Statistica Sinica **28** (2018), 1395-1414
doi:https://doi.org/10.5705/ss.202016.0423

CONSTRUCTION OF MAXIMIN DISTANCE DESIGNS VIA LEVEL PERMUTATION AND EXPANSION

Qian Xiao and Hongquan Xu

University of California, Los Angeles

Abstract: Maximin distance designs as an important class of space-filling designs are widely used in computer experiments, yet their constructions are challenging. We develop an efficient procedure to generate maximin Latin hypercube designs, as well as maximin multi-level fractional factorial designs, from existing orthogonal or nearly orthogonal arrays via level permutation and expansion. We show that the distance distributions of the generated designs are closely connected with the distance distributions and generalized word-length patterns of the initial designs. Examples are presented to show that our method outperforms many current prevailing methods.

Key words and phrases: Computer experiment, fractional factorial design, generalized minimum aberration, Latin hypercube design, orthogonal array, space-filling design.

1. Introduction

Computer experiments are widely used in scientific researches and product developments to simulate real-world problems with complex computer codes (Santner, Williams and Notz (2013); Fang, Li and Sudjianto (2006); Morris and Moore (2015)). The most suitable designs for computer experiments are space-filling Latin hypercube designs (LHDs), yet their construction are challenging, especially for those with a large number of runs and factors.

Many researchers have studied orthogonal LHDs; see, among others, Steinberg and Lin (2006), Cioppa and Lucas (2007), Lin, Mukerjee and Tang (2009), Sun, Liu and Lin (2010) and Yang and Liu (2012). However, orthogonal LHDs are not necessarily space-filling, e.g. design (a) in Figure 1. Another approach is through computer search using some optimality criteria based on discrepancy or distance. Hickernell (1998) defined several discrepancy criteria, and among them the centered L_2-discrepancy (CD) is the most widely accepted. Johnson, Moore and Ylvisaker (1990) proposed the maximin and minimax distance criteria. In

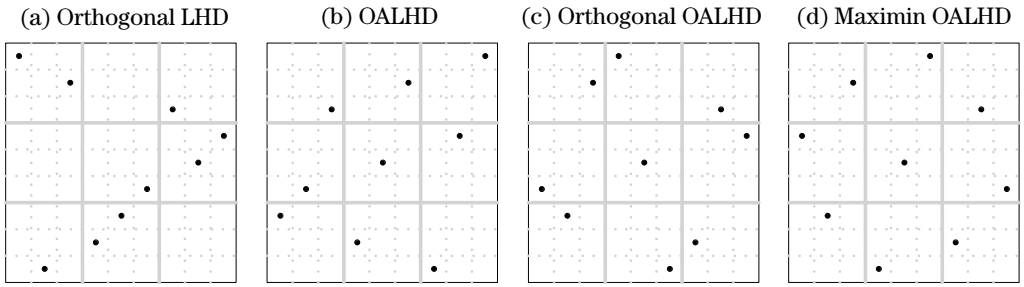

Figure 1. Comparison of 9-run 2-factor LHDs.

this paper, we adopt the maximin distance criterion which seeks to scatter design points over the experimental domain such that the minimum distance between points is maximized. Johnson, Moore and Ylvisaker (1990) showed that maximin distance designs are asymptotically optimal under a Bayesian setting. Morris and Mitchell (1995) proposed the criterion

$$\phi_p = \left(\sum_{i=2}^{n} \sum_{j=1}^{i-1} \frac{1}{d_{i,j}^p} \right)^{1/p}, \tag{1.1}$$

where $d_{i,j}$ is the distance between the i^{th} and j^{th} row of the design. When p is sufficiently large, ϕ_p is asymptotically identical to the maximin distance criterion. Morris and Mitchell (1995), Joseph and Hung (2008), Ba, Myers and Brenneman (2015), and many others proposed algorithms to construct maximin LHDs; see Lin and Tang (2015) for a summary. To the best of our knowledge, the R package SLHD by Ba, Myers and Brenneman (2015) is currently the most efficient algorithm.

Tang (1993) proposed to generate orthogonal array-based LHDs (OALHDs) by expanding levels in randomized orthogonal arrays (OAs). Though these OALHDs have desirable sampling and projection properties, most of them are not space-filling, e.g. designs (b) and (c) in Figure 1. A searching scheme can be applied to OALHDs (Leary, Bhaskar and Keane (2003)), but the results are not satisfactory. Ba, Myers and Brenneman (2015) used a level expansion procedure similar to that of Tang (1993) when generating SLHDs with multiple slices. They justified their method from a geometric perspective but did not provide theoretical support. We provide some theoretical results to complement the work of Tang (1993) and Ba, Myers and Brenneman (2015). We show that OAs, or nearly OAs if OAs do not exist, are good initial designs as they tend to generate robust space-filling designs. To avoid searching over the entire space of OALHDs

generated via level expansion, we propose to perform level permutations on the initial designs and restrict level expansions only to the maximin OAs. Tang, Xu and Lin (2012), Tang and Xu (2013), and Zhou and Xu (2014) used the level permutation method for constructing uniform and maximin fractional factorial designs, but their method cannot be used to construct LHDs and relies on the existence of multi-level OAs. We propose a procedure, the maximin distance level expansion (MDLE) method, to construct maximin designs by combining the strength of level permutation and expansion while avoiding their weaknesses. Our procedure is efficient, providing better designs using less time compared with existing methods. It is general, not only in the capability of constructing both maximin fractional factorial designs and maximin LHDs, but also in the flexibility to use multiple phases in level expansion that can significantly reduce the computation needed.

This paper is organized as follows. We present our theoretical results in Section 2. In Section 3, we introduce the procedure, searching algorithm, and justifications for our MDLE method. In Section 4, examples are given to show that our method outperforms the ordinary level expansion method, the OMLHD method, the R package SLHD, and the level permutation method. In Section 5, we introduce a multi-phase method for constructing large maximin designs. Section 6 concludes, and all proofs are given in the Appendix.

2. Some Theoretical Results

Let $D(n, s^k)$ be an n-run, k-factor, and s-level (labelled as $1, 2, \ldots, s$) balanced design where each level appears exactly n/s times in every column. From the initial design $D(n, s^k)$ we can generate a set of designs $D'(n, (ms)^k)$ with ms levels by a level expansion procedure. For each column in the initial design D, we replace the n/s positions of entry l ($l = 1, 2, \ldots, s$) by a random sequence of $n/(ms)$ replicates of numbers: $(l-1)m+1, (l-1)m+2, \ldots, (l-1)m+m$, where n, k, s, m are all integers larger than 1 and n is divisible by ms. When $m = n/s$, the generated D's are LHDs.

