Electronic Journal of Statistics
Vol. 17 (2023) 3942–3968
ISSN: 1935-7524
https://doi.org/10.1214/23-EJS2194

Construction of maximin L_1-distance Latin hypercube designs

Yuhao Yin*

Department of Statistics and Data Science, University of California, Los Angeles
e-mail: yuhaoyin1108@gmail.com

Lin Wang*

Department of Statistics, Purdue University
e-mail: linwang@purdue.edu

and

Hongquan Xu

Department of Statistics and Data Science, University of California, Los Angeles
e-mail: hqxu@stat.ucla.edu

Abstract: Computer experiments are increasingly being used to build high-quality surrogate models for complex emulation systems. Maximin distance Latin hypercube design is an efficient approach for designing computer experiments. Algorithmic search is commonly used for finding such designs but becomes ineffective when searching for large designs. Theoretical construction of such designs is fast but limited and challenging. In this paper, we propose a series of construction methods for maximin distance Latin hypercube designs. We use a piece-wise linear transformation to obtain balanced designs which are then rotated to generate Latin hypercube designs. Theoretical results guarantee that the generated designs are asymptotically optimal under the maximin distance criterion. The generated designs also exhibit low column correlations and mirror symmetry, which significantly benefits the identification of the main and interaction effects. Moreover, we present numerical comparisons with existing methods to demonstrate the superiority of the proposed methods.

MSC2020 subject classifications: Primary 62K99; secondary 62K15.
Keywords and phrases: Computer experiment, correlation, factorial design, orthogonal array, space-filling design.

Received August 2023.

Contents

*These two authors contribute equally.

1. Introduction

Computer experiments are increasingly being used to build high-quality surrogate models for complex emulation systems. A general experimental design for planning computer experiments is the space-filling design, which evenly spreads design points over the design region and thus explores the region efficiently. Representative space-filling designs include Latin hypercube designs (LHDs) [10], minimax and maximin distance designs [6], and uniform designs [5]. An LHD accommodates as many levels as the run size and therefore allows the study of complex systems. A maximin distance design maximizes the separation distance between design points and is asymptotically D-optimal for Gaussian process modeling when observations are nearly independent [6]. Uniform designs minimize the discrepancy between the empirical distribution of the design points and the uniform distribution, including the uniform projection design as a recent modification [16, 15]. Orthogonality between factors is another important criterion when designing computer experiments, which guarantees low correlation between factors and improves the identification of linear trend of factorial effects [24, 21]. Mirror symmetry of design points is also deemed as a good merit of a design because it guarantees that main and interaction effects are uncorrelated and can be accurately identified [25]. Recently, hybrid designs, such as maximin LHDs [19, 28], orthogonal maximin LHDs [8], orthogonal mirror-symmetric LHDs [18], and orthogonal uniform designs [27], have been extensively investigated and constructed to integrate merits of different design criteria.

Maximin LHDs that integrate the merits of maximin distance designs and LHDs are commonly used for designing intricate computer experiments. Metaheuristic algorithms, such as simulated annealing [12, 8], particle swarm optimization [11, 4], and the threshold-accepting method [23], have been adopted to search for good maximin LHDs. However, those stochastic algorithms are not efficient for constructing large and high-dimensional designs due to their computational complexity, yet large designs are commonly needed in computer experiments. Consequently, sophisticated systematic construction methods are highly valuable for being able to generate optimal maximin LHDs efficiently. Wang et al. [19] constructed a series of $p \times (p-1)$ (where p is a prime number) LHDs, which exactly or asymptotically minimize the separation L_1-distance.

Zhou et al. [28] constructed a class of $2^{t+2} \times 2^{t+1}$ (where t is some positive integer) maximin L_2-distance LHDs. Li et al. [9] developed a method of constructing big maximin distance designs by combining two small balanced designs.

This paper proposes a systematic approach to constructing mirror-symmetric maximin L_1-distance LHDs. The proposed method first constructs a class of balanced designs via a piece-wise linear transformation, akin to the Williams transformation used in [19] but modified for our purpose. We then rotate the generated balanced designs to obtain LHDs. When excluding the center point, the obtained LHDs are guaranteed to be asymptotically optimal under the maximin distance criterion. When including the center point, the obtained LHDs are asymptotically optimal among all mirror-symmetric LHDs. The proposed method is then extended to generate LHDs with more flexible sizes. In addition, when the run size of the design is relatively big (say, 100), we may not need as many levels as the run size (i.e., an LHD) to learn about the simulation system. In this case, the generated balanced designs in the first step, which are also asymptotically optimal with a maximin distance, can be directly used for designing complex simulation experiments. Although we focus on the maximum L_1-distance criterion, the constructed designs also perform well under the maximum L_2-distance criterion because the L_1-distance provides a lower bound for the L_2-distance by the Cauchy-Schwarz inequality [19].

The remainder of this paper proceeds as follows. Section 2 introduces notation and background. Section 3 proposes the construction methods and develops theoretical results to guarantee the asymptotical optimality of the constructed LHDs. Section 4 presents numerical comparisons to demonstrate the superiority of the proposed methods. Section 5 discusses the optimality of the constructed mirror-symmetric designs with an odd number of runs. Section 6 concludes this paper. All proofs are provided in the appendix.

2. Notation and background

For an $N \times n$ design matrix $D = (x_{ij})_{1 \leq i \leq N, 1 \leq j \leq n}$, the L_1-distance between the ith and kth rows is defined as $d_{ik}(D) = \sum_{j=1}^{n} |x_{ij} - x_{kj}|$. The L_1-distance of design D, denoted by $d(D)$, is the minimum L_1-distance between any two distinct rows of D, that is, $d(D) = \min\{d_{ik}(D) : i \neq k,\ i, k = 1, \ldots, N\}$. The maximin distance criterion [6] is to maximize $d(D)$ among all possible designs.

An $N \times n$ design with s levels is balanced if all levels appear equally often in each column, that is, each level appears exactly N/s times. In this paper, the levels of the generated designs are denoted by $-(s-1)/2, -(s-3)/2, \ldots, (s-3)/2, (s-1)/2$. An LHD is a special balanced design with $s = N$. A design D is mirror-symmetric if for any row x in D, its mirror-symmetric point, $-x$, is also a row of D. Denote $\text{sign}(D) = (\text{sign}(x_{ij}))$, where $\text{sign}(x)$ is 1 if $x \geq 0$ and -1 otherwise.

For an $N \times n$ balanced design, the average pairwise L_1-distance between rows is $N(s^2-1)n/[3s(N-1)]$ [29]. Because the minimum pairwise L_1-distance cannot exceed the integer part of the average, we have the following lemma.

Lemma 1. *For any $N \times n$ balanced design D with s levels,*

$$d(D) \leq d_{\text{upper}} = \left\lfloor \frac{N(s^2-1)n}{3s(N-1)} \right\rfloor,$$

where $\lfloor x \rfloor$ is the integer part of x.

Specifically, for an $N \times n$ LHD, $d_{\text{upper}} = \lfloor (N+1)n/3 \rfloor$. Define the distance efficiency of a balanced design D as

$$d_{\text{eff}}(D) = d(D)/d_{\text{upper}}.$$

The goal is to construct mirror-symmetric LHDs with $d_{\text{eff}}(D)$ equal or close to 1.

3. Construction methods

3.1. Basic construction method

We first introduce a transformation that will be used in the construction. Let p be an odd prime throughout the paper. For $x \in \{0, 1, \ldots, p-1\}$, define

$$\varphi(x) = \begin{cases} 2x, & \text{for } 0 \leq x < p/4; \\ -2x + p, & \text{for } p/4 < x < 3p/4; \\ 2x - 2p, & \text{for } 3p/4 < x < p. \end{cases} \tag{3.1}$$

Then φ defines a one-to-one map from $x = 0, \ldots, p-1$ to $-(p-1)/2, \ldots, (p-1)/2$. For example, for $p = 3$, φ maps $(0, 1, 2)$ to $(0, 1, -1)$; for $p = 5$, φ maps the levels $(0, 1, 2, 3, 4)$ to $(0, 2, 1, -1, -2)$. Figure 1 shows the cases for $p = 7$ and 11. For any pair of $x_1 \neq x_2$, $\varphi(x_1) \neq \varphi(x_2)$ so that the map is well defined. To see this, for instance, if $0 \leq x_1 < p/4$, $p/4 < x_2 < 3p/4$ and $\varphi(x_1) = \varphi(x_2)$, then $p = 2(x_1 + x_2)$, which contradicts with the fact that p is odd. It is easy to see that φ has the property

$$\varphi(p - x) = -\varphi(x). \tag{3.2}$$

Clearly, φ is a piece-wise linear transformation, akin to the Williams transformation applied to the construction of LHDs in [3] and [19]. It is modified from the Williams transformation to be directly applied in our methods for constructing mirror-symmetric maximin distance designs. Now we present our first algorithm for constructing the desired designs via the φ transformation.

Algorithm 1 (Construction of mirror-symmetric LHDs via rotation)**.**

Step 1. Let p be an odd prime number and X be a p-level full factorial design with 2 columns:

$$X = [0, G, p - G]^\top \pmod{p},$$

where 0, throughout the paper, is a conformable vector (or scalar) with all zero entries, and $G = [G_2, 2G_2, \ldots, ((p-1)/2)G_2]$ with

$$G_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & 2 & \cdots & p-1 \end{bmatrix}.$$

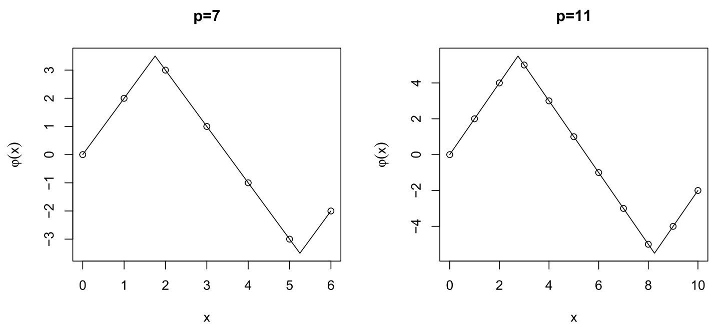

FIG 1. *The piece-wise linear transformation $\varphi(x)$ in (3.1) for $p = 7$ and 11.*

Step 2. Let $D = (x_{ij})$ be the design obtained by deleting the first column of $XX^\top \pmod p$ and

$$E = \varphi(D) = (\varphi(x_{ij}))\,.$$

Step 3. Define a rotation matrix

$$T_2 = \begin{bmatrix} p & -1 \\ 1 & p \end{bmatrix}, \tag{3.3}$$

and let $T = \mathrm{diag}\{T_2, \ldots, T_2\}$ with T_2 repeating $(p^2-1)/2$ times. Let

$$L = ET.$$

Step 4. Delete the first row of L and let

$$L^* = L - \mathrm{sign}(L)/2.$$

The T_2 in (3.3) is called a rotation matrix because $T_2^\top T_2 = cI$ for some constant c. Here $c = p^2 + 1$. This rotation matrix has been used to construct orthogonal LHDs [14, 13, 18], while we use the rotation to construct maximin distance designs. Below is a toy example to illustrate the use of Algorithm 1.

