$-values). Furthermore, UPLHDs demonstrate superior performance under the column-orthogonality and maximin L_1-distance criteria. As s increases, constructing optimal LHDs under the uniform projection criterion becomes more challenging, but the proposed Algorithms 1 and 4 provide a simple and effective method for generating asymptotically optimal uniform projection LHDs with small factor-to-run ratios.

Table 2 The $\phi_{\rm eff}$-values ($\times 100\%$) for the LHDs with $n = s^2$ and $n = s^3$ by the four methods

s	$RLHD(s^2, k)$							
	$k = 2$		$k = 3$		$k = 4$		$k = 5$	
	Max	Ave	Max	Ave	Max	Ave	Max	Ave
3	90.96	79.35	88.17	78.17	87.62	81.71		
5	97.53	92.70	96.01	92.69	95.01	92.37	94.32	92.20
7	98.75	96.61	98.44	96.52	97.69	96.07	97.30	96.04
11	99.56	98.64	99.32	98.52	99.15	98.48	98.85	98.46
13	99.72	99.01	99.51	98.90	99.30	98.90	99.26	98.89
17	99.81	99.45	99.72	99.38	99.63	99.32	99.57	99.32
19	99.87	99.56	99.77	99.49	99.72	99.49	99.60	99.46
s	$OBLHD(s^2, k)$							
	$k = 2$		$k = 3$		$k = 4$		$k = 5$	
	Max	Ave	Max	Ave	Max	Ave	Max	Ave9
3	**91.57**	87.15	**88.94**	85.82	**92.18**	85.62		
5	**98.11**	97.21	**97.60**	96.90	**97.56**	96.83	97.17	96.76
7	99.26	99.00	99.10	98.89	99.06	98.86	98.96	98.83

Table 2 (Continued) The ϕ_{eff}-values ($\times 100\%$) for the LHDs with $n = s^2$ and $n = s^3$ by the four methods

s	$OBLHD(s^2,k)$							
	$k=2$		$k=3$		$k=4$		$k=5$	
	Max	Ave	Max	Ave	Max	Ave	Max	Ave9
11	99.80	99.74	99.77	99.72	99.75	99.71	99.73	99.70
13	99.88	99.84	99.86	99.83	99.84	99.82	99.83	99.82
17	99.95	99.93	99.93	99.92	99.93	99.92	99.92	99.92
19	99.96	99.95	99.95	99.95	99.94	99.94	99.94	99.94
s	$SLHD(s^2,k)$							
	$k=2$		$k=3$		$k=4$		$k=5$	
	Max	Ave	Max	Ave	Max	Ave	Max	Ave
3	87.87	84.06	87.72	85.12	87.13	82.32		
5	97.81	95.39	95.90	93.94	95.22	93.47	94.76	92.75
7	98.87	97.76	98.02	96.75	97.53	96.49	97.57	96.31
11	99.67	99.21	99.32	98.91	99.06	98.68	98.98	98.55
13	99.72	99.39	99.54	99.21	99.36	99.05	99.33	98.97
17	99.82	99.57	99.73	99.51	99.63	99.46	99.60	99.40
19	99.83	99.62	99.74	99.59	99.71	99.54	99.65	99.51
s	$UPLHD(s^2,k)$							
	$k=2$		$k=3$		$k=4$		$k=5$	
3	88.00		87.00		89.64			
5	98.04		97.53		97.28		**97.24**	
7	**99.36**		**99.30**		**99.16**		**99.06**	
11	**99.87**		**99.86**		**99.80**		**99.77**	
13	**99.94**		**99.93**		**99.87**		**99.85**	
17	**99.97**		**99.97**		**99.94**		**99.93**	
19	**99.98**		**99.98**		**99.95**		**99.95**	
s	$RLHD(s^3,k)$							
	$k=2$		$k=3$		$k=4$		$k=5$	
	Max	Ave	Max	Ave	Max	Ave	Max	Ave
3	97.62	94.08	96.34	93.26	95.97	92.75		
5	99.43	98.64	99.19	98.58	98.98	98.53	98.99	98.40
7	99.80	99.54	99.71	99.48	99.67	99.48	99.61	99.44
11	99.95	99.88	99.92	99.87	99.91	99.87	99.89	99.86
13	99.96	99.93	99.96	99.92	99.94	99.92	99.94	99.92
17	99.99	99.96	99.98	99.96	99.97	99.96	99.97	99.96
19	99.99	99.99	99.98	99.99	99.98	99.97	99.98	99.97
s	$OBLHD(s^3,k)$							
	$k=2$		$k=3$		$k=4$		$k=5$	
	Max	Ave	Max	Ave	Max	Ave	Max	Ave
3	97.66	96.17	**97.06**	95.98	**96.76**	95.76		
5	99.70	99.52	99.61	99.46	99.55	99.45	**99.51**	99.43
7	99.91	99.87	99.89	99.86	99.87	99.85	99.86	99.85
11	99.98	99.98	99.98	99.98	99.97	99.98	99.97	99.97
13	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
17	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
19	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99