Example 1. As an illustration, we perform the level expansion procedure to generate a $D'(8, 4^2)$ from a $D(8, 2^2)$. For each column in D, we first replace all four entries of 1 with a random permutation of numbers: $1, 1, 2, 2$, and then replace all four entries of 2 with a random permutation of numbers: $3, 3, 4, 4$, thus generating a 4-level design D'. In all we have 1296 possible D's. Here is an example:

$$D = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{pmatrix}^T \Rightarrow D' = \begin{pmatrix} 1 & 2 & 1 & 2 & 3 & 4 & 4 & 3 \\ 2 & 3 & 1 & 3 & 1 & 4 & 2 & 4 \end{pmatrix}^T.$$

Let $x_{i,l}$ be the (i^{th},l^{th}) element and x_i be the i^{th} row of the initial design D. After level expansion, they are $x'_{i,l}$ and x'_i of the generated design D', respectively. Let $h_{i,j}$ be the Hamming distance (number of positions where the corresponding entries in the pair of rows are different) between rows x_i and x_j. Take $d_{il,jl} = |x_{i,l} - x_{j,l}|$. Denote the L_1-distance between two rows x_i and x_j as $d_{i,j} = \sum_{l=1}^{k} d_{il,jl}$. In this paper, we focus on constructing maximin designs in regard to the L_1-distance. Let $d_{min}(D)$ be the minimum L_1-distance among all pairs of rows in design D. In the same way, we define $h'_{i,j}$, $d'_{il,jl}$, $d'_{i,j}$ and $d_{min}(D')$ for the generated design D', respectively. For any balanced design D, we define the distance distribution as (# denotes the count)

$$B_l(D) = n^{-1}\#\{(i,j) : d_{i,j} = l; x_i, x_j \in D, i, j = 1, 2, \ldots, n\}.$$

It is easy to show that a design without repeated runs has $B_0(D) = 1$. The maximin design is defined as the one that sequentially minimizes the distance distribution $B_0(D), B_1(D)$, $B_2(D)$, $B_3(D)$, ... Designs with smaller ϕ_p values defined in (1.1) are more space-filling and have better distance distributions.

Lemma 1. *(a) For $i, j = 1, \ldots, n$ and $i \neq j$, upper and lower bounds for the L_1-distance between the i^{th} and j^{th} row in the generated design D' are*

$$md_{i,j} - (m-1)h_{i,j} \leqslant d'_{i,j} \leqslant md_{i,j} + (m-1)k.$$

(b) Upper and lower bounds for the minimum pairwise L_1-distance of the generated design D' are

$$md_{min}(D) - (m-1)h_{max}(D) \leqslant d_{min}(D') \leqslant md_{min}(D) + (m-1)k,$$

where $h_{max}(D)$ is the largest pairwise Hamming distance in design D.

Given n, s, and k, from different initial designs $D(n, s^k)$, by level expansion we can generate different sets of designs $D'(n, (ms)^k)$. By Lemma 1, the upper bound for $d_{min}(D')$ is determined by $d_{min}(D)$. If we can generate a design D'_{opt} with $d_{min}(D'_{opt}) = md_{min}(D_{Mm}) + (m-1)k$ where D_{Mm} is the maximin initial design, it is clear that D'_{opt} has the largest minimum distance among all possible D's from all possible initial designs D. In Lemma 1, the lower bound of $d_{min}(D')$ is also positively related with $d_{min}(D)$. Therefore, in order to generate good maximin designs via level expansion, initial designs with better distance distributions should be used.

From any initial design D, by level expansion we have $((n/s)!/(r!)^m)^{sk}$ pos-

sible generated designs D', where $r = n/(ms)$.

Theorem 1. *For $i, j = 1, \ldots, n$ and $i \neq j$, the expectation and variance of the pairwise L_1-distances in the generated designs D' via level expansion have the following relationship with the pairwise L_1-distance in the initial balanced design D:*

$$E(d'_{i,j}) = md_{i,j} + (k - h_{i,j})\gamma \quad \textit{and} \quad Var(d'_{i,j}) = C_{1,0} + C_{1,1}h_{i,j},$$

where $\gamma = n(m^2-1)/[3m(n-s)]$, $C_{1,0} = kn(m^2-1)(m^2n+2n-3m^2s)/[18m^2(n-s)^2]$, and $C_{1,1} = (m^2-1)[2n^2(m^2-1)-3m^2s(n-s)]/[18m^2(n-s)^2]$.

Thus the expected value of $d'_{i,j}$ is a function of both $d_{i,j}$ and $h_{i,j}$. For a 2-level design, the L_1-distance $d_{i,j}$ equals the Hamming distance $h_{i,j}$. For a design with more than 2 levels, $d_{i,j}$ is greater than or equal to $h_{i,j}$. In addition, the coefficient (m) for $d_{i,j}$ is almost three times as large as the absolute value of the coefficient (γ) for $h_{i,j}$. Therefore, the expected value of $d'_{i,j}$ is dominated by $d_{i,j}$. Generally speaking, a large $d_{i,j}$ value leads to a large $d'_{i,j}$ value on average.

When $s > 2$, we can improve designs' minimum distances by level permutation (Zhou and Xu (2014)). When permuting levels for one or more factors of a design, the pairwise Hamming distances do not change, but its pairwise L_1-distances vary. Given a design $D(n, s^k)$, we can generate in total $(s!)^k$ level-permuted designs (including isomorphic designs) and then consider all possible level expansions for each design. Let Θ denote the set of all designs generated by all level permutations and expansions.

Lemma 2. *When all possible level permutations and expansions are considered, for $i \neq j$, the expectation and variance of the pairwise L_1-distances in generated designs D' are*

$$E_\Theta(d'_{i,j}) = k\gamma + (m\frac{s+1}{3} - \gamma)h_{i,j},$$
$$Var_\Theta(d'_{i,j}) = C_{1,0} + \left(C_{1,1} + m^2\frac{(s+1)(s-2)}{18}\right)h_{i,j},$$

where γ, $C_{1,0}$, and $C_{1,1}$ are constants defined in Theorem 1.

Now we study the space-filling property for the generated design D'. For $D' \in \Theta$, let $\bar{d}' = \sum_{i\neq j=1}^{n} d'_{i,j}/(n(n-1))$ be the average distance in the generated design D'. It is easy to show that $\bar{d}' = kn(m^2s^2-1)/(3ms(n-1))$ because D' is level balanced for each column. Next, we show that the expectation of sum of squared distances in D' is minimized when the initial design is an OA. The concepts of generalized word-length pattern (GWLP) and generalized minimum

aberration (GMA) from Xu and Wu (2001) are needed to describe this result. For design $D(n, s^k)$, the GWLP is the vector $(A_1(D), A_2(D), \ldots A_k(D))$, where the value of $A_j(D)$ $(j = 1, 2 \ldots, k)$ represents the total aliasing between the general mean and all j-factor interactions in the full ANOVA model. The GMA criterion sequentially minimizes the GWLP.

