Optimal maximin $L_2$-distance Latin hypercube designs

Weiping Zhou, Jian-Feng Yang, Min-Qian Liu *

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

ARTICLE INFO

Article history:
Received 14 March 2019
Received in revised form 14 September 2019
Accepted 22 November 2019
Available online xxxx

MSC:
primary 62K05
secondary 62K15

Keywords:
Computer experiment
Orthogonality
Rotation method
Space-filling

ABSTRACT

Maximin distance Latin hypercube designs (LHDs) are extensively applied in computer experiments, but it is challenging to construct such designs. In this paper, based on a $2^2$ full factorial design and a series of saturated two-level regular designs, a number of maximin distance LHDs are constructed via the rotation method. Some of the constructed LHDs are exactly optimal and the others are asymptotically optimal under the maximin $L_2$-distance criterion. The constructed maximin distance LHDs have two prominent advantages: (i) no computer search is needed; and (ii) they are orthogonal or nearly orthogonal. Detailed comparisons with existing LHDs show that the constructed LHDs have larger minimum distances between design points.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

In computer experiments, complex systems are increasingly investigated through space-filling designs, which aim to distribute the design points over the design space as evenly as possible. Latin hypercube designs (LHDs), first introduced by McKay et al. (1979), are used as a popular class of space-filling designs. As we know, LHDs achieve one-dimensional space-filling property. One disadvantage of LHDs is that any such design is not necessarily space-filling in the full-dimensional space. To solve this problem, maximin distance criterion (Johnson et al., 1990) was proposed for constructing good LHDs. The maximin distance criterion is to maximize the minimum distance between design points, which guarantees the good space-filling property in the full-dimensional space. The maximin distance designs are asymptotically optimal for fitting Gaussian process models under a Bayesian setting (Johnson et al., 1990), and the maximin distance LHDs are well-suited for computer experiments (Lin and Tang, 2015).

There are many algorithms for constructing maximin distance LHDs, such as the simulated annealing (Morris and Mitchell, 1995; Joseph and Hung, 2008; Ba et al., 2015), swarm optimization algorithms (Moon et al., 2011; Chen et al., 2013) and the threshold-accepting method (Xiao and Xu, 2018). However, due to the computational complexity, these methods are not suitable to construct large LHDs which are needed in computer experiments (see for example, Morris, 1991; Kleijnen, 1997; Cioppa and Lucas, 2007; Gramacy et al., 2015). In order to overcome the challenges for constructing large LHDs, Zhou and Xu (2015) considered linear permutations to construct maximin $L_1$- and $L_2$-distance LHDs based on good lattice point sets; Xiao and Xu (2017) constructed LHDs with large minimum $L_1$-distance via Costas arrays; Wang et al. (2018) employed the Williams transformation to construct optimal maximin $L_1$-distance LHDs.

The rotation method, firstly presented by Beattie and Lin (2004, 2005), is simple and useful for constructing designs for computer experiments. This method was further employed to construct orthogonal LHDs, see e.g., Steinberg and Lin (2006), Lin et al. (2009), Pang et al. (2009), Sun and Tang (2017), and Wang et al. (2018a), among others. In this paper,
by combining the rotation method and the doubling operator of a design (Chen and Cheng, 2006), we propose several methods to construct maximin $L_2$-distance LHDs without any computer search. Firstly, based on a $2^d$ full factorial design, a class of asymptotically optimal maximin $L_2$-distance LHDs are constructed via the rotation method. Moreover, we show that these LHDs are orthogonal. Next, based on a series of saturated two-level regular designs, a good deal of maximin $L_2$-distance LHDs are constructed via the rotation method. Some of these LHDs are exactly optimal and the others are asymptotically optimal under the maximin $L_2$-distance criterion. Furthermore, the average correlations of these LHDs converge to zero as the design sizes increase, which is desirable for Gaussian process with linear trend (Wang et al., 2018a,b).

The rest of this paper is organized as follows. Section 2 provides relevant notation and definitions. Section 3 presents the construction methods, along with some discussions of asymptotic properties for the $L_2$-distance efficiency of the resulting designs. Section 4 discusses several convergence properties of the average correlations for the resulting designs. Section 5 provides some concluding remarks. All proofs are deferred to Appendix.

2. Preliminaries

Throughout, $J_{N \times n}$ is an $N \times n$ matrix of ones and $1_k$ is a $k \times 1$ vector of ones. Let $\lfloor x \rfloor$ denote the integer part of $x$. Let $D(N, s^t)$ denote a design with $N$ runs, $n$ factors, and $s$ levels, where each level occurs equally often in each factor. In this paper, an $N \times n$ matrix $L = (L_{ij})$ is called a Latin hypercube design (LHD), denoted by $L(N, n)$, when each column is a permutation of $-(N - 1)/2, -(N - 3)/2, \ldots, (N - 3)/2, (N - 1)/2$.

For any $N \times n$ design $D = (x_{ij})$, let $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ be the $i$th row of $D$. For any $N \times n$ LHD $D$, $d(D) \leq d(D)$.

From Lemma 1, for any $N \times n$ LHD $D$, define

$$d_{\text{eff}}(D) = d(D)/d(D) = d(D)\frac{N(N + 1)n/6}{n(n - 1)}$$

as the $L_2$-distance efficiency of $D$. For any $N \times n$ design $D = (x_{ij})$, define

$$\rho_{jk}(D) = \frac{(\phi_{k-1}(D) - \phi_{k-1}(D))}{(\phi_{k-1}(D) - \phi_{k-1}(D)) + 1}$$

for $k \geq 1$, where $\phi_{k-1}(D) + 1$ is the matrix obtained by adding 1 (mod 2) to all the entries of $\phi_{k-1}(D)$. Let

$$R_{u0} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad R_{uu} = \begin{pmatrix} 2^{2^u} & 0 \\ 0 & 2^{2^u} \end{pmatrix},$$

for $u = 2, 3, \ldots, $ then define

$$R_{v0} = \begin{pmatrix} 2R_{v0} & -Q_{vu} \\ Q_{vu} & 2R_{00} \end{pmatrix} \quad \text{and} \quad R_{uv} = \begin{pmatrix} 2R_{uv} & -Q_{vu} \\ Q_{vu} & 2R_{uv} \end{pmatrix}$$

for $u = 2, 3, \ldots, $.

