Statistica Sinica **26** (2016), 653-674
doi:http://dx.doi.org/10.5705/ss.2014.246

BI-DIRECTIONAL SLICED LATIN HYPERCUBE DESIGNS

Qiang Zhou[1], Tian Jin[2], Peter Z. G. Qian[3] and Shiyu Zhou[3]

[1]*City University of Hong Kong,* [2]*Shanghai University of Finance and Economics and* [3]*University of Wisconsin-Madison*

Abstract: We propose a new type of design for computer experiments called bi-directional sliced Latin hypercube design (BSLHD). The proposed design is a special Latin hypercube design (LHD) that simultaneously accommodates two slicing structures. It consists of multiple LHDs of smaller sizes, which can be joined in alternative ways to form two sets of standard sliced LHDs. These new structures are useful for computer experiments with qualitative factors, experiments running in batch mode, and ensembles of multiple computer models. Some sampling properties of the designs in estimating function means are proved and illustrated through numerical examples.

Key words and phrases: Bi-directional slicing, computer experiment, experimental design, qualitative factors, sliced Latin hypercube design.

1. Introduction

Latin hypercube designs are widely used in computer simulation (Fang, Li, and Sudjianto (2005); Santner, Williams, and Notz (2003)), stochastic optimization, and numerical integration. In many applications, the main goal is to estimate the expected output of a computer model given a distribution of inputs. McKay, Beckman, and Conover (1979) introduced the first class of Latin hypercube designs (LHDs), referred to as *ordinary Latin hypercube designs* hereafter. Qian (2012) developed a new type of design called sliced Latin hypercube designs (SLHDs). An SLHD is a special LHD that can be partitioned into several slices, each of which is a smaller LHD. The concept of SLHD is later extended to the general sliced Latin hypercube designs (GSLHDs) in Xie et al. (2014), where the slicing structure has multiple layers to facilitate the implementation of more complex computer experiments.

We propose a new type of design, called *bi-directional sliced Latin hypercube design* (BSLHD). This design simultaneously accommodates two slicing structures. The entire design is a special LHD consisting of elementary LHDs of a much smaller size. Joined in alternative ways, these elementary LHDs are able to form two sets of standard SLHDs. These properties of BSLHD provide significant advantages for computer experiments with various discrete natures. For example, SLHDs are considered to be a superior choice when emulating computer models with qualitative factor(s). Upon cross-validating an emulator for such computer models, the SLHD needs to be further partitioned into multiple folds, each of which consists of evenly partitioned design points at any level of the qualitative factor. It will be advantageous if each fold is itself an SLHD (of a smaller size), because it not only offers an even partition at each level, but also provides the slicing structure across all levels for better emulator fitting and testing during the cross-validation process. To the best of our knowledge, such a design structure can only be achieved by the BSLHDs proposed in this paper. Other potential applications of BSLHDs can be found in collective evaluation of computer models, experiments running in batch mode, etc.

The remainder of this article is organized as follows. Section 2 introduces a method for constructing BSLHDs. Section 3 derives some of their sampling properties. Section 4 presents numerical examples to corroborate the derived properties, and Section 5 concludes with some remarks. All proofs are deferred to supplementary material.

2. Construction

In this section, we present a method for constructing BSLHDs. For a positive integer b, let $\mathbf{Z}_b$ denote the set $\{1, 2, \ldots, b\}$. Drawing a uniform permutation on a set of b integers means randomly taking a permutation on the set, with all $b!$ possible permutations equally probable. For $x \in R$, $\lceil x \rceil$ denotes the smallest integer no less than x. Similarly define $\lceil \mathbf{D} \rceil$ for a real matrix $\mathbf{D}$. For a matrix $\mathbf{M}$, $\mathbf{M}(i, :)$ denotes its ith row, $\mathbf{M}(:, j)$ its jth column, and $\mathbf{M}(i_1 : i_2, j)$ a vector with its entries taken from rows i_1 to i_2 and column j.