Example 1. Let $p = 3$ and X be a 3-level full factorial:

$$X = [0, G, 3 - G]^\top \pmod 3, \quad \text{where } G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}.$$

The design E generated in Step 2 of Algorithm 1 is

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & -1 & 0 & -1 & -1 \\ 0 & 1 & 1 & -1 & 0 & -1 & -1 & 1 \\ 1 & 1 & -1 & 0 & -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & -1 & -1 & 1 & 0 & 1 \\ -1 & 0 & -1 & -1 & 1 & 0 & 1 & 1 \\ 0 & -1 & -1 & 1 & 0 & 1 & 1 & -1 \\ -1 & -1 & 1 & 0 & 1 & 1 & -1 & 0 \\ -1 & 1 & 0 & 1 & 1 & -1 & 0 & -1 \end{bmatrix}.$$

Let $T = \text{diag}\{T_2, T_2, T_2, T_2\}$ where

$$T_2 = \begin{bmatrix} 3 & -1 \\ 1 & 3 \end{bmatrix}.$$

Step 3 generates

$$L = ET = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & -1 & 4 & 2 & -3 & 1 & -4 & -2 \\ 1 & 3 & 2 & -4 & -1 & -3 & -2 & 4 \\ 4 & 2 & -3 & 1 & -4 & -2 & 3 & -1 \\ 2 & -4 & -1 & -3 & -2 & 4 & 1 & 3 \\ -3 & 1 & -4 & -2 & 3 & -1 & 4 & 2 \\ -1 & -3 & -2 & 4 & 1 & 3 & 2 & -4 \\ -4 & -2 & 3 & -1 & 4 & 2 & -3 & 1 \\ -2 & 4 & 1 & 3 & 2 & -4 & -1 & -3 \end{bmatrix}, \tag{3.4}$$

and Step 4 gives

$$L^* = \begin{bmatrix} 2.5 & -0.5 & 3.5 & 1.5 & -2.5 & 0.5 & -3.5 & -1.5 \\ 0.5 & 2.5 & 1.5 & -3.5 & -0.5 & -2.5 & -1.5 & 3.5 \\ 3.5 & 1.5 & -2.5 & 0.5 & -3.5 & -1.5 & 2.5 & -0.5 \\ 1.5 & -3.5 & -0.5 & -2.5 & -1.5 & 3.5 & 0.5 & 2.5 \\ -2.5 & 0.5 & -3.5 & -1.5 & 2.5 & -0.5 & 3.5 & 1.5 \\ -0.5 & -2.5 & -1.5 & 3.5 & 0.5 & 2.5 & 1.5 & -3.5 \\ -3.5 & -1.5 & 2.5 & -0.5 & 3.5 & 1.5 & -2.5 & 0.5 \\ -1.5 & 3.5 & 0.5 & 2.5 & 1.5 & -3.5 & -0.5 & -2.5 \end{bmatrix}. \tag{3.5}$$

To study the distance of L^* constructed in Algorithm 1, we need to first characterize the property of E and L in Steps 2 and 3.

Theorem 1. *The design E in Step 2 of Algorithm 1 is a $p^2 \times (p^2 - 1)$ mirror-symmetric balanced design with p levels. For $i \neq k$,*

$$d_{ik}(E) = \begin{cases} (p-1)p(p+1)/4 & \text{if } i = 1 \text{ or } k = 1, \\ (p-1)p(p+1)/2 & \text{if the } i\text{th and } k\text{th rows are mirror-symmetric}, \\ (p-1)p(p+1)/3 & \text{otherwise}. \end{cases}$$

Then $d(E) = (p-1)p(p+1)/4$.

Theorem 2. *The design L in Step 3 of Algorithm 1 is a $p^2 \times (p^2 - 1)$ mirror-symmetric LHD with*

$$d(L) \geq (p-1)^2 p(p+1)/4.$$

Now we provide the theoretical guarantee for the distance efficiency of the generated design L^* in Algorithm 1.

Theorem 3. *The design L^* constructed in Algorithm 1 is a $(p^2 - 1) \times (p^2 - 1)$ mirror-symmetric LHD with*

$$d(L^*) \geq \frac{(p-1)^2 p(p+1)}{3} - (p^2 - 1) \quad \text{and} \quad d_{\text{eff}}(L^*) \geq 1 - \frac{1}{p} - \frac{3}{p^2}.$$

As $p \to \infty, d_{\text{eff}}(L^*) \to 1$.

Theorem 3 shows that the L^* generated in Algorithm 1 is an asymptotically maximin distance LHD. Using Algorithm 1, we can obtain many big designs with high efficiencies. For example, using $p = 7, 11$, and 13, we obtain three LHDs, namely, a 48×48 LHD with $d_{\text{eff}} = 0.949$, a 120×120 LHD with $d_{\text{eff}} = 0.977$, and a 168×168 LHD with $d_{\text{eff}} = 0.981$. All those designs are obtained quickly without any computer search.

Another merit of the constructed L^* is that its columns are also mirror-symmetric. To see this, notice that $E^\top = E$. Then columns of E are mirror-symmetric because its rows are mirror-symmetric, and this symmetry is inherited by $L^{*\top}$ after rotation. In fact, let $N = p^2 - 1$ which is the run size of L^*, then the ith and $(i + N/2)$th columns of L^* are mirror-symmetric. Thanks to this symmetry, the first half columns of L^* form a mirror-symmetric LHD, denoted as L^*_{half}, with the same distance efficiency d_{eff} as L^*.

Corollary 1. *The first half columns of L^* constructed in Algorithm 1, denoted as L^*_{half}, is a $(p^2 - 1) \times (p^2 - 1)/2$ mirror-symmetric LHD with $d_{\text{eff}}(L^*_{\text{half}}) \to 1$ as $p \to \infty$.*

Existing systematic methods for constructing maximin distance LHDs can typically only generate saturated or supersaturated designs [22, 19], that is, the number of rows of the constructed designs is typically close to or the same as the number of columns. Those designs are suitable when the experiment budget (or time) is very limited. When the budget can allow more experimental runs, the L^*_{half} can be used to design the experiment. The number of rows is twice as the number of columns and it can better cover the design domain. In addition, when applying the leave-one-pair-out algorithm (to be developed in Section 3.4) to L^*_{half} and the column combination method (to be developed in Section 3.5), we can obtain more flexible designs with high distance efficiencies.

3.2. Correlation

We further study the pairwise correlation between columns of the designs in Algorithm 1. For any design D, define the average pairwise absolute correlation between columns by

$$\rho_{\text{ave}}(D) = \frac{\sum_{i \neq k} |\rho_{ik}(D)|}{n(n-1)}, \tag{3.6}$$

where $\rho_{ik}(D)$ is the correlation coefficient between the ith and kth columns of D. A low ρ_{ave} value generally indicates low correlations between columns, which results in low correlations between estimators for the linear trend of a simulation system.

Theorem 4. *For the designs E, L, and L^* in Steps 2–4 of Algorithm 1,*

$$\rho_{\text{ave}}(E) < \frac{2}{p^2 - 2} \to 0, \quad as \quad p \to \infty,$$
$$\rho_{\text{ave}}(L) < \left(1 + \frac{2}{p}\right) \frac{2}{p^2 - 2} \to 0, \quad as \quad p \to \infty,$$

and therefore,

$$\rho_{\text{ave}}(L^*) < \left(10 + \frac{8}{p}\right) \frac{1}{p^2 - 2} \to 0, \quad \textit{as} \quad p \to \infty.$$

Theorem 4 guarantees that a large L^* constructed via Algorithm 1 has a tiny average absolute correlation. Even so, it should be noted that L^* contains perfectly correlated pairs of columns, that is, mirror-symmetric columns in L^* have a perfect negative correlation. A simple way to reduce the correlation between them is to keep the first half columns (i.e., the 1st to the $(N/2)$th columns) of L^* and randomly permute rows of the second half columns (i.e., the $(N/2+1)$th to the last columns). Here $N = p^2 - 1$ is the run size of L^*. This process does not reduce the distance of L^* because it is column mirror-symmetric and the rows in the first half columns have the same distance distribution as the rows in the second half columns. The minimum distance will remain the same or potentially increase after permuting rows of the second half columns. If we also hope to retain the mirror symmetry of rows, we can permute the 1st to $(N/2)$th rows randomly and apply the same permutation to the $(N/2+1)$th to the last rows. For instance, for the design L^* in (3.5), by permuting row indices of the 5th to 8th columns from $(1,2,3,4,5,6,7,8)$ to $(2,1,4,3,6,5,8,7)$, we obtain a new design given by

$$L' = \begin{bmatrix} 2.5 & -0.5 & 3.5 & 1.5 & -0.5 & -2.5 & -1.5 & 3.5 \\ 0.5 & 2.5 & 1.5 & -3.5 & -2.5 & 0.5 & -3.5 & -1.5 \\ 3.5 & 1.5 & -2.5 & 0.5 & -1.5 & 3.5 & 0.5 & 2.5 \\ 1.5 & -3.5 & -0.5 & -2.5 & -3.5 & -1.5 & 2.5 & -0.5 \\ -2.5 & 0.5 & -3.5 & -1.5 & 0.5 & 2.5 & 1.5 & -3.5 \\ -0.5 & -2.5 & -1.5 & 3.5 & 2.5 & -0.5 & 3.5 & 1.5 \\ -3.5 & -1.5 & 2.5 & -0.5 & 1.5 & -3.5 & -0.5 & -2.5 \\ -1.5 & 3.5 & 0.5 & 2.5 & 3.5 & 1.5 & -2.5 & 0.5 \end{bmatrix}.$$

Design L' is a mirror-symmetric LHD with $d(L') = d(L^*)$, and the maximum pairwise absolute correlation between columns of L' is reduced to 0.76. Clearly, the improvement in correlation can be much better for a bigger design. In general, this process breaks the perfect negative correlation without reducing the distance of the design so that Theorem 3 also holds for L'.