For $d = 2^u$, $u = 1, 2, \ldots$, if $D$ with entries from $[0, 1]$ is a $2^d$ full factorial design, then $(\phi_{k}(D) - 1/2\mathbf{1}_{2^d+1}^T \mathbf{1}_{2^d+1})R_{kk}$ for $k \geq 1$ are the LHDs constructed by Sun and Tang (2017).

Lemma 2.

(i) For $d = 2^u$, $u = 1, 2, \ldots$, let $A$ be a $2^d$ full factorial design with entries from $[0, 1]$, and $A_k = \phi_k(A)$ for $k \geq 1$. If $x$ and $y$ are two rows of $A$, then

$$d((x - (1/2)\mathbf{1}_{2^d}^T)R_{00}, (y - (1/2)\mathbf{1}_{2^d}^T)R_{00}) = \frac{2^{2d} - 1}{3}d(x, y).$$
and if $\mathbf{x}^k$ and $\mathbf{y}^i$ are two rows of $A_k$, then
\[
d((\mathbf{x}^k - (1/2)R_{2d})_k, (\mathbf{y}^i - (1/2)R_{2d})_k) = \frac{2^{2(d+k)} - 1}{3} d(\mathbf{x}^k, \mathbf{y}^i).
\]

(ii) Let $E$ be an $N \times n$ matrix with entries from $\{0, 1\}$, and $E_k = \phi_k(E)$ for $k \geq 1$, then
\[
d(E_k) = 2^{k-1} \min\{2d(E), n\}.
\]

**Lemma 2** (i) tells us that the $L_2$-distance of the resulting design constructed via the rotation method is determined by that of the initial design, and **Lemma 2** (ii) shows that the $L_2$-distance of the large design $\phi_k(E)$ is determined by that of the small design $E$. These findings are important in calculating the maximin distances of the constructed designs in the following sections.

### 3. Construction methods

In this section, we propose several methods for constructing maximin $L_2$-distance LHDs without any computer search. The following lemma is useful for the construction.

**Lemma 3.** For $m = 1, 2, \ldots$, and $d = 2^u$ with $u = 1, 2, \ldots$, let $E_0 = (E_0^1, E_0^2, \ldots, E_0^m)$ be a $2^d \times md$ matrix, $F_0 = (F_0^1, F_0^2, \ldots, F_0^m)$ be a $2^d \times md$ matrix and $F_k = (F_k^1, F_k^2, \ldots, F_k^m)$ be a $2^{d+k} \times 2^k md$ matrix, where $E_0^i$ is a $2^d$ full factorial design with entries from $\{0, 1\}$, $F_0^i = (E_0^i - (1/2)R_{2d})_k$ and $F_k^i = (\phi_k(E_0^i) - (1/2)R_{2d})_k$ for $k \geq 1$. Then
\[
d(F_0) = \frac{2^{2d} - 1}{3} d(E_0), \quad d_{\text{eff}}(F_0) \geq \frac{2d(E_0)}{md}(1 - \frac{1}{2^d}),
\]
\[
d(F_k) = \frac{1}{3}(2^{2(d+k)} - 1)2^{k-1} \min\{2d(E_0), md\} \quad \text{and} \quad d_{\text{eff}}(F_k) \geq \alpha(1 - \frac{1}{2^{d+k}}) \quad \text{for} \quad k = 1, 2, \ldots,
\]
where
\[
\alpha = \begin{cases} \frac{2d(E_0)}{md}, & \text{if} \quad 2d(E_0) < md; \\ 1, & \text{if} \quad 2d(E_0) \geq md. \end{cases}
\]

It is worth noting that the choices of $m$ in **Lemma 3** are very broad. This makes it possible to generate many supersaturated LHDs. Obviously, the constructed LHDs $F_k$ for $k \geq 0$ are supersaturated if $2^d \leq md$ in **Lemma 3**. The following example is an illustration for **Lemma 3**.

**Example 1.** Consider $m = 1, 2, 3$ and $d = 2$. Let $b_1 = (0, 0, 1, 1)^T$, $b_2 = (0, 1, 0, 1)^T$, $b_3 = (0, 1, 1, 0)^T$. Then $(b_1, b_2, b_3)$ form a saturated $2^2$-1 Latin hypercube design. For $E_0$, $F_0$ and $F_1$ in **Lemma 3**, it can be calculated that (i) for $m = 1$, if $E_0 = (b_1, b_2)$, then $d(E_0) = 0$, $d(F_0) = 5$ and $d(F_1) = 42$; (ii) for $m = 2$, if $E_0 = (b_1, b_2, b_3, b_1)$, then $d(E_0) = 2, d(F_0) = 4, d(F_0) = 10$ and $d(F_1) = 84$; (iii) for $m = 3$, if $E_0 = (b_1, b_2, b_3, b_1, b_2, b_3)$, then $d(E_0) = 4, 2d(E_0) = 8 > 6, d(F_0) = 20$ and $d(F_1) = 126$, which all satisfy **Lemma 3**. The LHDs $F_0$’s and $F_1$’s are listed in Tables 1 and 2, respectively.

**Lemma 3** shows that, for $i = 0, 1, \ldots$, the $L_2$-distance of $F_i$ is determined by $E_0$, which means that $F_i$ may be a good design when we choose $E_0$ with the largest $L_2$-distance. From **Lemma 3**, we can obtain that if $2d(E_0) \geq md$, then $d_{\text{eff}}(F_k)$ converges to one as $k$ tends to infinity; so $F_k$ is asymptotically optimal under the maximin distance criterion. If $A$ is a $2^2$ full factorial design, then $d(A) = 1$ which attains the upper bound of $L_2$-distance in **Lemma 1**(i). Let $E_0 = A$, we can obtain the following result.