Table 2 (Continued) The ϕ_{eff}-values ($\times 100\%$) for the LHDs with $n = s^2$ and $n = s^3$ by the four methods

s	$SLHD(s^3, k)$							
	$k = 2$		$k = 3$		$k = 4$		$k = 5$	
	Max	Ave	Max	Ave	Max	Ave	Max	Ave
3	**98.16**	95.14	96.76	94.59	96.02	93.73		
5	99.69	99.20	99.33	98.97	99.20	98.80	99.03	98.63
7	99.86	99.61	99.78	99.59	99.68	99.55	99.63	99.52
11	99.93	99.86	99.91	99.86	99.89	99.85	99.89	99.86
13	99.96	99.92	99.93	99.90	99.93	99.91	99.93	99.91
17	99.98	99.97	99.98	99.96	99.97	99.96	99.97	99.96
19	99.98	99.97	99.98	99.97	99.98	99.97	99.98	99.97
s	$UPLHD(s^3, k)$							
	$k = 2$		$k = 3$		$k = 4$		$k = 5$	
3	97.17		94.05		91.80			
5	**99.89**		**99.79**		**99.77**		99.43	
7	**99.96**		**99.95**		**99.94**		**99.94**	
11	**99.99**		**99.99**		**99.99**		**99.99**	
13	**99.99**		**99.99**		**99.99**		**99.99**	
17	**99.99**		**99.99**		**99.99**		**99.99**	
19	**99.99**		**99.99**		**99.99**		**99.99**	

Table 3 The average value of correlation ρ_{ave} for the LHDs with $n = s^2$ and $n = s^3$ by the four methods

s	$RLHD(s^2, k)$				$OBLHD(s^2, k)$			
	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
3	**0.0167**	**0.0780**	0.0694		0.0333	0.1277	0.0667	
5	**0.0131**	0.0482	0.0751	0.0922	0.0269	**0.0323**	0.0532	**0.0286**
7	0.0117	0.0575	0.0662	0.0566	**0.0022**	0.0261	**0.0184**	**0.0147**
11	0.0264	0.0172	0.0416	0.0327	0.0046	**0.0051**	**0.0078**	**0.0079**
13	0.0085	0.0141	0.0325	0.0448	0.0049	**0.0036**	0.0073	0.0065
17	0.0082	0.0151	0.0195	0.0325	0.0043	0.0045	**0.0028**	0.0040
19	0.0114	0.0126	0.0229	0.0234	0.0032	0.0035	0.0032	**0.0024**
s	$SLHD(s^2, k)$				$UPLHD(s^2, k)$			
	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
3	0.1333	0.0888	**0.0666**		0.1000	0.1000	0.2167	
5	0.0192	0.0461	0.0435	0.1060	0.0385	0.0385	**0.0385**	0.0385
7	0.0063	0.0398	0.0441	0.0484	0.0200	**0.0200**	0.0200	0.0200
11	**0.0001**	0.0310	0.0302	0.0381	0.0082	0.0082	0.0082	0.0082
13	**0.0025**	0.0251	0.0184	0.0362	0.0059	0.0059	**0.0059**	**0.0059**
17	**0.0002**	0.0152	0.0223	0.0218	0.0034	**0.0034**	0.0034	**0.0034**
19	**0.0008**	0.0304	0.0217	0.0218	0.0028	**0.0028**	**0.0028**	0.0028
s	$RLHD(s^3, k)$				$OBLHD(s^3, k)$			
	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
3	**0.0128**	**0.0344**	0.0952		0.0470	0.0614	**0.0630**	
5	0.0243	0.0327	0.0457	0.0462	0.0101	**0.0155**	**0.0152**	**0.0138**
7	0.0375	0.0266	0.0365	0.0296	**0.0036**	**0.0121**	**0.0080**	**0.0084**

