CONSTRUCTION OF MAXIMIN DISTANCE LATIN HYPERCUBE DESIGNS VIA GOOD LATTICE POINT SETS

Xueru Zhang[1], Dennis K.J. Lin[2] and Wei Zheng[3]

[1] *University of Science and Technology Beijing,* [2] *Purdue University and* [3] *University of Tennessee*

Supplementary Material

The online Supplementary Material includes detailed proofs, algorithms based on the leave-one-out additive column expansion, quantitative comparisons between the proposed method and existing methods based on criteria such as the L_p-distance, column-orthogonality, projection uniformity and computational costs.

S1. Proofs

S1.1 Proof of Theorem 1

For any two rows $\boldsymbol{x}_1 = (x_{11}, \ldots, x_{1k})$ and $\boldsymbol{x}_2 = (x_{21}, \ldots, x_{2k})$ of D_0, it is clear that $\mid x_{1j} \oplus 0 - x_{2j} \oplus 0 \mid, \ldots, \mid x_{1j} \oplus (n-1) - x_{2j} \oplus (n-1) \mid$ consists

of $\mid x_{1j} - x_{2j} \mid$ and $n - \mid x_{1j} - x_{2j} \mid$ with the frequencies $n - \mid x_{1j} - x_{2j} \mid$ and $\mid x_{1j} - x_{2j} \mid$, respectively. Note that there has a row $\boldsymbol{x}_3$ satisfying that $\mid x_{1j} - x_{2j} \mid = x_{3j}$ or $n - x_{3j}$. Therefore, we have

$$d_p(\mathcal{E}((\boldsymbol{x}_1; \boldsymbol{x}_2))) = \tau_{p,n}(\boldsymbol{x}_3) = \sum_{j=1}^{k} (n - x_{3j}) x_{3j}^p + x_{3j}(n - x_{3j})^p$$

and $d_p(\mathcal{E}(D_0)) = \min\{\tau_{p,n}(\boldsymbol{x}_i) : \boldsymbol{x}_i \neq (0, \ldots, 0), \boldsymbol{x}_i \in D_0\}$.

Let $p = 1$ and D_0 be an $n \times \psi(n)$ GLP set generated by $\boldsymbol{h} = (h_1, \ldots, h_{\psi(n)})$ with $h_1 = 1$. For the ith row of D_0, it is shown that $\boldsymbol{x}_i = i\boldsymbol{h} \pmod{n}$. If i is coprime with n, it holds that

$$\tau_{1,n}(\boldsymbol{x}_i) = \tau_{1,n}(\boldsymbol{x}_1) = \frac{\left(n^3 - (-1)^t n \prod_{\ell=1}^{t} q_\ell\right) \prod_{\ell=1}^{t} (q_\ell - 1)}{3 \prod_{\ell=1}^{t} q_\ell}.$$

Let $\mathcal{P}$ be a random permutation of t components. Denote $\bar{q}_1, \ldots, \bar{q}_t$ and $\bar{r}_1, \ldots, \bar{r}_t$ as the permutations of $q_1, \ldots, q_t$ and $r_1, \ldots, r_t$ under $\mathcal{P}$, respectively. Therefore, we have $n = q_1^{r_1} \cdots q_t^{r_t} = \bar{q}_1^{\bar{r}_1} \cdots \bar{q}_t^{\bar{r}_t}$. Except $i = n$, any i can be represented by $i = \bar{q}_1^{\bar{r}_1} \ldots \bar{q}_j^{\bar{r}_j} \bar{q}_{j+1}^{u_{j+1}} \ldots \bar{q}_t^{u_t}$ with $0 \leq j \leq t-1$, $0 \leq u_{j+1} \leq \bar{r}_{j+1} - 1, \ldots, 0 \leq u_t \leq \bar{r}_t - 1$. Then, it holds that

$$\begin{aligned}
\tau_{1,n}(\boldsymbol{x}_i) = & \quad \frac{i^3 \prod_{\ell=1}^{j} (\bar{q}_\ell - 1)\, \tau_{1,n/i}(\boldsymbol{y}_1)}{\prod_{\ell=1}^{j} \bar{q}_\ell} \\
= & \quad \frac{\left(n^3 - (-1)^{t-j} \left(\prod_{\ell=j}^{t} \bar{q}_{\ell+1}^{\bar{r}_{\ell+1}+2u_{\ell+1}+1}\right) \prod_{\ell=1}^{j} \bar{q}_\ell^{3\bar{r}_\ell}\right) \prod_{\ell=1}^{t} (\bar{q}_\ell - 1)}{3 \prod_{\ell=1}^{t} \bar{q}_\ell},
\end{aligned}$$

where $\boldsymbol{y}_1$ is the first row of $n/i \times \psi(n/i)$ GLP set, in other word, equivalent to its generating vector.

If $t = 1$, $\tau_{1,n}(\boldsymbol{x}_i)$ arrives the minimum, that is $(q_1 - 1)(n^3/q_1 + n)/3$, when $j = 0$ and $u_1 = 0$. Otherwise, $\tau_{1,n}(\boldsymbol{x}_i)$ arrives the minimum, that is $n^3(1 - 1/(q_1q_2))(1 - 1/q_1)\cdots(1 - 1/q_t)/3$, when $j = t - 2$, $\bar{q}_{t-1} = q_2$, $\bar{q}_t = q_1$, $u_{t-1} = \bar{r}_{t-1} - 1 = r_2 - 1$ and $u_t = \bar{r}_t - 1 = r_1 - 1$. Then, we obtain the $d_1(\mathcal{E}(D_0))$ in Theorem 1.

The similar idea can be applied to the L_2-distance for two cases and is omitted here for saving space.

S1.2 Proof of Lemma 3

Let $\boldsymbol{x}_1 = i\boldsymbol{h} \pmod n$ and $\boldsymbol{x}_2 = l\boldsymbol{h} \pmod n$ be any two different rows of D with $i \neq l$. For $u \in \mathcal{U}_0$, it holds that

$$|ih_j \oplus u - lh_j \oplus u| = |(n - i)(n - h_j) \oplus u - (n - l)(n - h_j) \oplus u|. \quad \text{(S1.1)}$$

Denote $\boldsymbol{y}_1 = (n - i)\boldsymbol{h}' \pmod n$ and $\boldsymbol{y}_2 = (n - l)\boldsymbol{h}' \pmod n$. Based on (S1.1) and $\mathcal{U} \subset \mathcal{U}_0$, we have

$$\begin{aligned} d_p(\boldsymbol{x}_1 \oplus \mathcal{U}, \boldsymbol{x}_2 \oplus \mathcal{U}) &= \sum_{s=1}^{v}\sum_{j=1}^{k} |ih_j \oplus u_s - lh_j \oplus u_s| \\ &= \sum_{s=1}^{v}\sum_{j=1}^{k} |(n - i)(n - h_j) \oplus u_s - (n - l)(n - h_j) \oplus u_s| \\ &= d_p(\boldsymbol{y}_1 \oplus \mathcal{U}, \boldsymbol{y}_2 \oplus \mathcal{U}). \end{aligned}$$

Therefore, for p, we have

$$d_p(\mathcal{E}(D,\mathcal{U})) = \min_{1\le i<l\le n} \{d_p(\boldsymbol{x}_1 \oplus \mathcal{U}, \boldsymbol{x}_2 \oplus \mathcal{U})\}$$

$$= \min_{1\le i<l\le n}\{d_p(\boldsymbol{y}_1 \oplus \mathcal{U}, \boldsymbol{y}_2 \oplus \mathcal{U})\} = d_p(\mathcal{E}(D', \mathcal{U})).$$

S1.3 Proof of Theorem 2

Let's prove the case (i). According to Lemma 3, we have $d_p(\mathcal{E}(D_1)) = d_p(\mathcal{E}(D_1'))$. For $D_0 = (x_{ij})$, it is indicated that $\tau_{p,n}(x_{ij}) = \tau_{p,n}(n - x_{ij})$. Therefore, we have $\tau_{p,n}(i\boldsymbol{h} \pmod n) = \tau_{p,n}(i\boldsymbol{h}' \pmod n) = \tau_{p,n}(i(\boldsymbol{h}, \boldsymbol{h}'))/2$ for any $i \in \{1, \ldots, n\}$. Consequently, it holds that $d_p(\mathcal{E}(D_1)) = d_p(\mathcal{E}(D_1')) = d_p(\mathcal{E}(D_0))/2$.

Let's prove the case (ii). According to the proof of Lemma 4.1 of Zhang et al. (2024), we can analogously prove the case (ii) and omit it here.

S1.4 Proof of Theorem 3

Let's prove the case (ii). Assume $p = 1$. For $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ in the proof of Theorem 1, it holds that $|x_{1l} \oplus u - x_{2l} \oplus u| \in \{x_{31}, \ldots, x_{3\psi(n)}\}$ for $l = 1, \ldots, \psi(n)$. Let $\Psi(\boldsymbol{x}_3)$ be all pairs (i, j)'s of rows with $|x_{il} \oplus u - x_{jl} \oplus u| \in \{x_{31}, \ldots, x_{3\psi(n)}\}$ for $l = 1, \ldots, \psi(n)$. We have $\sum_{(i,j)\in\Psi(\boldsymbol{x}_3)} |x_{il} - x_{il}| = (\#\Psi(\boldsymbol{x}_3))\tau_{1,n}(\boldsymbol{x}_3)/(n\psi(n))$ for $l = 1, \ldots, \psi(n)$. Let $\mathcal{D}$ be an arbitrary $n\psi(n)/2$-column combination of $\mathcal{E}(D_0)$. It holds that $d_1(\mathcal{D}) \le \min\{\tau_{1,n}(\boldsymbol{x}_i)/2, i = 1, \ldots, n-1\} = d_1(\mathcal{E}(D_1))$. Then, $\mathcal{E}(D_1)$ is optimal among all $n\psi(n)/2$-column combinations of $\mathcal{E}(D_0)$ under the L_1-distance.

The similar idea can be applied to the L_p-distance for $p > 1$ and is omitted here for saving space.

S1.5 Proof of Lemma 4

Let $n \geq a > b > 0$. It holds that

$$|a \oplus u - b \oplus u| = \begin{cases} a - b & \text{if } \ 0 \leq u \leq n - a \text{ or } n - b < u, \\ n - a + b & \text{if } \ n - a < u \leq n - b. \end{cases}$$

Based on this, we discover that $|a \oplus u - b \oplus u| = |(n-a) \oplus (n+1-u) - (n-b) \oplus (n+1-u)|$. Therefore, we obtain $d_p(\mathcal{E}(D,\mathcal{U})) = d_p(\mathcal{E}(D',\mathcal{U}'))$ for any p.

S1.6 Proof of Theorem 5

According to Lemma 4, it is straightforward to acquire the case (i).