**Theorem 1.** Suppose $A$ is a $2^2$ full factorial design with entries from $\{0, 1\}$. For $k \geq 0$, let $L_k = (\psi_k(A) - (1/2)J_{2^{k+2}})_{2^{k+2}}$ be a $2^{k+2} \times 2^{k+1}$ matrix. Then $d(L_k) = 2^k(2^{k+2} - 1)/3$ and $d_{\text{eff}}(L_k) \geq 1 - 1/2^{k+2}$.

**Theorem 1** implies that $d(L_0) = 5$ and $d_{\text{eff}}(L_0) = 5/6$. Also, it is easy to see from **Theorem 1** that $d_{\text{eff}}(L_k)$ converges to one as $k$ tends to infinity. So $L_k$ is asymptotically optimal under the maximin distance criterion. **Table 3** compares the $L_2$-distances of $L_k$ for $k = 0, 1, \ldots, 10$ with that of the LHDs generated by the command `maximinSLHD` in the R package.
Table 2
The LHDs $F_i$’s in Example 1.

<table>
<thead>
<tr>
<th>$m = 1$</th>
<th>$m = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-3.5$</td>
<td>$-0.5$</td>
</tr>
<tr>
<td>$-1.5$</td>
<td>$2.5$</td>
</tr>
<tr>
<td>$1.5$</td>
<td>$-2.5$</td>
</tr>
<tr>
<td>$3.5$</td>
<td>$0.5$</td>
</tr>
<tr>
<td>$-2.5$</td>
<td>$-1.5$</td>
</tr>
<tr>
<td>$-0.5$</td>
<td>$3.5$</td>
</tr>
<tr>
<td>$0.5$</td>
<td>$-3.5$</td>
</tr>
<tr>
<td>$2.5$</td>
<td>$1.5$</td>
</tr>
</tbody>
</table>

$m = 3$

| $-3.5$ | $-0.5$ | $-2.5$ | $-1.5$ | $-3.5$ | $-0.5$ | $-2.5$ | $-1.5$ | $-3.5$ | $-0.5$ | $-2.5$ | $-1.5$ |
| $-1.5$ | $2.5$  | $-0.5$ | $3.5$  | $-1.5$ | $2.5$  | $-0.5$ | $3.5$  | $1.5$  | $-2.5$ | $0.5$  | $1.5$  |
| $1.5$  | $-2.5$ | $0.5$  | $-3.5$ | $1.5$  | $-2.5$ | $0.5$  | $-3.5$ | $3.5$  | $0.5$  | $2.5$  | $1.5$  |
| $3.5$  | $0.5$  | $2.5$  | $1.5$  | $3.5$  | $0.5$  | $2.5$  | $1.5$  | $-1.5$ | $2.5$  | $0.5$  | $-0.5$ |
| $-2.5$ | $-1.5$ | $3.5$  | $0.5$  | $-2.5$ | $-1.5$ | $3.5$  | $0.5$  | $-2.5$ | $-1.5$ | $3.5$  | $0.5$  |
| $-0.5$ | $3.5$  | $1.5$  | $-2.5$ | $-0.5$ | $3.5$  | $1.5$  | $-2.5$ | $0.5$  | $-3.5$ | $-1.5$ | $2.5$  |
| $0.5$  | $-3.5$ | $-1.5$ | $2.5$  | $0.5$  | $-3.5$ | $-1.5$ | $2.5$  | $2.5$  | $1.5$  | $-3.5$ | $-0.5$ |
| $2.5$  | $1.5$  | $-3.5$ | $-0.5$ | $3.5$  | $1.5$  | $-2.5$ | $0.5$  | $-3.5$ | $3.5$  | $1.5$  | $-2.5$ |

Table 3
Comparison of the $L_2$-distances for $2^{k+1} 	imes 2^{k+1}$ LHDs with $k \leq 10$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$L_k$</th>
<th>SLHD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Median</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>42</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>340</td>
<td>236</td>
</tr>
<tr>
<td>3</td>
<td>2728</td>
<td>1983</td>
</tr>
<tr>
<td>4</td>
<td>21840</td>
<td>16881</td>
</tr>
<tr>
<td>5</td>
<td>174752</td>
<td>141884</td>
</tr>
<tr>
<td>6</td>
<td>1398080</td>
<td>121858</td>
</tr>
<tr>
<td>7</td>
<td>11184768</td>
<td>9782026</td>
</tr>
<tr>
<td>8</td>
<td>89478400</td>
<td>67326353</td>
</tr>
<tr>
<td>9</td>
<td>715827712</td>
<td>58969264</td>
</tr>
<tr>
<td>10</td>
<td>5726622720</td>
<td>491137878</td>
</tr>
</tbody>
</table>

Note: $L_k$: constructed by Theorem 1; SLHD: constructed by the R package SLHD.

The distance efficiencies

Fig. 1. Design 1: Theorem 1; Design 2: Algorithm 1 for $b = 2$; Design 3: Algorithm 1 for $b = 3$.

SLHD provided by Ba et al. (2015). Here, we ran the command repeatedly 100 times. From Table 3, when $k \geq 2$, $L_k$ is better than SLHD under the maximin distance criterion. In Fig. 1, “Design 1” shows the values of $d_{\text{eff}}(L_k)$ for the $L_k$ constructed.
by Theorem 1, where \( k = 0, 1, \ldots, 10 \). The \( d_{\text{eff}}(L_k) \) increases fast as \( k \) increases and is greater than 0.9 when \( k = 2 \). When \( k \geq 3 \), the \( d_{\text{eff}}(L_k) \) values are far greater than 0.95 and converge to 1.

It is worth noting that the constructed designs in Theorem 1 have the same \( L_2 \)-distances with the designs constructed by Sun et al. (2009). Furthermore, by noting the existence of the mirror-symmetric structure, these designs can be shown to be optimal under the maximin \( L_2 \)-distance criterion (Wang et al., 2018c).

