Statistica Sinica **26** (2016), 675-690
doi:http://dx.doi.org/10.5705/ss.202015.0019

A GENERAL CONSTRUCTION FOR SPACE-FILLING LATIN HYPERCUBES

C. Devon Lin and L. Kang

Queen's University and Illinois Institute of Technology

Abstract: We propose a general method for constructing Latin hypercubes of flexible run sizes for computer experiments. The method makes use of arrays with a special structure and Latin hypercubes. By using different such arrays and Latin hypercubes, the proposed method produces various types of Latin hypercubes including orthogonal and nearly orthogonal Latin hypercubes, sliced Latin hypercubes, and Latin hypercubes in marginally coupled designs. In addition, the proposed algebraic design construction is particularly efficient as it does not need any optimization search but still produces Latin hypercubes whose space-filling properties are comparable with those generated by the common and latest methods in the literature.

Key words and phrases: Completely resolvable, computer experiment, fractional factorial design, orthogonal array, orthogonality, projection.

1. Introduction

In computer experiments, space-filling designs that aim to spread the design points evenly over the entire design space are generally used. A commonly used class of space-filling designs is the Latin hypercube. Their popularity comes from the feature that when projecting an n-point design onto any dimension, there is exactly one point in each of n equally-spaced intervals. This feature of Latin hypercubes is known as one-dimensional space-filling property. A random Latin hypercube may not be space-filling in two or higher dimensions. To achieve multi-dimensional space-filling property, various optimality criteria have been proposed. These include maximin distance criterion (Morris and Mitchell (1995)), multi-dimensional projection (Tang (1993); Moon, Dean, and Santner (2011)), orthogonality and near orthogonality (see, for example, Sun, Liu, and Lin (2009); Yang and Liu (2012); Georgiou and Efthimiou (2014)), and the discrepancy criterion (Fang et al. (2000)). Attempts have also been made to seek designs based on multiple optimality criteria. For example, Joseph and Hung (2008) searched for Latin hypercubes based on two criteria, maximin distance criterion and orthogonality; Leary, Bhaskar, and Keane (2003) considered both the maximin distance criterion and multi-dimensional projection by searching for maximin designs within the class of orthogonal array-based Latin hypercubes. A detailed account of Latin hypercubes can be found in Lin and Tang (2015) and the references therein. Despite the rich literature, an important yet challenging issue is to find optimal Latin hypercubes of large run sizes and high dimensions in a timely fashion.

For computer experiments with qualitative and quantitative factors, Qian (2012) introduced sliced Latin hypercube design, a special Latin hypercube design that can be partitioned into slices of smaller Latin hypercube designs. For run size economy, Deng, Hung, and Lin (2014) introduced marginally coupled designs in which a design for quantitative factors is a sliced Latin hypercube design with respect to each column of a design for qualitative factors.

The objective of this article is to introduce a general method of constructing Latin hypercubes. The method is shown to be flexible and powerful as it leads to several types of Latin hypercubes: orthogonal and nearly orthogonal Latin hypercubes, sliced Latin hypercubes, and those in marginally coupled designs. In addition, we empirically show that the proposed method provides a computationally efficient way to generate large-scale Latin hypercubes with near optimality of the maximin distance criterion.

The remainder of the paper is organized as follows. Section 2 presents notation, definitions and background. Section 3 introduces the new method for constructing Latin hypercubes. Applications of the proposed method to construct different classes of Latin hypercubes are considered in Section 4. We end with some concluding remarks in Section 5, and relegate proofs to the Appendix.

2. Notation, Definitions and Background

Two columns are said to be orthogonal if all possible level combinations appear equally often. An s-level orthogonal array of strength t for m factors, denoted by $\mathrm{OA}(n,m,s,t)$, is an $n \times m$ matrix with each column taking s levels $1,\ldots,s$ and, for every $n \times t$ submatrix of the array, each of all possible level combinations appears equally often. An $\mathrm{OA}(n,m,s,t)$, say $\mathbf{A}$, is said to be *completely resolvable* (CR) if it can be expressed as $\mathbf{A} = (\mathbf{A}_1^{\mathrm{T}},\ldots,\mathbf{A}_{n/s}^{\mathrm{T}})^{\mathrm{T}}$ such that each of $\mathbf{A}_1,\ldots,\mathbf{A}_{n/s}$ is an $\mathrm{OA}(s,m,s,1)$. We denote such an array by $\mathrm{CROA}(n,m,s,t)$.

A Latin hypercube $\mathbf{L}$ with n runs and p factors is an $n \times p$ matrix in which each column is a random permutation of n equally-spaced levels. Without loss of generality, we use the n levels $-(n-1)/2, -(n-3)/2, \ldots, (n-1)/2$. As such, the sum of the elements in each column of $\mathbf{L}$ is zero and the sum of squares of these elements is $n(n^2-1)/12$. Given an $\mathbf{L} = (l_{ij})$, a Latin hypercube design $\mathbf{X} = (x_{ij})$ on the design space $[0,1)^p$ is obtained via the linear transformation

$$x_{ij} = \frac{l_{ij} + (n-1)/2 + u_{ij}}{n}, \tag{2.1}$$

where u_{ij}'s are random numbers from $[0,1)$.

For any two design points $\mathbf{u} = (u_1, \ldots, u_p)$ and $\mathbf{v} = (v_1, \ldots, v_p)$ in the design space, the inter-point distance between $\mathbf{u}$ and $\mathbf{v}$ is

$$d(\mathbf{u}, \mathbf{v}) = \Big(\sum_{j=1}^{p} |u_j - v_j|^q \Big)^{1/q}. \tag{2.2}$$

When $q = 1$ and $q = 2$, the measure in (2.2) becomes the rectangular and Euclidean distances, respectively. The maximin distance criterion seeks a design of n points in the design space that maximizes the smallest inter-point distance. Morris and Mitchell (1995) introduced a computationally efficient scalar-value criterion of maximin distance criterion. It is based on a distance list $(d_1, \ldots, d_k)$ and an index list $(J_1, \ldots, J_k)$. The distance list contains the distinct values of inter-point distances, sorted from the smallest to the largest. For $i = 1, \ldots, k$, J_i in the index list is the number of pairs of design points in the design separated by the distance d_i. Note that $1 \leq k \leq n(n-1)/2$. Morris and Mitchell (1995) defined

$$\phi_\lambda = \Big(\sum_{i=1}^{k} J_i d_i^{-\lambda} \Big)^{1/\lambda}, \tag{2.3}$$

where λ is a positive integer. For large values of λ, the design that minimizes ϕ_λ in (2.3) is a maximin design. Thus in this paper we set $\lambda = 15$, which is also commonly used in the literature (Joseph and Hung (2008)).

For a matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$, where $\mathbf{x}_i$ is the ith column of $\mathbf{X}$, define the correlation between the ith column and the jth column to be $\rho_{ij}(\mathbf{X}) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)/(\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_i \mathbf{x}_j^{\mathrm{T}} \mathbf{x}_j)^{1/2}$. A matrix is called column-orthogonal if the correlation between any two distinct columns is zero. A Latin hypercube is said to be orthogonal if it is column-orthogonal. If all the pairwise correlations are relatively small, then the Latin hypercube is nearly orthogonal.

Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be the designs for qualitative factors and quantitative factors, respectively. A design $(\mathbf{D}_1, \mathbf{D}_2)$ is called a *marginally coupled design* if (i) $\mathbf{D}_2$ is a Latin hypercube design, and (ii) the rows in $\mathbf{D}_2$ corresponding to any level of any factor in $\mathbf{D}_1$ form a small Latin hypercube design. In the case of single qualitative factor, $\mathbf{D}_1$ is a vector of s levels, and $\mathbf{D}_2$ is a sliced Latin hypercube design of s slices such that $(\mathbf{D}_1, \mathbf{D}_2)$ is a marginally coupled design.