Table 3 (Continued) The average value of correlation ρ_{ave} for the LHDs with $n = s^2$ and $n = s^3$ by the four methods

s	$RLHD(s^3,k)$				$OBLHD(s^3,k)$			
	$k=2$	$k=3$	$k=4$	$k=5$	$k=2$	$k=3$	$k=4$	$k=5$
11	0.0170	0.0125	0.0129	0.0149	0.0083	**0.0046**	**0.0047**	**0.0029**
13	0.0026	0.0084	0.0103	0.0148	**0.0005**	**0.0020**	**0.0020**	**0.0013**
17	0.0017	0.0049	0.0036	0.0051	0.0007	**0.0010**	**0.0006**	**0.0009**
19	0.0017	0.0015	0.0072	0.0058	0.0010	**0.0007**	**0.0011**	**0.0006**
s	$SLHD(s^3,k)$				$UPLHD(s^3,k)$			
	$k=2$	$k=3$	$k=4$	$k=5$	$k=2$	$k=3$	$k=4$	$k=5$
3	0.0323	0.0421	0.0762		0.0659	0.2308	0.2985	
5	**0.0083**	0.0374	0.0384	0.0295	0.0154	0.0332	0.0373	0.0616
7	0.0084	0.0141	0.0226	0.0269	0.0057	0.0162	0.0186	0.0195
11	0.0078	0.0107	0.0134	0.0149	**0.0015**	0.0062	0.0073	0.0078
13	0.0096	0.0143	0.0082	0.0099	0.0009	0.0044	0.0052	0.0055
17	0.0066	0.0073	0.0053	0.0097	**0.0004**	0.0025	0.0030	0.0032
19	0.0048	0.0039	0.0058	0.0072	**0.0001**	0.0019	0.0024	0.0026

Table 4 The L_1-distances for the LHDs with $n = s^2$ and $n = s^3$ by the four methods

s	$RLHD(s^2,k)$				$OBLHD(s^2,k)$			
	$k=2$	$k=3$	$k=4$	$k=5$	$k=2$	$k=3$	$k=4$	$k=5$
3	2	5	7		3	6	9	
5	3	6	8	15	2	5	10	23
7	3	7	8	19	3	6	14	23
11	2	10	19	32	3	10	18	26
13	2	11	23	29	3	11	15	34
17	2	7	16	42	3	11	27	43
19	3	6	21	63	4	11	16	44
s	$SLHD(s^2,k)$				$UPLHD(s^2,k)$			
	$k=2$	$k=3$	$k=4$	$k=5$	$k=2$	$k=3$	$k=4$	$k=5$
3	3	7	9		**3**	**7**	**8**	
5	5	7	13	19	**5**	**12**	**20**	**25**
7	5	10	18	28	**7**	**17**	**28**	**35**
11	8	17	25	49	**11**	**25**	**44**	**55**
13	7	24	34	60	**13**	**31**	**52**	**65**
17	7	27	47	90	**17**	**51**	**68**	**85**
19	6	26	54	93	**19**	**57**	**76**	**95**
s	$RLHD(s^3,k)$				$OBLHD(s^3,k)$			
	$k=2$	$k=3$	$k=4$	$k=5$	$k=2$	$k=3$	$k=4$	$k=5$
3	2	6	11		3	7	11	
5	2	9	11		2	14	20	51
7	2	11	27	55	2	16	35	72
11	3	9	47	64	3	30	64	160
13	2	22	51	138	2	29	65	201
17	2	18	97	220	2	43	121	266
19	3	20	63	273	3	39	146	299
s	$SLHD(s^3,k)$				$UPLHD(s^3,k)$			
	$k=2$	$k=3$	$k=4$	$k=5$	$k=2$	$k=3$	$k=4$	$k=5$
3	5	8	13		**5**	**9**	**19**	
5	**6**	18	29	43	2	**28**	**37**	**89**
7	6	28	53	89	**9**	**38**	**98**	**146**
11	4	36	83	148	**13**	**60**	**218**	**488**
13	2	35	52	162	**15**	**70**	**296**	**680**
17	2	35	89	229	**19**	**90**	**488**	**1160**
19	2	22	114	254	**40**	**100**	**602**	**1448**