Let \( GF(2) = \{0, 1\} \) and \( GF(2^d) = \{a_0 + a_1 x + \cdots + a_{d-1} x^{d-1}, a_i \in GF(2)\} \). It is worth noting that there exists a primitive polynomial \( f(x) \) of degree \( d \) in \( GF(2) \) such that each nonzero element of \( GF(2^d) \) can be expressed as \( x^k \) modulo \( f(x) \) in \( GF(2^d) \) for \( k \in \{0, 1, \ldots, 2^d - 2\} \). Let \( \mathbf{1}, \mathbf{2}, \ldots, \mathbf{d} \) denote the \( d \) columns of a \( 2^d \) full factorial design. Each column, or a generated column, of \( \mathbf{1}, \mathbf{2}, \ldots, \mathbf{d} \), can be expressed by \( \mathbf{1}^{0} \mathbf{2}^{1} \mathbf{3}^{2 \cdot 1} \mathbf{d}^{d-1} \) for some \( a_i \in GF(2) \) and corresponds to a nonzero element \( a_0 + a_1 x + \cdots + a_{d-1} x^{d-1} \) of \( GF(2^d) \). As indicated in Steinberg and Lin (2006), Pang et al. (2009) and Wang et al. (2018a), the corresponding columns of the nonzero elements of \( GF(2^d) \), \( x^0, x, \ldots, x^{2^d-2} \) modulo \( f(x) \), form a saturated two-level regular design, denoted by \( B \), and any \( d \) successive columns of \( B \) form a full factorial design. From Steinberg and Lin (2006), we have the following general result.

**Lemma 4** (Steinberg and Lin, 2006). For any \( t \geq 0 \), the corresponding columns of the nonzero elements of \( GF(2^d) \), \( x^t, x^{t+1}, \ldots, x^{t+d-1} \) modulo \( f(x) \), form a full factorial design.

For the \( B \) defined above, \( d(B) = 2^d-1 \) (Mukerjee and Wu, 1995), which attains the upper bound of \( L_2 \)-distance in Lemma 1(i). Lemmas 3 and 4 show that we can obtain optimal maximin \( L_2 \)-distance LHDs based on this \( B \). Next, we propose a new method for constructing maximin \( L_2 \)-distance LHDs.

**Algorithm 1.**

1. **Step 1.** Given \( d = 2^d \) for \( u = 1, 2, \ldots \), obtain a saturated two-level regular design \( B \) as defined above, where \( B \) is a \( 2^d \times (2^d - 1) \) matrix.

2. **Step 2.** Let \( q = \min \{g : g(2^d - 1) \mod d = 0, g = 1, 2, \ldots, d\} \). Let \( C = \mathbf{1}_q^T \otimes B \). Write \( C \) as \( C = (C_1, C_2, \ldots, C_u) \), where \( \lambda = q(2^d - 1)/d \) and \( C_i \) is a \( 2^d \) full factorial design.

3. **Step 3.** For \( b = 1, 2, \ldots, \lambda \), and \( k = 0, 1, \ldots \), let \( I_k^b = (D_1^b, D_2^b, \ldots, D_{k}^{b}) \) be a \( 2^{d+k} \times 2^{k}bd \) design, where \( D_k^b = (\varphi_k(b))-1(0, 1, 2, 3) \) for \( k \geq 0 \).

**Remark 1.** Lemma 4 ensures that in Algorithm 1 the matrix \( C \) can be divided into \( \lambda \) groups of full factorial designs, and \( b \leq \lambda \) ensures that there are no identical columns in \( I_k^b \) for \( k = 0, 1, \ldots \).

**Theorem 2.** Let \( d = 2^d \) for \( u = 1, 2, \ldots \). From Algorithm 1, we have that

1. For \( h_1 = 1, 2, \ldots, q-1 \), if \( bd = h_1(2^d - 1) + h_2 \) with \( 0 < h_2 < 2^d - 1 \), then
   \[
   d_{\text{eff}}(I_k^b) \geq 1 - \frac{h_2}{h_1(2^d - 1) + h_2}, \text{ and } d_{\text{eff}}(I_k^b) > 1 - \frac{h_2 + 2^{-(d+k)}(2^d - 1)h_1}{h_1(2^d - 1) + h_2} \text{ for } k \geq 1;
   \]
2. If \( bd = q(2^d - 1) \), i.e. \( b = \lambda \), then \( I_0^b \) is a maximin \( L_2 \)-distance LHD with \( d(I_0^b) = 2^{d^2-1}q/6 \), and
   \[
   d_{\text{eff}}(I_k^b) \geq 1 - \frac{1}{2^{d+k}} \text{ for } k \geq 1.
   \]

**Theorem 2** shows that Algorithm 1 offers exact maximin \( L_2 \)-distance LHDs, \( L(2^2, 2(2^2 - 1)), L(2^4, 4(2^4 - 1)), L(2^8, 8(2^8 - 1)) \). By noting that \( d_{\text{eff}}(I_k^b) \) converges to one as \( k \) tends to infinity, **Theorem 2** also shows that Algorithm 1 offers a class of asymptotically optimal maximin \( L_2 \)-distance LHDs.

**Example 2.** Let \( d = 2 \). For the primitive polynomial \( f(x) = x^2 + 1 \) over \( GF(2) \), we have \( x^0 = 1, x = x \) and \( x^2 = 1 + x \) over \( GF(2^d) \). Thus we can obtain a saturated \( 2^{d-1} \) regular design \( B = (b_1, b_2, b_3) \) with \( b_1 = (0, 0, 1) \), \( b_2 = (0, 1, 0) \) and \( b_3 = (1, 0, 0) \). From Algorithm 1, it is clear that \( q = 2 \), \( C = (b_1, b_2, b_3, b_1, b_3, b_2, b_3) \) and \( b = 1, 2, 3 \). According to Algorithm 1, we can obtain LHDs \( I_i^b \) for \( i = 2, 3 \) and \( k \geq 0 \). Tables 4 and 5 compare the \( L_2 \)-distances of \( I_k^b \) with that of the LHDs generated by the command \texttt{maximinSLHD} in R package SLHD provided by Ba et al. (2015). Here, we ran the command repeatedly 100 times. From Table 4, for \( k \geq 2 \), \( I_k^b \) is better than SLHD under the maximin distance criterion.