A BSLHD $\mathbf{D}$ is associated with four positive integers: m, t, s, q. Let $n = mst$, $r = ms$, $h = mt$ and $p = st$. Throughout, we only consider designs in q dimensions and the dimension is dropped unless otherwise noted. The design $\mathbf{D}$ is an LHD of n runs in q dimensions that consists of p *element designs* with

$$\mathbf{D}_{ij} = \mathbf{D}([(i-1)r + (j-1)m + 1] : [(i-1)r + jm], :),$$

where $i = 1, \ldots, t$, $j = 1, \ldots, s$. Each element design $\mathbf{D}_{ij}$ is itself an LHD with m runs.

Interestingly, these p element designs can be joined in two ways to form LHDs of larger sizes, upon either index i or j. Take $\mathbf{D}_{i\bullet} = \bigcup_{j=1}^{s} \mathbf{D}_{ij}$ and $\mathbf{D}_{\bullet j} = \bigcup_{i=1}^{t} \mathbf{D}_{ij}$. A figurative illustration of $\mathbf{D}$'s structure is given in Figure 1, where the relations among the $\mathbf{D}_{ij}$'s, $\mathbf{D}_{i\bullet}$'s, $\mathbf{D}_{\bullet j}$'s, and $\mathbf{D}$ are presented. There each $\mathbf{D}_{i\bullet}$ is an LHD with r runs and each $\mathbf{D}_{\bullet j}$ is an LHD with h runs. In the notion of standard SLHDs (Qian (2012)), each $\mathbf{D}_{i\bullet}$ or $\mathbf{D}_{\bullet j}$ is an SLHD consisting of some $\mathbf{D}_{ij}$'s as

$$\begin{array}{cccc|c}
\mathbf{D}_{11} & \mathbf{D}_{12} & \cdots & \mathbf{D}_{1s} & \mathbf{D}_{1\bullet} \\
\mathbf{D}_{21} & \mathbf{D}_{22} & \cdots & \mathbf{D}_{2s} & \mathbf{D}_{2\bullet} \\
\vdots & \vdots & & \vdots & \vdots \\
\mathbf{D}_{t1} & \mathbf{D}_{t2} & \cdots & \mathbf{D}_{ts} & \mathbf{D}_{t\bullet} \\
\hline
\mathbf{D}_{\bullet 1} & \mathbf{D}_{\bullet 2} & \cdots & \mathbf{D}_{\bullet s} & \mathbf{D}
\end{array}$$

Figure 1. Structure of a BSLHD ($\mathbf{D}_{ij}$ is an LHD with m runs, $\mathbf{D}_{i\bullet}$ is an LHD with r runs, $\mathbf{D}_{\bullet j}$ is an LHD with h runs, $\mathbf{D}$ is an LHD with n runs).

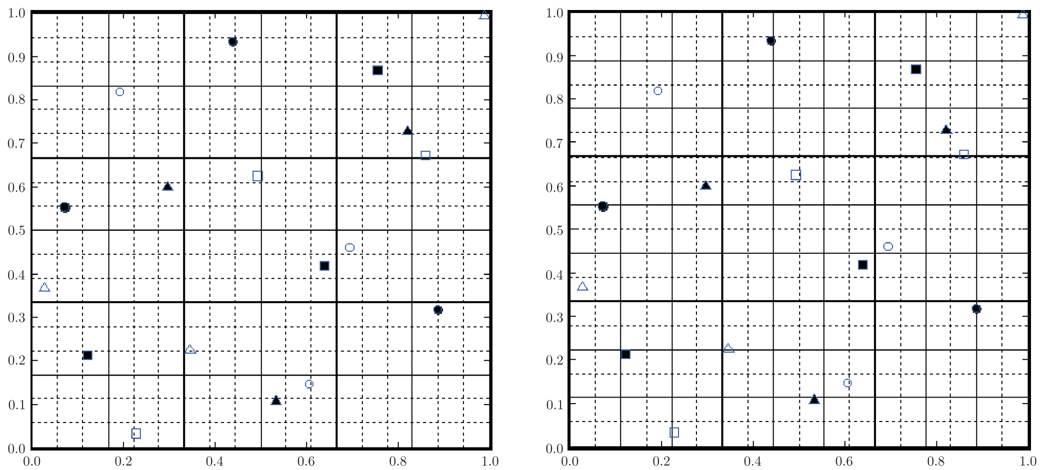
its slices. In addition, $\mathbf{D}$ is a special SLHD containing the slicing structures $\mathbf{D} = \bigcup_{i=1}^{t} \mathbf{D}_{i\bullet}$ and $\mathbf{D} = \bigcup_{j=1}^{s} \mathbf{D}_{\bullet j}$ (hence the name *bi-directional*).