Additionally, we present a numerical comparison between the LHDs generated by Algorithms 2 and 5 and the OBLHDs with better performance mentioned earlier. For comparability, we select the first five columns from the OAs $A, A_1, \cdots, A_m$ obtained through Algorithms 2 and 5, denoted as $A^{5m} = (A^5, A_1^5, \cdots, A_{m-1}^5)$, then use Tang's method[14] to generate OBLHDs. We performed 100 runs per design size, identified the run yielding the maximum ϕ_{eff}-value, and then evaluated its average value of correlation ρ_{ave} and the L_1-distances d_1. As shown in Tables A.1, A.2, and A.3 of Supplementary Material (excluded from the main text due to length considerations and similar trends), the LHDs generated by Algorithms 2 and 5 yield LHDs with superior uniform projection properties, smaller ρ_{ave}, and larger d_1 than the alternative method.

Furthermore, both the $LHD(s^2, k)$ and $LHD(s^3, k)$ designs, constructed by sequentially selecting k columns from $LHD(s^2, 5m)$ and $LHD(s^3, 5m)$, respectively, under the condition that $5(m_1 - 1) < k < 5m_1$ for $m_1 = 2, \cdots, m$, consistently exhibit higher ϕ_{eff}-values than OBLHDs of the same dimensions. The corresponding numerical comparisons are omitted here.

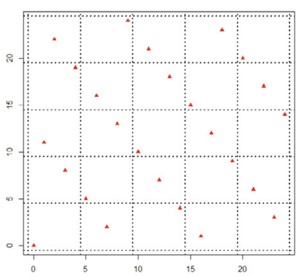
The following example further shows the stratification properties of the constructed LHDs.

Example 4.1 Consider three 25×3 LHDs in Table 5. Here $D^1 = (\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3)$ is the $LHD(25, 3)$ constructed by Algorithm 1, while D^2 and D^3 are two variants of D^1 with $\boldsymbol{l}_3$ replaced by another $5\boldsymbol{y}_1 + \boldsymbol{y}_2$, where $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are randomly selected columns from the Rao-Hamming $OA(25, 6, 5, 2)$. Figure 1 illustrates the bivariate projections of the three designs, showing that D^2 exhibits non-uniform projections, particularly with empty grids in $D^2[2, 3]$ ($D^i[j, k]$ is defined as the projection of design D^i onto columns j and k). In contrast, D^1 and D^3 demonstrate stratification properties across all 5×5 cells. The ϕ_{eff}-values are $\phi_{\text{eff}}(D^1) = 97.53\%$, $\phi_{\text{eff}}(D^2) = 75.78\%$, and $\phi_{\text{eff}}(D^3) = 97.12\%$, thus D^1 is better than D^3 under the uniform projection criterion. This confirms the effectiveness of Algorithm 1 in generating designs with good uniform projection and stratification properties.