Theorem 2. *When all possible level permutations and expansions are considered,*

$$E_\Theta(\sum_{i\neq j=1}^{n}(d'_{i,j})^2) = C_{2,1}A_2(D) + C_{2,0},$$

where $C_{2,1} = 2n^2(m(s+1)/3-\gamma)^2/s^2$ *and* $C_{2,0}$ *is a constant.*

From Theorem 2, we have $E_\Theta(\sum_{i\neq j=1}^{n}(d'_{i,j} - \bar{d}')^2) = E_\Theta(\sum_{i\neq j=1}^{n}(d'_{i,j})^2) - E_\Theta(\sum_{i\neq j=1}^{n}(\bar{d}')^2) = C_{2,1}A_2(D) + constant$. Since $C_{2,1} > 0$, the expectation of the variation of pairwise L_1-distances in D' is minimized when $A_2(D) = 0$. For a level balanced design, $A_1(D) = 0$. Xu and Wu (2001) showed that D is an OA of strength two if and only if $A_1(D) = A_2(D) = 0$. Thus, if the initial design is an OA of strength two or higher, generated designs tend to have small variations among all pairwise L_1-distances and large minimum pairwise L_1-distance. In other words, designs generated from OAs via level permutation and expansion tend to have robust space-filling properties.

Example 2. Consider constructing 32-run LHDs with 8 factors from five different 2-level designs with different A_2 or A_3 values. The first two designs are regular 2^{8-3} designs (with $A_2 = 0$) and the other three designs have 1, 2, 3 pairs of duplicated columns, indicated by $A_2 = 1$, 2, 3, respectively. Given a 2-level design, we randomly generated 10^5 LHDs via level permutation and expansion and computed the minimum pairwise L_1-distance for each of them. Table 1 compares the minimum, first quartile (Q1), median, third quartile (Q3) and maximum of the 10^5 minimum distances for five different initial designs. It is evident that initial designs with smaller A_2 values are more likely to generate designs with larger minimum distances via level permutation and expansion.

It is possible, but tedious, to extend Theorem 2 and link $E_\Theta(\sum_{i\neq j=1}^{n}(d'_{i,j})^r)$ with the values of $A_2(D), \ldots, A_r(D)$ for $r > 2$, similar to Theorem 4 of Zhou and Xu (2014). We do not pursue this here.

3. Maximin Distance Level Expansion (MDLE) Method

3.1. Procedures of MDLE

Based on the results in the previous section, we propose the MDLE method

Table 1. Summary of minimum pairwise L_1-distances.

Design	(A_1, A_2, A_3, A_4)	Min	Q_1	Median	Q_3	Max
Design 1	$(0, 0, 0, 3)$	15	36	39	42	52
Design 2	$(0, 0, 1, 2)$	15	35	38	41	51
Design 3	$(0, 1, 0, 2)$	14	33	36	39	49
Design 4	$(0, 2, 0, 1)$	11	32	35	38	48
Design 5	$(0, 3, 0, 3)$	10	29	32	34	45

that combines both level permutation and expansion. The method starts from OAs, or nearly-OAs if the corresponding OAs are not available, and expands their levels with one or more phases. Here we first discuss how to construct maximin designs from OAs with only one phase of level expansion. Refer to Section 5 for generalizations. To generate $D'(n, (ms)^k)$, we start from an $OA(n, s^{k_0})$ with $k_0 \geq k$. The MDLE method has three steps.

1. Select the GMA k-column subset from an $OA(n, s^{k_0})$ and denote this design by $D(n, s^k)$.

2. If $s > 2$, perform level permutation for design D from Step 1. Select the maximin design and denote it by $D_p(n, s^k)$.

3. For each column in D_p from Step 2, replace the n/s positions of entry l $(l = 1, 2, \ldots, s)$ by a random sequence of $n/(ms)$ replicates of numbers: $(l-1)m+1, (l-1)m+2 \ldots, (l-1)m+m$. Select the maximin design as the final design $D'(n, (ms)^k)$.

We usually start from saturated $OA(n, s^{k_0})$, or nearly saturated OAs with $k_0 \leq (n-1)/(s-1)$. When $k_0!/(k!(k_0-k)!)$ is small, we can enumerate and compare all subsets to find the GMA subset in Step 1; otherwise, we adopt a simple searching method: randomly generate and compare n_{gma} subsets and select the GMA subset where n_{gma} ranges from 1,000 to 5,000 based on the design size and computation available. We use the concept of minimum moment aberration (Xu (2003)) to efficiently determine GMA subsets. For 2-level regular designs we choose existing minimum aberration designs from the R package FrF2. In Steps 2 and 3, we adopt a threshold accepting (TA) algorithm modified from that of Dueck and Scheuer (1990). Compared with the simulated annealing algorithm by Kirkpatrick (1984) and Morris and Mitchell (1995), TA converges faster.

To implement the TA algorithm, we need to specify neighbour designs $\mathcal{N}(D_c)$ for a current design D_c in Steps 2 and 3. To generate neighbour designs $\mathcal{N}(D_c)$

in Step 2, we randomly choose two levels from a randomly chosen column of D_c and exchange all elements of these two levels. In Step 3, we define neighbour designs $\mathcal{N}(D_c)$ by exchanging the levels in two positions from a randomly chosen column of D_c, where these two positions have different values in D_c and the same value in D_p from Step 2.

We choose $\phi(D) = \phi_p(D)$ defined in (1.1) as the objective function to be minimized in our TA algorithm. The pseudo code for our TA algorithm is given in Algorithm 1. Based on the design size and time limits, typically we set n_{seq} equal to 2,000, choose n_{rounds} from 30 to 75, and choose n_{steps} from 3,000 to 7,500.

Algorithm 1 Pseudo code for threshold accepting (TA) algorithm

Initialize n_{seq} (number of steps to compute threshold sequences)
Initialize n_{rounds} (number of rounds) and n_{steps} (number of steps)
Initialize a starting design D_c and let $D_{min} = D_c$
for $i = 1$ to n_{seq} **do**
  Generate a new design D_n from its neighbors $\mathcal{N}(D_c)$ and let $\Delta_i = |\phi(D_c) - \phi(D_n)|$
end for
Compute the empirical distribution of Δ_i , $i = 1, 2, \ldots, n_{seq}$, denoted it as $F(x)$
for $r = 1$ to n_{rounds} **do**
  Generate threshold $\tau_r = F^{-1}(0.5(1 - r/n_{rounds}))$
  for $j = 1$ to n_{steps} **do**
    Generate a new design D_n from the neighbors $\mathcal{N}(D_c)$ and let $\delta = \phi(D_n) - \phi(D_c)$
    if $\delta < \tau_r$ **then** let $D_c = D_n$
    if $\phi(D_c) < \phi(D_{min})$ **then** let $D_{min} = D_c$
  end for
end for
Return D_{min}

3.2. Justifications for the procedures of MDLE

Zhou and Xu (2014) showed that from GMA initial designs we can generate designs with the best distance distributions on average via all possible level permutations. Thus, choosing GMA design D in Step 1 can benefit finding maximin design D_p in Step 2. Further, Lemma 1 and Theorem 1 in Section 2 show that from the maximin design D_p we can generate D's with the best distance distributions on average in Step 3. By Theorem 2, GMA initial designs minimize the expectation of the variation of distances in generated designs via level permutation and expansion. Therefore, this 3-step procedure is robust and efficient in generating good space-filling designs.