Let $t = 1$, and $\boldsymbol{x}_1, \boldsymbol{x}_2$ be two different rows of D_0. Let's start with the case (ii). Based on Lemma 4, we have $d_p(\mathcal{E}(D_0,\mathcal{U}_1)) = d_p(\mathcal{E}(D_0,\mathcal{U}_0\backslash\mathcal{U}_2))$. Analogous to the proof in Theorem 1, we have

$$\begin{aligned} & d_p(\mathcal{E}(D_0,\mathcal{U}_0\backslash\{(n+1)/2\})) \qquad \text{(S1.2)} \\ \leq \ & d_p(\mathcal{E}((\boldsymbol{x}_1;\boldsymbol{x}_2),\mathcal{U}_0\backslash\{(n+1)/2\})) \\ = \ & \tau_{p,n}(|\boldsymbol{x}_1 - \boldsymbol{x}_2| - \sum_{j=1}^{\psi(n)} \left| x_{1j} \oplus \frac{n+1}{2} - x_{2j} \oplus \frac{n+1}{2} \right|^p . \end{aligned}$$

If $|\boldsymbol{x}_1 - \boldsymbol{x}_2| \in q_1^u \mathcal{H}_{n/q_1^u}$ with $u \in \{0, \ldots, r_1 - 1\}$, it holds that

$$\begin{aligned}
&\sum_{j=1}^{\psi(n)} \left| x_{1j} \oplus \frac{n+1}{2} - x_{2j} \oplus \frac{n+1}{2} \right|^p \\
= &\sum_{j=1}^{\psi(n)} \left| q_1^u h_j \pmod n \oplus \frac{n+1}{2} - (n - q_1^u h_j) \pmod n \oplus \frac{n+1}{2} \right|^p \\
= &\ 2^{p+1} q_1^{u(p+1)} \sum_{j=1}^{\psi(n/q_1^u)/2} g_j^{*p}
\end{aligned}$$

where $p = 1, 2$, $g_1^* \leq \cdots \leq g_{\psi(n/q_1^u)}^* \in \mathcal{H}_{n/q_1^u}$. When $\boldsymbol{x}_1 = \boldsymbol{h}$ and $\boldsymbol{x}_2 = (n-1)\boldsymbol{h} \pmod n$, $\tau_{p,n}(|\boldsymbol{x}_1 - \boldsymbol{x}_2| - \sum_{j=1}^{\psi(n)} |x_{1j} \oplus (n+1)/2 - x_{2j} \oplus (n+1)/2|^p$ arrives the lower bound and the left and right sides of (S1.2) are equal for $p = 1, 2$. Then, we have

$$d_p(\mathcal{E}(D_0, \mathcal{U}_1)) = \frac{d_p(\mathcal{E}(D_0, \mathcal{U}_0 \backslash \{(n+1)/2\}))}{2} = \frac{(n - 2^{2p-3}3^{2-p}) d_p(\mathcal{E}(D_0))}{2n}.$$

Based on Lemma 4, we have $d_p(\mathcal{E}(D_0, \mathcal{U}_2)) = d_p(\mathcal{E}(D_0, \mathcal{U}_0 \backslash \mathcal{U}_1))$. Analogously, we have we have

$$\begin{aligned}
&d_p\left(\left(\mathcal{E}((\boldsymbol{x}_1; \boldsymbol{x}_2)), \mathcal{E}\left((\boldsymbol{x}_1; \boldsymbol{x}_2), \frac{n+1}{2}\right)\right)\right) \\
&\geq d_p\left(\left(\mathcal{E}(D_0), \mathcal{E}\left(D_0, \frac{n+1}{2}\right)\right)\right) \\
&\geq d_p(\mathcal{E}(D_0)) + \min\left\{ \sum_{j=1}^{\psi(n)} \left| x_{1j} \oplus \frac{n+1}{2} - x_{2j} \oplus \frac{n+1}{2} \right|^p \right\}.
\end{aligned} \tag{S1.3}$$

Besides, we have

$$\sum_{j=1}^{\psi(n)} \left| x_{1j} \oplus \frac{n+1}{2} - x_{2j} \oplus \frac{n+1}{2} \right|^p \ \geq \ \sum_{j=1}^{\psi(n)} \left| h_j \oplus \frac{n+1}{2} - n \oplus \frac{n+1}{2} \right|^p$$

$$= 2\sum_{j=1}^{\psi(n)/2}(h_j^*)^p$$
$$= \frac{2^{p-3}3^{2-p}d_p(\mathcal{E}(D_0))}{n}$$

where $h_1^*,\ldots,h_{\psi(n)}^*$ is a permutation of $h_1,\ldots,h_{\psi(n)}$ with $h_1^*<\cdots<h_{\psi(n)}^*$ for $p=1,2$. When $\boldsymbol{x}_1=\boldsymbol{h}$ and $\boldsymbol{x}_2=n\boldsymbol{h}\pmod n$, $\sum_{j=1}^{\psi(n)}|x_{1j}\oplus(n+1)/2-x_{2j}\oplus(n+1)/2|^p$ arrives the upper bound $2^{p-3}3^{2-p}d_p(\mathcal{E}(D_0))/n$ and the left and right sides of (S1.3) are equal for $p=1,2$. Then, we have

$$d_p(\mathcal{E}(D_0,\mathcal{U}_2))=\frac{d_p\left((\mathcal{E}(D_0),\mathcal{E}(D_0,(n+1)/2))\right)}{2}=\frac{(n-2^{p-3}3^{2-p})d_p(\mathcal{E}(D_0))}{2n}.$$

S1.7 Proof of Theorem 6

Without loss of generality, assume $1=h_1<\cdots<h_k=n$ for an even n. Let $\tilde{D}_0=(x_{ij})$ be an $n\times\psi(n+1)$ LGLP set. Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two arbitrary rows of $\tilde{D}_0$. If $u\neq\{n+1-x_{1j},n+1-x_{2j}\}$, $|\,x_{1j}\oplus_L u-x_{2j}\oplus_L u)\,|$ stays the same under the leave-one-out operation. Otherwise, $|\,(x_{1j}-x_{2j})\oplus_L(n+1-x_{1j})\,|+|\,(x_{1j}-x_{2j})\oplus_L(n+1-x_{2j})\,|=n+1$ becomes $n+1-|x_{1j}-x_{2j}|$ under the leave-one-out operation. Therefore, we have

$$d_1(\mathcal{L}((\boldsymbol{x}_1;\boldsymbol{x}_2)))=d_1(\mathcal{E}((\boldsymbol{x}_1;\boldsymbol{x}_2)))-\sum_{j=1}^{\psi(n+1)}|x_{1j}-x_{2j}|\ \text{ and}$$
$$d_1(\mathcal{L}((\boldsymbol{x}_1;\boldsymbol{x}_2)))\geq d_1(\mathcal{L}(\tilde{D}_0))\geq d_1(\mathcal{E}(D_0))-\max\left\{\sum_{j=1}^{\psi(n+1)}|x_{1j}-x_{2j}|\right\}\text{(S1.4)}$$

Taking $n+1=\tilde{q}_1^{\tilde{r}_1}$ as an example, we shall find the maximum of

$\sum_{j=1}^{\psi(n+1)} |x_{1j} - x_{2j}|$. It is obvious that $|x_{1j} - x_{2j}| \in \{1, \ldots, n\}$ with the maximum as $n-1$ and the minimum as 1. If $|x_{1j} - x_{2j}| \in \mathcal{H}_{n+1}$, we have

$$\begin{aligned}\sum_{j=1}^{\psi(n+1)} |x_{1j} - x_{2j}| &\leq \sum_{j=1}^{n} |(n+1-j) - j| - \sum_{j=1}^{(n+1)/\tilde{q}_1 - 1} |(n+1-j\tilde{q}_1) - j\tilde{q}_1| \\ &= \frac{\tilde{q}_1 n^2 - (n+1-\tilde{q}_1)^2}{2\tilde{q}_1}.\end{aligned}$$

If $|x_{1j} - x_{2j}| \in \tilde{q}_1^{\epsilon}\mathcal{H}_{(n+1)/\tilde{q}_1^{\epsilon}}$ with $\epsilon \in \{1, \ldots, r_1 - 1\}$, we have

$$\sum_{j=1}^{\psi(n+1)} |x_{1j} - x_{2j}| \leq \frac{\tilde{q}_1(n+1-\tilde{q}_1^{\epsilon})^2 - (n+1-\tilde{q}_1^{(\epsilon+1)})^2}{2\tilde{q}_1}.$$

When $\boldsymbol{x}_1 = \boldsymbol{h}$ and $\boldsymbol{x}_2 = (n+1)\boldsymbol{h} \pmod{n+1}$, $\sum_{j=1}^{\psi(n+1)} |x_{1j} - x_{2j}|$ arrives the upper bound $(\tilde{q}_1 n^2 - (n+1-\tilde{q}_1)^2)/(2\tilde{q}_1)$ and the left and right sides of (S1.4) are equal. Then, we obtain the closed form of $d_1(\mathcal{L}(\tilde{D}_0))$. Analogously, the similar idea can be used to prove the case with $t > 1$.

S1.8 Proof of Corollary 3

Let us prove the case (i). For $0 < b < a \leq n$, it holds that

$$|a \oplus_L u - b \oplus_L u| = \begin{cases} a - b & \text{if } \ 0 \leq u < n+1-a \ \text{ or } \ n+1-b < u, \\ b & \text{if } \ u = n+1-a, \\ n+1-a+b & \text{if } \ n+1-a < u < n+1-b, \\ n+1-a & \text{if } \ u = n+1-b. \end{cases}$$

Analogously, we have

$$|a \oplus_L u - b \oplus_L u| = |(n+1-a) \oplus_L (n+1-u) - (n+1-b) \oplus_L (n+1-u)|.$$

Therefore, we obtain $d_p(\mathcal{L}(\tilde{D},\mathcal{U})) = d_p(\mathcal{L}(\tilde{D}^+,\mathcal{U}^+))$ for any p.

The remaining cases can be similarly proven as Theorem 6 so that we omit their proofs.

S1.9 Proof of Theorem 7

Let's prove the case (i). Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two arbitrary rows of $\tilde{D}_0$. Based on Lemma 4, we have $d_p(\mathcal{L}(\tilde{D}_0,\mathcal{U}_1)) = d_p(\mathcal{L}(\tilde{D}_0,\mathcal{U}_1^+))$. Analogous to the proof of the case (i), we have

$$\begin{aligned} d_p(\mathcal{L}((\boldsymbol{x}_1;\boldsymbol{x}_2),\tilde{\mathcal{U}_0}\backslash\{0\})) &\geq d_p(\mathcal{L}(\tilde{D}_0,\tilde{\mathcal{U}_0}\backslash\{0\})) \\ &\geq d_p(\mathcal{L}(\tilde{D}_0)) - \max\left\{\sum_{j=1}^{\psi(n+1)} |x_{1j} - x_{2j}|^p\right\}. \end{aligned} \tag{S1.5}$$

Besides, we have

$$\begin{aligned} &\sum_{j=1}^{\psi(n+1)} |x_{1j} - x_{2j}|^p \\ \leq &\sum_{j=1}^{\psi(n+1)} |h_j - (n+1-h_j)|^p \\ = &\begin{cases} \dfrac{(n+1)^{p-1}((n+1)^2 - \tilde{q}_1)(\tilde{q}_1 - 1)}{2\tilde{q}_1} & \text{if } \ p = 1, \\ \dfrac{(n+1)(\tilde{q}_1 n(n-1) - (n+1-\tilde{q}_1)(n+1-2\tilde{q}_1))}{3\tilde{q}_1} & \text{if } \ p = 2. \end{cases} \end{aligned} \tag{S1.6}$$

When $\boldsymbol{x}_1 = \boldsymbol{h}$ and $\boldsymbol{x}_2 = n\boldsymbol{h} \pmod{(n+1)}$, $\sum_{j=1}^{\psi(n+1)} |x_{1j} \oplus_L (n+2)/2 - x_{2j} \oplus_L (n+2)/2|^p$ arrives the upper bound in (S1.6) and the left and right sides of (S1.5) are equal for $p = 1, 2$. Then, we have

$$\begin{cases} d_1(\mathcal{L}(\tilde{D}_0, \mathcal{U}_1)) = d_1(\mathcal{L}(\tilde{D}_0)) - d_1(\mathcal{E}(D_0))/2; \\ \\ d_2(\mathcal{L}(\tilde{D}_0, \mathcal{U}_1)) = (d_2(\mathcal{L}(\tilde{D}_0)) - \Lambda(n))/2, \end{cases}$$

where $d_p(\mathcal{L}(\tilde{D}_0))$ is defined in Theorem 6, $d_p(\mathcal{E}(D_0))$ is defined in Theorem 1 (replacing n by $n+1$ here) and

$$\Lambda(n) = \frac{(n+1)(\tilde{q}_1 n(n-1) - (n+1-\tilde{q}_1)(n+1-2\tilde{q}_1))}{3\tilde{q}_1}.$$

Let's prove the case (ii). Combined the proofs of Theorem 5 and case (i), we can derive the closed forms of $d_1(\mathcal{L}(\tilde{D}_0, \mathcal{U}_3))$ for an even n and $d_2(\mathcal{L}(\tilde{D}_0, \mathcal{U}_3))$ for an odd prime $n+1$. The detail is omitted for saving space. Similar idea can be applied to all cases for $t > 1$, which are omitted here for saving space.

S1.10 Proof of Lemma 5

For $\tau_{1,n}(\boldsymbol{x}_i)$ in the proof of Theorem 1, $\tau_{1,n}(\boldsymbol{x}_i)$ arrives the maximum when $j = t-1$, $\bar{q}_t = q_1$ and $u_t = \bar{r}_t - 1 = r_1 - 1$. The maximum value is

$$\begin{cases} \dfrac{n^{p+2}(q_1^2-1)}{2^{p-1}3q_1^2} & \text{if } \ t = 1; \\ \dfrac{n^{p+2}(q_1+1)\prod_{\ell=1}^{t}(q_\ell - 1)}{2^{p-1}3q_1\prod_{\ell=1}^{t} q_\ell} & \text{otheriwse.} \end{cases}$$

Based on the proof the Theorem 2, we have $d_p^{max}(\mathcal{E}(D_1)) = d_p^{max}(\mathcal{E}(D_0))/2$.