3. Design Construction

Let r, s, and f be positive integers with $s \geq 2$. Suppose that $\mathbf{A}$ is a $(rs^2) \times (2f)$ array for which each column has s levels $1, \ldots, s$, $\mathbf{B}$ is a $(rs) \times p$ Latin hypercube, and $\mathbf{C}_q$ is an $s \times p$ Latin hypercube for $q = 1, \ldots, r$. The array $\mathbf{A}$ can be partitioned as $\mathbf{A} = (\mathbf{A}_1^{\mathrm{T}}, \ldots, \mathbf{A}_r^{\mathrm{T}})^{\mathrm{T}}$ such that each of $\mathbf{A}_1, \ldots, \mathbf{A}_r$ is an $s^2 \times (2f)$ matrix. The Latin hypercube $\mathbf{B}$ can be partitioned as $\mathbf{B} = (\mathbf{B}_1^{\mathrm{T}}, \ldots, \mathbf{B}_r^{\mathrm{T}})^{\mathrm{T}}$ such that each of $\mathbf{B}_1, \ldots, \mathbf{B}_r$ is an $s \times p$ matrix. For $q = 1, \ldots, r$, let $b_{q,ij}$ and $c_{q,ij}$ be

the (i,j)th element of $\mathbf{B}_q$ and $\mathbf{C}_q$, respectively. The following construction steps are proposed.
I. For $j = 1,\ldots,p$, obtain a $(rs^2) \times (2f)$ matrix $\mathbf{U}_j$ from $\mathbf{A}$ by replacing level i in $\mathbf{A}_q$ by $b_{q,ij}$ for $i = 1,\ldots,s$ and $q = 1,\ldots,r$.
II. For $j = 1,\ldots,p$, obtain a $(rs^2) \times (2f)$ matrix $\mathbf{V}_j$ from $\mathbf{A}$ by replacing level i in $\mathbf{A}_q$ by $c_{q,ij}$ for $i = 1,\ldots,s$ and $q = 1,\ldots,r$.
III. For $j = 1,\ldots,p$, obtain a $(rs^2) \times (2f)$ matrix $\mathbf{L}_j$ by letting its $(2k-1)$th column be

$$\mathbf{l}_{j,2k-1} = \mathbf{v}_{j,2k-1} + s\mathbf{u}_{j,2k},$$

and its $(2k)$th column be

$$\mathbf{l}_{j,2k} = -s\mathbf{u}_{j,2k-1} + \mathbf{v}_{j,2k},$$

for $k = 1,\ldots,f$, where $\mathbf{u}_{j,h}$ and $\mathbf{v}_{j,h}$ are the hth column of $\mathbf{U}_j$ and $\mathbf{V}_j$, respectively, for $h = 1,\ldots,2f$. Obtain the matrix $\mathbf{L} = [\mathbf{L}_1,\ldots,\mathbf{L}_p]$ of order $(rs^2) \times (2fp)$.

For $r = 1$, if $\mathbf{B} = \mathbf{C}_r$ and $\mathbf{A}$ is an $\mathrm{OA}(s^2, 2f, s, 2)$, the proposed method is equivalent to that of Lin, Mukerjee, and Tang (2009) that was proposed to construct orthogonal and nearly orthogonal Latin hypercubes. By allowing different values of r and using Latin hypercubes $\mathbf{B}$ and $\mathbf{C}_q$'s of varying sizes, we are going to show that the proposed method is more flexible because it produces various classes of Latin hypercubes. The three-step procedure uses the same $\mathbf{A}_q$ for different values of j's in obtaining $\mathbf{U}_j$'s in Step I and $\mathbf{V}_j$'s in Step II. Alternatively, for different values of $j = 1,\ldots,p$, we can use different designs that are isomorphic to $\mathbf{A}_q$. In this case the proposed procedure is shown to lead to a vast class of space-filling Latin hypercubes with desirable projection properties. For convenience, we call the approach using isomorphic $\mathbf{A}_q$'s for different j's the *generalization method* and the one using identical $\mathbf{A}_q$ for different j's the *basic method*. Two designs are said to be isomorphic if one can be obtained from the other by reordering the runs, relabeling the factors having the same number of levels, and/or relabeling the levels of one or more factors.

Proposition 1. *The matrix $\boldsymbol{L}$ obtained via the basic method or the generalization method is a Latin hypercube if for $q = 1,\ldots,r$, $\boldsymbol{A}_q$ satisfies that every two consecutive columns of $\boldsymbol{A}$, $\boldsymbol{a}_{q,2k-1}$ and $\boldsymbol{a}_{q,2k}$ for $k = 1,\ldots,f$, are orthogonal.*

Proposition 1 indicates that the proposed methods provide new ways for constructing Latin hypercubes. We obtain Latin hypercubes by smaller ones $\mathbf{B}$ and $\mathbf{C}_q$'s and an array $\mathbf{A}$ that meets the condition in Proposition 1. To understand Proposition 1, one can think of $\mathbf{A}$ and $\mathbf{B}$ together providing a global layout of $\mathbf{L}$, while each $\mathbf{C}_q$ controls the local configurations of $\mathbf{L}$. Such a global layout can be viewed as the superposition of r components each of which is determined by each pair of $\mathbf{A}_q$ and $\mathbf{B}_q$. Therefore, a sliced Latin hypercube $\mathbf{B}$ is used to enhance space-filling property of $\mathbf{L}$. In addition, $\mathbf{A}$ does not have to be an orthogonal array, and thus it is less restrictive. One way to obtain such an $\mathbf{A}$ is to stack r replicates of an s-level orthogonal array of s^2 runs row by row. Another way is to choose columns satisfying the condition from an s-level saturated orthogonal array of rs^2 runs.

Example 1. Consider the case $s = 2, r = 2, f = 1, p = 2$. Let

$$\mathbf{A}^{\mathrm{T}} = \begin{pmatrix} 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 1 & 2 & 2 & 1 & 2 & 1 & 1 & 2 \end{pmatrix},$$

$$\mathbf{B}_1 = \frac{1}{2}\begin{pmatrix} 1 & 3 \\ -1 & -3 \end{pmatrix},\ \mathbf{B}_2 = \frac{1}{2}\begin{pmatrix} 3 & -1 \\ -3 & 1 \end{pmatrix},\ \text{and } \mathbf{C}_1 = \mathbf{C}_2 = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Then Steps I and II provide

$$\mathbf{U}_1 = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 3 & -3 \\ -3 & 3 \\ 3 & 3 \\ -3 & -3 \end{pmatrix}, \mathbf{U}_2 = \frac{1}{2}\begin{pmatrix} 3 & 3 \\ -3 & -3 \\ 3 & -3 \\ -3 & 3 \\ -1 & 1 \\ 1 & -1 \\ -1 & -1 \\ 1 & 1 \end{pmatrix}, \mathbf{V}_1 = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ 1 & 1 \\ -1 & -1 \end{pmatrix}, \mathbf{V}_2 = \frac{1}{2}\begin{pmatrix} -1 & -1 \\ 1 & 1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & -1 \\ 1 & 1 \end{pmatrix}.$$

Step III yields

$$\mathbf{L}_1 = \frac{1}{2}\begin{pmatrix} 3 & -1 \\ -3 & 1 \\ -1 & -3 \\ 1 & 3 \\ -5 & -7 \\ 5 & 7 \\ 7 & -5 \\ -7 & 5 \end{pmatrix}, \mathbf{L}_2 = \frac{1}{2}\begin{pmatrix} 5 & -7 \\ -5 & 7 \\ -7 & -5 \\ 7 & 5 \\ 1 & 3 \\ -1 & -3 \\ -3 & 1 \\ 3 & -1 \end{pmatrix}.$$

Because the $\mathbf{A}$ chosen satisfies the condition in Proposition 1, we obtain a Latin hypercube $\mathbf{L} = [\mathbf{L}_1, \mathbf{L}_2]$ of order 8×4.