It should be noted that L^*_{half} does not have the perfect correlation problem and all pairs of columns in L^*_{half} have very low correlations. It is easy to show that $\rho_{\text{ave}}(L^*_{\text{half}}) < \rho_{\text{ave}}(L^*)$ and therefore, by Theorem 4, $\rho_{\text{ave}}(L^*_{\text{half}}) \to 0$ as $p \to \infty$.

3.3. Generalization

We can generalize Algorithm 1 to obtain LHDs with $N = p^k - 1$ rows, where k can be any positive integer. To do so, consider starting with a p-level full factorial design with k columns, also denoted as X. Let $GF(p) = \{0, 1, \ldots, p-1\}$ and

$GF(p)[x] = \{a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} : a_0, \ldots, a_{k-1} \in GF(p)\}$. Delete the first column of $XX^\top \pmod{p}$, then each remaining column of $XX^\top \pmod{p}$ is a linear combination of columns of X, corresponds to a nonzero element $a_0 + a_1 x + \cdots + a_{k-1}x^{k-1}$ in $GF(p)[x]$, and can be expressed as x^d modulo a primitive polynomial over $GF(p)[x]$ for $d \in \{0, \ldots, p^k - 1\}$. As shown in [13] and [18], any k consecutive columns of $XX^\top \pmod{p}$, in the order of $x^0, x^1, \ldots, x^{p^k-1}$, form a full factorial design. Our generalized method detailed below uses this property to obtain a new class of maximin distance LHDs.

Algorithm 2 (Generalization of Algorithm 1)**.**

Step 1. Let X be a p-level full factorial design with k columns, where k can be any positive integer.

Step 2. Let $D = (x_{ij})$ be the design obtained by deleting the first column of $XX^\top \pmod{p}$. Arrange the order of columns of D such that any k consecutive columns form a full factorial design. Only keep the first $\lfloor (p^k - 1)/k \rfloor \cdot k$ columns of D and let

$$E = \varphi(D).$$

Step 3. Define a $k \times k$ matrix T_k, whose columns are permutations of $\{1, p, \ldots, p^{k-1}\}$ up to sign changes, and let $T = \text{diag}\{T_k, \ldots, T_k\}$ with T_k repeating $\lfloor (p^k - 1)/k \rfloor$ times. Let

$$L = ET. \tag{3.7}$$

Step 4. Delete the first row of L and let

$$L^* = L - \text{sign}(L)/2. \tag{3.8}$$

Algorithm 2 includes Algorithm 1 as a special case if the T_2 in Step 3 of Algorithm 2 is set to be the rotation matrix used in Algorithm 1. When k is a power of 2, that is $k = 2^c$ for some non-negative integer c, T_k can be constructed recursively from $T_{2^0} = 1$ and

$$T_{2^c} = \begin{bmatrix} p^{2^{c-1}} T_{2^{c-1}} & -T_{2^{c-1}} \\ T_{2^{c-1}} & p^{2^{c-1}} T_{2^{c-1}} \end{bmatrix}, \tag{3.9}$$

which is also a rotation matrix. For a general k, however, it is impossible to obtain an orthogonal T_k. In this case, we may use random permutations of $\{1, p, \ldots, p^{k-1}\}$ for each column of T_k, or search for the permutations that result in small correlations between columns. For example, through computer search, Wang et al. [18] found that the matrix

$$\begin{bmatrix} 1 & 1 & p^2 \\ p & -p^2 & 1 \\ p^2 & p & -p \end{bmatrix} \tag{3.10}$$

has small correlations between columns, so it is a good choice for T_3.

Example 2. Consider $k = 1$ and $p = 11$. Step 1 of Algorithm 2 generates the X with a single column whose entries are $0, 1, \ldots, 10$, and the design D in Step 2 is a good lattice point (GLP) design [29]. The matrix T_k in Step 3 reduces to 1 (a scalar), and therefore Step 4 generates the 10×10 mirror-symmetric LHD

$$L^* = \begin{bmatrix} 1.5 & 3.5 & 4.5 & 2.5 & 0.5 & -0.5 & -2.5 & -4.5 & -3.5 & -1.5 \\ 3.5 & 2.5 & -0.5 & -4.5 & -1.5 & 1.5 & 4.5 & 0.5 & -2.5 & -3.5 \\ 4.5 & -0.5 & -3.5 & 1.5 & 2.5 & -2.5 & -1.5 & 3.5 & 0.5 & -4.5 \\ 2.5 & -4.5 & 1.5 & 0.5 & -3.5 & 3.5 & -0.5 & -1.5 & 4.5 & -2.5 \\ 0.5 & -1.5 & 2.5 & -3.5 & 4.5 & -4.5 & 3.5 & -2.5 & 1.5 & -0.5 \\ -0.5 & 1.5 & -2.5 & 3.5 & -4.5 & 4.5 & -3.5 & 2.5 & -1.5 & 0.5 \\ -2.5 & 4.5 & -1.5 & -0.5 & 3.5 & -3.5 & 0.5 & 1.5 & -4.5 & 2.5 \\ -4.5 & 0.5 & 3.5 & -1.5 & -2.5 & 2.5 & 1.5 & -3.5 & -0.5 & 4.5 \\ -3.5 & -2.5 & 0.5 & 4.5 & 1.5 & -1.5 & -4.5 & -0.5 & 2.5 & 3.5 \\ -1.5 & -3.5 & -4.5 & -2.5 & -0.5 & 0.5 & 2.5 & 4.5 & 3.5 & 1.5 \end{bmatrix}$$

with $d(L^*) = 34$ and $d_{\text{eff}}(L^*) = 94.4\%$.

Example 3. Consider $k = 3$ and $p = 5$. Step 1 of Algorithm 2 provides a 125×3 full factorial design X with levels 0, 1, 2, 3, 4 so that D in Step 2 has 125 rows and 124 columns. Step 2 only keeps 123 columns of D and generates $E = \varphi(D)$. Step 3 uses the matrix

$$T_3 = \begin{bmatrix} 1 & 1 & 25 \\ 5 & -25 & 1 \\ 25 & 5 & -5 \end{bmatrix},$$

which is the matrix in (3.10) with $p = 5$, and $T = \text{diag}\{T_3, \ldots, T_3\}$ with T_3 repeating 41 times, and generates design L. Step 4 generates L^* with 124 rows and 123 columns, which is also mirror-symmetric. The generated L^* has $d(L^*) = 4901$ and $d_{\text{eff}}(L^*) = 95.6\%$.

Examples 2 and 3 illustrate the use and effectiveness of Algorithm 2. Theoretical results about the distance efficiency of the generated L^* from Algorithm 2 requires tedious notations and thus are omitted here. The theory for a special case $k = 1$ is simple to present.

Theorem 5. *If $k = 1$, the design L^* constructed in Algorithm 2 is a $(p-1) \times (p-1)$ mirror-symmetric LHD with*

$$d(L^*) \geq \frac{p^2 - 1}{3} - (p - 1) \quad \text{and} \quad d_{\text{eff}}(L^*) \geq 1 - \frac{2}{p}.$$

As $p \to \infty, d_{\text{eff}}(L^*) \to 1$.

Wang et al. [19] provided a construction method for LHDs with $p - 1$ runs and $p - 1$ factors as well. A lower bound of d_{eff} for their design E_b^* is $1 - 2.43/p$ for a prime $p \geq 7$, which is slightly worse than the lower bound for L^* by Theorem 5. For example, when $p = 89$, $d_{\text{eff}}(L^*) = 0.9890 > d_{\text{eff}}(E_b^*) = 0.9852$;

when $p = 103$, $d_{\text{eff}}(L^*) = 0.9903 > d(E_b^*) = 0.9886$, etc. In addition, L^* is also mirror-symmetric among columns, so that L^*_{half} formed by the first half columns has the same distance efficiency as L^*. By contrast, the E_b^* does not have this merit. The E_b^* is approximately maximin only if all columns are present, and its distance efficiency decreases dramatically when the first half columns are used. Section 4 provides more details on the comparison of these two types of designs.

3.4. Leave-one-pair-out method

We present a method to generate many asymptotic maximin distance LHDs with flexible run and column sizes.

Algorithm 3 (Leave-one-pair-out method)**.**

Step 1. Specify an $N \times n$ mirror-symmetric LHD, denoted as D.

Step 2. Delete any row x, denoted as $x = (x_1, x_2, \dots, x_n)$, and its mirror-symmetric row $-x$. Rearrange levels of each of the remaining $N - 2$ rows, denoted as $y = (y_1, y_2, \dots, y_n)$, by letting

$$y_i^* = \begin{cases} y_i - \text{sign}(y_i) & \text{if } |y_i| > |x_i|, \\ y_i & \text{otherwise.} \end{cases} \tag{3.11}$$

where $y^* = (y_1^*, y_2^*, \dots, y_n^*)$ is the corresponding row after level rearrangement. The resulting $(N - 2) \times n$ design is still a mirror-symmetric LHD.

Step 3. Repeat Step 2 for k_r times, where k_r is a fixed constant, to generate an $(N - 2k_r) \times n$ mirror-symmetric LHD.

Step 4. Delete any k_c columns, where k_c is a fixed constant, to generate an $(N - 2k_r) \times (n - k_c)$ mirror-symmetric LHD, denoted as F.

Algorithm 3 can be regarded as a generalization of the leave-one-out method introduced in [19]. The key difference is that we leave out a pair of mirror-symmetric rows to keep the mirror-symmetric structure. Whenever two mirror-symmetric rows are deleted from the original $N \times n$ design, the L_1-distance will decrease at most by $2n$ after the level rearrangement detailed in (3.11), and whenever a column gets deleted, the L_1-distance of the remaining design will reduce at most by $N - 1$. Hence, we have the following result.