S2. ACE-based and LACE-based algorithms

S2.1 Algorithm 0

To further accommodate an arbitrary number of factors (m), Algorithm 0 applies simulated annealing (SA) to find the best collection of m columns among the $n\psi(n)$ columns within the largest ACE. It systematically searches all columns within $\mathcal{E}(D_0)$ in such a way that, during each iteration, one column is randomly selected from the current design $\mathcal{D}$, and another is randomly chosen from $\mathcal{E}(D_0)\backslash\mathcal{D}$ for exchange. The resulting new design is denoted as $\mathcal{D}_{\text{try}}$. The process initiates by randomly selecting m columns from $\mathcal{E}(D_0)$ and updates the current design $\mathcal{D}$ to $\mathcal{D}_{\text{try}}$ using the probability $\pi = 1$ if $\text{Eff}_p(\mathcal{D}) < \text{Eff}p(\mathcal{D}\text{try})$, otherwise with $\pi = \exp\left((\text{Eff}_p(\mathcal{D}_{\text{try}}) - \text{Eff}_p(\mathcal{D}))/T\right)$. Algorithm 0 stops when the tuning parameter T falls below a pre-specified threshold. In the context of SA, the tuning parameter refers to a temperature parameter that controls the likelihood of accepting a worse solution than the current one.

S2.2 The D_1-based and (D_1, D_1')-based algorithms

Algorithm 2 often produces more efficient LHDs than Algorithm 1 when $m \approx n\psi(n)/2$. Particularly when $m = n\psi(n)/2$, Algorithm 2 ensures the output of the optimal design $\mathcal{E}(D_1)$ as proven in Lemma 4. Note the D_0 used in Algorithm 1 has the partition of $D_1 \cup D_1'$. On the other hand, Algorithm 2 either ignores D_1' or includes all linear permutations of D_1'. Here, we study the more flexible framework where D_1 and D_1' work together to form an efficient LHD.

Specifically, we draw an integer partition of m as $m = m_1 + m_2$ for non-negative $m_1, m_2 \leq n\psi(n)/2$ with a certain degree of randomness in each iteration. Denote by $v_i = \lfloor 2m_i/\psi(n) \rfloor$ the length of $\mathcal{U}^i$ and let $K_i = m_i \pmod{(\psi(n)/\ 2)}$ for $i = 1, 2$. The m_1 columns are comprised of the main part $\mathcal{E}(D_1, \mathcal{U}^1)$ together with a random selection of K_1 columns from $\mathcal{E}(D_1, \mathcal{U}_0 \backslash \mathcal{U}^1)$. The m_2 columns are constructed in a similar way based on D_1' and $\mathcal{U}^2$. The index sets $\mathcal{U}^1$ and $\mathcal{U}^2$ are optimized in a similar way as in previous algorithms. Then the output design will be fed into Algorithm 0 as the initial design if the tuning parameter at the end of the first iteration loop is above δ_1. The full details will be provided in Algorithm 3.

Note that Algorithms 1–2 are two special cases of Algorithm 3. Algorithm 3 reduces to Algorithm 1 if $\mathcal{U}^1 = \mathcal{U}^2$ and $v_1 = v_2 = \lfloor m/\psi(n) \rfloor$. Al-

gorithm 3 reduces to Algorithm 2 if $\mathcal{U}^2 = \emptyset$ with $(v_1, v_2) = (\lfloor 2m/\psi(n) \rfloor, 0)$ when $m \leq n\psi(n)/2$ or $\mathcal{U}^2 = \mathcal{U}_0$ with $(v_1, v_2) = (\lfloor 2m/\psi(n) \rfloor - n, n)$ when $m > n\psi(n)/2$. Unlike Algorithm 2, Algorithm 3 does not ensure the optimality of the output design when $m = n\psi(n)/2$. However, Algorithm 3 is more flexible in general. Overall, the designs produced by all the algorithms have high L_1- or L_2-distance efficiencies, mostly exceeding 90%. If more iterations are allowed, the produced design can be used as an initial design for the further search of the globally optimal LHD with columns selected from all possible $n!$ permutations.

S2.3 LACE-based algorithms

Let's begin the discussion with Algorithm 0_L. It uses the simulated annealing to find the best m columns among the $(n+1)\psi(n+1)$ columns in $\mathcal{L}(\tilde{D}_0)$, instead of the $n\psi(n)$ columns in $\mathcal{E}(D_0)$ as in Algorithm 0. Regarding Algorithm 1_L, it changes the elementary design from the GLP set D_0 in Algorithm 1 to the LGLP set $\tilde{D}_0$ to produce the skeleton LACE expansion in the first iteration loop. Moreover, Algorithm 1_L searches the $\mathcal{U} \subset \tilde{\mathcal{U}}_0$ that best works with $\tilde{D}_0$ to produce a highly efficient design. Algorithm 2_L still uses $\tilde{D}_0$ as the elementary LGLP set but searches an optimal index set $\mathcal{U} \subset \mathcal{U}_3$ coupled with $\tilde{D}_0$ in the first iteration loop. This algorithm nei-

ther uses D_1 in Algorithm 2 nor $\tilde{D}_1$ since the sub-design $\mathcal{L}(\tilde{D}_0, \mathcal{U}_3)$ often has larger L_1- and L_2-distance efficiencies than $\mathcal{E}(D_1)$ and $\mathcal{L}(\tilde{D}_1)$ when n is even and $m \approx (n+1)\psi(n+1)/2$. It will only adapt to $m \leq (n+2)\psi(n+1)/2$. For the cases of $m > (n+2)\psi(n+1)/2$, the first $(n+2)\psi(n+1)/2$ columns are set to $\mathcal{L}(\tilde{D}_1, \mathcal{U}_3)$ and the remaining $m - (n+2)\psi(n+1)/2$ columns will be optimized by the same algorithm among columns of $\mathcal{L}(\tilde{D}_0, \tilde{\mathcal{U}}_0 \backslash \mathcal{U}_3)$. Lastly, Algorithm 3_L changes two elementary designs from the GLP set D_1 and its strictly dual D_1' in Algorithm 3 to the LGLP set $\tilde{D}_1$ and its strictly dual as $\tilde{D}_1^+$ in the first iteration loop. This algorithmic searches two index sets $\mathcal{U}^1, \mathcal{U}^2 \subset \tilde{\mathcal{U}}_0$ that best work with $\tilde{D}_1$ and $\tilde{D}_1^+$ respectively in the similar way as Algorithm 3. Like the LACE-based algorithm, for Algorithms 1_L-3_L, the resulting design at the end of the first iteration loop is fed into Algorithm 0_L as the initial design for optimizing m columns.

S3. Quantitative comparisons

S3.1 Quantitative comparisons between the proposed method and existing works

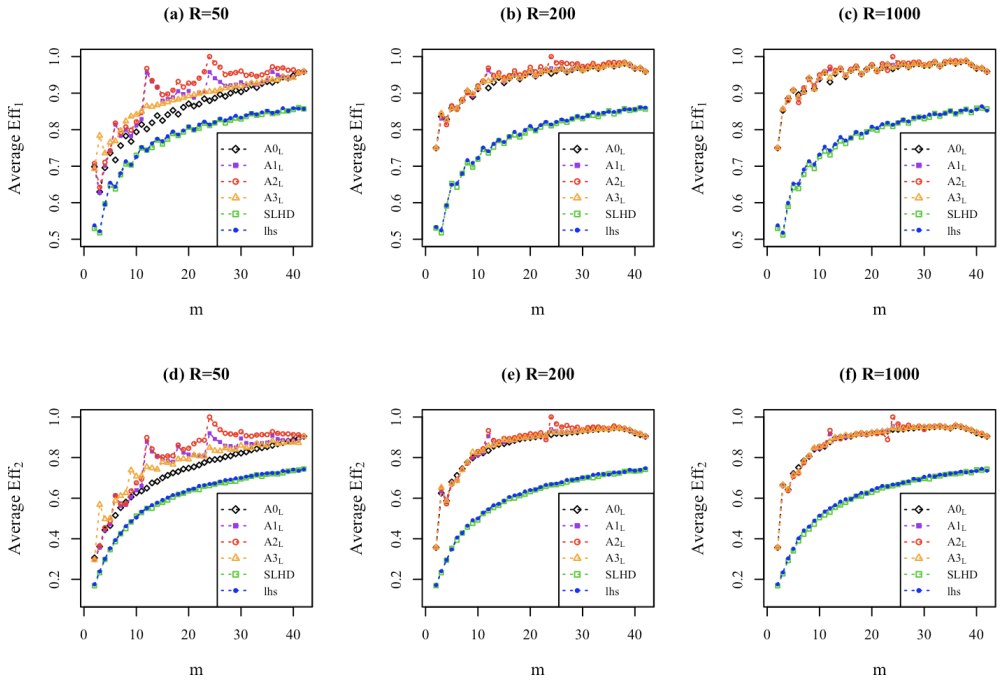

Figure S1: The L_1- and L_2-distance efficiencies under various scenarios: Algorithms 0_L, 1_L, 2_L, 3_L, "SLHD" and "lhs" in the R-packages (marked as A0_L, A1_L, A2_L, A3_L, SLHD and lhs in sequence).

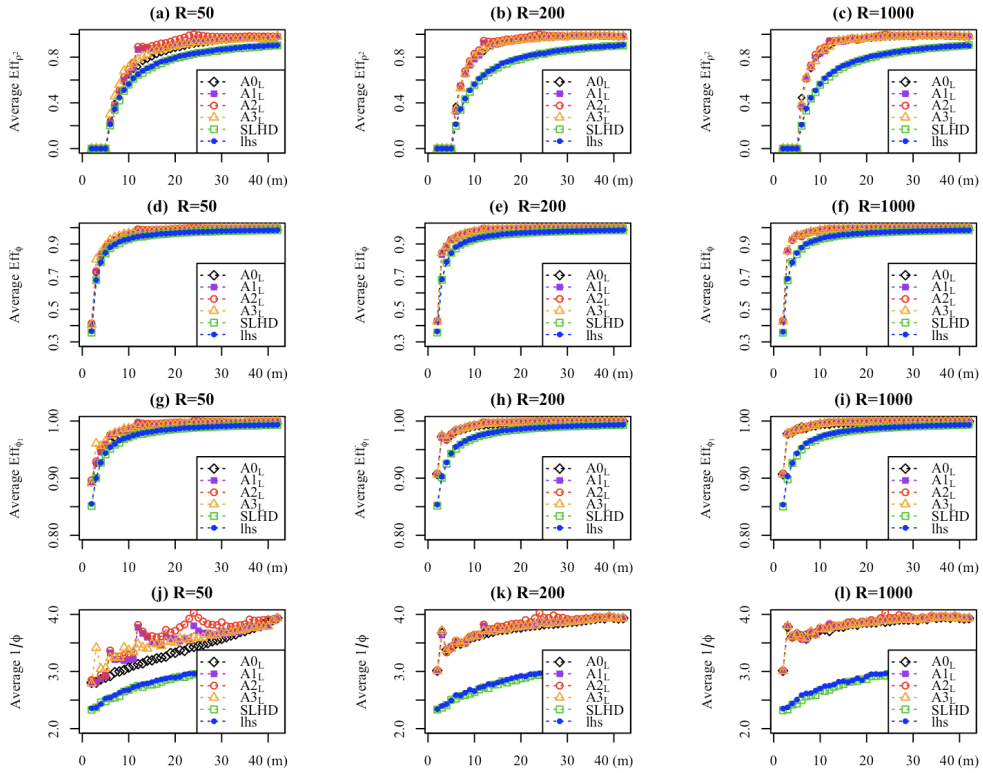

Figure S2: The ρ^2, ϕ, ϕ_1 and $\tilde{\phi}$ efficiencies under various scenarios: Algorithms 0_L, 1_L, 2_L, 3_L, "SLHD" and "lhs" in the R-packages (marked as A0$_L$, A1$_L$, A2$_L$, A3$_L$, SLHD and lhs in sequence).

Table S1: Quantitative comparisons under the L_2-distance criterion.