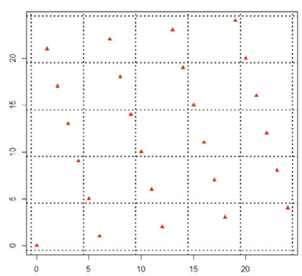
Table 5 Three $LHD(25, 3)s$ in Example 4.1

D^1			D^2			D^3		
0	0	0	0	0	0	0	0	0
1	11	21	1	11	14	1	11	19
2	22	17	2	22	23	2	22	8
3	8	13	3	8	7	3	8	22
4	19	9	4	19	16	4	19	11
5	5	5	5	5	6	5	5	6
6	16	1	6	16	15	6	16	20
7	2	22	7	2	4	7	2	14
8	13	18	8	13	13	8	13	3
9	24	14	9	24	22	9	24	17
10	10	10	10	10	12	10	10	12
11	21	6	11	21	21	11	21	1
12	7	2	12	7	5	12	7	15
13	18	23	13	18	19	13	18	9
14	4	19	14	4	3	14	4	23
15	15	15	15	15	18	15	15	18
16	1	11	16	1	2	16	1	7
17	12	7	17	12	11	17	12	21
18	23	3	18	23	20	18	23	10
19	9	24	19	9	9	19	9	4
20	20	20	20	20	24	20	20	24
21	6	16	21	6	8	21	6	13
22	17	12	22	17	17	22	17	2
23	3	8	23	3	1	23	3	16
24	14	4	24	14	10	24	14	5

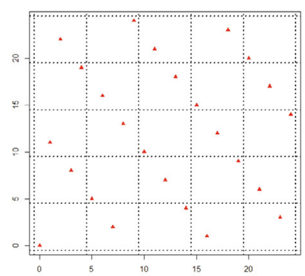
The $LHD(s^2, k)$ generated by Algorithm 1 satisfies $S_1(D_k) = S_2(D_k) = 0$, and its two-dimensional stratification property is visualized in Figure 1 via the distribution of D^1 on the 5×5 cells. In contrast, the $LHD(s^3, k)$ from Algorithm 4 achieves a further improvement with $S_1(D_k) = S_2(D_k) = S_3(D_k) = 0$. Figure 2 displays the distribution of the $LHD(27, 2)$ from Example 3.6 across 3×9 and 9×3 cells, where the design points occupy all cells in both projections. Together, Figures 1 and 2 confirm the excellent low-dimensional stratification property of the proposed design.