**Table 4**, \( d(I_k^b) = 20 \), which attains the upper bound of \( L_2 \)-distance in Lemma 1. For \( k \geq 3 \), \( L_k^1 \) is better than SLHD under the maximin distance criterion. In Fig. 1, “Design 2” and “Design 3” show the values of \( d_{\text{eff}}(I_k^1) \) and \( d_{\text{eff}}(I_k^2) \) respectively for \( k = 0, 1, \ldots, 10 \). It can be seen that both \( d_{\text{eff}}(I_k^1) \) and \( d_{\text{eff}}(I_k^2) \) increase fast as \( k \) increases and both are greater than 0.9 when \( k = 2 \). When \( k \geq 3 \), the \( d_{\text{eff}}(I_k^1) \) and \( d_{\text{eff}}(I_k^2) \) values are all far greater than 0.95 and converge to 1.

According to Theorem 1 and Algorithm 1, we can obtain a wealth of (asymptotically or exactly) optimal maximin \( L_2 \)-distance LHDs. Table 6 presents a collection of optimal maximin \( L_2 \)-distance LHDs of \( N \) runs and \( n \) factors with \( N \leq 128 \). In Table 6, the designs \( L(N, n) \) with \( n = N/2 \) and \( N = 4, 8, 16, 32, 64, 128 \) are constructed using **Theorem 1** and the others are constructed by **Algorithm 1**.
Table 4
Comparison of the $L_2$-distances for $2^{k+2} \times 2^{k+2}$ LHDs with $k \leq 10$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$L_k^2$</th>
<th>SLHD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Median</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>84</td>
<td>78</td>
</tr>
<tr>
<td>2</td>
<td>680</td>
<td>605</td>
</tr>
<tr>
<td>3</td>
<td>5 456</td>
<td>4 796</td>
</tr>
<tr>
<td>4</td>
<td>43 680</td>
<td>38 759</td>
</tr>
<tr>
<td>5</td>
<td>349 504</td>
<td>321 028</td>
</tr>
<tr>
<td>6</td>
<td>2 796 160</td>
<td>2 630 278</td>
</tr>
<tr>
<td>7</td>
<td>22 369 536</td>
<td>20 631 324</td>
</tr>
<tr>
<td>8</td>
<td>178 956 800</td>
<td>164 035 301</td>
</tr>
<tr>
<td>9</td>
<td>1 431 655 424</td>
<td>1 252 525 609</td>
</tr>
<tr>
<td>10</td>
<td>11 453 245 440</td>
<td>10 336 809 965</td>
</tr>
</tbody>
</table>

Note: $L_k^2$: constructed by Algorithm 1; SLHD: constructed by the R package SLHD.

Table 5
Comparison of the $L_3$-distances for $2^{k+2} \times (6 \times 2^k)$ LHDs with $k \leq 10$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$L_k^3$</th>
<th>SLHD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Median</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>126</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>1 020</td>
<td>948</td>
</tr>
<tr>
<td>3</td>
<td>8 184</td>
<td>7 644</td>
</tr>
<tr>
<td>4</td>
<td>65 520</td>
<td>63 168</td>
</tr>
<tr>
<td>5</td>
<td>524 256</td>
<td>508 802</td>
</tr>
<tr>
<td>6</td>
<td>4 194 240</td>
<td>4 013 245</td>
</tr>
<tr>
<td>7</td>
<td>33 554 304</td>
<td>31 613 152</td>
</tr>
<tr>
<td>8</td>
<td>268 435 200</td>
<td>228 219 327</td>
</tr>
<tr>
<td>9</td>
<td>2 147 483 136</td>
<td>1 921 440 435</td>
</tr>
<tr>
<td>10</td>
<td>17 179 868 160</td>
<td>15 856 028 901</td>
</tr>
</tbody>
</table>

Note: $L_k^3$: constructed by Algorithm 1; SLHD: constructed by the R package SLHD.

Table 6
Some optimal maximin $L_2$-distance LHDs of $N$ runs and $n$ factors with $N \leq 128$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$n$</th>
<th>$N$</th>
<th>$n$</th>
<th>$N$</th>
<th>$n$</th>
<th>$N$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>60</td>
<td>32</td>
<td>88</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>16</td>
<td>28</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>16</td>
<td>36</td>
<td>32</td>
<td>40</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>16</td>
<td>40</td>
<td>32</td>
<td>48</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>16</td>
<td>44</td>
<td>32</td>
<td>56</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>16</td>
<td>48</td>
<td>32</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>16</td>
<td>52</td>
<td>32</td>
<td>72</td>
<td>64</td>
<td>80</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>16</td>
<td>56</td>
<td>32</td>
<td>80</td>
<td>64</td>
<td>96</td>
</tr>
</tbody>
</table>

Note: $L_k^2$: constructed by Algorithm 1; SLHD: constructed by the R package SLHD.

4. Orthogonality of the resulting designs

The $\rho_{ave}(D)$ measures the overall orthogonality of $D$. The design $D$ with a small $\rho_{ave}(D)$ value is good for fitting the Gaussian process model with potential linear trend (Wang et al., 2018a,b). In this section, we consider the $\rho_{ave}$ values of the LHDs with large $L_2$-distances constructed via Theorem 1 and Algorithm 1.

**Proposition 1.** For the designs $L_k$ constructed in Theorem 1, we have $\rho_{ave}(L_k) = 0$ for $k \geq 0$.

**Proposition 1** shows that orthogonal LHDs with large $L_2$-distances can be directly generated via Theorem 1 without any computer search. For the LHDs constructed by Algorithm 1, we have the following result.