We further justify our method from a geometric point of view. Ba, Myers and Brenneman (2015) discussed a relevant geometric idea, but it only applies to SLHDs with multiple slices. Here we discuss the situations for fractional factorial designs and general LHDs, including SLHDs with only one slice. We relate a design's geometric structure with its GMA structure. To get a space-filling n-run and k-factor design, a straightforward idea is to divide the design space equally into n k-dimensional lattices, put one point in each lattice, and properly adjust each point's position within its lattice. This geometric structure of "one point per lattice" can be achieved by performing level expansion to full factorial initial designs. For example, see OALHDs in Figure 1(b), (c), and (d) generated by the level expansion process from full factorial $D(9, 3^2)$. These designs have only one point per lattice formed by the solid lines, but the positions of points within the lattices are different. By either the level permutation or level expansion process, the "one point per lattice" structure is not changed, but their positions within the lattice are adjusted, and thus the distance distribution of the design can be improved. By our MDLE method with full factorial initials, we can find the design with best distance distribution while keeping the "one point per lattice" structure, e.g., design (d) in Figure 1.

As a generalization, when $n < s^k$, an initial design with the most low-dimensional projections that are full factorials is ideal for our MDLE method, and GMA designs have such a property in many cases. Box and Hunter (1961) pointed out that any p-dimensional ($p < r$) projection of a 2-level regular design with resolution r is a full-factorial design. Chen (1998) proved that for a 2-level regular design, $\binom{k}{p} - \sum_{j=r}^{p} \binom{k-j}{p-j} A_j(D)$ p-dimensional projections ($p = r, \ldots, \lfloor r + (r-1)/2 \rfloor$) are full-factorial designs. Under these cases, since the GMA initials have largest resolutions and sequentially minimize $A_j(D)$ ($j = 1, 2, \ldots k$), they have the most parts that are full-factorials in p-dimensional projection spaces ($p \leqslant \lfloor r+(r-1)/2 \rfloor$). GMA nonregular designs have similar projection properties; see Xu, Phoa and Wong (2009) for a review. As a result, GMA initial designs tend to generate better space-filling designs via level expansion.

4. Results and Comparisons

4.1. Construction of maximin LHDs

First, we compared our MDLE method with the ordinary level expansion (OLE) method of Tang (1993) and Leary, Bhaskar and Keane (2003) in generating maximin OALHDs. The OLE method first randomly selects a required number

Table 2. Comparisons of constructions of maximin LHDs.

		MDLE			OLE	OMLHD		SLHD	
n	k	d(pair)	ψ_p	d_2	d(pair)	d(pair)	ψ_p	d(pair)	d_2
27	9	**72(2)**	**0.012**	**28.8**	68(5)	60(1)	0.025	63(1)	28.2
32	20	**205(1)**	**0.005**	**55.6**	205(2)	177(1)	0.012	190(1)	55.1
54	5	**54(1)**	**0.0311**	**28.1**	45(1)	47(2)	0.0393	44(1)	27.8
54	20	**329(1)**	**0.0034**	**88.4**	317(1)	279(1)	0.0083	294(1)	88.2
54	25	**425(3)**	**0.0022**	**102.7**	399(1)	360(1)	0.012	382(1)	100.9
64	6	**83(1)**	**0.0209**	**40.6**	61(1)	70(3)	0.0299	67(1)	39.1
64	20	**378(1)**	**0.0034**	**105.2**	369(1)	310(1)	0.0084	340(1)	102.6
64	40	**813(1)**	**0.0025**	**157.4**	804(1)	698(1)	0.0048	771(1)	155.7
81	8	**152(1)**	**0.0111**	**64.2**	102(1)	123(2)	0.0198	121(1)	62.7
81	25	**604(2)**	**0.0022**	**147.9**	577(1)	504(1)	0.0028	540(1)	146.7
81	40	**1,016(1)**	**0.0016**	**194.9**	962(1)	899(1)	0.0016	934(1)	194.5
125	10	**284(2)**	**0.0072**	**111.8**	199(1)	237(3)	0.0136	232(1)	110.6
125	23	**797(1)**	**0.0021**	**206.9**	640(1)	668(1)	0.0021	726(1)	206.8
125	31	**1,126(1)**	**0.0014**	**251.1**	971(1)	955(1)	0.0076	1,038(1)	250.8
128	12	**378(1)**	**0.0051**	**135.5**	284(1)	314(1)	0.0092	313(1)	132.6
128	49	**1,893(1)**	**0.0014**	**337.6**	1,873(1)	1,643(1)	0.0057	1,801(1)	335.3
128	64	**2,512(1)**	**0.0017**	**395.2**	2,479(1)	2,239(1)	0.0061	2,497(1)	392.1

of columns from a saturated or nearly saturated OA to be the initial design, then performs level expansion, and searches for the maximin generated LHDs. In order to make a fair comparison, we replaced the simulated annealing algorithm in Leary, Bhaskar and Keane (2003) with our more efficient TA algorithm.

Table 2 lists some arbitrarily chosen cases for comparison, where "d(pair)" represents the minimum pairwise L_1-distance (and the number of pairs with the minimum distance). For all tables, we use bold font to represent the better results. For the 32, 64 and 128-run cases, the MDLE method starts from the respective 2-level minimum aberration initial designs that are available in R package FrF2, whereas the OLE method starts from the corresponding saturated OAs. For the 27, 54, 81, and 125-run cases, both methods start from initial designs $OA(27, 3^{13})$, $OA(54, 3^{25})$, $OA(81, 3^{40})$, and $OA(125, 5^{31})$, respectively; these are available in R package DoE.base. All codes were run in R on a laptop with an Intel 2.50GHz I7 CPU. Time used by our MDLE method ranged from 5 minutes to an hour for the different cases here. For all cases, we let the OLE method use at least twice as much time as the MDLE method.