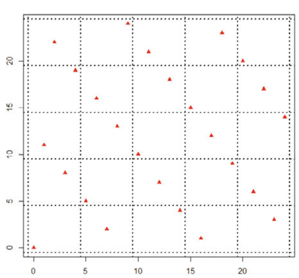
(a) $D^1[1, 2]$



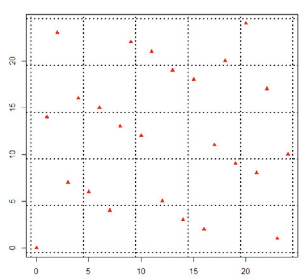
(b) $D^1[1, 3]$



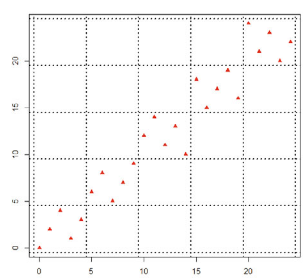
(c) $D^1[2, 3]$



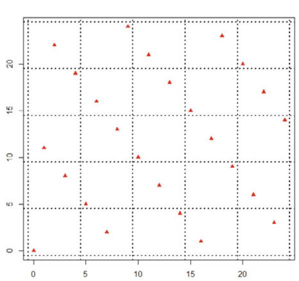
(d) $D^2[1, 2]$



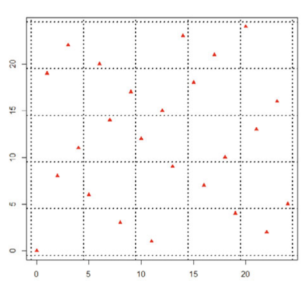
(e) $D^2[1, 3]$



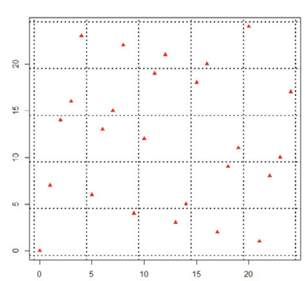
(f) $D^2[2, 3]$



(g) $D^3[1, 2]$

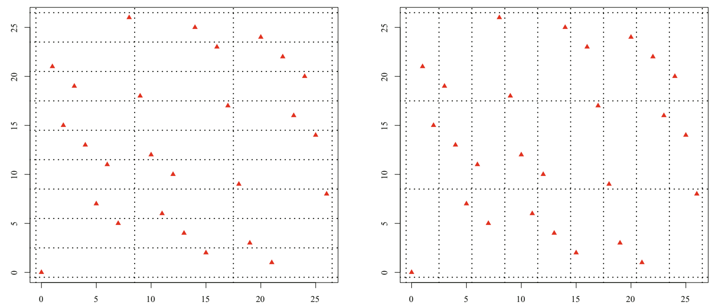


(h) $D^3[1, 3]$



(i) $D^3[2, 3]$

Figure 1 Bivariate projection plots of three $LHD(25, 3)$s in Example 4.1



(a) 3×9 cells (b) 9×3 cells

Figure 2 Bivariate projection plots of $LHD(27, 2)$ in Example 3.6

The constructed design is further validated through Gaussian process modeling (also referred to as Kriging) in the following example.

Example 4.2 Assume that the true response of the computer simulation code is $f(\boldsymbol{x})$, where $\boldsymbol{x} = (x_1, \cdots, x_k)$. It is common to fit a Kriging model with linear trends

$$\widehat{f}(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{k} \beta_j \boldsymbol{x}_j + Z(\boldsymbol{x}),$$

as a surrogate model (metamodel) to approximate $f(\boldsymbol{x})$. Here, $\beta_0, \beta_1, \cdots, \beta_k$ are some unknown constants, and $Z(\boldsymbol{x})$ is a stationary Gaussian process with mean 0 and covariance function $\sigma^2 R(\cdot)$. A common choice for the correlation function $R(\cdot)$ is the Gaussian kernel function $K(h_l; \theta_l) = \exp\left(-(\theta_l h_l)^2/2\right)$.

We modify the four-dimensional function of Ai, et al.[30] as $f(\boldsymbol{x})$:

$$\begin{aligned} f(\boldsymbol{x}) =& 3\sin(2\pi x_1 - \pi) + 2(x_2 - 0.5) - 5(x_3 - 0.5) + 2(x_4 - 0.5) \\ &+ 2(x_2 - 0.5)\sin(2\pi x_1 - \pi) + 3(x_1 - 0.5)(x_2 - 0.5)\sin(2\pi x_1 - \pi) \\ &- 2(x_1 - 0.5)(x_3 - 0.5) - 1.5(x_2 - 0.5)(x_3 - 0.5)\sin(2\pi x_3 - \pi) \\ &+ 2(x_1 - 0.5)(x_2 - 0.5)(x_4 - 0.5) + 10, \end{aligned}$$

where the range of the independent variables is $\boldsymbol{x} = (x_1, x_2, x_3, x_4) \in [0, 1]^4$. We select the constructed $UPLHD(s^2, 4)$ as the experimental design and compare it with three other designs: an $RLHD(s^2, 4)$, an $OBLHD(s^2, 4)$ and an $SLHD(s^2, 4)$ (here we take $s = 5, 7$). It should be noted that the three $LHD(s^2, 4)s$ used for comparison are the LHDs with the largest ϕ_{eff}-values obtained by repeating $n_1 = 100$ times. Here our simulation is divided into two parts. Part one: Repeat $n_1 = 100$ times for the three methods of generating LHDs, and find the $LHD(s^2, 4)s$ with the largest ϕ_{eff}-value as the comparison object. Part two: Repeat Part one $n_2 = 100$ times to obtain the $LHD(s^2, 4)s$ with the largest ϕ_{eff}-value in each replicate.