**Theorem 3.** Let $d = 2^u$ for $u = 1, 2, \ldots$. From Algorithm 1, we have that

\[
\rho_{ave}(L_k^2) \leq \frac{3 \times (2^{2d+k}-1)(2^{d-1}+1)^2}{bd(2^{2d-1}+1)} - \frac{1}{2^{kbd-1}} \quad \text{for } k \geq 0; \quad \text{and}
\]

\[
\rho_{ave}(L_k^3) \leq \frac{3 \times (2^{2d+k}-1)(2^{d-1}+1)^2}{bd(2^{2d-1}+1)} - \frac{1}{2^{kbd-1}} \quad \text{for } k \geq 0; \quad \text{and}
\]
Fig. 2. Design 1: Theorem 1; Design 2: Algorithm 1 for $b = 2$; Design 3: Algorithm 1 for $b = 3$.

(ii) if $bd = q(2^d - 1)$, i.e. $b = \lambda$, then

$$\rho_{\text{ave}}(L^b_k) \leq \frac{3 \times (2^{d+k} - 1)(2^d - 1)q^2}{bd(2^dbd - 1)(2^{d+k} + 1)} - \frac{1}{2^kbd - 1} \text{ for } k \geq 0.$$  

From Theorem 3, we can show that $\rho_{\text{ave}}(L^b_k)$ converges to zero as $k$ tends to infinity. Thus, a class of LHDs with large $L_2$-distances and small $\rho_{\text{ave}}$’s can be easily generated via Algorithm 1 without any computer search. In Fig. 2, “Design 1”, “Design 2” and “Design 3” show the values of $\rho_{\text{ave}}(L_k)$, $\rho_{\text{ave}}(L^2_k)$ and $\rho_{\text{ave}}(L^3_k)$ respectively, for $k = 0, 1, \ldots, 10$, where $L_k$ is constructed by Theorem 1; $L^2_k$ and $L^3_k$ are constructed by Algorithm 1. Both $\rho_{\text{ave}}(L^2_k)$ and $\rho_{\text{ave}}(L^3_k)$ decrease fast as $k$ increases and are less than 0.1 when $k$ is 2. When $k \geq 3$, the $\rho_{\text{ave}}(L^2_k)$ and $\rho_{\text{ave}}(L^3_k)$ values are far less than 0.05 and converge to 0.

5. Concluding remarks

In this paper, we propose some methods for constructing maximin $L_2$-distance LHDs via the rotation method. The methods do not need any computer search and are more efficient especially for constructing large designs. They can lead to a class of asymptotically optimal maximin $L_2$-distance LHDs and exactly optimal maximin $L_2$-distance LHDs. Furthermore, some resulting designs are orthogonal and the average correlations of the other designs converge to zero as the design sizes increase.

The rotation method used in this paper has two major drawbacks. The first one is the limitation on the run size, which must be the power of two. If one can relax the requirement to work with LHDs, an alternative is to rotate non-regular two-level designs to generate nearly LHDs with flexible run sizes (Steinberg and Lin, 2015). Such designs are still desirable for many practical situations (Bingham et al., 2009; Sun et al., 2011; Ding et al., 2013; Jaynes et al., 2013). The second one is the low coverage in low dimensional subspaces. The new factors in the resulting design naturally divide into pairs, where each pair has the two largest rotation weights on the same original factors in the two-level design. In the projection onto these two factors, all the design points concentrate in just a few cells of a coarser binary grid. In order to overcome this drawback, Steinberg and Lin (2015) recommended choosing just one factor from each of such pairs. This will improve the two-dimensional coverage a lot, although the optimality under the maximin distance criterion cannot be guaranteed any more.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 11771220, 11871033, 11431006, 11671386 and 11971204), National Ten Thousand Talents Program, China, Tianjin Development Program for Innovation and Entrepreneurship, China, Tianjin “131” Talents Program, China, and Project 61331903. The first two authors contributed equally to this work.
Appendix. Proofs

Proof of Lemma 2. (i) Let $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ be two rows of $A$. It is clear that
\[
d((x - (1/2)1_d)_{u_0}, (y - (1/2)1_d)_{u_0}) = (x - y)_{u_0}R_{u_0}^T_{u_0}(x - y)^T = \frac{2^{2d} - 1}{3}d(x, y),
\]
since $R_{u_0}R_{u_0}^T = (2^{d} - 1)/3I_d$. Similarly,
\[
d((x^k - (1/2)1_d)_{u_k}, (y^k - (1/2)1_d)_{u_k}) = \frac{2^{2(d+k)} - 1}{3}d(x^k, y^k),
\]
since $R_{u_k}R_{u_k}^T = (2^{2(d+k)} - 1)/3I_{2d}$. (ii) Let $x$ and $y$ be two rows of $E$, then $(x, x, (x, x + 1(\operatorname{mod} 2)), (y, y, (y, y + 1(\operatorname{mod} 2)))$ are four rows of $E_1$. It is clear that $d((x, x, (y, y) = d((x, x + 1(\operatorname{mod} 2)), (y, y + 1(\operatorname{mod} 2))) = 2d(x, y)$ and $d((x, x, (x, x + 1(\operatorname{mod} 2))) = d((y, y, (y, y + 1(\operatorname{mod} 2))) = d((x, x + 1(\operatorname{mod} 2)), (y, y)) = n$. Thus $d(E_1) = \min(2d(E), n)$. For $k > 1$, by recursion we can obtain $d(E_k) = 2^{k-1} \min(2d(E), n)$. This completes the proof.

Proof of Lemma 3. From Lemma 2, it is clear that Lemma 3 is true.

Proof of Theorem 1. The results just follow from Lemma 3.

Proof of Theorem 2. Let $C_k(b) = (\varphi_k(C_1), \ldots, \varphi_k(C_b))$ and $B_k = \varphi_k(B)$ with $k \geq 0$. It is clear from Lemma 1(ii) that $d_{upper} = (2^{d+k}(2^{d+k} + 1)2^{bd}/6)$ for the $L_k^b$ with $k \geq 0$.