Method	n							
	$t=1$				$t=2$			
	$r_1=1$		$r_1=2$	$r_1=3$	$r_1=r_2=1$		$r_1=2$ $r_2=1$	$r_1=1$ $r_2=2$
	5	11	9	8	10	22	12	18
A1	0.883	0.892	0.828	0.815	0.699	0.818	0.647	0.691
	(0.400, 1)	**(0.227, 1)**	**(0.167, 0.933)**	(**0.306, 0.917**)	(0.200, 0.819)	(0.119, 0.913)	(0.154, 0.769)	(**0.158**, 0.789)
$A1_L$	–	–	–	0.817	0.901	0.922	0.906	0.916
				(0.208, 0.856)	(0.277, 0.945)	(0.132, 0.964)	(0.240, 0.948)	(0.145, 0.960)
A2	0.880	**0.937**	**0.855**	0.815	0.707	0.859	0.681	0.728
	(0.400, 1)	**(0.227, 1)**	**(0.167, 0.933)**	(**0.306, 0.917**)	(0.200, 0.819)	(0.119, 0.913)	(0.154, 0.769)	(**0.158**, 0.789)
$A2_L$	–	–	–	0.825	**0.933**	**0.959**	**0.939**	**0.955**
				(0.208, 0.900)	(**0.278**, **1**)	(**0.138, 1**)	(0.240, **1**)	(0.146, **1**)
A3	0.883	0.893	0.833	0.815	0.700	0.812	0.645	0.689
	(0.400, 1)	**(0.227, 1)**	**(0.167, 0.933)**	(**0.306, 0.917**)	(0.200, 0.819)	(0.119, 0.913)	(0.154, 0.769)	(**0.158**, 0.789)
$A3_L$	–	–	–	0.815	0.900	0.916	0.902	0.912
				(0.208, 0.855)	(0.277, 0.946)	(0.134, 0.966)	(0.236, 0.950)	(0.146, 0.962)
SLHD	0.583	0.688	0.612	0.522	0.531	0.716	0.537	0.633
$m \le n\psi(n)$	(0.215, 0.688)	(0.052, 0.781)	(0.081, 0.712)	(0.096, 0.657)	(0.066, 0.655)	(0.013, 0.798)	(0.050, 0.658)	(0.021, 0.735)
SLHD	–	–	–	0.627	0.700	0.809	0.727	0.782
$m \le (n+1)\psi(n+1)$				(0.101, 0.728)	(0.070, 0.788)	(0.014, 0.865)	(0.048, 0.802)	(0.021, 0.844)
lhs	0.596	0.694	0.608	0.524	0.527	0.717	0.536	0.631
$m \le n\psi(n)$	(0.248, 0.701)	(0.051, 0.775)	(0.078, 0.709)	(0.102, 0.652)	(0.067, 0.656)	(0.015, 0.799)	(0.047, 0.665)	(0.021, 0.732)
lhs	–	–	–	0.623	0.702	0.809	0.722	0.783
$m \le (n+1)\psi(n+1)$				(0.093, 0.732)	(0.068, 0.790)	(0.014, 0.867)	(0.042, 0.805)	(0.021, 0.851)

Note: Algorithms 1, 1_L, …, 3, 3_L, "SLHD" and "lhs" in the R-packages (marked as A1, $A1_L$, …,A3, $A3_L$, SLHD and lhs in sequence). For each case, the listed data is the median along with minimum and maximum (as specified in parentheses) of $\bar{d}_1(\mathcal{D})$'s with the numbers of columns of $\mathcal{D}$'s being either $2,\ldots,n\psi(n)$ or $2,\ldots,(n+1)\psi(n+1)$.

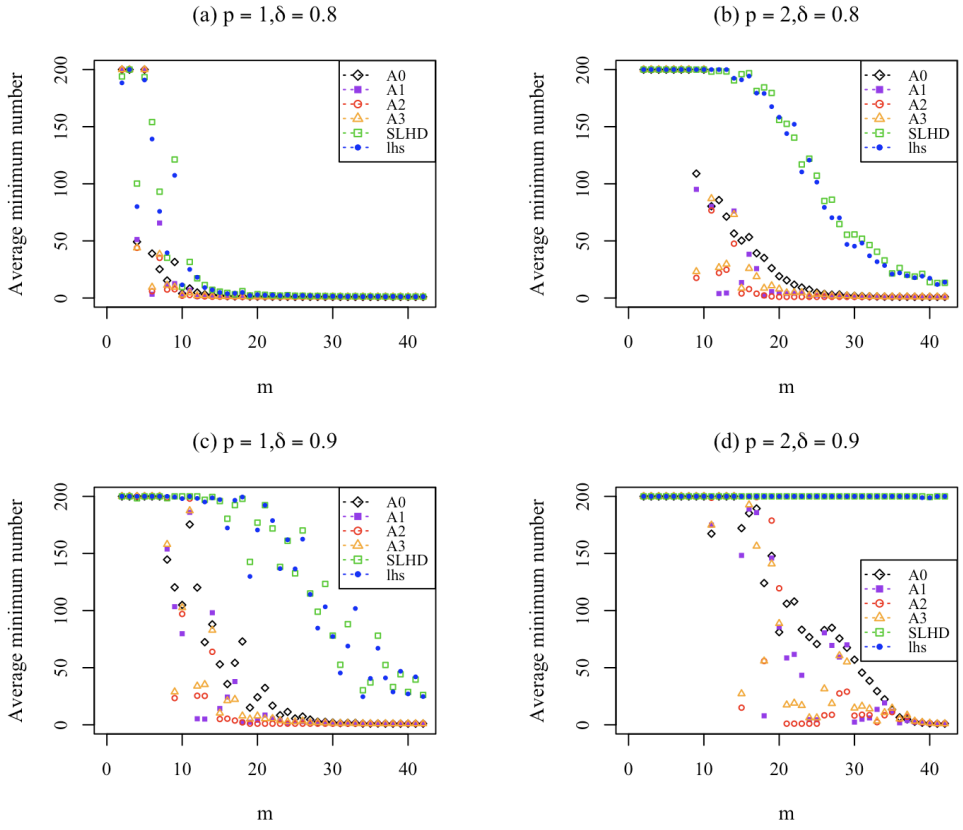

Figure S3: The computational cost under various scenarios: Algorithms 0, 1, 2, 3, "SLHD" and "lhs" in the R-packages (marked as A0, A1, A2, A3, SLHD and lhs in sequence).

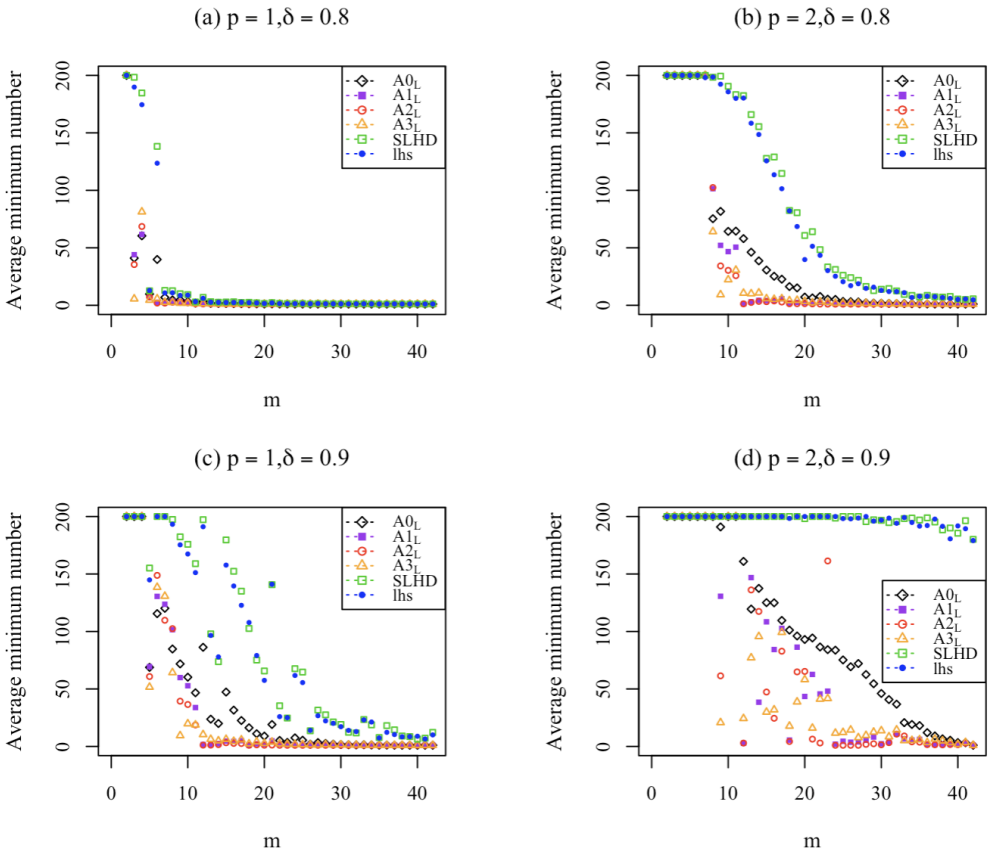

Figure S4: The computational cost under various scenarios: Algorithms 0_L, 1_L, 2_L, 3_L, "SLHD" and "lhs" in the R-packages (marked as $\mathrm{A0}_L$, $\mathrm{A1}_L$, $\mathrm{A2}_L$, $\mathrm{A3}_L$, SLHD and lhs in sequence).

Table S2: Quantitative comparisons under the ρ^2 criterion.

Method	n							
	$t=1$				$t=2$			
	$r_1=1$		$r_1=2$	$r_1=3$	$r_1=r_2=1$		$r_1=2$ $r_2=1$	$r_1=1$ $r_2=2$
	5	11	9	8	10	22	12	18
A1	0.943	0.945	0.804	0.755	0.595	0.697	0.490	0.549
	(0, 1)	**(0, 1)**	**(0, 0.877)**	(**0**, 0.831)	(**0**, 0.701)	(**0**, 0.753)	(**0**, 0.629)	(**0**, 0.646)
$A1_L$	–	–	–	0.786	0.952	0.969	0.956	0.964
				(**0**, 0.838)	(**0**, 0.988)	(**0**, 0.996)	(**0**, 0.992)	(**0**, 0.995)
A2	**0.944**	**0.981**	**0.836**	0.760	0.622	0.728	0.544	0.596
	(0, 1)	**(0, 1)**	**(0, 0.877)**	(**0**, 0.831)	(**0**, 0.701)	(**0**, 0.753)	(**0**, 0.629)	(**0**, 0.646)
$A2_L$	–	–	–	**0.807**	**0.986**	**0.991**	**0.987**	**0.990**
				(**0**, 0.838)	**(0, 1)**	**(0, 1)**	**(0, 1)**	**(0, 1)**
A3	0.939	0.945	0.809	0.755	0.584	0.695	0.486	0.549
	(0, 1)	**(0, 1)**	**(0, 0.877)**	(**0**, 0.831)	(**0**, 0.701)	(**0**, 0.753)	(**0**, 0.629)	(**0**, 0.646)
$A3_L$	–	–	–	0.785	0.951	0.965	0.954	0.961
				(**0**, 0.838)	(**0**, 0.988)	(**0**, 0.996)	(**0**, 0.992)	(**0**, 0.995)
SLHD	0.939	0.945	0.809	0.755	0.584	0.695	0.486	0.549
$m \le n\psi(n)$	(**0**, 0.842)	(**0**, 0.919)	(**0**, 0.867)	(0, 0.813)	(**0**, 0.801)	(**0**, 0.909)	(**0**, 0.789)	(**0**, 0.851)
SLHD	–	–	–	0.774	0.852	0.922	0.872	0.907
$m \le (n+1)\psi(n+1)$				(**0**, 0.888)	(**0**, 0.929)	(**0**, 0.960)	(**0**, 0.935)	(**0**, 0.953)
lhs	0.728	0.838	0.740	0.626	0.594	0.817	0.582	0.702
$m \le n\psi(n)$	(**0**, 0.853)	(**0**, 0.916)	(**0**, 0.870)	(**0**, 0.810)	(**0**, 0.795)	(**0**, 0.909)	(**0**, 0.789)	(**0**, 0.852))
lhs	–	–	–	0.780	0.857	0.920	0.869	0.907
$m \le (n+1)\psi(n+1)$				(**0, 0.889**)	(**0**, 0.928)	(**0**, 0.961)	(**0**, 0.937)	(**0**, 0.954)

Note: Algorithms 1, 1_L, ..., 3, 3_L, "SLHD" and "lhs" in the R-packages (marked as A1, $A1_L$, ...,A3, $A3_L$, SLHD and lhs in sequence). For each case, the listed data is the median along with minimum and maximum (as specified in parentheses) of $\bar{d}_1(\mathcal{D})$'s with the numbers of columns of $\mathcal{D}$'s being either $2, \ldots, n\psi(n)$ or $2, \ldots, (n+1)\psi(n+1)$.