Theorem 6. *The design F generated via Algorithm 3 is an $(N - 2k_r) \times (n - k_c)$ mirror-symmetric LHD with*

$$d_{\text{eff}}(F) \geq d_{\text{eff}}(D) - \frac{6k_r}{N+1} - \frac{3k_c}{n} - \frac{2}{(N+1)n}.$$

Theorem 6 shows that if $d_{\text{eff}}(D) \to 1$ as $N, n \to \infty$, then $d_{\text{eff}}(F) \to 1$ given that k_r and k_c are much smaller than N and n, respectively. Hence, Algorithm 3 provides a method to construct a series of asymptotically optimal mirror-symmetric LHDs with flexible sizes. For simplicity, we apply Algorithm 3 by deleting the last row and the corresponding mirror-symmetric row in Step 2 and the last k_c columns in Step 4. For example, L^* constructed via Algorithm 1 with $p = 17$ has 288 rows and 288 columns, and $d_{\text{eff}}(L^*) = 0.989$. Set $k_r = 2$ and $k_c = 5$. Applying Algorithm 3 to this L^* generates a 284×283 mirror-symmetric LHD with $d_{\text{eff}} = 0.975$. Applying Algorithm 3 to its half design L^*_{half}, we obtain a 284×139 mirror-symmetric LHD with $d_{\text{eff}} = 0.955$. As another example, start with the 498×498 L^* constructed via Algorithm 2 with $k = 1$ and $p = 499$. Applying Algorithm 3 to the L^* with $k_r = 4$ and $k_c = 15$, we obtain a 490×483 mirror-symmetric LHD with $d_{\text{eff}} = 0.973$. Applying Algorithm 3 to its half design L^*_{half} with $k_r = 20$ and $k_c = 15$, we obtain a 458×234 mirror-symmetric LHD with $d_{\text{eff}} = 0.956$. We can easily obtain many flexible mirror-symmetric LHDs by applying Algorithm 3 to the designs generated from Algorithms 1 and 2 and their half designs.

3.5. Combining designs

We present a general method to generate asymptotic maximin distance LHDs with flexible column sizes by combining designs.

Theorem 7. *Let D_1 and D_2 be LHDs with N rows and n_1 and n_2 columns, respectively, and $F = [D_1, D_2]$ be the design formed by combining columns of D_1 and D_2. Then F is an LHD with $n_1 + n_2$ columns and*

$$d_{\text{eff}}(F) \geq \frac{n_1 d_{\text{eff}}(D_1) + n_2 d_{\text{eff}}(D_2)}{n_1 + n_2} - \frac{4}{(N+1)(n_1+n_2)}.$$

If both $d_{\text{eff}}(D_1) \to 1$ and $d_{\text{eff}}(D_2) \to 1$, then $d_{\text{eff}}(F) \to 1$ as $N \to \infty$.

By Theorem 7, we can combine two maximin LHDs to obtain a maximin LHD with a flexible number of columns. Let D_1 be a design constructed in the previous sections with $n_1 = N$ or $N/2$ columns, and let D_2 be a maximin LHD generated with existing stochastic algorithms (e.g., the R package `SLHD` [1]) with flexible n_2 columns. Then the design F formed by combining columns of D_1 and D_2 would have a large distance. This method has two advantages. First, compared with a direct search using stochastic algorithms to obtain $n = n_1 + n_2$ columns, the design F typically has a higher distance efficiency because of the high distance efficiency of D_1. Second, the stochastic search is only applied for searching D_2 with n_2 columns, which greatly saves the computation compared to searching F with n columns. In addition, if both D_1 and D_2 are mirror-symmetric, so does the resulting design F.

Example 4. Consider $k = 1$ and $p = 97$. Algorithm 2 generates a 96×96 maximin LHD, L^*, with $d(L^*) = 3072$ and $d_{\text{eff}}(L^*) = 99.0\%$. The half design L^*_{half} is 96×48 and has $d(L^*_{\text{half}}) = 1536$ and $d_{\text{eff}}(L^*_{\text{half}}) = 99.0\%$. If we need a design with 53 columns, we can use the R package `SLHD` [1] to generate a 96×5 maximin LHD and combine it with L^*_{half} by columns, which can generate a 96×53 LHD with a distance 1608 and $d_{\text{eff}} = 93.9\%$. When we combine L^*

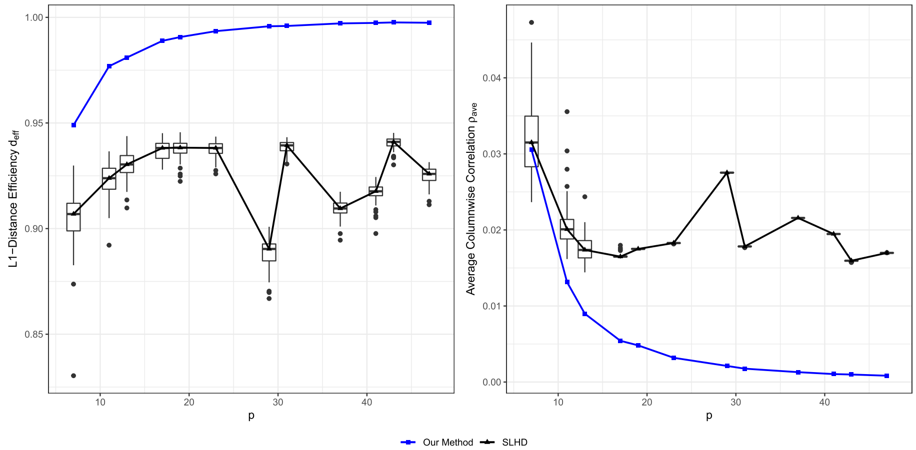

FIG 2. *Comparison of the values of d_{eff} (left) and ρ_{ave} (right) for the L^* constructed via Algorithm 1 and the LHDs obtained via the `SLHD` package.*

with the 96×5 design, we obtain a 96×101 LHD, which has a distance 3151 and $d_{\text{eff}} = 96.5\%$. By comparison, the best 96×53 design we obtain from 100 repetitions of `SLHD` has $d_{\text{eff}} = 87.2\%$, and the best 96×101 design we obtain from 100 repetitions of `SLHD` has $d_{\text{eff}} = 92.8\%$. Clearly, using L^* and L^*_{half} can generate better designs with larger distance efficiencies.

4. Numerical comparisons

In this section, we evaluate the performance of the generated designs by comparing them with existing ones.

We first compare the distance efficiency d_{eff} and average pairwise correlation ρ_{ave} of the design L^* generated from Algorithm 1 and the LHDs generated from the R package `SLHD` [1], which is a well-applied tool for generating maximin LHDs. Because `SLHD` contains randomness, we run the function `maximinSLHD` 100 times, with option $t = 1$ and default settings of other arguments, and examine the boxplots of the values of d_{eff} and ρ_{ave} for the obtained LHDs. For each odd prime p, Figure 2 plots the values of d_{eff} (left) and ρ_{ave} (right) of the $(p^2-1) \times (p^2-1)$ design L^* and the LHDs obtained via the `SLHD` package. It is clear that L^* outperforms the LHDs from the `SLHD` package in the sense of maximizing d_{eff} and minimizing ρ_{ave}. Specifically, $d_{\text{eff}}(L^*)$ is always above 95% and converges to 1 quickly as p increases. By contrast, the d_{eff} values of the LHDs from the `SLHD` package are all below 95% and usually fluctuate around 90%. In addition, except for $p = 7$, $\rho_{\text{ave}}(L^*)$ is always smaller than the ρ_{ave} values of the LHDs from the `SLHD` package and vanishes quickly as p increases.

Next, we evaluate the performance of the half designs. The half design of L^* generated from Algorithm 1 has the same d_{eff} as L^* and its superiority is demonstrated by the above comparison. Therefore, we instead evaluate the half

TABLE 1
Comparison of the L_1-distances of $N \times n$ LHDs.

N	n	Our Method	LP-GLP	WP-GLP		SLHD	
				Median	Max	Median	Max
6	3	6	5	4	4	6	6
10	5	17	14	10	12	14	15
12	6	24	20	15	20	20	22
16	8	43	37	26	36	34.5	37
18	9	54	45	34	43	43	48
22	11	81	69	51	65	64	69
28	14	131	115	86	103	105	111
30	15	150	125	100	118	121	127
36	18	216	180	146	173	174	183
40	20	267	231	183	211	218	228
42	21	294	245	203	239	241	250
46	23	353	304	246	284	289	303
52	26	451	387	320	362	370.5	385
58	29	561	480	403	446	465	483
60	30	600	500	432	479	497	515
66	33	726	605	529	590	607.5	625
70	35	817	696	599	656	685	706
72	36	864	720	636	698	725	753
78	39	1014	845	752	824	857	877
82	41	1121	952	837	912	946	986
88	44	1291	1095	971	1047	1096.5	1127
96	48	1536	1280	1164	1255	1310	1352

design of L^* obtained from Algorithm 2, denoted as L^*_{half}. We illustrate the merits of L^*_{half} constructed in Algorithm 2 with $k = 1$, which is an LHD with $N = p-1$ runs and $n = N/2$ factors. We compare it with LHDs of the same size generated using different methods, including the linearly permuted good lattice point sets ("LP-GLP", [29]), Williams transformation of linearly permuted good lattice point sets ("WP-GLP", [19]), and the R package `SLHD` [1]. For the "LP-GLP" method, we apply the linear permutation $x \to x + (p-1)/2 \pmod p$ to the $p \times (p-1)$ good lattice point design following [17] and use the leave-one-out method to obtain a $(p-1) \times (p-1)$ LHD, which is mirror-symmetric among both rows and columns and then we take the first half columns. The $(p-1) \times (p-1)$ LHD generated by the "WP-GLP" method is not mirror-symmetric among columns, and the design consisting of the first half columns has a small L_1-distance. To make a meaningful comparison we randomly select half of its columns with 10,000 repetitions and show the median and maximum distance. For the `SLHD` package, we run the command `maximinSLHD` with option $t = 1$ and default settings for other arguments 100 times and also consider the median and maximum distance. Table 1 displays the L_1-distance of different designs. For all N investigated, our constructed design L^*_{half} consistently possesses the largest L_1-distance. Moreover, this superiority of L^*_{half} holds for a large N, considering the asymptotic maximin optimality of L^*_{half}.