(i) From Lemma 2(ii), it is known that
\[
d(C_k(b)), \geq h_1(d(B)) = 2^{d-1}h_1 \quad \text{and} \quad d(L_k(b)), = \frac{2^{d} - 1}{3}d(C_k(b)), \geq 2^{d-1} \frac{2^{d} - 1}{3}h_1.
\]
From Lemma 2(ii), $d(B_k) = 2^{k-1}(2^d - 1)$ for $k \geq 1$. We can obtain that
\[
d(L_k^b) = \frac{2^{2(d+k)} - 1}{3}d(C_k(b)) \geq \frac{(2^{2(d+k)} - 1)h_1}{d(B_k)},
\]
since $d(C_k(b)) \geq h_1(d(B_k))$ for $k \geq 1$. Thus Theorem 2(i) is true from (1).

(ii) If $bd = q(2^d - 1)$, then
\[
d(C_k(b)) = q(d(B)) = 2^{d-1}q \quad \text{and} \quad d(L_k(b)) = \frac{2^{d} - 1}{3}d(C_k(b)) = \frac{2^{k-1}(2^d - 1)q}{3}.
\]
From Lemma 1, it is known that $L_k^b$ is a maximin distance LHD. Furthermore, it is clear that for $k \geq 1$,
\[
d(L_k^b) = \frac{2^{2(d+k)} - 1}{3}d(C_k(b)) = \frac{(2^{2(d+k)} - 1)q}{d(B_k)} = \frac{2^{k-1}(2^d - 1)q}{3}.
\]
Thus Theorem 2(ii) is true from (1). This completes the proof.

Let $D = (x_j)$ be an $N \times n$ matrix, define $\operatorname{Sum}(D) = \sum_{j=1}^{N} \sum_{i=1}^{n} x_{ij}$, and $\operatorname{Abs}(D) = \{|x_j|\}$ where $|x_j|$ is the absolute value of $x_j$. To prove Theorem 3, the following lemma is crucial.

Lemma 5. In Algorithm 1, let $M_0(b) = (C_1 - (1/2)J_{2d \times d}, \ldots, C_b - (1/2)J_{2d \times d})$, $M_k(b) = (\varphi_k(C_1) - (1/2)J_{2^{d+k} \times 2^{d+k}}, \ldots, \varphi_k(C_b) - (1/2)J_{2^{d+k} \times 2^{d+k}})$ with $k \geq 1$. We have that

(i) for $h_1 = 1, 2, \ldots, q - 1$, if $bd = h_1(2^d - 1) + h_2$ with $0 < h_2 < 2^d - 1$, then
\[
\sum_{h_1} \operatorname{Abs}((M_k(b)) - (M_k(b))) \leq 4^{k-1}2(2^d - 1)(h_1 + 1)^2 \quad \text{for } k = 0, 1, \ldots;
\]

(ii) if $bd = q(2^d - 1)$, then $\sum_{h_1} \operatorname{Abs}((M_k(b)) - (M_k(b))) = 4^{k-1}2(2^d - 1)q^2$ for $k = 0, 1, \ldots$.

Proof of Lemma 5. For the $B$ in Algorithm 1, $(B - (1/2)J_{2d \times (2^d-1)})^T(B - (1/2)J_{2d \times (2^d-1)}) = 2^{d-2}I_{2d-1}$. For $k = 1, 2, \ldots, \text{let } M_k(b) = (M_k(b), \ldots, M_k(b))$ with $M_k(b) = \varphi_k(C_i) - (1/2)J_{2^{d+k} \times 2^{d+k}}$, then
\[
M_k(b) = \begin{pmatrix} M_k(b) \\ -M_k(b) \end{pmatrix}, \text{ and } (M_k(b)) = \begin{pmatrix} 2(M_k(b)) - (M_k(b)) \\ 0 \end{pmatrix}.
\]
Thus

\[ \text{Sum}(\text{Abs}(M_{k+1}^{(b_i)\top}M_k^{(b_j)})) = 4\text{Sum}(\text{Abs}((M_{k-1}^{(b_i)\top}M_k^{(b_j)})) \text{ and } \text{Sum}(\text{Abs}((M_k^{(b_i)\top}M_k^{(b_j)}))) = 4\text{Sum}(\text{Abs}((M_{k+1}^{(b_i)\top}M_{k-1}^{(b_j)}))). \]

Let \( M_0^{(b)} = (1^T_{h+1} \otimes B) - (1/2)J_{2^d \times (h_1+1)2^{d-1}} \).

(i) If \( bd = h_1(2^d - 1) + h_2 \) with \( 0 < h_2 < 2^d - 1 \), then for \( k \geq 1 \),

\[ \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})) \leq \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})) = 2^{d-2}(2^d - 1)(h_1 + 1)^2, \]

and

\[ \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})) = 4\text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})) \leq 4^{k-1}2^d(2^d - 1)(h_1 + 1)^2. \]

(ii) If \( bd = q(2^d - 1) \), then \( M_0^{(b)} = C - (1/2)J_{2^d \times q(2^d-1)} = (1/2)J_{2^d \times q(2^d-1)} \), and

\[ \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})) = 2^{d-2}(2^d - 1)q^2. \]

Thus

\[ \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})) = 4^{k-1}2^d(2^d - 1)q^2 \text{ for } k \geq 1. \]

This completes the proof. \( \square \)

**Proof of Theorem 3.** Let \( M_k^{(b)} = (M_k^{(b_1)}, \ldots, M_k^{(b_j)}) \) with \( M_0^{(b)} = C_i - (1/2)J_{2^d \times 2^d} \).