From Table 2, it is clear that the MDLE method generates better OALHDs than the OLE method for all cases. Compared with the MDLE method, the OLE method only includes Step 3, but does not have the first two steps of the

MDLE method. Thus, Table 2 shows the usefulness of the first two steps in the MDLE method which provides good initial designs for level expansion. When the MDLE method starts with 2-level initial designs, Step 2 is skipped since level permutations do not change designs' distance distributions. Thus, the usefulness of Step 1 alone can be seen from the 32, 64 and 128-run cases in Table 2. From the 54-run/25-factor, 81-run/40-factor and 125-run/31-factor cases, we can see the usefulness of Step 2 alone since Step 1 is skipped.

Next, we compared our MDLE method with the OMLHD method of Joseph and Hung (2008) and the R package SLHD of Ba, Myers and Brenneman (2015) in generating space-filling LHDs. Joseph and Hung (2008) proposed the multi-objective criterion

$$\psi_p = \omega\rho^2 + (1-\omega)\frac{\phi_p - \phi_{p,lb}}{\phi_{p,ub} - \phi_{p,lb}}, \tag{4.1}$$

where ϕ_p is defined in (1.1) with $p = 15$, ρ^2 is the average of squared column-wise correlations, ω is the weight which is set to 0.5, $\phi_{p,lb}$ and $\phi_{p,ub}$ are the smallest and largest possible ϕ_p values. Joseph and Hung (2008) used a modified simulated annealing algorithm to search for LHDs that minimize ψ_p values. Table 2 lists some cases for comparison, where ψ_p is defined in (4.1) and d_2 represents designs' minimum pairwise L_2-distances. For the OMLHD method, we ran the code from Y. Hung's homepage (`http://stat.rutgers.edu/home/yhung/index.htm`) with $nstart = 5$ and default settings, and chose the best results. For the SLHD method, we ran the command `maximinSLHD` with slice parameter $t = 1$ and default settings for 200 times, and chose the best results.

For all cases in Table 2, the MDLE method generates better space-filling designs than the OMLHD method in regard to both the L_1-distance and the ψ_p criterion. Our MDLE method searches designs toward the L_1-distance alone. Designs from our method have small pairwise correlations, since they can always collapse to OAs or nearly OAs. Further, the MDLE method generates better maximin designs than the SLHD method under both the L_1- and L_2-distances. In order to make a fair comparison with the SLHD package, the ϕ_p criterion used in the MDLE method adopted the L_1- and L_2-distance for each case respectively. Our MDLE method was implemented in R whereas the SLHD and OMLHD methods were implemented in C++. The R package SLHD provides an interface to call the C++ program. Our MDLE method used less than half of the time used by the SLHD and OMLHD methods, although C++ is more efficient than R in terms of computation.

Table 3. Comparisons in the constructions of four-level maximin FFDs.

(a)

		MDLE	LP
n	k	d(pair)	d(pair)
16	3	2(12)	2(12)
16	4	4(60)	4(56)
16	5	4(1)	4(4)
32	3	2(156)	2(156)
32	4	2(8)	2(8)
32	5	4(100)	4(106)
32	6	5(48)	5(58)
32	7	6(24)	6(28)
32	8	8(132)	8(128)
32	9	**9(62)**	8(6)

(b)

		MDLE	LP
n	k	d(pair)	d(pair)
48	10	**9(6)**	8(3)
48	13	**13(15)**	12(10)
64	9	**8(395)**	6(38)
64	11	**10(77)**	9(12)
80	7	**5(177)**	4(48)
80	11	**9(1)**	8(29)
128	29	**30(42)**	29(79)
128	40	**43(1)**	40(2)

4.2. Construction of maximin fractional factorial designs

First, we compared our MDLE method with the level permutation (LP) method of Zhou and Xu (2014) in generating maximin fractional factorial designs (FFDs). Zhou and Xu (2014) included a table of 10 maximin designs with $n \leqslant 32$ that are comparable here, and we list them in Table 3 (a). We further selected another eight larger cases with $n \geqslant 48$ in Table 3 (b) to compare the two methods. All designs are 4-level FFDs. For the MDLE method, in the 16, 32, 64, and 128-run cases, 2-level minimum aberration initial designs were used; in the 48 and 80-run cases, $OA(48, 2^{47})$ and $OA(80, 2^{79})$ were used as the initial designs. For the LP method, in the 48, 64, 80 and 128-run cases, initial designs $OA(48, 4^{13})$, $OA(64, 4^{11})$, $OA(80, 4^{11})$ and $OA(128, 4^{40})$ were used, respectively. Both methods' codes were run in R. For all cases, the LP method used at least twice as much time as the MDLE method.

In Table 3 (a), for the first nine cases both methods generated designs with the same minimum pairwise distances. For the last case in Table 3 (a) and all cases in Table 3 (b), the MDLE method outperforms the LP method. Furthermore, the LP method relies on existing OA initials that have the same number of runs, factors and levels as the generated designs. These OAs are often difficult to find or even do not exist. For example, there is no $OA(24, 6^8)$ that can be used to generate maximin $D'(24, 6^8)$. Compared with the LP method, our MDLE method has more flexibility in design size, since we can start from 2-level designs to generate multi-level designs. For example, we can start from a 2-level $OA(24, 2^{23})$ to generate the 24-run/6-level maximin design with up to 23 factors.

Table 4. Comparison in the construction of four-level and six-level uniform designs.

			MDLE		MDLE-CD		UD-page designs	
n	k	s	CD	d(pair)	CD	d(pair)	CD	d(pair)
32	7	4	0.074	**6(18)**	**0.070**	5(6)	0.071	4(1)
32	13	4	**0.0343**	**13(5)**	**0.0343**	**13(5)**	0.0344	12(2)
40	13	4	0.3186	**13(4)**	**0.3067**	12(5)	0.3068	11(1)
40	15	4	0.5080	**16(56)**	**0.4969**	13(1)	0.4987	14(1)
48	11	4	0.1841	**10(11)**	**0.1758**	8(1)	0.1767	7(1)
48	15	4	0.461	**15(13)**	**0.447**	13(1)	0.449	12(1)
36	12	6	0.1744	**20(15)**	**0.1673**	19(14)	0.1691	17(1)
48	12	6	0.1416	**19(21)**	**0.1362**	16(2)	0.1374	16(1)
54	9	6	0.0601	**12(13)**	**0.0564**	8(1)	0.0568	10(3)
54	12	6	0.1362	**17(14)**	**0.1268**	16(1)	0.1299	16(2)
60	9	6	0.0576	**12(11)**	**0.0544**	8(1)	0.0546	9(2)

Next, we compared designs from our MDLE method with some existing uniform designs listed on the uniform design homepage (`http://uic.edu.hk/isci/`). These uniform designs (UD-page designs) were searched by Kaitai Fang and his collaborators toward the centered L_2-discrepancy (CD) criterion where smaller CD values indicate more space-filling designs. In order to make a fair comparison, in Table 4 we also include a modified version of our MDLE method (MDLE-CD) which searches best designs using the CD criterion in Step 3. We selected some 4-level and 6-level cases for comparison in Table 4. Both the MDLE and MDLE-CD methods started from the initial designs $OA(32, 2^{31})$, $OA(40, 2^{39})$, and $OA(48, 2^{47})$ to generate the 4-level designs, and $OA(36, 3^{13})$, $OA(48, 2^{47})$, $OA(54, 3^{18})$, and $OA(60, 2^{30})$ to generate the 6-level designs for the corresponding cases.