Next, we compare our method for constructing flexible designs proposed in Section 3.5 with the R package `SLHD`. Specifically, we consider adding n_2 columns to two designs with n_1 columns: the 96×48 LHD (half design generated by

TABLE 2
L_1-distances of $N \times n$ designs with varying n_2/n_1 ratios.

N	n	53	58	62	67	72	77	82	86	91
96	Ours	1608	1732	1849	2000	2149	2299	2459	2579	2731
	SLHD	1493	1659	1779	1933	2090	2249	2407	2542	2704
N	n	92	101	109	118	126	134	143	151	160
168	Ours	4912	5336	5749	6212	6640	7063	7545	7977	8476
	SLHD	4640	5114	5547	6020	6465	6920	7453	7866	8411

Algorithm 2 with $p = 97$ and $k = 1$) and the 168×84 LHD (half design generated by Algorithm 1 with $p = 13$). The added columns are also generated by `SLHD` with varying n_2/n_1 ratios. Selected $n = n_1 + n_2$ values are such that the ratio n_2/n_1 is approximately equal to $0.1, 0.2, \ldots, 0.9$. Table 2 compares our designs with those constructed by the R package SLHD. When the `SLHD` package is used, we run the command `maximinSLHD` with option $t = 1$ and default settings for other arguments 100 times and choose the best design. We see that our designs consistently have a larger L_1-distance than the designs generated from `SLHD`.

5. Mirror-symmetric designs with an odd number of runs

The constructed designs in Section 3 always have an even number of runs. If the required run size N is odd, the upper bound d_{upper} in Lemma 1 is not achievable with a mirror-symmetric design, meaning that a mirror-symmetric maximin design does not exist. This is because a mirror-symmetric design with an odd number of runs must include the center $(0, \ldots, 0)$, which restricts its distribution of design points. In this case, if a mirror-symmetric design is required, we need to derive a tight upper bound for the L_1-distance among mirror-symmetric designs to study their distance efficiency.

Theorem 8. *For any $N \times n$ mirror-symmetric balanced design D with s levels, where N is an odd number,*

$$d(D) \leq d^*_{\text{upper}} = \left\lfloor \frac{N(s^2-1)n}{4s(N-1)} \right\rfloor. \tag{5.1}$$

Clearly, Theorem 8 provides a tighter bound than Lemma 1. By Theorem 8, we define a new distance efficiency for mirror-symmetric designs with an odd number of runs. For such a design D, define

$$d^*_{\text{eff}}(D) = d(D)/d^*_{\text{upper}}.$$

Then for an odd run size, our goal is to generate mirror-symmetric designs with d^*_{eff} equal or close to 1.

The design L generated in Step 3 of Algorithm 1 has p^2 rows and $p^2 - 1$ columns. By Theorem 2, it is a mirror-symmetric LHD, and

$$d^*_{\text{eff}}(L) \geq \frac{(p-1)^2 p(p+1)/4}{d^*_{\text{upper}}} \geq \frac{(p-1)^2 p(p+1)/4}{(p^4-1)/4} = 1 - \frac{p+1}{p^2+1} \to 1,$$

meaning that L is asymptotically maximin among all mirror-symmetric LHDs. For example, the L in (3.4) has 9 rows and 8 columns with $d(L) = 20$. The corresponding $d^*_{\text{upper}} = 20$. Therefore, $d^*_{\text{eff}}(L) = 1$. Note that L also has mirror-symmetric columns so that the first half columns of L form a design L_{half} with the same d^*_{eff} as L. Similarly, the design L generated in Step 3 of Algorithm 2 and when $k = 1$, its half design, are also mirror-symmetric LHDs with a high efficiency d^*_{eff}.

The design E in Step 2 of Algorithm 1 has p^2 rows, $p^2 - 1$ columns, and p levels, and is mirror-symmetric. Although it is not an LHD, it has balanced levels. Using a big prime p, we can obtain a large design with many levels. For example, with $p = 23$, we obtain a 529×528 balanced design with 23 levels each repeating 23 times. These many levels are typically adequate to study complex and intricate systems.

Theorem 1 shows that $d(E) = (p-1)p(p+1)/4$, which exactly equals d^*_{upper} for such a design size. Therefore, we always have

$$d^*_{\text{eff}}(E) = 1,$$

meaning that E is exactly maximin among all mirror-symmetric designs with the same size and number of levels. The E also has mirror-symmetric columns so that the first half columns of E form a design E_{half} with $d^*_{\text{eff}}(E_{half}) = 1$.

We evaluate the performance of L_{half}, the half design of L obtained from Step 3 of Algorithm 1. We compare L_{half} with designs constructed by existing methods:

(a) The maximin distance designs using the R package `SLHD`. These designs are obtained by maximizing the minimum Euclidean distance between distinct design points. A design with a bigger Euclidean distance is more space-filling.

(b) The maximum projection designs using the R package `MaxPro` [2]. These designs are obtained by minimizing the maximum projection criterion $\psi(D)$ proposed in [7]. A design D with a smaller $\psi(D)$ tends to be more space-filling in all projection dimensions of the design domain.

(c) The uniform designs [5] implemented via the R package `UniDOE` [26]. These designs are obtained by minimizing the centered L_2-discrepancy (CD_2). Design points with a smaller CD_2 tend to be more uniformly distributed in the design domain.

We compare the space-filling properties in all projection dimensions for the above designs. For any $N \times n$ design D, there are $\binom{n}{q}$ possible projected designs on q dimensions, $q = 1, \ldots, n$. For each projected design, we consider four space-filling criteria: the Euclidean distance of the design (the minimum Euclidean distance between its distinct rows), $\psi(D)$, CD_2, and ρ_{ave}. The design points are scaled to $[0, 1]^n$ to compute $\psi(D)$ and CD_2. When comparing CD_2, we use L_{half} as a baseline and compute the relative CD_2, which is the difference of CD_2 values between other designs and L_{half}. For each projection dimension q, we evaluate all $\binom{n}{q}$ projected designs and present the worst-case scenarios for the above criteria.

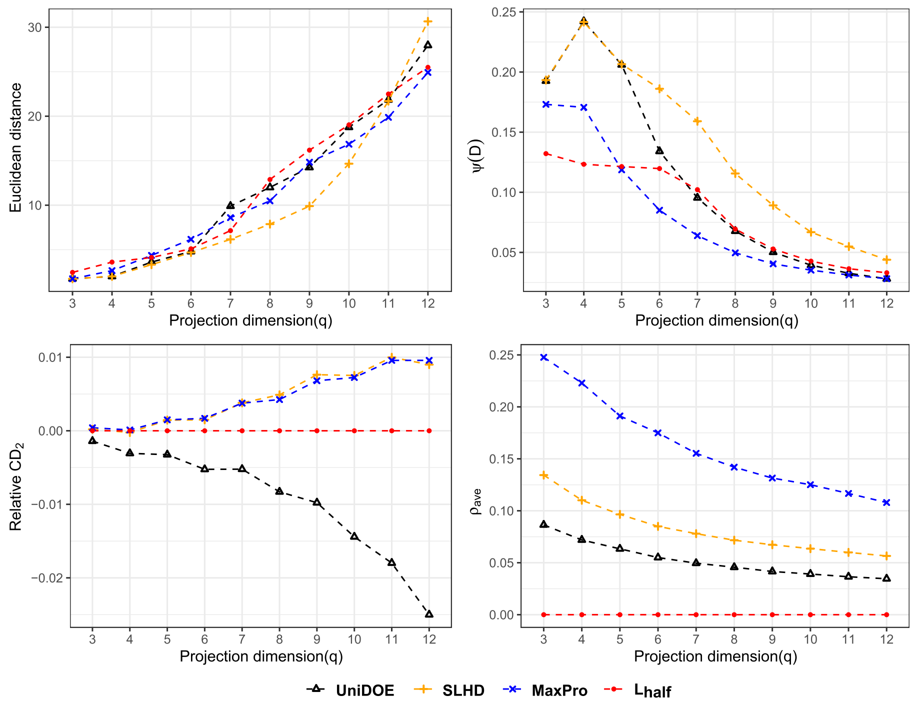

FIG 3. *Comparison of space-filling criteria for projections of* 25×12 *designs: Euclidean distance (the larger the better),* $\psi(D)$ *(the smaller the better), relative* CD_2 *(the smaller the better), and* ρ_{ave} *(the smaller the better).*

Figure 3 presents the comparison for 25×12 designs generated using different methods. The L_{half} is the half design of L generated in Step 3 of Algorithm 1 with $p = 5$. Figure 3 exhibits the robustness of L_{half} under different space-filling criteria. First, from the comparison of Euclidean distance, we see that SLHD performs the worst for all low-dimensional projections, and all other three methods are comparable. Note that L_{half} has relatively a big Euclidean distance in all projection dimensions although it targets to have the maximum L_1-distance. For $\psi(D)$ or CD_2, L_{half} performs comparatively well and may even outperform the corresponding optimal LHDs (i.e., maximum projection designs for $\psi(D)$) in low dimensions. Moreover, L_{half} has the smallest average pairwise correlation ρ_{ave} for all projection dimensions. Figure 4 presents the comparison for 49×24 designs, where L_{half} is the half design of L obtained from Step 3 of Algorithm 1 with $p = 7$. We see the same robustness of L_{half} as the comparison of 25×12 designs.