Based on Figure 1, we shall call the two slicing structures *row slicing* and *column slicing* for easy reference and, correspondingly, call $\mathbf{D}_{i\bullet}$ a *row-slice* and $\mathbf{D}_{\bullet j}$ a *column-slice.* The roles of row-slices and column-slices, as well as t and s, are symmetric. Therefore, without loss of generality, assume $t \geq s$ hereinafter.

Figure 2 presents a BSLHD with $m = 3$, $t = 3$, $s = 2$, and $q = 2$ where (1) the design denoted by all the symbols has $n = 18$ total runs in a $q = 2$ dimension space; (2) there are $p = 6$ element designs each with $m = 3$ runs denoted by a distinct symbol; (3) there are $t = 3$ row-slices each with $r = 6$ runs; and (4) there are $s = 2$ column-slices each with $h = 9$ runs. Due to the random nature, Figure 2 shows only one possible realization of the BSLHD for illustration. In practice, additional optimality rules for regular LHDs (e.g., *maximin*) may also be applied here to select a design with the best space-fillingness from a pool of randomly generated candidates.

To facilitate the construction of the BSLHD, we define a bi-directional sliced permutation vector (BSPV) $\pi(m,t,s)$ on $\mathbf{Z}_n$, which is the building block for BSLHD. A BSLHD can be constructed based on q independently generated BSPVs, each of which corresponds to one dimension of the design. Let $\mathbf{S}$ be a BSPV with lth element $\pi(l)$, $l = 1, \ldots, n$. Then (1) $\mathbf{S}$ consists of p *element blocks* each with m numbers: $\mathbf{S}_{ij} = \{\pi((i-1)r + (j-1)m + 1), \ldots, \pi((i-1)r + jm)\}$, $i = 1, \ldots, t$, $j = 1, \ldots, s$, where $\lceil \mathbf{S}_{ij}/p \rceil$ is a permutation on $\mathbf{Z}_m$; (2) with $\mathbf{S}_{i\bullet} = \bigcup_{j=1}^{s} \mathbf{S}_{ij}$ as a *row-block*, and $\mathbf{S}_{\bullet j} = \bigcup_{i=1}^{t} \mathbf{S}_{ij}$ as a *column-block*, $i = 1, \ldots, t$, $j = 1, \ldots, s$, then $\lceil \mathbf{S}_{i\bullet}/t \rceil$ is a permutation on $\mathbf{Z}_r$ and $\lceil S_{\bullet j}/s \rceil$ is a permutation on $\mathbf{Z}_h$.

We now introduce a special table, called nested permutation table (NPT), for storing numbers, which is essential to our construction algorithm. The NPT, $\mathbf{T}$, has $t \times t$ cells, each of which contains $s \times 1$ subcells, making it a table with $p \times t$ subcells. We call the rows formed by cells *rows*, and call the rows formed by subcells *subrows*, so there are t rows / p subrows in an NPT. Numbers from $\mathbf{Z}_p$ are assigned to selected subcells of the $\mathbf{T}$ table. Table 1 shows an example of an NPT table with $t = 3$ and $s = 2$, filled with numbers from $\mathbf{Z}_6$. For the NPT,