Goodness of fit (R^2, the larger the better) and mean squared prediction error (MSPE, the smaller the better) were chosen as the evaluation criteria for the fitting and prediction performance of the metamodel $\widehat{f}$. For computing the MSPE, the runs of a random $LHD(10^5, 4)$

are used as the test points. The results of Part one are presented in Table 6, where the bolded parts represent the designs with better predictive performance. In order to reduce the instability of the simulation results of Part one, the simulation of Part one is repeated 100 times in Part two, and the obtained results are averaged and listed in Table 7. It can be easily observed from Tables 6 and 7 that the UPLHD has a slightly higher R^2 and lower MSPE, indicating superior performance.

Table 6 Fitting and prediction performance of $\widehat{f}$ under four different $LHD(s^2, 4)$s in Example 4.2

	$RLHD(49,4)$	$OBLHD(49,4)$	$SLHD(49,4)$	$UPLHD(49,4)$
R^2	0.9984	0.9981	0.9978	**0.9987**
MSPE($\times 10^{-3}$)	25.0	30.9	33.0	**20.8**
	$RLHD(25,4)$	$OBLHD(25,4)$	$SLHD(25,4)$	$UPLHD(25,4)$
R^2	0.9981	0.9978	0.9979	**0.9983**
MSPE($\times 10^{-3}$)	30.6	34.5	32.7	**26.7**

Note: The R^2 and MSPE values presented in the table are derived from design simulations using the largest ϕ_{eff}-value among $n_1 = 100$ RLHD, OBLHD, and SLHD configurations.

Table 7 Replicate the simulation in Table 6 100 ($n_2 = 100$) times to obtain the average R^2 and MSPE values

	$RLHD(49,4)$	$OBLHD(49,4)$	$SLHD(49,4)$	$UPLHD(49,4)$
R^2	0.9984	0.9981	0.9981	**0.9987**
MSPE($\times 10^{-3}$)	24.6	31.3	29.2	**20.8**
	$RLHD(25,4)$	$OBLHD(25,4)$	$SLHD(25,4)$	$UPLHD(25,4)$
R^2	0.9979	0.9980	0.9977	**0.9983**
MSPE($\times 10^{-3}$)	38.0	33.1	35.8	**26.7**

5 Concluding Remarks

In computer experiments where only a few factors are active, a design with good low-dimensional space-filling properties is crucial. This paper proposes an algebraic method to construct asymptotically optimal Latin hypercube designs (LHDs) with small factor-to-run ratios under the uniform projection criterion. Our approach is simple and does not require computer search algorithms. Theoretical and numerical results show the obtained LHDs have good performance under the uniform projection, low-dimensional stratification, maximin distance, and column-orthogonality criteria. Furthermore, based on the constructed UPLHDs, we extended the designs to obtain LHDs accommodating a greater number of factors.

One limitation of our construction is the requirement of s being a prime number to generate LHDs with s^2 and s^3 runs. Future research could extend the method to cover cases where s is a prime power or non-prime number to obtain uniform projection LHDs. Another future work is to generalize the method to construct optimal LHDs under the uniform projection criteria with respect to other commonly used discrepancies[24].

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Santner T J, Williams B J, and Notz W I, *The Design and Analysis of Computer Experiments*, 2nd Edition, Springer, New York, 2018.

[2] Fang K T, Li R, and Sudjianto A, *Design and Modeling for Computer Experiments*, Chapman and Hall/CRC, New York, 2005.

[3] Garud S S, Karimi I A, and Kraft M, Design of computer experiments: A review, *Computers and Chemical Engineering*, 2017, **106**: 71–95.

[4] McKay M, Beckman R, and Conover W, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, 1979, **21**: 239–245.

[5] Johnson M E, Moore L M, and Ylvisaker D, Minimax and maximin distance designs, *Journal of Statistical Planning and Inference*, 1990, **26**: 131–148.