6. Concluding remarks

In this paper, we have proposed a series of systematic construction methods to generate mirror-symmetric maximin distance designs. Algorithm 1 applies

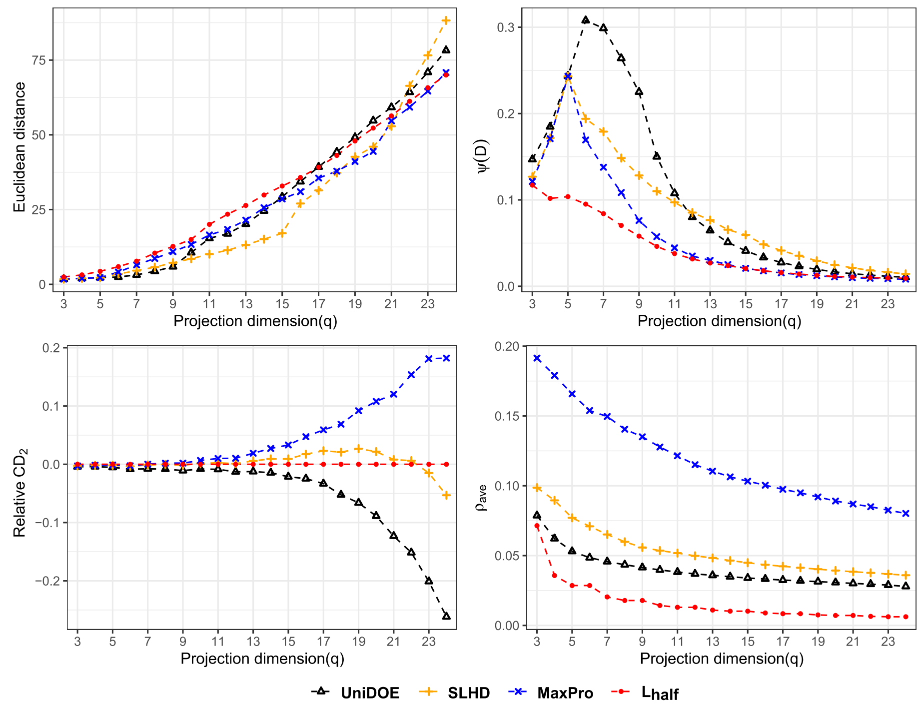

FIG 4. *Comparison of space-filling criteria for projections of* 49×24 *designs: Euclidean distance (the larger the better),* $\psi(D)$ *(the smaller the better), relative* CD_2 *(the smaller the better), and* ρ_{ave} *(the smaller the better).*

the piece-wise linear transformation in (3.1) to a factorial design to obtain a maximin balanced design, which is then rotated to generate an LHD with p^2 or $p^2 - 1$ runs and $p^2 - 1$ columns, where p is an odd prime. Algorithms 2 extends Algorithm 1 to generate designs with p^k or $p^k - 1$ runs and up to $p^k - 1$ columns for an interger k. Half columns of the generated designs have the same distance efficiency and are also maximin distance designs. Algorithm 3 and Section 3.5 further extend Algorithms 1 and 2 from different aspects to generate more flexible designs. The generated designs are asymptotically optimal under the maximin L_1-distance criterion, as guaranteed by the theoretical results. The average absolute correlation between columns of the generated designs rapidly converges to zero as the design size increases. Moreover, existing systematic methods for constructing maximin distance LHDs can typically only generate saturated or supersaturated designs. By contrast, our generated designs can have much more rows than columns and therefore can better cover the design domain. Further, we compare the generated designs with existing designs under various space-filling criteria, which often shows the superiority of the newly generated ones.

We have also constructed a class of maximin balanced designs. As discussed before, when the design size is big, we may not need as many levels as in an LHD to learn about the simulation system. In this case, maximin balanced designs are good choices for designing experiments. However, despite some recent research [e.g., 9, 20], construction methods for maximin balanced designs are not well studied yet and are particularly worthy of further study.

Appendix A: Proofs

Proof of Theorem 1. Design E is balanced because D in Step 2 of Algorithm 1 is balanced. Since

$$XX^\top = \begin{bmatrix} 0 & 0 & 0 \\ 0 & G^\top G & p - G^\top G \\ 0 & p - G^\top G & G^\top G \end{bmatrix} \pmod p,$$

and

$$E = \begin{bmatrix} \varphi(0) & \varphi(0) \\ \varphi(G^\top G) & \varphi(p - G^\top G) \\ \varphi(p - G^\top G) & \varphi(G^\top G) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \varphi(G^\top G) & -\varphi(G^\top G) \\ -\varphi(G^\top G) & \varphi(G^\top G) \end{bmatrix},$$

where the second equation holds due to the property (3.2), then the ith and $(i+(N-1)/2)$th rows of E are mirror-symmetric, for $i = 2, \dots, (N+1)/2$, and therefore E is mirror-symmetric. Here $N = p^2$ is the number of rows of E.

The first column of $XX^\top$ is a vector of all zeros. Let $E' = \varphi(XX^\top)$, then the first column of E' is also a vector of all zeros because $\varphi(0) = 0$. Therefore, E' is a $p^2 \times p^2$ matrix with the same distance distribution as E, and it suffices to study the distance between rows in E'. Denote $XX^\top = (x'_{ij})$, then $E' = (\varphi(x'_{ij}))_{1\le i\le N,\, 1\le j\le n}$, where $N = n = p^2$.

Case 1: $i = 1$ or $k = 1$. Since $(\varphi(x'_{11}), \dots, \varphi(x'_{1n}))$ is a row of all zeros, and $(\varphi(x'_{i1}), \dots, \varphi(x'_{in}))$ for any $i \neq 1$ has every level in $\{-(p-1)/2, \dots, (p-1)/2\}$ appearing p times, then

$$d_{ik}(E) = p \sum_{t=0}^{p-1} |t - (p-1)/2| = (p-1)p(p+1)/4.$$

Case 2: the ith and kth rows are mirror-symmetric. Also considering that $(\varphi(x'_{i1}), \dots, \varphi(x'_{in}))$ for any $i \neq 1$ has every level in $\{-(p-1)/2, \dots, (p-1)/2\}$ appearing p times,

$$d_{ik}(E) = p \sum_{t=0}^{p-1} |t - (p-1-t)| = p \sum_{t=0}^{p-1} |2t - (p-1)| = (p-1)p(p+1)/2.$$

Case 3: other than the above two cases. Denote $X = (A, B)$, where A and B respectively are the two design columns of X. Rows in $XX^\top$ are in the linear space spanned by $A^\top$ and $B^\top$ over the finite field $GF(p) = \{0, 1, \dots, p-1\}$. The first row in $XX^\top$ is the zero vector of this linear space, and any row in D can be represented as $c_1A^\top + c_2B^\top \pmod p$, where $c_1, c_2 \in GF(p)$. Consequently, we can divide rows in $XX^\top$, excluding the first row, into $p+1$ groups, denoted as $R_1, \ldots, R_{p+1}$, with $R_j = \{c(A^\top + jB^\top)(\text{mod } p) : c \in GF(p)\backslash\{0\}\}$ for $j = 1, \ldots, p-1$, $R_p = \{cA^\top(\text{mod } p) : c \in GF(p)\backslash\{0\}\}$, and $R_{p+1} = \{cB^\top(\text{mod } p) : c \in GF(p)\backslash\{0\}\}$. The number of elements in every group is $p-1$. Vectors within the same group are linearly dependent, and those between different groups are linearly independent. There are two situations.

(i) For any two row vectors in distinct groups, there are p^2 possible level combinations, and each appears exactly once. Their corresponding rows in E' also have each level combination appearing once since the transformation (3.1) is a one-to-one mapping over $GF(p)$. Therefore,

$$\begin{aligned} d_{ik}(E) &= \sum_{s=0}^{p-1}\sum_{t=0}^{p-1}|s-t| = \sum_{s=0}^{p-1}\left(\sum_{t=0}^{s}(s-t) + \sum_{t=s+1}^{p-1}(t-s)\right) \\ &= \sum_{s=0}^{p-1}\left[\frac{(s+1)s}{2} + \frac{(p-s)(p-s-1)}{2}\right] = (p-1)p(p+1)/3. \end{aligned}$$

(ii) All row vectors in the same group, plus a zero vector, form p $p \times (p-1)$ good lattice point (GLP) designs. Also, the transformation (3.1) can be regarded as a special case of the linearly permuted Williams transformation since $\varphi(x) = W(x \oplus (p-1)/4) - (p-1)/2$ if $p = 1 \pmod 4$ and $\varphi(x) = -W(x \oplus (3p-1)/4) + (p-1)/2$ if $p = 3 \pmod 4$, where $W(x)$ is the Williams transformation defined in [19] and $x \oplus y = x + y \pmod p$. Then, according to Theorem 1 of [19], if the ith and kth rows of $XX^\top$ are in the same group and are not mirror-symmetric, then

$$d_{ik}(E) = (p^2-1)/3 \times p = (p-1)p(p+1)/3.$$

This completes the proof. □

Proof of Theorem 2. The design L is an LHD since two consecutive columns of E in Step 2 form an orthogonal array of strength two and T_2 is a rotation matrix [14, 18]. In addition, L is mirror-symmetric since E is mirror-symmetric and T is block-diagonal. Denote $L = (l_{ij})$ and $E = (e_{ij})$. Since $L = ET$ and the rotation matrix T is block diagonal, for any row index $i \neq k$, we have

$$\begin{aligned} d_{ik}(L) &= \sum_{j=1}^{p^2-1}|l_{ij} - l_{kj}| \\ &= \sum_{t=1}^{(p^2-1)/2}\big\{|(pe_{i(2t-1)} + e_{i(2t)}) - (pe_{k(2t-1)} + e_{k(2t)})| \\ &\quad + |(-e_{i(2t-1)} + pe_{i(2t)}) - (-e_{k(2t-1)} + pe_{k(2t)})|\big\} \\ &= \sum_{t=1}^{(p^2-1)/2}\big\{|p(e_{i(2t-1)} - e_{k(2t-1)}) + (e_{i(2t)} - e_{k(2t)})| \end{aligned}$$

$$+ |p(e_{i(2t)} - e_{k(2t)}) - (e_{i(2t-1)} - e_{k(2t-1)})|\}$$
$$\geq \sum_{t=1}^{(p^2-1)/2} \{p|e_{i(2t-1)} - e_{k(2t-1)}| - |e_{i(2t)} - e_{k(2t)}|$$
$$+ p|e_{i(2t)} - e_{k(2t)}| - |e_{i(2t-1)} - e_{k(2t-1)}|\}$$
$$= \sum_{j=1}^{p^2-1} \{p|e_{ij} - e_{kj}| - |e_{ij} - e_{kj}|\} = (p-1)d_{ik}(E). \quad \text{(A.1)}$$

Then by Theorem 1,

$$d(L) = \min_{i \neq k} d_{ik}(L) \geq (p-1)d(E) = (p-1)^2 p(p+1)/4.$$

This completes the proof. □

Proof of Theorem 3. First, we have $d_{ik}(L^*) \geq d_{(i+1)(k+1)}(L) - (p^2 - 1)$. By (A.1), we have $d_{(i+1)(k+1)}(L) \geq (p-1)d_{(i+1)(k+1)}(E)$ for all $i \neq k$, then

$$d_{ik}(L^*) \geq (p-1)d_{(i+1)(k+1)}(E) - (p^2-1), \quad \forall\, i \neq k.$$

Therefore,

$$d(L^*) \geq (p-1) \min_{i,k \neq 1} d_{ik}(E) - (p^2-1) = \frac{(p-1)^2 p(p+1)}{3} - (p^2-1).$$

By Lemma 1, we have $d_{\text{upper}}(L^*) = \lfloor p^2(p^2-1)/3 \rfloor$. Therefore,

$$d_{\text{eff}}(L^*) = \frac{d(L^*)}{d_{\text{upper}}(L^*)} \geq \frac{p(p-1)^2(p+1)/3 - (p^2-1)}{p^2(p^2-1)/3} = 1 - \frac{1}{p} - \frac{3}{p^2} \to 1,$$

as $p \to \infty$. □

Proof of Theorem 4. First consider $\rho_{\text{ave}}(E)$. Since $XX^\top$ is symmetric with $(XX^\top)^\top = XX^\top$, the ith column of E is the $(i+1)$th row of $E' = \varphi(XX^\top)$. Following the proof of Theorem 1, all rows of $XX^\top$, excluding the first row which contains all zeros, are partitioned to $p+1$ groups $R_1, \ldots, R_{p+1}$, which is also a partition for the columns of the design D (in Step 2 of Algorithm 1).