(a) three 6-run row-slices ($\mathbf{D}_{1\bullet}$, $\mathbf{D}_{2\bullet}$, $\mathbf{D}_{3\bullet}$) (b) two 9-run column-slices ($\mathbf{D}_{\bullet 1}$, $\mathbf{D}_{\bullet 2}$)

Figure 2. An example of BSLHD $(3, 3, 2, 2)$ viewed as (a) row-slices and (b) column-slices ($\mathbf{D}_{11}$ – ■; $\mathbf{D}_{12}$ – □; $\mathbf{D}_{21}$ – ▲; $\mathbf{D}_{22}$ – △; $\mathbf{D}_{31}$ – ●; $\mathbf{D}_{32}$ – ○; $\mathbf{D} = \bigcup_{i,j} \mathbf{D}_{ij}$ is an 18-run LHD, each $\mathbf{D}_{i\bullet} = \bigcup_{j=1}^{s} \mathbf{D}_{ij}$ is a 6-run LHD; each $\mathbf{D}_{\bullet j} = \bigcup_{i=1}^{t} \mathbf{D}_{ij}$ is a 9-run LHD; each $\mathbf{D}_{ij}$ is a 3-run LHD).

Table 1. An example of NPT with $t = 3$ and $s = 2$.

		1
2		
	3	
	4	
		5
6		

we use $t_{i,j}$ to denote the subcell at the ith subrow and jth column, $i = 1, \ldots, p$, $j = 1, \ldots, t$. For example, we have $t_{4,2} = 4$ and $t_{6,1} = 6$ in Table 1.

Divide $\mathbf{Z}_p$ into s blocks each with t consecutive numbers, with the jth block being $\{(j-1)t+1, \ldots, jt\}$, $j = 1, \ldots, s$. We describe the algorithm to construct a BSPV $\pi(m, t, s)$.

Step 1. For $j = 1, \ldots, s$, fill the empty NPT $\mathbf{T}$ with $t_{(j-1)t+i,\pi(i)} = (j-1)t + i$, $i = 1, \ldots, t$, where $\{\pi(1), \ldots, \pi(t)\}$ is a uniform permutation on $\mathbf{Z}_t$, with the permutations carried out independently for each j. Based on $\mathbf{T}$, fill an $s \times t$ empty matrix $\mathbf{Q}$ with the p numbers from $\mathbf{Z}_p$, such that, for any $j = 1, \ldots, s$, $\lceil \mathbf{Q}(j, :)/s \rceil$ is a permutation on $\mathbf{Z}_t$, and for any $i = 1, \ldots, t$, $\lceil \mathbf{Q}(:, i)/t \rceil$ is a permutation on $\mathbf{Z}_s$. (See Appendix for details of finding such $\mathbf{Q}$). Randomly permute the rows and then columns of $\mathbf{Q}$ to obtain $\mathbf{Q}'$.

Table 2. An NPT with $t = 4$ and $s = 3$.

		1	
2			
	3		
			4
5			
			6
		7	
	8		
	9		
		10	
			11
12			

Step 2. Repeat Step 1 independently m times to obtain m matrices $\mathbf{Q}_1', \ldots, \mathbf{Q}_m'$. For $\mathbf{Q}_l$, let $\bar{\mathbf{Q}}_l = \mathbf{Q}_l' + p(l-1)$, $l = 1, \ldots, m$.

Step 3. For an empty $r \times t$ matrix $\mathbf{W}$ whose (i,j)th element is $w_{i,j}$, let $\mathbf{W}(((j-1)m + l), :) = \bar{\mathbf{Q}}_l(j, :)$, $j = 1, \ldots, s$, $l = 1, \ldots, m$. For each group of m elements $\mathbf{W}(((j-1)m + 1) : jm, i)$, $i = 1, \ldots, t$ and $j = 1, \ldots, s$, randomly permute them within those m locations. Let $\pi((j-1)r + i) = w_{i,j}$, $i = 1, \ldots, r$, $j = 1, \ldots, t$, then $\{\pi(1), \ldots, \pi(n)\}$ is a BSPV $\pi(m, t, s)$ on $\mathbf{Z}_n$.