Example 2. Consider the case $s = 4$, $r = 2$, $f = 4$. Take the $\mathrm{OA}(32, 9, 4, 2)$ from the website Sloane (2014), remove the first column of this orthogonal array and use the remaining eight columns as $\mathbf{A}$. It satisfies the condition in Proposition 1. Let

$$\mathbf{B}_1 = \frac{1}{2}\begin{pmatrix} 5 & 7 \\ -1 & -5 \\ -7 & 1 \\ 3 & -3 \end{pmatrix},\ \mathbf{B}_2 = \frac{1}{2}\begin{pmatrix} 7 & 5 \\ -5 & -1 \\ 1 & -7 \\ -3 & 3 \end{pmatrix},\ \mathbf{C}_1 = \frac{1}{2}\begin{pmatrix} 1 & -3 \\ 3 & 1 \\ -1 & 3 \\ -3 & -1 \end{pmatrix},\ \text{and } \mathbf{C}_2 = \frac{1}{2}\begin{pmatrix} -3 & -3 \\ -1 & 1 \\ 1 & -1 \\ 3 & 3 \end{pmatrix}.$$

The construction provides a group of 32×16 Latin hypercubes. To illustrate the improved projection property offered by the generalization method over the basic

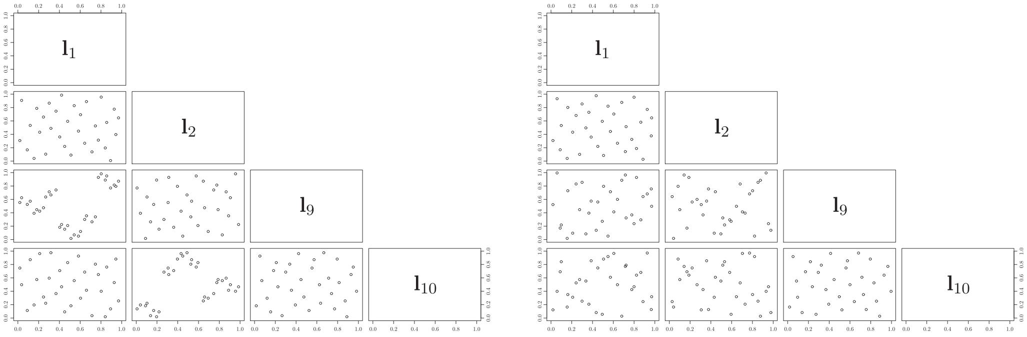

Figure 1. (a) the pairwise scatter plot of the $(1, 2, 9, 10)$th columns in a Latin hypercube design based on $\mathbf{L}$ in Example 2, obtained using (a) the basic method, (b) the generalization method.

method, Figure 1 displays the pairwise scatter plots of the $(1, 2, 9, 10)$th columns of one design generated from the basic method and one from the generalization method. Note that there are no clustered points in Figure 1b. Pairwise plots of other columns exhibit similar patterns. In addition, the rectangular distance measurement ϕ_λ in (2.3) with $\lambda = 15$ of Latin hypercube designs produced by the basic and generalization method are 0.421 and 0.347, respectively, indicating that the generalization method produces Latin hypercube designs that are more space-filling in terms of maximin distance criterion.

4. Applications

This section applies the proposed basic and generalization methods to construct various types of Latin hypercubes. By choosing different $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}_q$'s, the proposed methods produce a vast class of Latin hypercubes. We show empirically that this class contains designs whose maximin criterion values can be comparable to those produced by the *R* packages *lhs* (Carnell (2012)) and *SLHD* (Ba (2013); Ba, Brenneman, and Myers (2014)). The first package has been used frequently in practice and in the literature and the latter is relatively new and has been shown to perform better than some existing ones. We also provide the conditions of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}_q$'s for the design constructed to be an orthogonal Latin hypercube, and illustrate how new orthogonal Latin hypercubes can be constructed. In addition, the proposed methods are shown to construct space-filling Latin hypercubes in marginally coupled designs.

4.1. Construction of maximin Latin hypercubes

This section investigates the performance of the proposed methods by comparing them with some existing algorithmic approaches on maximin Latin hypercube designs. For the proposed methods, we consider four approaches. Approach

I is the basic method that uses identical $\mathbf{A}_q$ for different values of j's in Steps I and II. Denote this orthogonal array by $\mathbf{A}_0$. Approaches II, III, and IV correspond to the generalization method when the $\mathbf{A}_q$'s are obtained from the $\mathbf{A}_0$ by row permutations, column permutations, and both row and column permutations, respectively.

To compare Approaches I $-$ IV with the packages *lhs* and *SLHD*, we considered the rectangular distance criterion ϕ_λ in (2.3) with $\lambda = 15$. Table 1 lists such rectangular distance measurements of maximin Latin hypercubes of various run sizes and factors. Approaches I - IV used $\mathbf{A}_0 = \text{OA}(s^2, 2f, s, 2)$ from the website Sloane (2014). A random sliced Latin hypercube was used for $\mathbf{B} = (\mathbf{B}_1^{\mathrm{T}}, \ldots, \mathbf{B}_r^{\mathrm{T}})^{\mathrm{T}}$ and a maximin Latin hypercube generated by the package *lhs* was used for $\mathbf{C}_q$ for $q = 1, \ldots, r$. Because $\mathbf{B}$ was randomly generated and Approaches II, III and IV used the random row, column, row with column permutations, different designs were generated for the same design size. We repeated each approach 100 times, and choose the minimum ϕ_λ value of the 100 designs shown Table 1. For fair comparison, for the package *lhs*, we also generated 100 maximin Latin hypercube designs of n runs on $(0,1)^m$ using the R command $maximinLHS(n,m)$, and show the minimum ϕ_λ value. For the package *SLHD*, we used R command $maximinSLHD(t = 1, m = n, k = m, itermax = 100)$ to generate a maximin Latin hypercube design of n runs and m factors. Here $itermax$ represents the maximum allowable iterations in the optimization. Since $itermax$=100 is large enough, repeating this command leads to almost identical ϕ_λ value for each design size shown in Table 1. There is no need to repeat this command. Table 1 also lists the computing time of the packages *lhs* and *SLHD* on a personal computer with Intel i7 CPU 2.8 GHz. Since the proposed methods generate designs algebraically, its computing time is a few seconds and thus is negligible. For example, the computing time for the last case in Table 1 was 2.9 seconds.

Table 1 reveals that among the six methods, Approaches II, IV and the package *SLHD* produce Latin hypercube designs with similar measurements of the maximin distance criterion. From the computational efficiency point of view, the package *SLHD* requires much more computing time, especially for large n and m, whereas Approaches II and IV are much more efficient.

4.2. Construction of orthogonal and nearly orthogonal Latin hypercubes

This section shows that the proposed methods can be used to construct new orthogonal and nearly orthogonal Latin hypercubes. When $\mathbf{A}_q$'s are taken to be orthogonal arrays in the basic method, the correlation between any pair of columns in the resulting Latin hypercube can be quantified by the correlation between columns of $\mathbf{B}_q$, the correlation between columns of $\mathbf{C}_q$, and the correlation of columns between $\mathbf{B}_q$ and $\mathbf{C}_q$.