Table S3: Quantitative comparisons under the ϕ criterion.

Method	n							
	$t=1$				$t=2$			
	$r_1=1$		$r_1=2$	$r_1=3$	$r_1=r_2=1$		$r_1=2$ $r_2=1$	$r_1=1$ $r_2=2$
	5	11	9	8	10	22	12	18
A1	**0.994**	0.990	0.916	0.900	0.801	0.793	0.724	0.706
	(0.638, 1)	**(0, 1)**	(**0**, 0.930)	(0, 0.911)	(0, 0.818)	(0, 0.807)	(0, 0.755)	(0, 0.734)
$A1_L$	–	–	–	0.933	0.991	0.992	0.991	0.991
				(0, 0.944)	(0, 0.999)	(0, 0.999)	(0, 0.999)	(0, 0.999)
A2	**0.994**	**0.997**	0.923	0.901	0.805	0.801	0.738	0.722
	(0.637, **1**)	**(0, 1)**	(**0**, 0.930)	(0, 0.911)	(0, 0.818)	(0, 0.807)	(0, 0.755)	(0, 0.734)
$A2_L$	–	–	–	0.938	**0.998**	**0.998**	**0.998**	**0.998**
				(0, 0.944)	**(0, 1)**	**(0, 1)**	**(0, 1)**	**(0, 1)**
A3	**0.994**	0.990	0.917	0.899	0.797	0.793	0.723	0.705
	(0.633, **1**)	**(0, 1)**	(**0**, 0.930)	(0, 0.911)	(0, 0.818)	(0, 0.807)	(0, 0.75)	(0, 0.734)
$A3_L$	–	–	–	0.934	0.991	0.991	0.990	0.991
				(0, 0.944)	(0, 0.999)	(0, 0.999)	(0, 0.999)	(0, 0.999)
SLHD	0.958	0.962	**0.946**	0.925	0.914	0.947	0.900	0.919
$m \le n\psi(n)$	(0.560, 0.977)	(**0**, 0.982)	(**0, 0.973**)	(0, 0.963)	(0, 0.958)	(0, 0.973)	(0, 0.951)	(0, 0.960)
SLHD	–	–	–	0.956	0.968	0.977	0.970	0.974
$m \le (n+1)\psi(n+1)$				(0, 0.978)	(0, 0.985)	(0, 0.988)	(0, 0.985)	(0, 0.987)
lhs	0.962	0.964	**0.946**	0.928	0.912	0.947	0.901	0.919
$m \le n\psi(n)$	(0.586, 0.979)	(**0**, 0.981)	(**0,0.973**)	(0, 0.963)	(0, 0.955)	(0, 0.973)	(0, 0.950)	(0, 0.959)
lhs	–	–	–	**0.958**	0.969	0.977	0.969	0.975
$m \le (n+1)\psi(n+1)$				(0, **0.979**)	(0, 0.984)	(0, 0.989)	(0, 0.985)	(0, 0.987)

Note: Algorithms 1, 1_L, ..., 3, 3_L, "SLHD" and "lhs" in the R-packages (marked as A1, $A1_L$, ...,A3, $A3_L$, SLHD and lhs in sequence). For each case, the listed data is the median along with minimum and maximum (as specified in parentheses) of $\bar{d}_1(\mathcal{D})$'s with the numbers of columns of $\mathcal{D}$'s being either $2, \ldots, n\psi(n)$ or $2, \ldots, (n+1)\psi(n+1)$.

Table S4: Quantitative comparisons under the ϕ_1 criterion.

Method	n							
	$t=1$				$t=2$			
	$r_1=1$		$r_1=2$	$r_1=3$	$r_1=r_2=1$		$r_1=2$ $r_2=1$	$r_1=1$ $r_2=2$
	5	11	9	8	10	22	12	18
A1	**0.997** **(0.921, 1)**	0.998 (0.847, **1**)	0.986 (0.842, 0.988)	0.982 (**0.882**, 0.984)	0.966 (0.853, 0.969)	0.988 (0.796, 0.989)	0.960 (0.845, 0.965)	0.975 (**0.815**, 0.978)
$A1_L$	–	–	–	0.986 (0.859, 0.989)	0.998 **(0.868, 1)**	0.999 **(0.801, 1)**	0.999 (0.861, **1**)	0.999 (0.814, **1**)
A2	**0.997** **(0.921, 1)**	**0.999** (0.843, **1**)	**0.987** (0.832, 0.988)	0.982 (**0.882**, 0.984)	0.967 (0.854, 0.969)	0.989 (0.795, 0.989)	0.963 (0.844, 0.965)	0.977 (0.814, 0.978)
$A2_L$	–	–	–	**0.988** (0.861, 0.989)	**1** **(0.868, 1)**	**1** **(0.801, 1)**	**1** **(0.863, 1)**	**1** **(0.815, 1)**
A3	**0.997** (0.920, **1**)	0.998 **(0.848, 1)**	0.986 (**0.843**, 0.988)	0.981 (**0.882**, 0.984)	0.965 (0.853, 0.969)	0.988 (0.796, 0.989)	0.959 (0.845, 0.965)	0.975 (**0.815**, 0.978)
$A3_L$	–	–	–	0.987 (0.861, 0.989)	0.998 **(0.868, 1)**	0.999 **(0.801, 1)**	0.998 (0.861, **1**)	0.999 (**0.815, 1**)
SLHD $m \le n\psi(n)$	0.977 (0.864, 0.988)	0.993 (0.788, 0.997)	**0.987** (0.807, **0.993**)	0.979 (0.817, 0.990)	0.983 (0.797, 0.991)	0.996 (0.730, 0.998)	0.984 (0.778, 0.992)	0.992 (0.747, 0.996)
SLHD $m \le (n+1)\psi(n+1)$	–	–	–	**0.988** (0.822, **0.994**)	0.993 (0.798, 0.997)	0.998 (0.732, 0.999)	0.995 (0.781, 0.998)	0.998 (0.749, 0.999)
lhs $m \le n\psi(n)$	0.980 (0.876, 0.989)	0.993 (0.783, 0.997)	**0.987** (0.808, **0.993**)	0.980 (0.819, 0.989)	0.982 (0.795, 0.991)	0.996 (0.736, 0.998)	0.984 (0.781, 0.992)	0.992 (0.748, 0.996)
lhs $m \le (n+1)\psi(n+1)$	–	–	–	**0.988** (0.814, **0.994**)	0.994 (0.796, 0.997)	0.998 (0.736, 0.999)	0.995 (0.777, 0.998)	0.998 (0.743, 0.999)

Note: Algorithms 1, $1_L, \ldots, 3, 3_L$, "SLHD" and "lhs" in the R-packages (marked as A1, $A1_L, \ldots$,A3, $A3_L$, SLHD and lhs in sequence). For each case, the listed data is the median along with minimum and maximum (as specified in parentheses) of $\bar{d}_1(\mathcal{D})$'s with the numbers of columns of $\mathcal{D}$'s being either $2, \ldots, n\psi(n)$ or $2, \ldots, (n+1)\psi(n+1)$.

Table S5: Quantitative comparisons under the $1/\tilde{\phi}$ criterion.

Method	n							
	$t=1$				$t=2$			
	$r_1=1$		$r_1=2$	$r_1=3$	$r_1=r_2=1$		$r_1=2$ $r_2=1$	$r_1=1$ $r_2=2$
	5	11	9	8	10	22	12	18
A1	0.344 (0.322, 0.424)	0.106 (0.097, 0.172)	0.178 (0.166, 0.246)	0.227 (0.212, 0.281)	0.217 (**0.196**, 0.260)	0.046 (**0.042**, 0.075)	**0.170** (**0.142**, 0.220)	**0.080** (**0.064**, 0.108)
$A1_L$	–	–	–	0.209 (0.202, 0.269)	0.124 (0.117, 0.184)	0.033 (0.031, 0.075)	0.093 (0.087, 0.137)	0.047 (0.044, 0.095)
A2	0.342 (0.322, 0.422)	0.102 (0.097, 0.173)	0.174 (0.165, 0.252)	0.224 (0.208, 0.279)	0.217 (0.195, 0.259)	0.044 (**0.042**, 0.075)	0.159 (0.140, 0.218)	0.070 (0.063, 0.104)
$A2_L$	–	–	–	0.204 (0.194, 0.268)	0.119 (0.114, 0.184)	0.032 (0.030, 0.074)	0.088 (0.084, 0.134)	0.045 (0.043, 0.093)
A3	0.343 (0.322, 0.426)	0.106 (0.097, 0.171)	0.178 (0.166, 0.245)	0.227 (0.212, 0.281)	**0.218** (0.195, 0.262)	**0.047** (**0.042**, 0.075)	**0.170** (**0.142**, 0.212)	**0.080** (**0.064**, 0.107)
$A3_L$	–	–	–	0.209 (0.202, 0.267)	0.125 (0.117, 0.184)	0.033 (0.031, 0.074)	0.094 (0.087, 0.136)	0.048 (0.044 , 0.095)
SLHD $m \le n\psi(n)$	**0.441** **(0.417, 0.515)**	**0.142** (0.128, 0.256)	0.207 (0.184, 0.301)	**0.260** (**0.232**, 0.338)	0.191 (0.169, 0.277)	0.045 (0.040, **0.140**)	0.147 (0.129, 0.235)	0.071 (0.061, 0.163)
SLHD $m \le (n+1)\psi(n+1)$	–	–	–	0.239 (0.216, 0.332)	0.162 (0.147, 0.273)	0.04 (0.037, 0.137)	0.118 (0.108, 0.234)	0.058 (0.053, 0.161)
lhs $m \le n\psi(n)$	0.431 (0.413, 0.497)	0.141 **(0.129, 0.260)**	**0.208** **(0.185, 0.302)**	0.259 (0.231, 0.337)	0.191 (0.169, **0.279**)	0.045 (0.040, 0.133)	0.146 (0.129, 0.234)	0.071 (0.062, 0.163)
lhs $m \le (n+1)\psi(n+1)$	–	–	–	0.239 (0.215, **0.343**)	0.161 (0.146, 0.276)	0.04 (0.037, 0.133)	0.120 (0.108, **0.243**)	0.058 (0.053, **0.166**)

Note: Algorithms 1, 1_L, ..., 3, 3_L, "SLHD" and "lhs" in the R-packages (marked as A1, $A1_L$, ...,A3, $A3_L$, SLHD and lhs in sequence). For each case, the listed data is the median along with minimum and maximum (as specified in parentheses) of $\check{d}_1(\mathcal{D})$'s with the numbers of columns of $\mathcal{D}$'s being either $2,\ldots,n\psi(n)$ or $2,\ldots,(n+1)\psi(n+1)$.

S3.2 Hybrid approaches combing existing algebraic constructions and a stochastic algorithmic search

While any existing algebraic construction method can handle arbitrary design sizes by adding a stochastic algorithmic search step, this does not guarantee high efficiency when the gap between the targeted design size and the designed design size is too large. Compared to other algebraic methods, the major benefit of combining ACE and LACE in our construction is the provision of a denser allocation among integers for the number of runs (n). Specifically, existing methods are mostly limited to prime n and $m = n$ or $n - 1$. Our construction is able to accommodate any integer n and $m = kv$ so that it is much easier to fill the gap with algorithm than existing construction methods. Theoretically, we shall ensure the (asymptotic) optimality of our designs for n being prime (prime power) and prime (prime power) minus one, which also cover more cases than existing results for prime n. Lastly, even for other n's, the proposed design achieves an efficiency of at least 93.33%.