We illustrate the construction of $\pi(m, t, s)$ step by step for $m = 2$, $t = 4$ and $s = 3$. Based on the relations previously defined, we have $n = 24$, $r = 6$, $h = 8$, and $p = 12$.

First, use Step 1 to construct an $s \times t$ matrix $\mathbf{Q}$ on $\mathbf{Z}_{12}$. The $s = 3$ blocks from $\mathbf{Z}_{12}$ are $\{1, 2, 3, 4\}$, $\{5, 6, 7, 8\}$, and $\{9, 10, 11, 12\}$. The $s = 3$ uniform random permutations on $\mathbf{Z}_4$ we have obtained are $\{3, 1, 2, 4\}$, $\{1, 4, 3, 2\}$, and $\{2, 3, 4, 1\}$. The corresponding NPT $\mathbf{T}$ based on the permutations is given in Table 2.

To find a $\mathbf{Q}$ matrix with the desired properties described in Step 1, we use the procedures in the Appendix. First, we find the 4×4 $\mathbf{G}$ matrix whose entries equal the number of elements inside each *cell* (not *subcell*) of $\mathbf{T}$ given in Table 2:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

For example, number "2" at the 2nd row and 4th column of $\mathbf{G}$ means there are two numbers ("4" and "6") in the corresponding *cell* of $\mathbf{T}$. The matrix $\mathbf{G}$ is then decomposed into three permutation matrices, with one possible result being

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$= \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Corresponding to these permutation matrices, we might extract these three vectors from $\mathbf{T}$: $(5,8,1,11)$, $(2,9,10,6)$, $(12,3,7,4)$. This set of extracted vectors is not unique when there is at least one entry larger than 1 in matrix $\mathbf{G}$ (refer to the Appendix for details). These vectors are then joined row by row to obtain

$$\mathbf{Q}_1 = \begin{bmatrix} 5 & 8 & 1 & 11 \\ 2 & 9 & 10 & 6 \\ 12 & 3 & 7 & 4 \end{bmatrix}.$$

Randomly permute the rows and then columns of $\mathbf{Q}_1$, with possible outcome being

$$\mathbf{Q}_1' = \begin{bmatrix} 11 & 5 & 8 & 1 \\ 4 & 12 & 3 & 7 \\ 6 & 2 & 9 & 10 \end{bmatrix}.$$

For $l = 1$, we have $\bar{\mathbf{Q}}_1 = \mathbf{Q}_1'$. Repeat the above steps for $l = 2$, with one possible outcome being

$$\bar{\mathbf{Q}}_2 \begin{bmatrix} 15 & 21 & 17 & 24 \\ 19 & 13 & 23 & 16 \\ 22 & 20 & 14 & 18 \end{bmatrix}.$$

In Step 3, we obtain a possible $\mathbf{W}$ matrix as

$$\mathbf{W} = \begin{bmatrix} 11 & 21 & 8 & 24 \\ 15 & 5 & 17 & 1 \\ 4 & 12 & 3 & 16 \\ 19 & 13 & 23 & 7 \\ 22 & 20 & 9 & 18 \\ 6 & 2 & 14 & 10 \end{bmatrix}.$$

Finally, we use $\mathbf{W}$ to obtain the following vector:

$$\begin{aligned} \boldsymbol{v}^T &= \pi(2,4,3) \\ &= (11,15,4,19,22,6,21,5,12,13,20,2,8,17,3,23,9,14,24,1,16,7,18,10). \end{aligned}$$

Now we verify the properties of $\boldsymbol{v}$. It has $p = 12$ element blocks defined by $\mathbf{S}_{ij} = (\pi(6(i-1)+2j-1), \pi(6(i-1)+2j))$, $i = 1, \ldots, 4$, and $j = 1, 2, 3$.