Table 1. The rectangular distance measurement ϕ_λ in (2.3) with $\lambda = 15$ and computing time of maximin Latin hypercube designs of of n runs and m factors produced by Approaches I, II, III, IV, the packages *lhs* and *SLHD*.

n	m	r	s	f	p	ϕ_λ with $\lambda = 15$						Computing Time ('s)	
						I	*II*	*III*	*IV*	*lhs*	*SLHD*	t_{lhs}	t_{SLHD}
98	24	2	7	4	3	0.2601	0.2556	0.2644	0.2557	0.2656	0.2337	1.55	32.58
98	48	2	7	4	6	0.1164	0.1169	0.1174	0.1171	0.1202	0.1122	3.08	49.90
245	24	5	7	4	3	0.4067	0.3075	0.3643	0.3034	0.3337	0.2751	15.68	193.60
245	48	5	7	4	6	0.1443	0.1352	0.1409	0.1349	0.1433	0.1291	31.49	519.23
490	24	10	7	4	3	0.5476	0.3456	0.4468	0.3434	0.3922	0.3068	105.04	2362.77
490	48	10	7	4	6	0.1740	0.1497	0.1645	0.1492	0.1649	0.1433	212.85	3536.49
338	42	2	13	7	3	0.2277	0.1644	0.2239	0.1641	0.1796	0.1564	66.88	960.39
338	84	2	13	7	6	0.0797	0.0774	0.0804	0.0774	0.0810	0.0756	129.64	2381.34
845	42	5	13	7	3	0.3541	0.1896	0.3121	0.1886	0.2246	0.1792	887.86	12161.57
845	84	5	13	7	6	0.1080	0.0879	0.1019	0.0879	0.0952	0.0863	1712.38	23875.75
1690	42	10	13	7	3	0.4989	0.2099	0.4163	0.2083	0.2720	0.1982	6631.00	61581.84
1690	84	10	13	7	6	0.1325	0.0967	0.1174	0.0966	0.1088	0.0951	12755.74	105372.90

Proposition 2. *Suppose that each of $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_r$ is an $OA(s^2, 2f, s, 2)$. For $m \neq m'$, the correlation $\rho_{mm'}(\boldsymbol{L})$ between the m and m'th columns of $\boldsymbol{L}$ obtained via the basic method is*

$$\begin{cases} \dfrac{12\sum_{q=1}^{r} t_j^{\boldsymbol{B}_q} t_{j'}^{\boldsymbol{B}_q}}{r(r^2s^4-1)}, & \begin{array}{l} m=(j-1)2f+2k-1, m'=(j'-1)2f+2k'-1, k\neq k' \textit{ or} \\ m=(j-1)2f+2k, m'=(j'-1)2f+2k', k\neq k'; \end{array} \\ \dfrac{12\sum_{q=1}^{r} (s^2 t_{jj'}^{\boldsymbol{B}_q} + t_{jj'}^{\boldsymbol{C}_q})}{rs(r^2s^4-1)}, & \begin{array}{l} m=(j-1)2f+2k-1, m'=(j'-1)2f+2k'-1, j\neq j', k=k' \\ \textit{or } m=(j-1)2f+2k, m'=(j'-1)2f+2k', j\neq j', k=k'; \end{array} \\ -\dfrac{12\sum_{q=1}^{r} t_j^{\boldsymbol{B}_q} t_{j'}^{\boldsymbol{B}_q}}{r(r^2s^4-1)}, & \begin{array}{l} m=(j-1)2f+2k-1, m'=(j'-1)2f+2k', k\neq k' \textit{ or} \\ m=(j-1)2f+2k, m'=(j'-1)2f+2k'-1, k\neq k'; \end{array} \\ -\dfrac{12\sum_{q=1}^{r} (t_j^{\boldsymbol{B}_q} t_{j'}^{\boldsymbol{B}_q} - t_{jj'}^{\boldsymbol{C}_q,\boldsymbol{B}_q} + t_{jj'}^{\boldsymbol{B}_q,\boldsymbol{C}_q})}{r(r^2s^4-1)}, & m=(j-1)2f+2k-1, m'=(j'-1)2f+2k', k=k'; \\ -\dfrac{12\sum_{q=1}^{r} (t_j^{\boldsymbol{B}_q} t_{j'}^{\boldsymbol{B}_q} - t_{jj'}^{\boldsymbol{B}_q,\boldsymbol{C}_q} + t_{jj'}^{\boldsymbol{C}_q,\boldsymbol{B}_q})}{r(r^2s^4-1)}, & m=(j-1)2f+2k, m'=(j'-1)2f+2k'-1, k=k', \end{cases}$$

where $j \in \{1, \ldots, p\}$, $j' \in \{1, \ldots, p\}$, $k \in \{1, \ldots, f\}$, $k' \in \{1, \ldots, f\}$, $t_j^{\boldsymbol{B}_q} = \sum_{i=1}^{s} b_{q,ij}$, $t_j^{\boldsymbol{C}_q} = \sum_{i=1}^{s} c_{q,ij}$, $t_{jj'}^{\boldsymbol{B}_q} = \sum_{i=1}^{s} b_{q,ij} b_{q,ij'}$, $t_{jj'}^{\boldsymbol{C}_q} = \sum_{i=1}^{s} c_{q,ij} c_{q,ij'}$, $t_{jj'}^{\boldsymbol{B}_q,\boldsymbol{C}_q} = \sum_{i=1}^{s} b_{q,ij} c_{q,ij'}$, $t_{jj'}^{\boldsymbol{C}_q,\boldsymbol{B}_q} = \sum_{i=1}^{s} c_{q,ij} b_{q,ij'}$, and $b_{q,ij}$ and $c_{q,ij}$ are the (i,j)th element of $\boldsymbol{B}_q$ and $\boldsymbol{C}_q$, respectively.

The following Corollaries 1 and 2, respectively, provide the conditions for **L** to be an orthogonal Latin hypercube obtained via the basic method and the generalization method that applies column permutations to $\mathbf{A}_q$ in Steps I and II for different j's.

Corollary 1. *The matrix $\boldsymbol{L}$ in the basic method is an orthogonal Latin hypercube if for $q = 1, \ldots, r$, the following all hold: (i) $\boldsymbol{A}_q$ is an $\mathrm{OA}(s^2, 2f, s, 2)$; (ii) $\boldsymbol{B}_q$ is column-orthogonal and each column of $\boldsymbol{B}_q$ has mean zero; (iii) $\boldsymbol{C}_q$ is an orthogonal Latin hypercube; and (iv) $\boldsymbol{B}^T\boldsymbol{C}$ is a symmetric matrix where $\boldsymbol{C} = (\boldsymbol{C}_1^T, \ldots, \boldsymbol{C}_r^T)^T$.*

Example 3. Given a $r \geq 1$, $p = 1$ and, for $q = 1, \ldots, r$, let $\mathbf{A}_q$ be an $\mathrm{OA}(25, 6, 5, 2)$, $\mathbf{B} = (\mathbf{B}_1^T, \ldots, \mathbf{B}_r^T)^T$ be a permutation of a vector $\{-(5r-1)/2, -(5r-3)/2, \ldots, (5r-1)/2\}$ such that $\mathbf{B}_q$ sums to zero, and $\mathbf{C}_q$ be a random permutation of $\{-2, -1, 0, 1, 2\}$. By Corollary 1, we obtain a class of $(25r) \times 6$ orthogonal Latin hypercubes which are new to the literature for $r \neq 5^{2^c-2}$ with a positive integer c.