Table 4 shows that designs by the MDLE method are always better than the UD-page designs in regard to maximin distance criterion. Designs from the MDLE-CD method are better than the UD-page designs toward the CD criterion.

5. Multi-Phase MDLE Method

In constructing maximin designs $D'(n, (ms)^k)$ from initial designs $D(n, s^k)$, when m is very large, the one-phase MDLE method introduced in Section 3 is not efficient because level expansion produces too many designs. In addition, when n and k are too large given the computation constraint, we need to further restrict the searching space in the MDLE method. Under such situations, we can apply a multi-phase MDLE method. The multi-phase MDLE method shares the same

Table 5. Comparison of one-phase and two-phase MDLE methods in constructing LHDs.

		One-phase		Two-phase		
n	k	d(pair)	time	d(pair)	time	sequence
27	3	14(4)	67	14(5)	107	$3 \to 9 \to 27$
32	5	37(1)	103	37(3)	101	$2 \to 8 \to 32$
64	6	**83(1)**	**301**	81(1)	306	$2 \to 8 \to 64$
81	4	50(1)	478	50(3)	490	$3 \to 9 \to 81$
125	3	**38(5)**	**603**	37(3)	950	$5 \to 25 \to 125$
32	15	**151(1)**	**211**	150(2)	218	$2 \to 8 \to 32$
54	12	173(1)	886	**178(2)**	**806**	$3 \to 9 \to 54$
54	20	309(1)	1,346	**322(2)**	**1,275**	$3 \to 9 \to 54$
64	40	805(1)	1,062	**810(1)**	**995**	$2 \to 8 \to 64$
81	40	1,005(1)	1,479	**1,014(1)**	**936**	$3 \to 9 \to 54$
125	31	1,111(1)	2,085	**1,116(1)**	**1,548**	$5 \to 25 \to 125$

Note: Time in seconds.

Steps 1 and 2 as the one-phase MDLE. The difference lies in Step 3: instead of directly generating $D'(n, (ms)^k)$ from $D_p(n, s^k)$, we gradually expand the levels from s to ms in multiple phases. For example, in a two-phase MDLE method with $m = m_1 m_2$, in Step 3 we first generate maximin design $D_1(n, (m_1 s)^k)$ via level expansion from D_p in Step 2; then from D_1 we generate maximin design $D_2(n, (m_2 m_1 s)^k)$ which is $D'(n, (ms)^k)$ via level expansion again. It is straightforward to generalize and justify this process with more phases in both theory and geometry, as in Section 3.2.

The more phases we use, the more restrictions are put on the searching space. The number of designs needed to be compared decreases dramatically with multiple phases. For example, to generate $D'(16, 8^2)$ from $D(16, 2^2)$ for the one-phase MDLE method, we have in total about 4×10^{13} possible D's to be compared; for the two-phase MDLE method, we only need to compare about 1.7×10^6 designs. More restrictions on the searching space also means that we are more likely to miss good designs, at least in theory. In practice, with limited computations, the multiple-phase method can be more efficient than the one-phase method, especially for large designs.

Table 5 compares the one-phase and two-phase MDLE methods in generating maximin LHDs with time constraints. For both methods, we started from the respective full factorial designs for the first five cases, the minimum aberration designs for the 32- and 64-run cases, $OA(54, 3^{24})$, $OA(81, 3^{40})$, and $OA(125, 5^{31})$ for the rest of cases, respectively. For the last five cases, where the numbers of runs and factors are relatively large, the two-phase method generates better

designs in a shorter time than the one-phase method. Given adequate computation time, the one-phase method eventually generates better designs than the two-phase method; see the last four cases in Table 5 and corresponding results in Table 2 where we ran the one-phase MDLE method for a longer time.

When OAs with different levels exist, generally speaking, it is better to use OA initials with fewer levels given abundant computations. As an illustration, for the 128-run/12-factor case, starting from the 2-, 4- and 8-level OA initials, the one-phase MDLE method generates LHDs with the minimum L_1-distances of 378, 375 and 368, respectively. The 2-level OA initial gives the best result here, but requires more than 5 times the computations to achieve a stable result compared with the 8-level initial. Since any 8-level OA can be collapsed to a 2-level OA, the MDLE method is less likely to miss good results from 2-level OAs. For large designs with computation constraints, OA initials with larger levels may work better since the searching space is much smaller and less phases are needed.

6. Summary

We propose the MDLE method which can efficiently generate maximin LHDs and maximin fractional factorial designs. To justify our method, we establish a relationship of the L_1-distance distributions between the initial and generated designs via level expansion. When all possible level permutations of the initial designs are considered, we give expectations and variances of the pairwise L_1-distances for the generated designs.Various comparisons show that our MDLE method outperforms the ordinary level expansion process, the OMLHD algorithm, the SLHD package, and the level permutation method. We also find many more space-filling designs compared to the existing uniform designs.

The MDLE method is easy to generalize for mixed-level cases. Starting from a mixed-level initial design, we can individually set the level expansion path for each factor. In this way, we can generate mixed-level factorial designs. Although the MDLE method cannot generate maximin designs with any run size, it works well from nearly OAs (Xu (2002)) or optimal supersaturated designs (Xu and Wu (2005)), when suitable OA initials are not available.

Acknowledgment

This research was supported in part by National Science Foundation DMS Grant 1407560. The authors thank two reviewers for their helpful comments.

Appendix: Proofs

Proof of Lemma 1. (a) For $i \neq j$, when $x_{i,l} = x_{j,l}$, $d'_{il,jl}$ takes on values of $0, 1, \ldots, m-1$; when $x_{i,l} \neq x_{j,l}$, $d'_{il,jl}$ takes on values of $m(d_{il,jl}-1)+1, \ldots, m(d_{il,jl}-1)+2m-1$. Therefore, the smallest possible $d'_{i,j}$ value is

$$\min\ d'_{i,j} = 0 * (k - h_{i,j}) + \sum_{l=1}^{h_{i,j}} (m(d_{il,jl} - 1) + 1) = md_{i,j} - (m-1)h_{i,j}$$

and the largest possible $d'_{i,j}$ value is

$$\max\ d'_{i,j} = (m-1)(k - h_{i,j}) + \sum_{l=1}^{h_{i,j}} (m(d_{il,jl} - 1) + 2m - 1) = md_{i,j} + k(m-1).$$

Thus, we have $md_{i,j} - (m-1)h_{i,j} \leqslant d'_{i,j} \leqslant md_{i,j} + k(m-1)$.