For \( i = 1, 2, \ldots, b, j = 1, 2, \ldots, b, \)

\[ \text{Sum}(\text{Abs}((D_0^T D_0^T D_0^T)) = \text{Sum}(\text{Abs}(R_0^T M_0^{(b)})^T M_0^{(b)})) \]

\[ \leq \left( \sum_{k=0}^{2^{d-1}2^d} \right)^2 \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})) \]

\[ = (2^d - 1)^2 \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})), \]

and

\[ \text{Sum}(\text{Abs}(L_0^{(b)})^T L_0^{(b)})) \leq (2^d - 1)^2 \text{Sum}(\text{Abs}((M_0^{(b)})^T M_0^{(b)})). \]

From Lemma 5, we have that for \( h_1 = 1, 2, \ldots, q - 1 \), if \( bd = h_1(2^d - 1) + h_2 \) with \( 0 < h_2 < 2^d - 1 \), then

\[ \text{Sum}(\text{Abs}(L_0^{(b)})^T L_0^{(b)}) \leq 2^{d-2}(2^d - 1)^2(h_1 + 1)^2; \]

and if \( bd = q(2^d - 1) \), then

\[ \text{Sum}(\text{Abs}(L_0^{(b)})^T L_0^{(b)}) \leq 2^{d-2}(2^d - 1)^2q^2. \]

It is clear that

\[ \rho_{ave}(L_0^{(b)}) = \frac{1}{bd(bd - 1)} - \frac{12}{2(2^d - 1)} \text{Sum}(\text{Abs}(L_0^{(b)})^T L_0^{(b)}) - bd) \]

Thus for \( h_1 = 1, 2, \ldots, q - 1 \), if \( bd = h_1(2^d - 1) + h_2 \) with \( 0 < h_2 < 2^d - 1 \), then

\[ \rho_{ave}(L_0^{(b)}) \leq \frac{3(2^d - 1)^2(h_1 + 1)^2}{bd(bd - 1)(2^d - 1)} - \frac{1}{bd - 1}; \]

if \( bd = q(2^d - 1) \), then

\[ \rho_{ave}(L_0^{(b)}) \leq \frac{3q(2^d - 1)}{(q(2^d - 1) - 1)(2^d + 1)} - \frac{1}{q(2^d - 1) - 1}; \]

For \( k = 1, 2, \ldots, (L_0^{(b)})^T L_0^{(b)} = \left( (D_k^{(b)})^T D_k^{(b)}) \right)_{i,j=1,2,\ldots,b,} \) where \( (D_k^{(b)})^T D_k^{(b)} = R_k^{(b)}(M_k^{(b)})^T M_k^{(b)} R_k \cdot \)

\[ (D_0^{(b)})^T D_k^{(b)} = \begin{pmatrix} \sum_{h=0}^{2^{d+1}2^d} 2^h \sum_{h=0}^{2^{d+1}2^d} 2^h \\ 8U_{k-1}^{(b)} + 2V_{k-1}^{(b)} \end{pmatrix}, \]

where \( U_{k-1}^{(b)} = R_{u k-1}^{(b)}(M_{k-1}^{(b)})^T M_{k-1}^{(b)} R_{u k-1} \), \( V_{k-1}^{(b)} = Q_{u k-1}^{(b)}(M_{k-1}^{(b)})^T M_{k-1}^{(b)} Q_{u k-1} \), and

\[ Z_{k-1}^{(b)} = R_{u k-1}^{(b)}(M_{k-1}^{(b)})^T M_{k-1}^{(b)} R_{u k-1} \].

We can obtain that

\[ \text{Sum}(\text{Abs}(U_{k-1}^{(b)})) \leq (2^{d+1}2^d - 1)^2 \text{Sum}(\text{Abs}((M_{k-1}^{(b)})^T M_{k-1}^{(b)})), \]

\[ \text{Sum}(\text{Abs}(V_{k-1}^{(b)})) \leq \text{Sum}(\text{Abs}((M_{k-1}^{(b)})^T M_{k-1}^{(b)})), \]

\[ \text{Sum}(\text{Abs}(W_{k-1}^{(b)})) \leq \text{Sum}(\text{Abs}((M_{k-1}^{(b)})^T M_{k-1}^{(b)})), \]

and

\[ \text{Sum}(\text{Abs}(Z_{k-1}^{(b)})) \leq \text{Sum}(\text{Abs}((M_{k-1}^{(b)})^T M_{k-1}^{(b)})). \]

Thus

\[ \text{Sum}(\text{Abs}((D_0^{(b)})^T D_k^{(b)})) \leq 4(2^{d+1}2^d - 1)^2 \text{Sum}(\text{Abs}((M_{k-1}^{(b)})^T M_{k-1}^{(b)})), \]

and

\[ \text{Sum}(\text{Abs}((L_0^{(b)})^T L_k^{(b)})) \leq 4(2^{d+1}2^d - 1)^2 \text{Sum}(\text{Abs}((M_{k-1}^{(b)})^T M_{k-1}^{(b)})). \]
It is clear that

\[
\rho_{\text{ave}}(L_k^h) \leq \frac{1}{2^k (2b - 1)} \left( \frac{12}{2^k + 1} \sum_{\mathbf{d} = 1}^{2^k} \left( \left( \sum_{\mathbf{L}^{(i)}_{k-1}} \mathbf{L}^{(i)}_{L_k^h} \right)^2 - 2^k b d \right) \right)
\]

From Lemma 5, it is known that for \( k \geq 1 \) and \( h_1 = 1, 2, \ldots, q - 1 \), if \( bd = h_1 (2^d - 1) + h_2 \) with \( 0 < h_2 < 2^d - 1 \), then

\[
\rho_{\text{ave}}(L_k^h) \leq \frac{1}{2^k (2b - 1)} \left( \frac{12 \times 4^{k-1} (2^d - 1)^2}{2^k (2^k + 1)} \right) - 2^k b d
\]

and if \( bd = q (2^d - 1) \), then

\[
\rho_{\text{ave}}(L_k^h) \leq \frac{1}{2^k (2b - 1)} \left( \frac{12 \times 4^{k-1} (2^d - 1)^2}{2^k (2^k + 1)} \right) - 2^k b d
\]

This completes the proof. \( \square \)

References