Corollary 2. *For a generalization method that applies column permutations to $\boldsymbol{A}_q$ in Steps I and II for different j's, the resulting matrix $\boldsymbol{L}$ is an orthogonal Latin hypercube if the following all hold:* (i) *the $\boldsymbol{A}_q$'s, $\boldsymbol{B}_q$'s, and $\boldsymbol{C}_q$'s satisfy the conditions of Corollary 1 for $q = 1, \ldots, r$; and* (ii) *$\boldsymbol{u}_{j,2k-1}^T\boldsymbol{v}_{j',2k'} + \boldsymbol{v}_{j,2k-1}^T\boldsymbol{u}_{j',2k'} = 0$, $\boldsymbol{u}_{j,2k}^T\boldsymbol{v}_{j',2k'-1} + \boldsymbol{v}_{j,2k}^T\boldsymbol{u}_{j',2k'-1} = 0$, $\boldsymbol{u}_{j,2k}^T\boldsymbol{v}_{j',2k'} - \boldsymbol{v}_{j,2k-1}^T\boldsymbol{u}_{j',2k'-1} = 0$, $\boldsymbol{u}_{j,2k-1}^T\boldsymbol{v}_{j',2k'-1} - \boldsymbol{v}_{j,2k}^T\boldsymbol{u}_{j',2k'} = 0$ for $j = 1, \ldots, p, j' = 1, \ldots, p, j \neq j', k = 1, \ldots, f, k' = 1, \ldots, f$.*

Example 4. Use an $\mathrm{OA}(16, 4, 4, 2)$ as $\mathbf{A}_1$ and $\mathbf{A}_2$ and let

$$\mathbf{B}_1 = \frac{1}{2}\begin{pmatrix} 1 & 5 \\ 5 & -1 \\ -1 & -5 \\ -5 & 1 \end{pmatrix}, \ \mathbf{B}_2 = \frac{1}{2}\begin{pmatrix} 3 & 7 \\ 7 & -3 \\ -3 & -7 \\ -7 & 3 \end{pmatrix}, \text{ and } \mathbf{C}_1 = \mathbf{C}_2 = \frac{1}{2}\begin{pmatrix} 1 & -3 \\ 3 & 1 \\ -1 & 3 \\ -3 & -1 \end{pmatrix}.$$

The generalization method in Corollary 2 provides many 32×8 orthogonal Latin hypercubes.

A few remarks on Corollaries 1 and 2 are in order. Comparing the proposed methods with the approach in Lin, Mukerjee, and Tang (2009), all orthogonal Latin hypercubes by the latter can be produced by the former with $r = 1$. Corollaries 1 and 2 highlight an important contribution of the proposed methods as they provide many new orthogonal Latin hypercubes of rs^2 runs where s is odd and $r \neq s^{2^c-2}$ for any positive integer c. (Orthogonal Latin hypercubes with odd s and $r = s^{2^c-2}$ were given by Pang, Liu, and Lin (2009).) Such run sizes of orthogonal Latin hypercubes cannot be found by existing methods. For example, the run sizes of orthogonal Latin hypercubes in Steinberg and Lin (2006), Sun, Liu, and Lin (2010), and Lin et al. (2010) have $n = 2^{2^c}$ for any integer $c \geq 2$, $n = r2^{c+1}$ or $r2^{c+1} + 1$ for any positive integers c and r, and $n = 8k$ or $8k + 1$ for any positive integer k, respectively. In addition, Corollary 2 points out that the generalization method provides orthogonal Latin hypercubes that are space-filling in low dimensions. Not all column permutations to $\mathbf{A}_q$'s in the generalization method lead to orthogonal Latin hypercubes. Condition (ii) of Corollary 2 is used to find such permutations.

4.3. Construction of Latin hypercubes in marginally coupled designs

This subsection discusses how the proposed methods can be used to construct marginally coupled designs $(\mathbf{D}_1, \mathbf{D}_2)$, where $\mathbf{D}_1$ and $\mathbf{D}_2$ are the designs for qualitative factors and quantitative factors, respectively. The result of the basic method is summarized in Proposition 3. Such a result holds for the generalization method that applies column permutations to $\mathbf{A}_q$ for different j's in Steps I and II. When used for constructing marginally coupled designs, the merit of the proposed methods is that they provide more space-filling Latin hypercubes $\mathbf{D}_2$'s of rs^2 runs with $r \geq 2$ than those by the existing method, namely, Construction 2 of Deng, Hung, and Lin (2014). Their method obtains columns of $\mathbf{D}_2$ based on the same (rs)-level column in a mixed orthogonal array and thus the points of the resulting $\mathbf{D}_2$ are not well spread-out in the design space.

Proposition 3. *Let $\boldsymbol{E} = (\boldsymbol{E}_1^{\mathrm{T}}, \ldots, \boldsymbol{E}_r^{\mathrm{T}})^{\mathrm{T}}$ and $\boldsymbol{F}$ be the designs for qualitative factors and quantitative factors, respectively, where each of $\boldsymbol{E}_1, \ldots, \boldsymbol{E}_r$ has s^2 runs and $\boldsymbol{F}$ is obtained from $\boldsymbol{L}$ in the basic method via the linear transformation in* (2.1)*. Design $(\boldsymbol{E}, \boldsymbol{F})$ is a marginally coupled design if the following properties of $\boldsymbol{A} = (\boldsymbol{A}_1^{\mathrm{T}}, \ldots, \boldsymbol{A}_r^{\mathrm{T}})^{\mathrm{T}}$ all hold: (i) $\boldsymbol{A}$ satisfies the condition in Proposition 1; and (ii) for $q = 1, \ldots, r$ and for each level of each column of $\boldsymbol{E}_q$, the corresponding rows of $\boldsymbol{A}_q$ is an $\mathrm{OA}(s, 2f, s, 1)$.*

Example 5. Let $\mathbf{H}$ be an $\mathrm{OA}(16, 5, 4, 2)$ and

$$\mathbf{H}^{\mathrm{T}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 1 & 2 & 3 & 4 & 2 & 1 & 4 & 3 & 3 & 4 & 1 & 2 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 3 & 4 & 1 & 2 & 4 & 3 & 2 & 1 & 2 & 1 & 4 & 3 \\ 1 & 2 & 3 & 4 & 4 & 3 & 2 & 1 & 2 & 1 & 4 & 3 & 3 & 4 & 1 & 2 \end{pmatrix}.$$

For given p and r and $q = 1, \ldots, r$, let $\mathbf{E}_q$ and $\mathbf{A}_q$ be the first three and last two columns of $\mathbf{H}$, respectively, and let $\mathbf{C}_q$ be a $4 \times p$ Latin hypercube. Such $\mathbf{E}_q$'s and $\mathbf{A}_q$'s meet condition (ii) of Proposition 3. Let $\mathbf{B}$ be a $(4r) \times p$ Latin hypercube. By Proposition 3, we obtain a marginally coupled design $(\mathbf{E}, \mathbf{F})$ with $\mathbf{E} = (\mathbf{E}_1^{\mathrm{T}}, \ldots, \mathbf{E}_r^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{F}$ being a $(16r) \times (2p)$ Latin hypercube design from $\mathbf{L}$ via (2.1), where $\mathbf{L}$ is constructed using such $\mathbf{A}_q$'s via the basic or generalization method. Here design $\mathbf{E}$ for qualitative factors is a replicated orthogonal array. Figure 2 displays the pairwise scatter plots of four columns in such a 48×24 Latin hypercube $\mathbf{L}$ with $r = 3$ and $p = 12$. The pairwise plots for other columns are similar. Figure 2 graphically shows that the proposed method provides marginally coupled designs that are space-filling. To quantify the space-filling property, we compute the rectangular distance measurement ϕ_λ in (2.3) with $\lambda = 15$ of $\mathbf{L}$ and

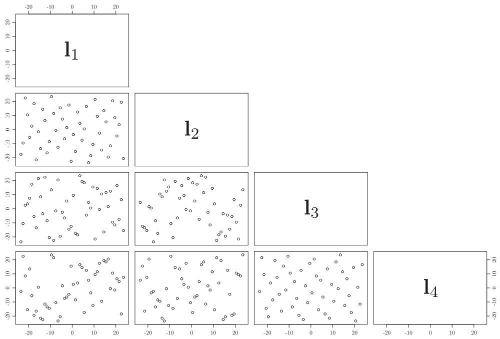

Figure 2. The pairwise scatter plot of four columns in design $\mathbf{L}$ in Example 5, where $\mathbf{l}_i$ represents the ith column of $\mathbf{L}$, for $i = 1, \ldots, 4$.

the resulting distance is 0.224. As a comparison, we used each of the R packages *lhs* and *SLHD* to generate 100 designs, and the minimum ϕ_λ values were 0.228 and 0.206, respectively. Clearly, the proposed method provides marginally coupled designs with similar space-filling property.