Furthermore, the hybrid approach which combines existing algebraic construction and a stochastic algorithmic search is subject to a stringent restriction on the space-filling property of design points. Note that n_1 and k_1 represent the numbers of runs and columns respectively in the design

generated, limited to $n_1 \leq n$ and $k_1 \leq k$. The algebraic construction method produces a highly efficient LHD in which the design points are confined to the space $[0, n_1]^{k_1}$. Subsequently, a stochastic algorithmic search is employed to explore the remaining $n - n_1$ runs in the space $(n_1, n]^{k_1} \times [0, n]^{k-k_1}$ and the remaining $k - k_1$ columns in the space $[0, n]^{k-k_1}$.

Consider $n = n_1 = 6$ and $2 \leq m \leq 42$ as an illustration. Wang, Xiao and Xu (2018) constructed an optimal $n_1 \times n_1$ LHD (D) with the L_1-distance when $2n_1 + 1$ is a prime number. An $n \times m$ design can be constructed with $m = vn_1 + v'$, combining v maximin LHDs and selecting $v' < n_1$ columns from D. For any m, we compare the L_1-distances between the designs using Algorithm 2 and the approach based on Wang, Xiao and Xu (2018) . The results are summarized in Table S6. In 20 cases of m, the proposed designs exhibit larger L_1-distances compared to that by the hybrid approach based on Wang, Xiao and Xu (2018) . In 13 cases of m, they show equal L_1-distances. In 8 cases of m, the proposed designs have smaller L_1-distances than that by the hybrid approach based on Wang, Xiao and Xu (2018)). Notably, for these 8 cases, m is almost a multiple of n_1. Violating this constraint of m and n_1, the proposed method surpasses the hybrid approach based on Wang, Xiao and Xu (2018) in terms of L_1-distance.

When $n > n_1$, for the hybrid approach based on Wang, Xiao and Xu

Table S6: The comparisons on the L_1-distance between the method based on Wang, Xiao and Xu (2018) and Algorithm 2 when $n = 6$.

m	The method based on Wang, Xiao and Xu (2018)	Algorithm 2	m	The method based on Wang, Xiao and Xu (2018)	Algorithm 2
2	2	**3**	22	49	**50**
3	5	**6**	23	**51**	**51**
4	7	**8**	24	**56**	**56**
5	9	**10**	25	**57**	**57**
6	**14**	13	26	58	**59**
7	**15**	**15**	27	61	**62**
8	16	**17**	28	63	**64**
9	**19**	**19**	29	65	**66**
10	**21**	**21**	30	**70**	68
11	23	**24**	31	**71**	**71**
12	**28**	26	32	72	**73**
13	**29**	**29**	33	**75**	**75**
14	30	**31**	34	77	**78**
15	33	**34**	35	79	**80**
16	35	**36**	36	**84**	**84**
17	37	**38**	37	**85**	**85**
18	**42**	40	38	86	**87**
19	**43**	42	39	**89**	**89**
20	44	**45**	40	**91**	90
21	**47**	**47**	41	**93**	92
			42	**98**	94

(2018) , the first n_1 runs fall within $(0, n_1]^{k_1} \times [0, n]^{k-k_1}$ while the last $n-n_1$ runs fall within $(n_1, n]^{k_1} \times [0, n]^{k-k_1}$. As a result, there are numerous areas

that consistently remain unexplored, such as $[0, n_1] \times (n_1, n]^{k_1-1} \times [0, n]^{k-k_1}$ and $[0, n_1]^2 \times (n_1, n]^{k_1-2} \times [0, n]^{k-k_1}$. This drawback will result in poor space-filling properties, e.g., the projection property. The proposed algorithms, however, have no significant gaps left in the experimental space. Take a 6×6 optimal L_1-distance LHD as an example. Wang, Xiao and Xu (2018) constructed an $n_1 \times n_1$ optimal L_1-distance LHD when n_1 is prime. The optimal LHD with $n_1 = 5$ is shown on the left side of Table S7 (1st-5th rows and columns). In the 6th row, columns 1-5 have to be level 6. Moreover, the upper bound of the L_1-distance is 11 regardless of the 6th column added. By randomly selecting a column, we obtained the corresponding LHD listed on the left side of Table S7. The first 5 runs fall within $[0, 5]^5 \times (0, 6]$ while the last run fall within $(5, 6]^5 \times (0, 6]$. There are numerous areas that consistently remain unexplored, such as $[0, 5] \times (5, 6]^4 \times [0, 6]$ and $[0, 5]^2 \times (5, 6]^3 \times [0, 6]$. In terms of projection uniformity, this LHD has $\phi(D) = 0.0095$ and $\tilde{\phi}(D) = 0.3128$. We construct the target design by Algorithm 2, listed on the right side of Table S7. The L_1-distance, ϕ and $\tilde{\phi}$ values of the proposed LHD are 12, 0.0093 and 0.2931, respectively.

These results demonstrate that the proposed design has a better space-filling property than that obtained via a stochastic algorithmic search based on the design in Wang, Xiao and Xu (2018) .

Table S7: The design by a stochastic algorithmic search based on the design in Wang, Xiao and Xu (2018) and the propsed design by Algorithm 2.

The approach based on Wang, Xiao and Xu (2018)						Algorithm 2					
1	2	3	4	5	4	1	6	2	4	5	4
2	4	5	3	1	2	2	2	5	6	3	5
3	5	2	1	4	1	5	1	1	3	4	3
4	3	1	5	2	6	6	5	3	1	2	6
5	1	4	2	3	3	4	4	6	5	6	1
6	6	6	6	6	5	3	3	4	2	1	2

S3.3 Comparisons the proposed designs and random samples to be the initial points in multimodal optimization

The proposed design stands out as an excellent choice for initial points in multi-start methods for global optimization, especially beneficial for multimodal optimization. To avoid local optima, multi-start methods restart optimization from different starting points to explore the landscape and identify what appears to be the global optimum. Compared with random starting points, a space-filling LHD often offers multi-start methods greater chances of reaching global optima, as it distributes more uniformly in space and exhibits superior projection uniformity. For a high-dimension optimization, the proposed designs with $n < m$ are more economical and suitable in practice.

We compare the proposed design with random samples as initial points in multi-start methods for multimodal optimization. Typically, an optimization problem is formulated in terms of an objective function $f(\boldsymbol{x})$ that requires minimization, where the vector $\boldsymbol{x}$ represents a candidate solution to the problem. Here, we consider three test functions: Levy function, Alpine 1 function, and Salomon function. Their mathematical formulations, suggested search space and the location of the global minimum are as follows.

- Levy function: $f(\boldsymbol{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{m-1} (w_i - 1)^2 \left[1 + 10\sin^2(\pi w_i + 1)\right] + (w_m - 1)^2 \left[1 + \sin^2(2\pi w_m)\right]$ where $\boldsymbol{x} \in [-10, 10]^m$ and $w_i = 1 + (x_i - 1)/4$ for all $i = 1, \ldots, m$. The global minimum is $f(\boldsymbol{x}^*) = 0$ at $\boldsymbol{x}^* = (1, \ldots, 1)$.

- Alpine 1 function: $f(\boldsymbol{x}) = \sum_{i=1}^{m} |x_i \sin(x_i) + 0.1x_i|$ where $\boldsymbol{x} \in [-5, 5]^m$. The global minimum is $f(\boldsymbol{x}^*) = 0$ at $\boldsymbol{x}^* = (0, \ldots, 0)$.

- Salomon function: $f(\boldsymbol{x}) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{m} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{m} x_i^2}$ where $\boldsymbol{x} \in [-5, 5]^m$. The global minimum is $f(\boldsymbol{x}^*) = 0$ at $\boldsymbol{x}^* = (0, \ldots, 0)$.

All three test functions are multimodal.

For each sampling method, we consider six design sizes: $(n, m) =$

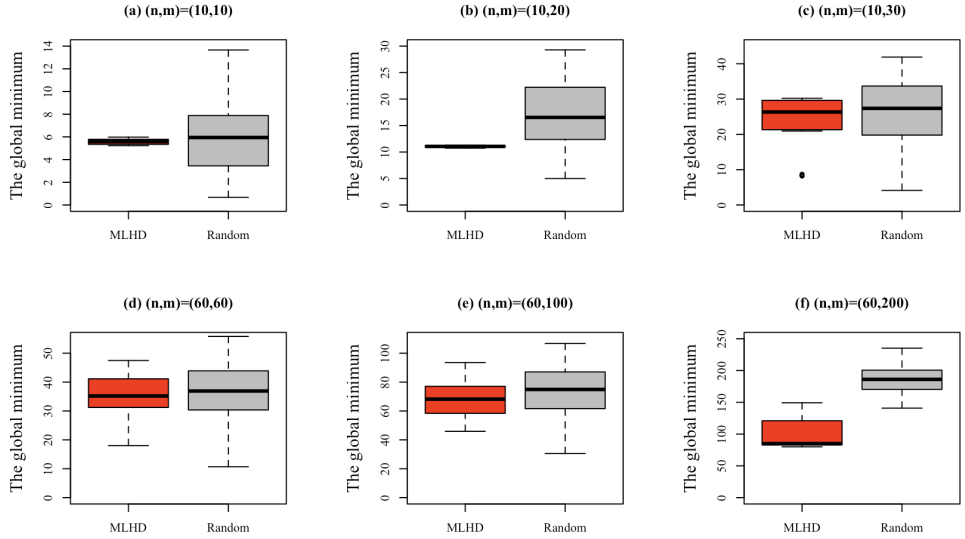

Figure S5: Comparisons the proposed design and random sampling under Levy function.

$(10, 10), (10, 20)$, $(10, 30), (60, 60), (60, 100)$, and $(60, 200)$. Random samples are uniformly drawn from the domain, while random column permutations are applied to the proposed design. The multimodal optimization is solved by the R package "nloptr" with each point of $n \times m$ initial points. Among the n candidate solutions, the best minimizer with the minimum value of f is considered as the global minimizer. This process is repeated 100 times. For each scenario of samples and test function, boxplots of the global minima are presented in Figures S5–S7.

For each scenario, the multi-start method employing the proposed design exhibits a smaller worst-case value (maximum value) of global minima

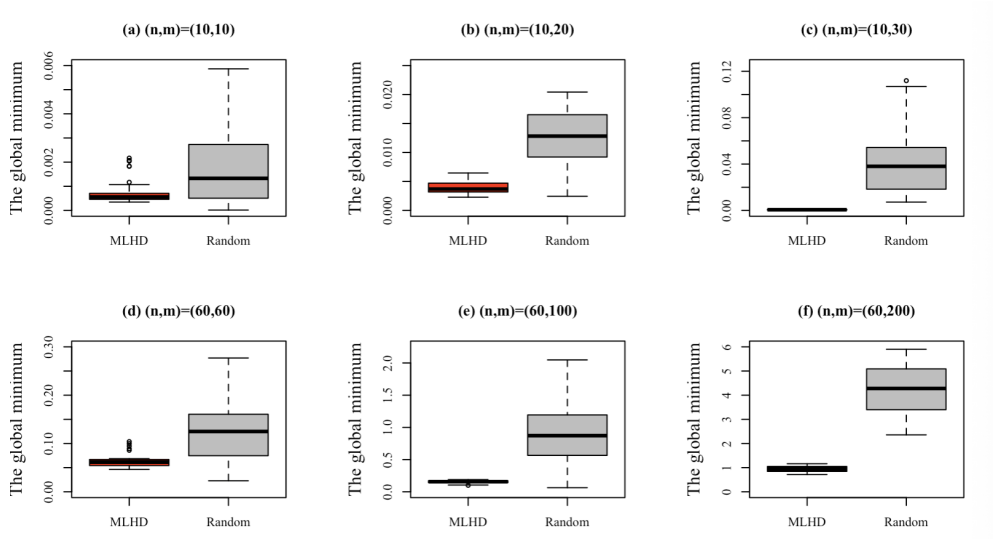

Figure S6: Comparisons the proposed design and random sampling under Alpine 1 function.