[6] Li W L, Liu M Q, and Tang B X, A method of constructing maximin distance designs, *Biometrika*, 2021, **108**: 845–855.

[7] Fang K T, Uniform design: An application of number-theoretic methods to experimental designs, *Acta Mathematicae Applicatae Sinica*, 1980, **4**: 363–372.

[8] Wang Y and Fang K T, A note on uniform distribution and experimental design, *Kexue Tongbao*, 1981, **26**: 485–489.

[9] He Y Z and Tang B X, Strong orthogonal arrays and associated Latin hypercubes for computer experiments, *Biometrika*, 2013, **100**: 254–260.

[10] Steinberg D M and Lin D K J, A construction method for orthogonal Latin hypercube designs, *Biometrika*, 2006, **93**: 279–288.

[11] Sun F S, Liu M Q, and Lin D K J, Construction of orthogonal Latin hypercube designs, *Biometrika*, 2009, **96**: 971–974.

[12] Morris M D and Mitchell T J, Exploratory designs for computational experiments, *Journal of Statistical Planning and Inference*, 1995, **43**: 381–402.

[13] Wang L, Xiao Q, and Xu H Q, Optimal maximin L_1-distance Latin hypercube designs based on good lattice point designs, *The Annals of Statistics*, 2018, **46**: 3741–3766.

[14] Tang B X, Orthogonal array-based Latin hypercubes, *Journal of the American Statistical Association*, 1993, **88**: 1392–1397.

[15] Fang K T, Lin D K J, Winker P, et al., Uniform design: Theory and application, *Technometrics*, 2000, **42**: 237–248.

[16] Liang Y Z, Fang K T, and Xu Q S, Uniform design and its applications in chemistry and chemical engineering, *Chemometrics and Intelligent Laboratory Systems*, 2001, **58**: 43–57.

[17] Hickernell F J and Liu M Q, Uniform designs limit aliasing, *Biometrika*, 2002, **89**: 893–904.

[18] Moon H, Dean A M, and Santner T J, Two-stage sensitivity-based group screening in computer experiments, *Technometrics*, 2012, **54**: 376–387.

[19] Kleijnen J P, Design and analysis of simulation experiments: Tutorial, *Advances in Modeling and Simulation*, Eds. by Tolk A, Fowler J, Shao G, et al., Springer, New York, 2017.

[20] Moon H, Dean A, and Santner T, Algorithms for generating maximin Latin hypercube and orthogonal designs, *Journal of Statistical Theory and Practice*, 2011, **5**: 81–98.

[21] Joseph V R, Gul E, and Ba S, Maximum projection designs for computer experiments, *Biometrika*, 2015, **102**: 371–380.

[22] Sun F S, Wang Y P, and Xu H Q, Uniform projection designs, *The Annals of Statistics*, 2019, **47**: 641–661.

[23] Wang Y P, Sun F S, and Xu H Q, On design orthogonality, maximin distance, and projection uniformity for computer experiments, *Journal of the American Statistical Association*, 2022, **117**: 375–385.

[24] Liu S X, Wang Y P, and Sun F S, Two-dimensional projection uniformity for space-filling designs, *Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 2023, **51**: 293–311.

[25] Zhou Y D and Xu H Q, Space-filling properties of good lattice point sets, *Biometrika*, 2015, **102**: 959–966.

[26] Tian Y and Xu H Q, A minimum aberration-type criterion for selecting space-filling designs, *Biometrika*, 2022, **109**: 489–501.

[27] Bush K A, Orthogonal arrays of index unity, *The Annals of Mathematical Statistics*, 1952, **23**: 426–434.

[28] Hedayat A S, Sloane N J A, and Stufken J, *Orthogonal Arrays: Theory and Applications*, Springer, New York, 1999.

[29] Ba S, Myers W R, and Brenneman W A, Optimal sliced Latin hypercube designs, *Technometrics*, 2015, **57**: 479–487.

[30] Ai M Y, Kong X S, and Li K, A general theory for orthogonal array based Latin hypercube sampling, *Statistica Sinica*, 2016, **26**: 761–777.