(b) Let x_a and x_b be the pair of rows in design D that forms the minimum pairwise L_1-distance $d_{min}(D)$ (there could be many such pairs). Let x'_c and x'_d be the pair of rows in design D' that forms the minimum pairwise L_1-distance $d_{min}(D')$ (there could be many such pairs). Then

$$\begin{aligned} d_{min}(D') = d'_{c,d} &\leqslant d'_{a,b} \leqslant md_{a,b} + k(m-1) = md_{min}(D) + (m-1)k, \\ d_{min}(D') = d'_{c,d} &\geqslant md_{c,d} - (m-1)h_{c,d} \geqslant md_{c,d} - (m-1)h_{max}(D) \\ &\geqslant md_{a,b} - (m-1)h_{max}(D) = md_{min}(D) - (m-1)h_{max}(D). \end{aligned}$$

Thus, we have $md_{min}(D) - (m-1)h_{max}(D) \leqslant d_{min}(D') \leqslant md_{min}(D) + (m-1)k$.

Proof of Theorem 1. We first calculate the probability distribution for $d'_{il,jl}$ with its range given in Lemma 1. For $i \neq j$, when $x_{i,l} = x_{j,l}$, the probability distribution is

$$\begin{aligned} P(d'_{il,jl} = 0) &= \frac{m\binom{n/(ms)}{2}}{m(m-1)(n/(ms))^2 + m\binom{n/(ms)}{2}} = \frac{n - ms}{m(n-s)}, \\ P(d'_{il,jl} = t) &= \frac{2(m-t)(n/(ms))^2}{m(m-1)(n/(ms))^2 + m\binom{n/(ms)}{2}} = \frac{2n(m-t)}{m^2(n-s)} \end{aligned}$$

for $t = 1, 2, \ldots, m-1$. Thus,

$$E(d'_{il,jl}) = \sum_{t=1}^{m-1} tP(d'_{il,jl} = t) = \frac{n(m^2-1)}{3m(n-s)} = \gamma, \tag{A.1}$$

$$E((d'_{il,jl})^2) = \sum_{t=1}^{m-1} t^2 P(d'_{il,jl} = t) = \frac{n(m^2-1)}{6(n-s)} = \frac{m}{2}\gamma. \tag{A.2}$$

When $x_{i,l} \neq x_{j,l}$, the probability distribution is

$$P(d'_{il,jl} = d_0 + t) = \frac{t+1}{m^2}, \text{ for } t = 0, 1, \ldots, m-1,$$
$$P(d'_{il,jl} = d_0 + t) = \frac{2m-t-1}{m^2}, \text{ for } t = m, \ldots, 2m-2,$$

where $d_0 = m(d_{il,jl} - 1) + 1$. It is straightforward to verify that

$$E(d'_{il,jl}) = \sum_{t=0}^{2m-2} (d_0 + t) P(d'_{il,jl} = d_0 + t) = m d_{il,jl}, \tag{A.3}$$

$$E((d'_{il,jl})^2) = \sum_{t=0}^{2m-2} (d_0 + t)^2 P(d'_{il,jl} = d_0 + t) = m^2 d_{il,jl}^2 + \frac{m^2-1}{6}. \tag{A.4}$$

It is clear that $d_{il,jl} = 0$ when $x_{i,l} = x_{j,l}$. Combining (A.1) and (A.3), we have

$$E(d'_{i,j}) = \sum_{l=1}^{k} E(d'_{il,jl}) = \sum_{l=1}^{k} m d_{il,jl} + (k - h_{i,j})\gamma = m d_{i,j} + (k - h_{i,j})\gamma. \tag{A.5}$$

Next, combining (A.2) and (A.4), we have

$$E\left(\sum_{l=1}^{k} (d'_{il,jl})^2\right) = \sum_{l=1}^{k} E((d'_{il,jl})^2) = m^2 \sum_{l=1}^{k} d_{il,jl}^2 + \frac{m^2-1}{6} h_{i,j} + (k - h_{i,j})\frac{m\gamma}{2}. \tag{A.6}$$

Further, we have

$$E((d'_{i,j})^2) = E\left(\left(\sum_{l=1}^{k} d'_{il,jl}\right)^2\right) = E\left(\sum_{l=1}^{k} (d'_{il,jl})^2\right) + E\left(\sum_{p \neq q=1}^{k} d'_{ip,jp} d'_{iq,jq}\right). \tag{A.7}$$

Since $d'_{ip,jp}$ and $d'_{iq,jq}$ $(p \neq q)$ are independently determined by the p^{th} and q^{th} columns in the initial design D, with (A.1) and (A.3), we have

$$E\left(\sum_{p \neq q=1}^{k} d'_{ip,jp} d'_{iq,jq}\right) = \sum_{p \neq q=1}^{k} E(d'_{ip,jp}) E(d'_{iq,jq}) = m^2 \sum_{p \neq q=1}^{k} d_{ip,jp} d_{iq,jq} + \\ + 2(k - h_{i,j})\gamma \sum_{l=1}^{k} m d_{il,jl} + (k - h_{i,j})(k - h_{i,j} - 1)\gamma^2. \tag{A.8}$$

Combining (A.6), (A.7), and (A.8), after some simple algebra, we have

$$Var(d'_{i,j}) = E\left((d'_{i,j})^2\right) - \left(E(d'_{i,j})\right)^2 = C_{1,0} + C_{1,1} h_{i,j},$$

where $C_{1,0}$ and $C_{1,1}$ are constants given in Theorem 1.

Proof of Lemma 2. We need to distinguish two types of operations: level permutation and level expansion. Let σ denote a level permutation and π denote a level expansion. Let E_σ denote the expectation toward designs generated by all possible level permutations and E_π denote the expectation toward designs generated by all level expansions. As we perform level permutation first and level expansion second, using the properties of conditional expectations, we have

$$E_\Theta(d'_{i,j}) = E_\sigma[E_\pi(d'_{i,j}|\sigma)], \tag{A.9}$$

$$Var_\Theta(d'_{i,j}) = E_\sigma[Var_\pi(d'_{i,j}|\sigma)] + Var_\sigma[E_\pi(d'_{i,j}|\sigma)]. \tag{A.10}$$