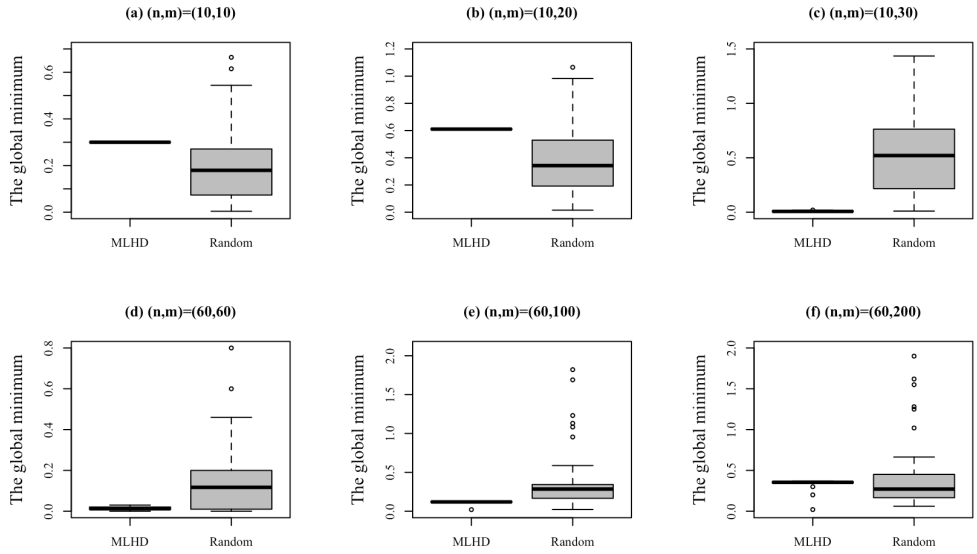

Figure S7: Comparisons the proposed design and random sampling under Salomon function.

compared to the method using random samples. This suggests that the proposed design is more robust than random samples in multimodal optimization. Furthermore, the multi-start method with the proposed design generally outperforms the method with random samples in terms of the mean value of global minima, especially evident in the scenarios depicted in Figure S5 and those with $n = 60$ in Figures S5–S7.

S3.4 A comparison of the limiting efficiency and the actual efficiency

In practice, designs of small to moderate sizes are often of great interest due to budget considerations. The following numerical results demonstrate how the bounds change as n increases, providing a clearer picture of their behavior for finite sizes. Following our recommendation to use ACE or LACE based on the parity of run sizes, we calculated the L_p-efficiencies for the designs in Theorem 4 with odd run sizes and those in Theorem 7 with even run sizes for $p = 1, 2$ and $3 \leq n \leq 100$. The results are presented in Figure S8, alongside the corresponding theoretical bounds for comparison. It is shown that the efficiencies of all designs exceed 0.8. And the L_1-distance efficiencies are even greater than 0.85. Specifically, the designs with prime n's achieve optimal efficiencies. If n is a prime power,

the resulting efficiency gradually approaches the theoretical bound (1) as n increases. Otherwise, for $t > 2$ and odd $\tilde{n}$, the efficiency gradually reaches the theoretical bound $(1 - 1/(\tilde{q}_1\tilde{q}_2))$ as $\tilde{n}$ divisible by $\tilde{q}_1$ and $\tilde{q}_2$ increases where $\psi(\cdot)$ is the Euler function, and $\tilde{n} = n$ for odd n and $\tilde{n} = n+1$ for even n. To further illustrate how these theoretical bounds vary with "small to moderate" sizes, we performed similar calculations for $n = 25, 50, \ldots, 250$. The results, shown in Figure S9, demonstrate conclusions consistent with those in Figure S8.

S3.5 The L_p-distance efficiencies with $p \geq 2$

In fact, some ACEs arrive the optimal L_p-distance for any p, such as the ACE with a prime n in Theorem 1. There ACEs are equal-distance designs, making them optimal under the L_p-distance for any p. However, the efficiencies of other ACEs depend on the "distance" between their L_p-distance and the L_p-distance of the resulting equal-distance designs. The theoretical results of the L_1- and L_2-distances presented in this paper can be extended to the case of $p > 2$. The main theoretical results involves calculating $\sum_{i=1}^{n} i^p$ for any p. However, it remains challenging to derive a unified formulation for $\sum_{i=1}^{n} i^p$. For example, we have

1. $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

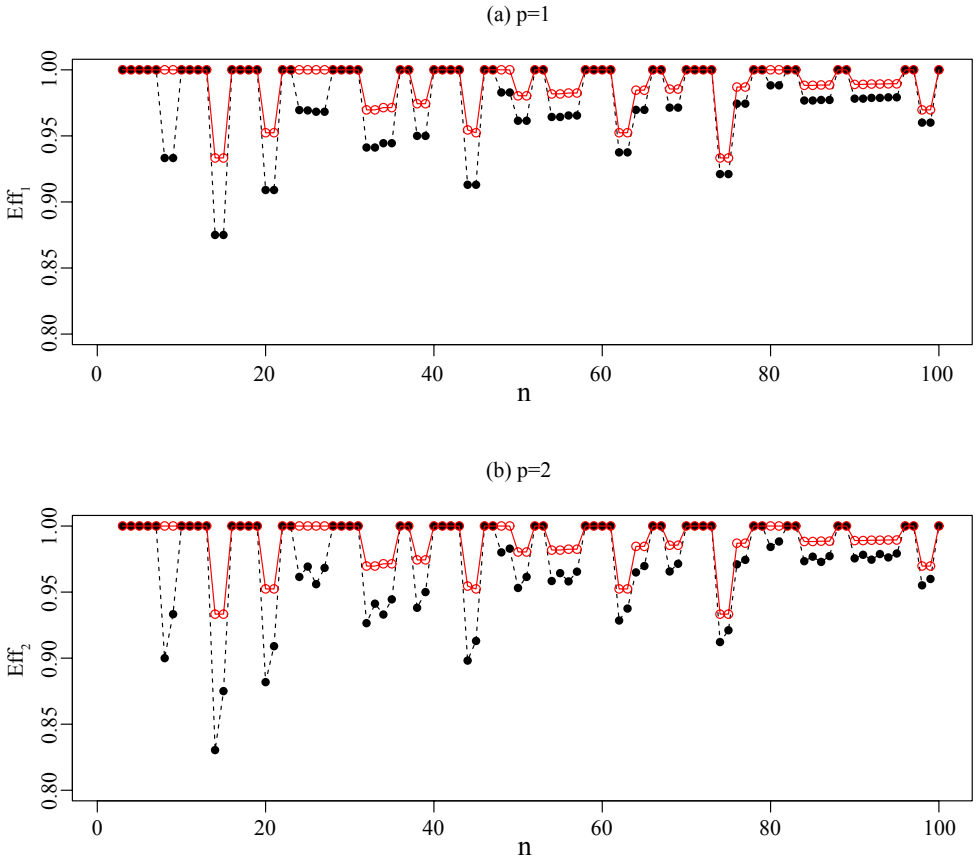

Figure S8: A comparison of the limiting efficiency and the actual efficiency for $n \leq 100$ (the symbols of ○ and ● represent the limiting efficiency and the actual efficiency, respectively).

2. $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

3. $\sum_{i=1}^{n} i^3 = (\frac{n(n+1)}{2})^2$

4. $\sum_{i=1}^{n} i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$.

Therefore, it is impractical to derive a unified expression for the L_p-distances of ACEs or LACEs for arbitrary p, although we successfully

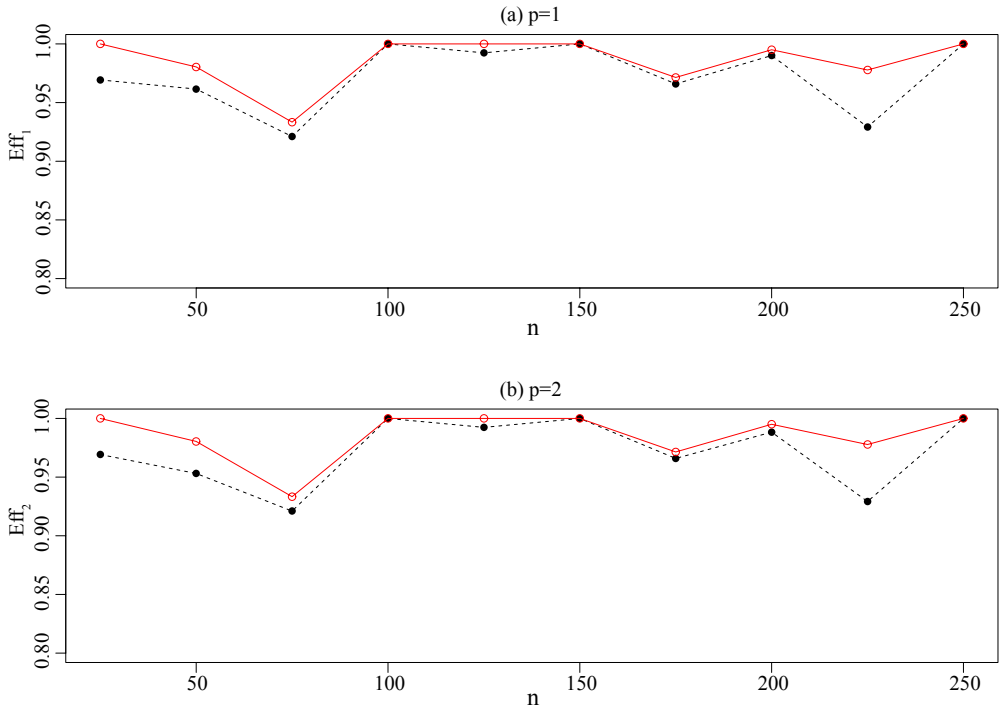

Figure S9: A comparison of the limiting efficiency and the actual efficiency for small to moderate sizes (the symbols of ○ and ● represent the limiting efficiency and the actual efficiency, respectively).

achieved this for the most commonly used values, $p = 1$ and $p = 2$.

S3.6 The sensitivity analysis for tuning parameter selection

To evaluate the impact of parameter variations on efficiency and solution quality, we conduct random sampling of parameter values around their recommended settings. Specifically:

Table S8: The L_2-distance efficiencies of some ACEs in Table 1.

n	m	D	$\mathcal{U}$	d_2	Eff_2	n	m	D	$\mathcal{U}$	d_2	Eff_2
3	2	D_0	$\{0\}$	2	0.5	7	6	D_0	$\{3\}$	35	0.625
	4		$\{0,2\}$	7	0.875		12		$\{0,2\}$	98	0.875
	6		$\{0,1,2\}$	12	1		18		$\{0,3,4\}$	154	0.917
5	4	D_0	$\{0\}$	10	0.5		24		$\{0,2,3,4\}$	210	0.938
	8		$\{0,2\}$	30	0.75		30		$\{0,2,3,4,6\}$	266	0.95
	12		$\{0,2,3\}$	55	0.917		36		$\{0,1,2,3,4,5\}$	322	0.958
	16		$\{0,1,2,3\}$	70	0.875		42		$\{0,1,2,3,4,5,6\}$	392	1
	20		$\{0,1,2,3,4\}$	100	1		3	D_1	$\{0\}$	14	0.5
	2	D_1	$\{0\}$	5	0.5		6		$\{3,5\}$	35	0.625
	4		$\{0,1\}$	10	0.5		9		$\{2,4,6\}$	70	0.833
	6		$\{0,1,3\}$	20	0.667		12		$\{0,1,2,6\}$	98	0.875
	8		$\{0,1,2,4\}$	30	0.75		15		$\{0,1,3,4,5\}$	126	0.9
	10		$\{0,1,2,3,4\}$	50	1		18		$\{0,1,2,3,4,5\}$	140	0.833
							21		$\{0,1,2,3,4,5,6\}$	196	1

Note: d_2 represents the L_2-distance and Eff_2 represents the L_2-distance efficiency.

1. $T \sim U(0.8, 1)$ (continuous uniform distribution),

2. $r \sim U(0.91, 0.99)$ (continuous uniform distribution),

3. $\kappa \sim U_{dis}(1, 20)$ (discrete uniform distribution),

4. $\delta_2 = |\tilde{\delta}_2|$ with $\tilde{\delta}_2 \sim N(0.01, 0.01)$, (normal distribution).

In theory, the algorithm's efficiency improves as δ_1 decreases or R in-

creases. We conduct numerical experiments with $R = 200, 500$ and 1000 and $(n, m) = (7, 7)$. For each R, Algorithm 2 is executed to generate an $n \times m$ LHD using the random parameter sampling. This process was repeated 100 times. Figure S10 displays the boxplot of criterion values across parameter configurations, demonstrating a consistent improvement in efficiency as R increases. Notably, efficiency variability decreases significantly for larger R values, particularly under the ϕ, ϕ_1 and $\tilde{\phi}$ criteria. Similar simulations for $m = 14$ (Figure S11) further corroborate these findings. These results confirm the robustness of our algorithm against parameter perturbations around the recommended values.