Example 6. Let $\mathbf{E} = (\mathbf{E}_1^{\mathrm{T}}, \mathbf{E}_2^{\mathrm{T}}, \mathbf{E}_3^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{A} = (\mathbf{A}_1^{\mathrm{T}}, \mathbf{A}_2^{\mathrm{T}}, \mathbf{A}_3^{\mathrm{T}})^{\mathrm{T}}$ be the $(2, 8, 12, 13)$th and $(3, 4, 5, 10, 6, 9, 7, 11)$th columns of the $\mathrm{OA}(27, 13, 3, 2)$ from the website Sloane (2014), respectively, where each of $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ and each of $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ have 9 runs. Such $\mathbf{E}_q$'s and $\mathbf{A}_q$'s meet condition (ii) of Proposition 3. Use a $9 \times p$ Latin hypercube $\mathbf{B}$ and $3 \times p$ Latin hypercubes $\mathbf{C}_q$'s, for $q = 1, 2, 3$. We obtain a marginally coupled design $(\mathbf{E}, \mathbf{F})$ where $\mathbf{F}$ is a $27 \times (8p)$ Latin hypercube design obtained from $\mathbf{L}$ via (2.1), and $\mathbf{L}$ is constructed using such $\mathbf{A}_q$'s by the proposed methods. Here, the design for qualitative factors is a unreplicated three-level orthogonal array.

Corollary 3 is an application of Proposition 3 when design $\mathbf{E}$ is for a single qualitative factor. It also provides a mechanism to determine the slices of a sliced Latin hypercube and leads to a rich class of sliced Latin hypercubes. Its proof is straightforward and thus omitted.

Corollary 3. *For $j = 1, \ldots, s$ and $k = 1, \ldots, r$, let $\boldsymbol{\omega}_{jk} = \{(j-1)s + (k-1)s^2 + h : h = 1, \ldots, s\}$ and $\boldsymbol{\pi}_k$ be a random permutation of $\{1, \ldots, s\}$. Let $\pi_{k,i}$ be the ith entry of $\boldsymbol{\pi}_k$. The matrix $\boldsymbol{L}$ is a sliced Latin hypercube of s slices $\boldsymbol{L}_1, \ldots, \boldsymbol{L}_s$ where $\boldsymbol{L}_i$ consists of the $\{\boldsymbol{\omega}_{\pi_{k,i}k} : k = 1, \ldots, r\}$th rows of $\boldsymbol{L}$ if the following properties of $\boldsymbol{A}$ all hold: (i) $\boldsymbol{A}$ satisfies the condition in Proposition 1; and (ii) $\boldsymbol{A}$ is a $\mathrm{CROA}(rs^2, 2f, s, t)$ $(t \geq 1)$ in the sense that $\boldsymbol{A}$ can be expressed as $\boldsymbol{A} = (\tilde{\boldsymbol{A}}_1^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{A}}_{rs}^{\mathrm{T}})^{\mathrm{T}}$ such that each of $\tilde{\boldsymbol{A}}_1, \ldots, \tilde{\boldsymbol{A}}_{rs}$ is an $\mathrm{OA}(s, 2f, s, 1)$.*

Example 7. Consider an $\mathrm{OA}(32, 8, 4, 2)$, say $\mathbf{H}$,

$$\mathbf{H}^{\mathrm{T}} = \left(\begin{array}{cccccccccccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\
4 & 3 & 2 & 1 & 3 & 4 & 1 & 2 & 2 & 1 & 4 & 3 & 1 & 2 & 3 & 4 & 4 & 3 & 2 & 1 & 3 & 4 & 1 & 2 & 2 & 1 & 4 & 3 & 1 & 2 & 3 & 4 \\
4 & 1 & 3 & 2 & 3 & 2 & 4 & 1 & 4 & 1 & 3 & 2 & 3 & 2 & 4 & 1 & 2 & 3 & 1 & 4 & 1 & 4 & 2 & 3 & 2 & 3 & 1 & 4 & 1 & 4 & 2 & 3 \\
4 & 3 & 1 & 2 & 2 & 1 & 3 & 4 & 1 & 2 & 4 & 3 & 3 & 4 & 2 & 1 & 2 & 1 & 3 & 4 & 4 & 3 & 1 & 2 & 3 & 4 & 2 & 1 & 1 & 2 & 4 & 3 \\
4 & 2 & 1 & 3 & 4 & 2 & 1 & 3 & 3 & 1 & 2 & 4 & 3 & 1 & 2 & 4 & 1 & 3 & 4 & 2 & 1 & 3 & 4 & 2 & 2 & 4 & 3 & 1 & 2 & 4 & 3 & 1 \\
2 & 4 & 3 & 1 & 4 & 2 & 1 & 3 & 4 & 2 & 1 & 3 & 2 & 4 & 3 & 1 & 3 & 1 & 2 & 4 & 1 & 3 & 4 & 2 & 1 & 3 & 4 & 2 & 3 & 1 & 2 & 4 \\
3 & 2 & 4 & 1 & 4 & 1 & 3 & 2 & 1 & 4 & 2 & 3 & 2 & 3 & 1 & 4 & 2 & 3 & 1 & 4 & 1 & 4 & 2 & 3 & 4 & 1 & 3 & 2 & 3 & 2 & 4 & 1 \\
1 & 4 & 3 & 2 & 2 & 3 & 4 & 1 & 4 & 1 & 2 & 3 & 3 & 2 & 1 & 4 & 4 & 1 & 2 & 3 & 3 & 2 & 1 & 4 & 1 & 4 & 3 & 2 & 2 & 3 & 4 & 1
\end{array}\right).$$

Take a $8 \times p$ Latin hypercube $\mathbf{B}$ and $4 \times p$ Latin hypercubes $\mathbf{C}_q$'s, for $q = 1, 2$. By Corollary 3, the proposed method gives many $32 \times (8p)$ sliced Latin hypercubes of four slices $\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{L}_3$, and $\mathbf{L}_4$. For example, if $\boldsymbol{\pi}_1 = \{1, 3, 4, 2\}$ and $\boldsymbol{\pi}_2 = \{2, 4, 1, 3\}$, slices $\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{L}_3$, and $\mathbf{L}_4$ correspond to $\{1 - 4, 21 - 24\}$, $\{9 - 12, 29 - 32\}$, $\{13 - 16, 17 - 20\}$, and $\{5 - 8, 25 - 28\}$ of $\mathbf{L}$, respectively.

5. Conclusions and Discussion

We have introduced a general and flexible method for constructing several types of Latin hypercubes. The merits of the proposed methods lie in obtaining many new maximin Latin hypercubes, orthogonal and nearly orthogonal Latin hypercubes, and those in marginally coupled designs, particularly of large run sizes and high dimensions.

There are a few issues worth noting. Section 4.1 only considers the random row and/or column permutations in the generalization method. To construct maximin Latin hypercube designs, optimization algorithms such as genetic algorithm and simulated annealing can be used to find optimal permutations. In addition, the optimal choices of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}_q$'s are worth investigating. Aside from Latin hypercube designs, the proposed method may be useful for constructing such other space-filling designs as uniform designs, and orthogonal and nearly orthogonal designs (Bingham, Sitter, and Tang (2009)).

Acknowledgement

Lin is supported by the Natural Sciences and Engineering Research Council of Canada. The authors thank the editor, an associate editor and the referees for their helpful comments.