For a given level permutation σ, let $d^\sigma_{i,j}$ denote the L_1-distance of a level permuted design generated by σ. Level permutation does not change pairwise Hamming distances of a design. Applying Theorem 1 to each level permutation σ, we have

$$E_\pi(d'_{i,j}|\sigma) = md^\sigma_{i,j} + (k - h_{i,j})\gamma, \tag{A.11}$$

$$Var_\pi(d'_{i,j}|\sigma) = C_{1,0} + C_{1,1}h_{i,j}. \tag{A.12}$$

Similar to the proof of Theorem 1, when considering all possible level permutations,

$$E_\sigma(d^\sigma_{i,j}) = \frac{s+1}{3}h_{i,j}, \tag{A.13}$$

$$Var_\sigma(d^\sigma_{i,j}) = E_\sigma((d^\sigma_{i,j})^2) - [E_\sigma(d^\sigma_{i,j})]^2 = \frac{(s+1)(s-2)}{18}h_{i,j}. \tag{A.14}$$

Combining (A.9), (A.11), and (A.13), we have

$$\begin{aligned} E_\Theta(d'_{i,j}) &= E_\sigma[md^\sigma_{i,j} + (k - h_{i,j})\gamma] = m\frac{s+1}{3}h_{i,j} + (k - h_{i,j})\gamma \\ &= k\gamma + (m\frac{s+1}{3} - \gamma)h_{i,j}. \end{aligned} \tag{A.15}$$

Combining (A.10), (A.11), (A.12), and (A.14), we have

$$\begin{aligned} Var_\Theta(d'_{i,j}) &= E_\sigma[C_{1,0} + C_{1,1}h_{i,j}] + Var_\sigma[md^\sigma_{i,j} + (k - h_{i,j})\gamma] \\ &= C_{1,0} + C_{1,1}h_{i,j} + m^2 Var_\sigma[d^\sigma_{i,j}] \\ &= C_{1,0} + (C_{1,1} + m^2\frac{(s+1)(s-2)}{18})h_{i,j}. \end{aligned} \tag{A.16}$$

Proof of Theorem 2. From (A.15) and (A.16), we have

$$\begin{aligned} E_\Theta(\sum_{i\neq j=1}^{n}(d'_{i,j})^2) &= \sum_{i\neq j=1}^{n} E_\Theta((d'_{i,j})^2) = \sum_{i\neq j=1}^{n}[Var_\Theta(d'_{i,j}) + (E_\Theta(d'_{i,j}))^2] \\ &= \sum_{i\neq j=1}^{n}[C_{1,0} + (C_{1,1} + m^2\frac{(s+1)(s-2)}{18})h_{i,j}] \end{aligned}$$

$$+ \sum_{i \neq j=1}^{n} [k\gamma + (m\frac{s+1}{3} - \gamma)h_{i,j}]^2. \tag{A.17}$$

Xu (2003) showed that the GWLP is related to moments of Hamming distances. In particular, for a balanced design with $A_1(D) = 0$, we have the following relationships:

$$\sum_{i \neq j=1}^{n} h_{i,j} = \frac{kn^2(s-1)}{s}, \tag{A.18}$$

$$\sum_{i \neq j=1}^{n} h_{i,j}^2 = \frac{n^2}{s^2}\{2A_2(D) + (s-1)k[1 + (s-1)k]\}. \tag{A.19}$$

Then the result follows from (A.17), (A.18), and (A.19).

References

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

Box, G. E. and Hunter, J. S. (1961). The 2^{k-p} fractional factorial designs. *Technometrics* **3**, 311–351.

Chen, H. (1998). Some projective properties of fractional factorial designs. *Statistics & Probability Letters* **40**, 185–188.

Cioppa, T. M. and Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* **49**, 45–55.

Dueck, G. and Scheuer, T. (1990). Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing, *Journal of Computational Physics* **90**, 161–175.

Fang, K.-T., Li, R. and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*, CRC Press.

Hickernell, F. J. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of Computation* **67**, 299–322.

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

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

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics* **34**, 975–986.

Leary, S., Bhaskar, A. and Keane, A. (2003). Optimal orthogonal-array-based Latin hypercubes. *Journal of Applied Statistics* **30**, 585–598.

Lin, C. D., Mukerjee, R. and Tang, B. (2009). Construction of orthogonal and nearly orthogonal Latin hypercubes. *Biometrika* **96**, 243–247.

Lin, C. D. and Tang, B. (2015). Latin hypercubes and space-filling designs, in *Handbook of Design and Analysis of Experiments*, eds. Dean, A., Morris, M., Stufken, J. and Bingham,

D., New York: Chapman and Hall/CRC, pp. 593–625.

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

Morris, M. D. and Moore, L. M. (2015). Design of computer experiments: Introduction and Background, in *Handbook of Design and Analysis of Experiments*, eds. Dean, A., Morris, M., Stufken, J. and Bingham, D., New York: Chapman and Hall/CRC, pp. 577–591.

Santner, T. J., Williams, B. J. and Notz, W. I. (2013). *The Design and Analysis of Computer Experiments*, New York: Springer.

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

Sun, F., Liu, M.-Q. and Lin, D. K. J. (2010). Construction of orthogonal Latin hypercube designs with flexible run sizes. *Journal of Statistical Planning and Inference* **140**, 3236–3242.

Tang, B. (1993). Orthogonal array-based Latin hypercubes, *Journal of the American Statistical Association* **88**, 1392–1397.

Tang, Y. and Xu, H. (2013). An effective construction method for multi-level uniform designs. *Journal of Statistical Planning and Inference* **143**, 1583–1589.

Tang, Y., Xu, H. and Lin, D. K. J. (2012). Uniform fractional factorial designs. *The Annals of Statistics* **40**, 891–907.

Xu, H. (2002). An algorithm for constructing orthogonal and nearly orthogonal arrays with mixed levels and small runs. *Technometrics* **44**, 356–368.

Xu, H. (2003). Minimum moment aberration for nonregular designs and supersaturated designs. *Statistica Sinica*, 691–708.

Xu, H., Phoa, F. K. H. and Wong, W. K. (2009). Recent developments in nonregular fractional factorial designs. *Statistics Surveys* **3**, 18–46.

Xu, H. and Wu, C. F. J. (2001). Generalized minimum aberration for asymmetrical fractional factorial designs. *The Annals of Statistics* **29**, 1066–1077.

Xu, H. and Wu, C. F. J. (2005). Construction of optimal multi-level supersaturated designs. *The Annals of Statistics*, **33**, 2811–2836.

Yang, J. and Liu, M.-Q. (2012). Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs. *Statistica Sinica* **22**, 433–442.

Zhou, Y.-D. and Xu, H. (2014). Space-filling fractional factorial designs. *Journal of the American Statistical Association* **109**, 1134–1144.

Department of Statistics, University of California, Los Angeles, CA 90095-1554, USA.

E-mail: xiaoqian1990v@gmail.com

Department of Statistics, University of California, Los Angeles, CA 90095-1554, USA.

E-mail: hqxu@stat.ucla.edu

(Received September 2016; accepted February 2017)