S3.7 The prediction accuracy of the proposed design

To further evaluate the prediction accuracy of the proposed design, we compare our method against "SLHD" from the R package using Gaussian process (GP) models. The Ackley function, a widely used benchmark for testing optimization algorithms, is selected as the test function. Its analytical expression is given by:

$$f(\boldsymbol{x}) = -a \exp\left(-b\sqrt{\frac{1}{k}\sum_{i=1}^{k} x_i^2}\right) - \exp\left(\frac{1}{k}\sum_{i=1}^{k}\cos(cx_i)\right) + a + \exp(1),$$

where $a = 20, b = 0.2$, $c = 2\pi$ and $\boldsymbol{x} = (x_1, \ldots, x_k) \in [0, 1]^k$. Typically, the number of factors (m) exceeds k. In our analysis, we evaluate configurations

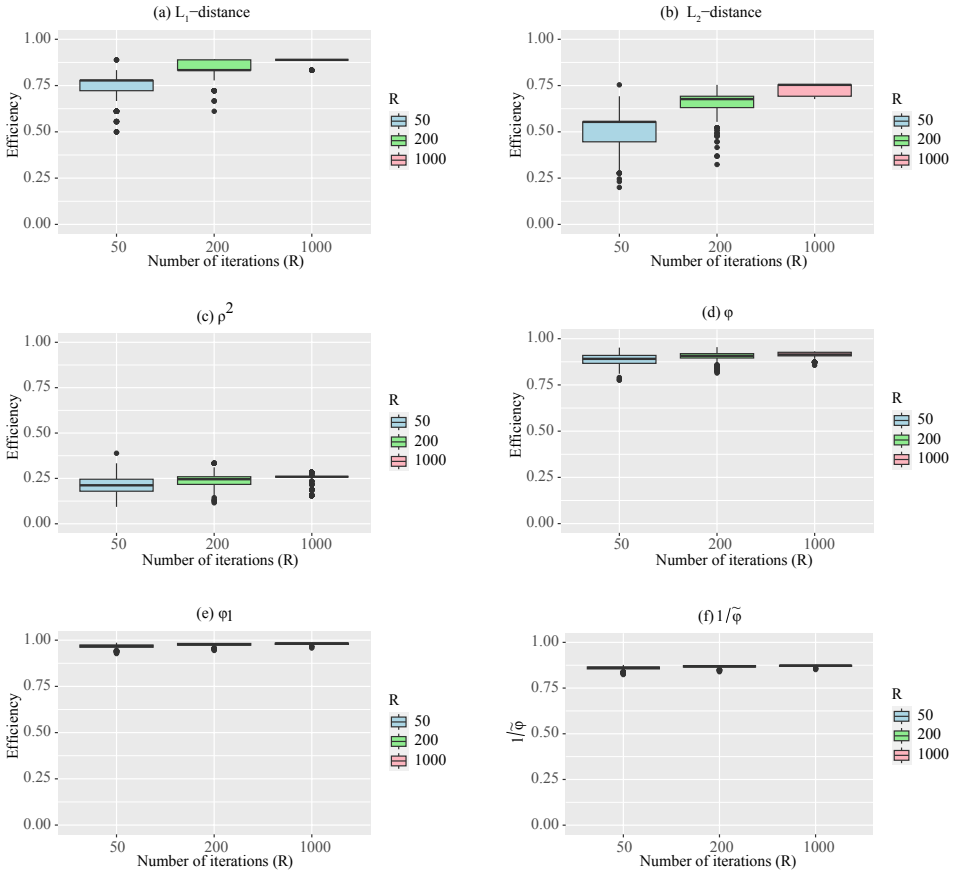

Figure S10: Boxplot of criterion values across different selections of tuning parameters when $(n, m) = (7, 7)$.

with $m \in \{7, 17\}$, sample sizes $n \in \{k, 2k, 3k\}$ and $k = \lfloor \sqrt{m} \rfloor$. For the proposed design, we use Algorithm 2 to construct it when n is odd and Algorithm 2_L when n is even.

The GP model is trained using the R package "hetGP" for both the proposed designs and the design via "SLHD". For every design, the process

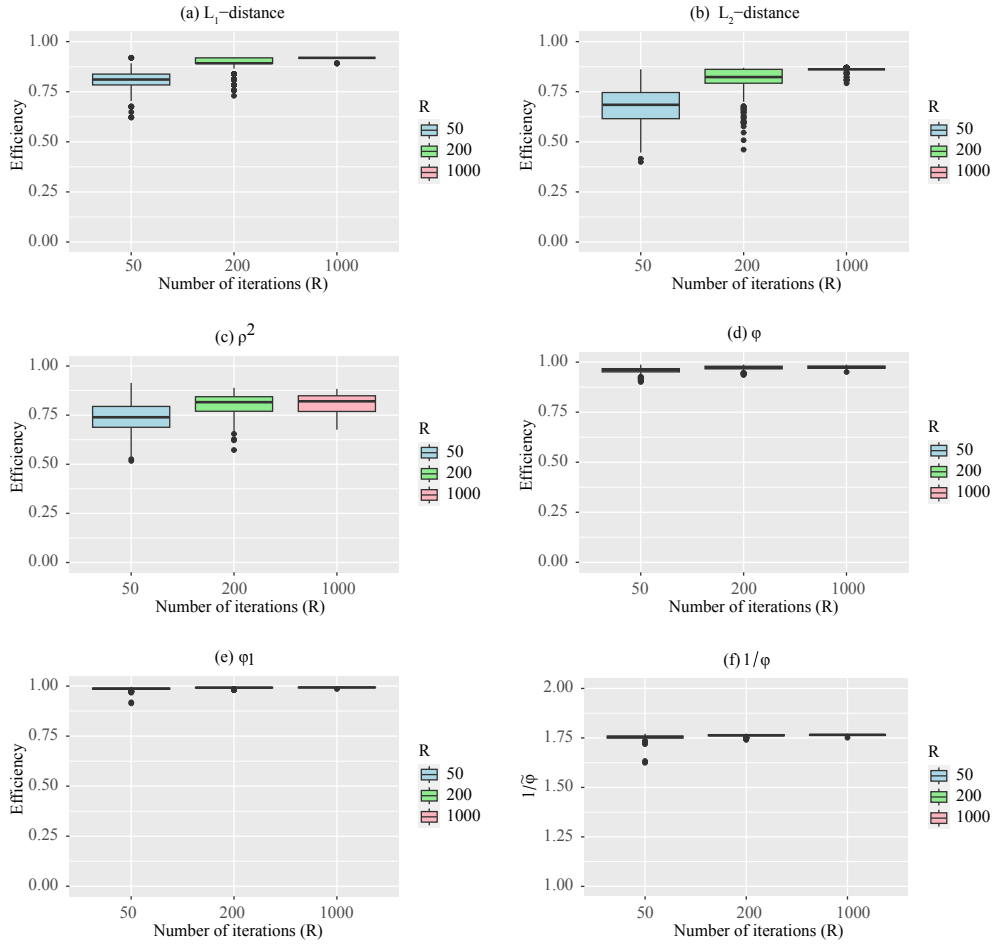

Figure S11: Boxplot of criterion values across different selections of tuning parameters when $(n, m) = (7, 14)$.

is repeated 100 times. We randomly select 10 sets of k factors from the total m factors in these designs, and this selection process is repeated 20 times for each set. For each scenario of m, the test data consists of $10^{\lceil log(m) \rceil + 2}$ random points uniformly sampled from $[0, 1]^k$. To evaluate design performance, we

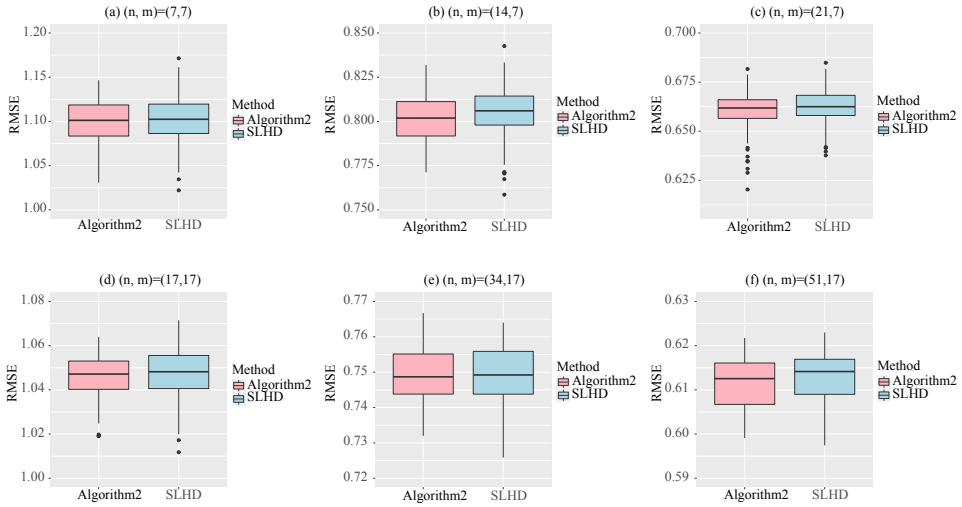

Figure S12: Comparisons on RMSE among Algorithm 2 and "SLHD" in R-package.

use the RMSE value to measure the discrepancy between predicted and true values. For each combination of n and set of k factors, we apply each design to fit the GP model and compute the RMSE value between the GP model's predictions and the true values from the test data. For each combination of n and k, the boxplots of RMSE values across 100 replications for the proposed designs and the design via "SLHD" are shown in Figure S12. For each combination of n and the set of k factors, we calculate the RMSE difference between the design via "SLHD" and the proposed designs. Figure S13 displays the boxplots of all pairwise RMSE differences across 100 replications, comparing the design via "SLHD" with the proposed designs.

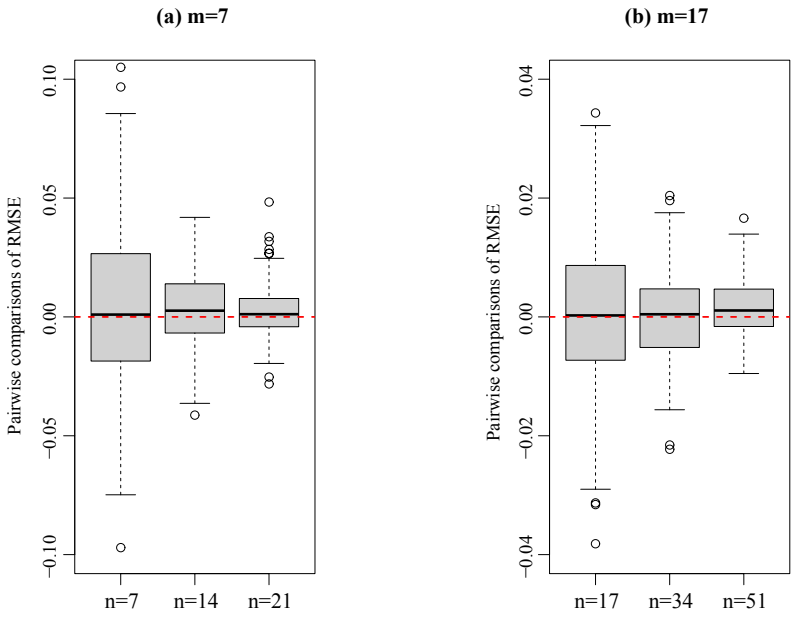

Figure S13: Pairwise comparisons on RMSE among Algorithm 2 and "SLHD" in R-package.

Figure S12 shows that each case yields a small RMSE value, and for a fixed m, the RMSE values decrease as n increases. As illustrated in Figure S13, more than half of the pairwise RMSE differences are greater than zero, indicating that the proposed design consistently achieves a lower RMSE. These findings indicate the better performance of the proposed designs in terms of prediction accuracy under GP modeling.

References

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

Zhang, X. R., Y. D. Zhou, M.-Q. Liu and D. K. J. Lin (2024). Sequential good lattice point sets for computer experiments. *Science China Mathematics* **67**, 2153–2170.

School of Mathematics and Physics, University of Science and Technology Beijing, Beijing, BJ 100083, China

Email: xueruzhang@ustb.edu.cn

Department of Statistics, Purdue University, West Lafayette, IN 47907, U.S.A

E-mail: dkjlin@purdue.edu

Department of Business Analytics and Statistics, University of Tennessee, Knoxville, TN 37996, U.S.A

E-mail: wzheng9@utk.edu