(i) For any two columns in distinct groups, all level combinations appear exactly once. Their corresponding columns in E also have all level combinations appearing once since there is a one-to-one mapping over $GF(p)$. Therefore, $\rho_{ik}(E) = 0$.

(ii) All columns in the same group, plus a zero vector, form p $p \times (p-1)$ GLP designs, as argued in Case 3 of the proof of Theorem 1. Also, the transformation (3.1) can be regarded as a special case of the linearly permuted Williams transformation. Then by Theorem 5 of [19], for any R_t, $t = 1, \ldots, p+1$,

$$\sum_{i \neq k \in R_t} |\rho_{ik}(E)| < 2(p-1).$$

In summary,

$$\rho_{\text{ave}}(E) = \frac{\sum_{i\neq k}|\rho_{ik}(E)|}{(p^2-1)(p^2-2)} = \frac{\sum_{t=1}^{p+1}\sum_{i\neq k\in R_t}|\rho_{ik}(E)|}{(p^2-1)(p^2-2)} < \frac{2(p-1)(p+1)}{(p^2-1)(p^2-2)}$$
$$= \frac{2}{p^2-2}. \tag{A.2}$$

Next, we consider $\rho_{\text{ave}}(L)$. We will prove that

$$\sum_{i\neq k}|\rho_{ik}(L)| \leq \frac{(p+1)^2}{p^2+1}\sum_{i\neq k}|\rho_{ik}(E)|. \tag{A.3}$$

We regroup the summation of column-wise correlations on the left-hand side of (A.3) as

$$\sum_{i\neq k}|\rho_{ik}(L)| = \sum_{t\neq s}\{|\rho_{(2t-1)(2s-1)}(L)| + |\rho_{(2t)(2s-1)}(L)| + |\rho_{(2t-1)(2s)}(L)|$$
$$+ |\rho_{(2t)(2s)}(L)|\} + 2\sum_{t=1}^{(p^2-1)/2}|\rho_{(2t-1)(2t)}(L)|,$$

where $i,k \in \{1,2,\dots,p^2-1\}$ and $t,s\in\{1,2,\dots,(p^2-1)/2\}$. Since $E=(e_{ij})$ is a balanced design and $L=(l_{ij})$ is an LHD, $\sum_{j=1}^{p^2} e_{jk} = \sum_{j=1}^{p^2} l_{jk} = 0$, and $\sum_{j=1}^{p^2} e_{jk}^2 = c_0$ for every k, where c_0 is a constant. For any t, columns $2t-1$ and $2t$ of E are always combinatorially orthogonal; therefore, after rotation, they are still orthogonal and $\rho_{(2t-1)(2t)}(L)=0$. In addition, we have

$$|\rho_{(2t-1)(2s-1)}(L)|$$
$$= \frac{|\sum_j(pe_{j(2t-1)}+e_{j(2t)})(pe_{j(2s-1)}+e_{j(2s)})|}{\sqrt{\sum_j(pe_{j(2t-1)}+e_{j(2t)})^2\sum_j(pe_{j(2s-1)}+e_{j(2s)})^2}}$$
$$= \frac{1}{(p^2+1)c_0}\Big|\sum_j\{p^2e_{j(2t-1)}e_{j(2s-1)} + pe_{j(2t-1)}e_{j(2s)} + pe_{j(2t)}e_{j(2s-1)}$$
$$+ e_{j(2t)}e_{j(2s)}\}\Big|$$
$$\leq \frac{1}{(p^2+1)c_0}\left\{p^2\Big|\sum_j e_{j(2t-1)}e_{j(2s-1)}\Big| + p\Big|\sum_j e_{j(2t-1)}e_{j(2s)}\Big|\right.$$
$$\left. + p\Big|\sum_j e_{j(2t)}e_{j(2s-1)}\Big| + \Big|\sum_j e_{j(2t)}e_{j(2s)}\Big|\right\}$$
$$= \frac{1}{p^2+1}\{p^2|\rho_{(2t-1)(2s-1)}(E)| + p|\rho_{(2t-1)(2s)}(E)| + p|\rho_{(2t)(2s-1)}(E)|$$
$$+ |\rho_{(2t)(2s)}(E)|\}.$$

Similarly,

$$
\begin{aligned}
|\rho_{(2t)(2s-1)}(L)| \le \frac{1}{p^2+1}\big\{&p|\rho_{(2t-1)(2s-1)}(E)| + |\rho_{(2t-1)(2s)}(E)| \\
&+ p^2|\rho_{(2t)(2s-1)}(E)| + p|\rho_{(2t)(2s)}(E)|\big\}, \\
|\rho_{(2t-1)(2s)}(L)| \le \frac{1}{p^2+1}\big\{&p|\rho_{(2t-1)(2s-1)}(E)| + p^2|\rho_{(2t-1)(2s)}(E)| \\
&+ |\rho_{(2t)(2s-1)}(E)| + p|\rho_{(2t)(2s)}(E)|\big\}, \\
|\rho_{(2t)(2s)}(L)| \le \frac{1}{p^2+1}\big\{&|\rho_{(2t-1)(2s-1)}(E)| + p|\rho_{(2t-1)(2s)}(E)| \\
&+ p|\rho_{(2t)(2s-1)}(E)| + p^2|\rho_{(2t)(2s)}(E)|\big\}.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
\sum_{i\neq k}|\rho_{ik}(L)| &= \sum_{t\neq s}\big\{|\rho_{(2t-1)(2s-1)}(L)| + |\rho_{(2t)(2s-1)}(L)| + |\rho_{(2t-1)(2s)}(L)| \\
&\quad + |\rho_{(2t)(2s)}(L)|\big\} \\
&\le \frac{(p+1)^2}{p^2+1}\sum_{t\neq s}\big\{|\rho_{(2t-1)(2s-1)}(E)| + |\rho_{(2t)(2s-1)}(E)| \\
&\quad + |\rho_{(2t-1)(2s)}(E)| + |\rho_{(2t)(2s)}(E)|\big\} \\
&= \frac{(p+1)^2}{p^2+1}\sum_{i\neq k}|\rho_{ik}(E)|,
\end{aligned}
$$

which completes the proof of (A.3). Then by (A.2), we have

$$
\rho_{\mathrm{ave}}(L) \le \frac{(p+1)^2}{p^2+1}\rho_{\mathrm{ave}}(E) \le \frac{(p+1)^2}{p^2+1}\frac{2}{p^2-2} < \left(1+\frac{2}{p}\right)\frac{2}{p^2-2}. \tag{A.4}
$$

Lastly, we consider $\rho_{\mathrm{ave}}(L^*)$. Denote $L^* = (l^*_{ij})$. Then $\sum_{j=1}^{p^2-1} l^*_{jk} = \sum_{j=1}^{p^2} l_{jk} = 0$ for any k. Moreover,

$$
\sum_{j=1}^{p^2-1}(l^*_{jk})^2 = 2\left[\left(\frac{1}{2}\right)^2 + \left(\frac{3}{2}\right)^2 + \cdots + \left(\frac{p^2-2}{2}\right)^2\right] = \frac{p^2(p^2-1)(p^2-2)}{12}, \tag{A.5}
$$

and

$$
\sum_{j=1}^{p^2}(l_{jk})^2 = 2\left[1^2 + 2^2 + \cdots + \left(\frac{p^2-1}{2}\right)^2\right] = \frac{p^2(p^2-1)(p^2+1)}{12}. \tag{A.6}
$$

Then

$$
|\rho_{ik}(L^*)| = \frac{\left|\sum_j l^*_{ji}l^*_{jk}\right|}{\sqrt{\sum_j (l^*_{ji})^2\sum_j (l^*_{jk})^2}} = \left|\sum_j l^*_{ji}l^*_{jk}\right| \Big/ \left(\frac{p^2(p^2-1)(p^2-2)}{12}\right), \tag{A.7}
$$

where

$$
\begin{aligned}
&|\sum_j l^*_{ji} l^*_{jk}| \\
&= \Bigg| \sum_{j:\, l_{ji}>0,\, l_{jk}>0} \left(l_{ji} - \frac{1}{2}\right)\left(l_{jk} - \frac{1}{2}\right) + \sum_{j:\, l_{ji}>0,\, l_{jk}<0} \left(l_{ji} - \frac{1}{2}\right)\left(l_{jk} + \frac{1}{2}\right) \\
&\qquad + \sum_{j:\, l_{ji}<0,\, l_{jk}>0} \left(l_{ji} + \frac{1}{2}\right)\left(l_{jk} - \frac{1}{2}\right) + \sum_{j:\, l_{ji}<0,\, l_{jk}<0} \left(l_{ji} + \frac{1}{2}\right)\left(l_{jk} + \frac{1}{2}\right)\Bigg| \\
&= \Bigg| \sum_j l_{ji} l_{jk} - \frac{1}{2} \sum_{j:\, l_{ji}>0,\, l_{jk}>0} (l_{ji} + l_{jk}) + \frac{1}{2} \sum_{j:\, l_{ji}<0,\, l_{jk}<0} (l_{ji} + l_{jk}) \\
&\quad + \frac{1}{2} \sum_{j:\, l_{ji}>0,\, l_{jk}<0} (l_{ji} - l_{jk}) - \frac{1}{2} \sum_{j:\, l_{ji}<0,\, l_{jk}>0} (l_{ji} - l_{jk}) + \sum_{j:\, l_{ji}l_{jk}>0} \frac{1}{4} - \sum_{j:\, l_{ji}l_{jk}<0} \frac{1}{4}\Bigg| \\
&\le |\sum_j l_{ji} l_{jk}| + \frac{1}{2} \sum_j \{|l_{ji}| + |l_{jk}|\} + \sum_j \frac{1}{4} \\
&= |\sum_j l_{ji} l_{jk}| + \frac{(p^2-1)(p^2+1)}{4} + \frac{p^2-1}{4}.
\end{aligned}
$$