Appendix

Proof of Proposition 1. We first show that Proposition 1 holds for the basic method. For $j = 1, \ldots, p$ and $h = 1, \ldots, 2f$, each column $\mathbf{v}_{j,h}$ has s levels $\{-(s-1)/2, -(s-3)/2, \ldots, (s-1)/2\}$, and each column $\mathbf{u}_{j,h}$ has rs levels $\{-(rs-1)/2, -(rs-3)/2, \ldots, (rs-1)/2\}$. Consider the $(2k-1)$th column $\mathbf{l}_{j,2k-1}$ and the

$(2k)$th column $\mathbf{l}_{j,2k}$ of $\mathbf{L}_j$. The former is uniquely determined by $\mathbf{v}_{j,2k-1}$ and $\mathbf{u}_{j,2k}$ while the latter is uniquely determined by $\mathbf{v}_{j,2k}$ and $\mathbf{u}_{j,2k-1}$. Since $\mathbf{A}_q$ satisfies that the pair of columns, $\mathbf{a}_{q,2k-1}$ and $\mathbf{a}_{q,2k}$, is orthogonal for all $q = 1, \ldots, r$ and $k = 1, \ldots, f$, both the pair of columns, $\mathbf{v}_{j,2k-1}$ and $\mathbf{u}_{j,2k}$, and the pair of columns, $\mathbf{v}_{j,2k}$ and $\mathbf{u}_{j,2k-1}$, have the property that the level combinations appear exactly once. Thus both the column $\mathbf{l}_{j,2k-1}$ and the column $\mathbf{l}_{j,2k}$ are a permutation of $\{-(rs^2-1)/2, -(rs^2-3)/2, \ldots, (rs^2-1)/2\}$ by definition. Applying similar arguments, Proposition 1 holds for the generalization method.

Proof of Proposition 2. First, we consider the case $m = (j-1)2f + 2k - 1$ and $m' = (j'-1)2f + 2k' - 1$. The correlation between the column $\mathbf{l}_{j,2k-1}$ and the column $\mathbf{l}_{j',2k'-1}$ is $\rho_{mm'}(\mathbf{L}) = \mathbf{l}_{j,2k-1}^{\mathrm{T}}\mathbf{l}_{j',2k'-1}/\{12^{-1}rs^2(r^2s^4-1)\}$, where

$$\mathbf{l}_{j,2k-1}^{\mathrm{T}}\mathbf{l}_{j',2k'-1} = \mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{v}_{j',2k'-1} + s\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{v}_{j',2k'-1} + s\mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{u}_{j',2k'} + s^2\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{u}_{j',2k'}. \tag{A.1}$$

Here $\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{v}_{j',2k'-1} = \mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{u}_{j',2k'} = 0$ because of the pairwise projection property of orthogonal arrays $\mathbf{A}_q$'s and the fact that each column of $\mathbf{C}_q$ sums to zero, for $q = 1, \ldots, r$. Furthermore, in (A.1), for $k \neq k'$, we have $\mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{v}_{j',2k'-1} = 0$ and

$$\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{u}_{j',2k'} = \sum_{q=1}^{r}(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'})$$

followed by the pairwise projection property of orthogonal arrays $\mathbf{A}_q$'s, and for $k = k'$, we have (i) $j \neq j'$; (ii) because the two columns, $\mathbf{v}_{j,2k-1}$ and $\mathbf{v}_{j',2k'-1}$, use the same column from $\mathbf{A} = (\mathbf{A}_1^{\mathrm{T}}, \ldots, \mathbf{A}_r^{\mathrm{T}})^{\mathrm{T}}$, we have $\mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{v}_{j',2k'-1} = s\sum_{q=1}^{r}\sum_{i=1}^{s} c_{q,ij}c_{q,ij'}$; and (iii) because the two columns, $\mathbf{u}_{j,2k}$ and $\mathbf{u}_{j',2k'}$, use the same column from $\mathbf{A}$, we have

$$\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{u}_{j',2k'} = \sum_{q=1}^{r} s \sum_{i=1}^{s} b_{q,ij}b_{q,ij'}.$$

As a result, for the case $m = (j-1)2f + 2k - 1$ and $m' = (j'-1)2f + 2k' - 1$, we have

$$\mathbf{l}_{j,2k-1}^{\mathrm{T}}\mathbf{l}_{j',2k'-1} = \begin{cases} s^2\sum_{q=1}^{r}(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'}), & k \neq k'; \\ \sum_{q=1}^{r}(s^3\sum_{i=1}^{s} b_{q,ij}b_{q,ij'} + s\sum_{i=1}^{s} c_{q,ij}c_{q,ij'}), & k = k', j \neq j'. \end{cases} \tag{A.2}$$

Similar arguments lead to, for the case $m = (j-1)2f + 2k$ and $m' = (j'-1)2f + 2k'$, $\mathbf{l}_{j,2k}^{\mathrm{T}}\mathbf{l}_{j',2k'}$ has the same value as in (A.2). We now consider the case $m = (j-1)2f + 2k - 1$ and $m' = (j'-1)2f + 2k'$. The correlation between the column $\mathbf{l}_{j,2k-1}$ and the column $\mathbf{l}_{j',2k'}$ is $\rho_{mm'}(\mathbf{L}) = \mathbf{l}_{j,2k-1}^{\mathrm{T}}\mathbf{l}_{j',2k'}/\{12^{-1}rs^2(r^2s^4-1)\}$, where

$$\mathbf{l}_{j,2k-1}^{\mathrm{T}}\mathbf{l}_{j',2k'} = -s^2\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{u}_{j',2k'-1} - s\mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{u}_{j',2k'-1} + \mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{v}_{j',2k'}$$

$$+s\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{v}_{j',2k'}. \tag{A.3}$$

Here $\mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{v}_{j',2k'} = 0$ because of the pairwise projection property of orthogonal arrays $\mathbf{A}_q$'s and the fact that each column of $\mathbf{C}_q$ sums to zero, for $q = 1, \ldots, r$. Furthermore, in (A.3), for $k \neq k'$, we have $\mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{u}_{j',2k'-1} = \mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{v}_{j',2k'} = 0$ and

$$\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{u}_{j',2k'-1} = \sum_{q=1}^{r}(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'})$$

followed by the pairwise projection property of orthogonal arrays $\mathbf{A}_q$'s, and for $k = k'$, we have (i) $j \neq j'$; (ii) $\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{u}_{j',2k'-1} = \sum_{q=1}^{r}(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'})$ followed by the pairwise projection property of orthogonal arrays $\mathbf{A}_q$'s; (iii) because the two columns, $\mathbf{v}_{j,2k-1}$ and $\mathbf{u}_{j',2k'-1}$, use the same column from $\mathbf{A} = (\mathbf{A}_1^{\mathrm{T}}, \ldots, \mathbf{A}_r^{\mathrm{T}})^{\mathrm{T}}$, we have $\mathbf{v}_{j,2k-1}^{\mathrm{T}}\mathbf{u}_{j',2k'-1} = \sum_{q=1}^{r}\sum_{i=1}^{s} c_{q,ij}b_{q,ij'}$; and (iv) because both the two columns, $\mathbf{u}_{j,2k}$ and $\mathbf{v}_{j',2k'}$, use the same column from $\mathbf{A}$, we have $\mathbf{u}_{j,2k}^{\mathrm{T}}\mathbf{v}_{j',2k'} = \sum_{q=1}^{r} s\sum_{i=1}^{s} b_{q,ij}c_{q,ij'}$. Consequently, for the case $m = (j-1)2f + 2k - 1$ and $m' = (j'-1)2f + 2k'$, we have $\mathbf{l}_{j,2k-1}^{\mathrm{T}}\mathbf{l}_{j',2k'} =$

$$\begin{cases} -s^2\displaystyle\sum_{q=1}^{r}(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'}), & k \neq k'; \\ -s^2\displaystyle\sum_{q=1}^{r}\{-s(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'}) - \sum_{i=1}^{s} c_{q,ij}b_{q,ij'} + \sum_{i=1}^{s} b_{q,ij}c_{q,ij'}\}, & k = k', j \neq j'. \end{cases}$$