Therefore, by (A.5), (A.6) and (A.7),

$$
|\rho_{ik}(L^*)| \le \frac{p^2+1}{p^2-2} |\rho_{ik}(L)| + \frac{3(p^2+2)}{p^2(p^2-2)} < 2|\rho_{ik}(L)| + \frac{6}{p^2-2},
$$

and

$$
\rho_{\text{ave}}(L^*) = \frac{\sum_{i \ne k} |\rho_{ik}(L^*)|}{(p^2-1)(p^2-2)} < \frac{\sum_{i \ne k} \left\{2|\rho_{ik}(L)| + \frac{6}{p^2-2}\right\}}{(p^2-1)(p^2-2)} = 2\rho_{\text{ave}}(L) + \frac{6}{p^2-2}.
$$

By (A.4), we have

$$
\rho_{\text{ave}}(L^*) < 2\left(1 + \frac{2}{p}\right)\frac{2}{p^2-2} + \frac{6}{p^2-2} = \left(10 + \frac{8}{p}\right)\frac{1}{p^2-2}. \qquad \square
$$

Proof of Theorem 5. As explained in Case 3 of the proof of Theorem 1, the transformation (3.1) can be regarded as a special case of the linearly permuted Williams transformation. When $k = 1$, L in Step 3 of Algorithm 2 is a special case of the E_b in Theorem 1 of [19] with $b = (p-1)/4$ if $p = 1 \pmod 4$ or $b = (3p-1)/4$ if $p = 3 \pmod 4$. Therefore, $d(L) \ge (p^2-1)/3$ and $d(L^*) \ge d(L) - (p-1) \ge (p^2-1)/3 - (p-1)$. $\square$

Proof of Theorem 6. We have $d(F) \geq d(D) - 2nk_r - (N-1)k_c$. Thus,

$$\begin{aligned} d_{\text{eff}}(F) &= \frac{d(F)}{\lfloor (N-2k_r+1)(n-k_c)/3 \rfloor} \\ &\geq \frac{d(D) - 2nk_r - (N-1)k_c}{(N+1)n/3} \\ &= \frac{d(D)}{(N+1)n/3} - \frac{6k_r}{N+1} - \frac{3(N-1)k_c}{(N+1)n} \\ &\geq \frac{d_{\text{eff}}(D)[(N+1)n-2]/3}{(N+1)n/3} - \frac{6k_r}{N+1} - \frac{3k_c}{n} \\ &= d_{\text{eff}}(D) - \frac{6k_r}{N+1} - \frac{3k_c}{n} - \frac{2}{(N+1)n}. \end{aligned}$$

□

Proof of Theorem 7. We have $d(D_1) = \lfloor (N+1)n_1/3 \rfloor d_{\text{eff}}(D_1) \geq [(N+1)n_1 - 2]d_{\text{eff}}(D_1)/3$ and $d(D_2) \geq [(N+1)n_2 - 2]d_{\text{eff}}(D_2)/3$. It is obvious that $d(F) \geq d(D_1) + d(D_2)$. Thus,

$$\begin{aligned} d_{\text{eff}}(F) &= \frac{d(F)}{\lfloor (N+1)(n_1+n_2)/3 \rfloor} \geq \frac{d(F)}{(N+1)(n_1+n_2)/3} \\ &\geq \frac{n_1 d_{\text{eff}}(D_1) + n_2 d_{\text{eff}}(D_2)}{n_1+n_2} - \frac{2d_{\text{eff}}(D_1) + 2d_{\text{eff}}(D_2)}{(N+1)(n_1+n_2)} \\ &\geq \frac{n_1 d_{\text{eff}}(D_1) + n_2 d_{\text{eff}}(D_2)}{n_1+n_2} - \frac{4}{(N+1)(n_1+n_2)}. \end{aligned}$$

□

Proof of Theorem 8. Because N is odd, the mirror-symmetric design $D = (x_{ij})_{N\times n}$ has a row of all zeros. Without loss of generality, we assume the first row consists of all zeros. As design D is balanced, all levels in $\{-(s-1)/2, -(s-3)/2, \ldots, (s-3)/2, (s-1)/2\}$ appear exactly N/s times in each column of D. The total L_1-distance between the first row (the row of all zeros) and all other rows is

$$\begin{aligned} \sum_{i=2}^{N} d_{1i}(D) &= \sum_{i=2}^{N}\sum_{j=1}^{n} |x_{1j} - x_{ij}| = \sum_{j=1}^{n}\sum_{i=2}^{N} |x_{ij}| \\ &= n \cdot 2\left(|1| + \cdots + \left|\frac{s-3}{2}\right| + \left|\frac{s-1}{2}\right|\right) \cdot \frac{N}{s} = \frac{N(s^2-1)n}{4s}. \end{aligned}$$

Because $d(D)$ cannot exceed the average distance between the first row and any other row, then

$$d(D) \leq \frac{N(s^2-1)n}{4s(N-1)}.$$

□

References

[1] Ba, S. (2015). SLHD: Maximin-Distance (Sliced) Latin Hypercube Designs. R package version 2.1-1.

[2] Ba, S. and Joseph, V. R. (2018). MaxPro: Maximum Projection Designs. R package version 4.1-2.

[3] Butler, N. A. (2001). Optimal and orthogonal Latin hypercube designs for computer experiments. *Biometrika* **88** 847–857. MR1859414

[4] Chen, R.-B., Hsieh, D.-N., Hung, Y. and Wang, W. (2013). Optimizing Latin hypercube designs by particle swarm. *Statistics and Computing* **23** 663–676. MR3094806

[5] Fang, K.-T., Li, R. and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Chapman and Hall/CRC. MR2510302

[6] Johnson, M. E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26** 131–148. MR1079258

[7] Joseph, V. R., Gul, E. and Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika* **102** 371–380. MR3371010

[8] Joseph, V. R. and Hung, Y. (2008). Orthogonal-maximin Latin hypercube designs. *Statistica Sinica* **18** 171–186. MR2416907

[9] Li, W., Liu, M.-Q. and Tang, B. (2021). A method of constructing maximin distance designs. *Biometrika* **108** 845–855. MR4341355

[10] McKay, M., Beckman, R. and Conover, W. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** 239–245. MR0533252

[11] Moon, H., Dean, A. and Santner, T. (2011). Algorithms for generating maximin Latin hypercube and orthogonal designs. *Journal of Statistical Theory and Practice* **5** 81–98. MR2829824

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

[13] Pang, F., Liu, M.-Q. and Lin, D. K. J. (2009). A construction method for orthogonal Latin hypercube designs with prime power levels. *Statistica Sinica* **19** 1721–1728. MR2589206

[14] Steinberg, D. M. and Lin, D. K. J. (2006). A construction method for orthogonal Latin hypercube designs. *Biometrika* **93** 279–288. MR2278083

[15] Sun, C.-Y. and Tang, B. (2023). Uniform projection designs and strong orthogonal arrays. *Journal of the American Statistical Association* **118** 417–423. MR4571131

[16] Sun, F., Wang, Y. and Xu, H. (2019). Uniform projection designs. *The Annals of Statisitcs* **47** 641–661. MR3909945

[17] Tang, Y. and Xu, H. (2014). Permuting regular fractional factorial designs for screening quantitative factors. *Biometrika* **101** 333–350. MR3215351

[18] Wang, L., Sun, F., Lin, D. K. J. and Liu, M.-Q. (2018). Construction of orthogonal symmetric Latin hypercube designs. *Statistica Sinica* **28** 1503–1520. MR3821015

[19] Wang, L., Xiao, Q. and Xu, H. (2018). Optimal maximin L_1-distance Latin hypercube designs based on good lattice point designs. *The Annals of Statistics* **46** 3741–3766. MR3852667

[20] Wang, L. and Xu, H. (2022). A class of multilevel nonregular designs for

studying quantitative factors. *Statistica Sinica* **32** 825–845. MR4405099

[21] Wang, L., Yang, J.-F., Lin, D. K. and Liu, M.-Q. (2015). Nearly orthogonal Latin hypercube designs for many design columns. *Statistica Sinica* **25** 1599–1612. MR3409083

[22] Xiao, Q. and Xu, H. (2017). Construction of maximin distance Latin squares and related Latin hypercube designs. *Biometrika* **104** 455–464. MR3698265

[23] Xiao, Q. and Xu, H. (2018). Construction of maximin distance designs via level permutation and expansion. *Statistica Sinica* **28** 1395–1414. MR3821010

[24] Ye, K. Q. (1998). Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association* **93** 1430–1439. MR1666638

[25] Ye, K. Q., Li, W. and Sudjianto, A. (2000). Algorithmic construction of optimal symmetric Latin hypercube designs. *Journal of Statistical Planning and Inference* **90** 145–159. MR1791586

[26] Zhang, A., Li, H., Quan, S. and Yang, Z. (2018). UniDOE: Uniform Design of Experiments. R package version 1.0.2.

[27] Zhang, X.-R., Liu, M.-Q. and Zhou, Y.-D. (2020). Orthogonal uniform composite designs. *Journal of Statistical Planning and Inference* **206** 100–110. MR4036698

[28] Zhou, W., Yang, J.-F. and Liu, M.-Q. (2020). Optimal maximin L_2-distance Latin hypercube designs. *Journal of Statistical Planning and Inference* **207** 113–122. MR4066128

[29] Zhou, Y. and Xu, H. (2015). Space-filling properties of good lattice point sets. *Biometrika* **102** 959–966. MR3431565