Followed by similar arguments, for the case $m = (j-1)2f + 2k$ and $m' = (j'-1)2f + 2k' - 1$, we have $\mathbf{l}_{j,2k}^{\mathrm{T}}\mathbf{l}_{j',2k'-1} =$

$$\begin{cases} -s^2\displaystyle\sum_{q=1}^{r}(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'}), & k \neq k'; \\ -s^2\displaystyle\sum_{q=1}^{r}\{-s(\sum_{i=1}^{s} b_{q,ij})(\sum_{i=1}^{s} b_{q,ij'}) - \sum_{i=1}^{s} b_{q,ij}c_{q,ij'} + \sum_{i=1}^{s} c_{q,ij}b_{q,ij'}\}, & k = k', j \neq j'. \end{cases}$$

Therefore, we complete the proof after defining $t_j^{\mathbf{B}_q}$, $t_j^{\mathbf{C}_q}$, $t_{jj'}^{\mathbf{B}_q}$, $t_{jj'}^{\mathbf{C}_q}$, $t_{jj'}^{\mathbf{B}_q,\mathbf{C}_q}$ and $t_{jj'}^{\mathbf{C}_q,\mathbf{B}_q}$.

Proof of Proposition 3. To show Proposition 3, we need to show that for each level of each column of $\mathbf{E}$, the corresponding rows of $\mathbf{L}$ form a Latin hypercube design after the linear transformation in (2.1) That is, for each level of each column of $\mathbf{E}$, the corresponding rows of $\lceil\{\mathbf{L} + (rs^2+1)/2\}/s\rceil - (rs+1)/2$ is a permutation of $\{-(rs-1)/2, -(rs-3)/2, \ldots, (rs-1)/2\}$, where $\lceil x\rceil$ represents the smallest integer not less than x. For $j = 1, \ldots, p$ and $k = 1, \ldots, f$, consider the column $\mathbf{l}_{j,2k-1}$ of $\mathbf{L}$. We have

$$\begin{aligned}
\left\lceil \frac{\mathbf{l}_{j,2k-1}+(rs^2+1)/2}{s} \right\rceil - \frac{rs+1}{2} &= \left\lceil \frac{\mathbf{v}_{j,2k-1}+s\mathbf{u}_{j,2k}+(rs^2+1)/2}{s} \right\rceil - \frac{rs+1}{2} \\
&= \left\lceil \frac{\mathbf{v}_{j,2k-1}+1/2}{s} + \mathbf{u}_{j,2k} + \frac{rs}{2} \right\rceil - \frac{rs+1}{2} \\
&= \left\lceil \frac{\mathbf{v}_{j,2k-1}+1/2}{s} + \frac{1}{2} + \mathbf{u}_{j,2k} + \frac{rs}{2} - \frac{1}{2} \right\rceil - \frac{rs+1}{2} \\
&= \left\lceil \frac{\mathbf{v}_{j,2k-1}+1/2}{s} + \frac{1}{2} \right\rceil + \mathbf{u}_{j,2k} + \frac{rs}{2} - \frac{1}{2} - \frac{rs+1}{2} \\
&= \mathbf{u}_{j,2k}, \qquad (A.4)
\end{aligned}$$

where the second last step follows because $\mathbf{u}_{j,2k}$ has levels $\{-(rs-1)/2, -(rs-3)/2, \ldots, (rs-1)/2\}$ and $\mathbf{u}_{j,2k} + rs/2 - 1/2$ is always an integer, and the last step follows because $\mathbf{v}_{j,2k-1}$ has levels $\{-(s-1)/2, -(s-3)/2, \ldots, (s-1)/2\}$ and $\lceil(\mathbf{v}_{j,2k-1} + 1/2)/s + 1/2\rceil = 1$. Similarly, for $j = 1, \ldots, p$ and $k = 1, \ldots, f$, the column $\mathbf{l}_{j,2k}$ of $\mathbf{L}$ has

$$\left\lceil \frac{\mathbf{l}_{j,2k}+(rs^2+1)/2}{s} \right\rceil - \frac{rs+1}{2} = -\mathbf{u}_{j,2k-1}. \qquad (A.5)$$

Thus, (A.4) and (A.5) indicate that, for the ith column $\mathbf{l}_i$ of $\mathbf{L}$, $\lceil\{\mathbf{l}_i + (rs^2 + 1)/2\}/s\rceil - (rs+1)/2$ is completely determined by the corresponding column of matrix $\mathbf{U}$ consisting of $\{(\mathbf{u}_{j,2k-1}, \mathbf{u}_{j,2k}) : j = 1, \ldots, p, k = 1, \ldots, f\}$, which in turn is completely determined by the $\mathbf{A}_q$'s and $\mathbf{B}$. If condition (ii) holds, for each level of each column of $\mathbf{E}$, the corresponding rows of the ith column $\mathbf{l}_i$ of $\mathbf{L}$ satisfy that $\lceil\{\mathbf{l}_i+(rs^2+1)/2\}/s\rceil-(rs+1)/2$ is equal to a column of $\mathbf{B}$, and thus a permutation of $\{-(rs-1)/2, -(rs-3)/2, \ldots, (rs-1)/2\}$. We thus complete the proof.

References

Ba, S. (2013). SLHD: Maximin-distance (sliced) Latin hypercube designs. R package version 1.2-1. `http://CRAN.R-project.org/package=SLHD`.

Ba, S., Brenneman, W. A. and Myers, W. R. (2014). Optimal sliced Latin hypercube designs. *Technometrics*, accepted.

Bingham, D., Sitter, R. R. and Tang, B. (2009). Orthogonal and nearly orthogonal designs for computer experiments. *Biometrika* **96**, 51-65.

Carnell, R. (2012). lhs: Latin hypercube samples. R package version 0.10. `http://CRAN.R-project.org/package=lhs`.

Deng, X., Hung, Y. and Lin, C. D. (2014). Design for computer experiments with qualitative and quantitative factors. *Statist. Sinica*, accepted.

Fang, K. T., Lin, D. K., Winker, P. and Zhang, Y. (2000). Uniform design: theory and application. *Technometrics*, **42**, 237-248.

Georgiou, S. D. and Efthimiou, I. (2014). Some classes of orthogonal Latin hypercube designs. *Statist. Sinica* **24**, 101-120.

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

Leary, S., Bhaskar, A. and Keane, A. (2003). Optimal orthogonal-array-based Latin hypercubes. *J. Appl. Statist.* **30**, 585-598.

Lin, C. D., Bingham, D., Sitter, R. R. and Tang, B. (2010). A new and flexible method for constructing designs for computer experiments. *Ann. Statist.* **38**, 1460-1477.

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* (Edited by D. Bingham, A. Dean, M. Morris, and J. Stufken). CRC Press.

Moon, H., Dean, A. and Santner, T. (2011). Algorithms for generating maximin Latin hypercube and orthogonal designs. *J. Statist. Theory Pract.* **5**, 81-98.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computational experiments. *J. Statist. Plann. Inference* **43**, 381-402.

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

Qian, P. Z. G. (2012). Sliced Latin hypercube designs. *J. Amer. Statist. Assoc.* **107**, 393-399.

Sloane, N. J. A. (2014). `http://neilsloane.com/oadir/`.

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. (2009). Construction of orthogonal Latin hypercube designs. *Biometrika* **96**, 971-974.

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

Tang, B. (1993). Orthogonal array-based Latin hypercubes. *J. Amer. Statist. Assoc.* **88**, 1392-1397.

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

Department of Mathematics and Statistics, Queen‘s University, Kingston, ON, K7L 3N6, Canada.

E-mail: cdlin@mast.queensu.ca

Mathematics Department, Illinois Institute of Technology, Chicago, IL 60616, USA.

E-mail: lkang2@iit.edu

(Received January 2015; accepted June 2015)