(1) $\lceil \boldsymbol{v}^T/12 \rceil = ((1,2), (1,2), (2,1), (2,1), (1,2), (2,1), (1,2), (1,2), (1,2), (2,1), (2,1), (2,1))$. This verifies that each $\lceil \mathbf{S}_{ij}/12 \rceil$ is a permutation on $\mathbf{Z}_2$.

(2) For easier illustration with $t = 4$ row-blocks and $s = 3$ column-blocks, we define $\mathbf{W}_1$ and $\mathbf{W}_2$ as:

$$\mathbf{W}_1 = \begin{bmatrix} \mathbf{S}_{1\bullet}^T \\ \mathbf{S}_{2\bullet}^T \\ \mathbf{S}_{3\bullet}^T \\ \mathbf{S}_{4\bullet}^T \end{bmatrix}, \quad \mathbf{W}_2 = \begin{bmatrix} \mathbf{S}_{\bullet 1}^T \\ \mathbf{S}_{\bullet 2}^T \\ \mathbf{S}_{\bullet 3}^T \end{bmatrix},$$

where each row-block is a row in $\mathbf{W}_1$ and each column-block is a row in $\mathbf{W}_2$. We then obtain

$$\lceil \mathbf{W}_1/4 \rceil = \begin{bmatrix} 3 & 4 & 1 & 5 & 6 & 2 \\ 6 & 2 & 3 & 4 & 5 & 1 \\ 2 & 5 & 1 & 6 & 3 & 4 \\ 6 & 1 & 4 & 2 & 5 & 3 \end{bmatrix}, \quad \lceil \mathbf{W}_2/3 \rceil = \begin{bmatrix} 4 & 5 & 7 & 2 & 3 & 6 & 8 & 1 \\ 2 & 7 & 4 & 5 & 1 & 8 & 6 & 3 \\ 8 & 2 & 7 & 1 & 3 & 5 & 6 & 4 \end{bmatrix}.$$

This verifies that each row-block $\mathbf{S}_{i\bullet}$ is a permutation on $\mathbf{Z}_6$, and each column-block $\mathbf{S}_{\bullet j}$ is a permutation on $\mathbf{Z}_8$. The vector $\boldsymbol{v}$ here is indeed a BSPV.

To construct a BSLHD (m, t, s, q), we independently generate q BSPVs: $\pi_k(m, t, s)$, and fill an empty $n \times q$ matrix by letting its lth row and kth column entry be $\pi_k(l)$, $l = 1, \ldots, n$, $k = 1, \ldots, q$. Based on this matrix, an $n \times q$ BSLHD $\mathbf{D}$ is generated through

$$d_{lk} = \frac{(\pi_k(l) - u_{lk})}{n}, \quad l = 1, \ldots, n, \; k = 1, \ldots, q, \tag{2.1}$$

where the u_{lk}'s are independent $U(0, 1]$ random variables, $\pi_k(l)$'s and u_{lk}'s are mutually independent.

3. Sampling Properties

In this section, we present some sampling properties of the proposed design. We have assumed $t \geq s$. Due to its complex combinatorial construction, we derive the sampling properties of BSLHD for $t = cs$, where c is any positive integer. If $t \neq cs$, the distribution function of $\pi(m, t, s)$ is more complicated because of the dependence among the rows in the nested permutation table. Performances of the proposed designs for $t \neq cs$ will be demonstrated through the numerical example.

In a similar spirit as SLHD and GSLHD, one major advantage of BSLHD lies in the collective evaluation of computer models. For example, we are interested in a physical system with s variants (e.g., s operating modes or system configurations). There are t different computer models available for simulating this system, which might be built based on t different algorithms. Each computer model contains a categorical variable with s levels, corresponding to the s variants of the physical system it describes. Denote the computer models as f_{ij}, $i = 1, \ldots, t$, $j = 1, \ldots, s$, where f_{ij} is the jth variant of the ith computer model. Assume each f_{ij} has factors $\mathbf{x} = (x^1, \ldots, x^q)$ whose distribution is the uniform measure on $(0, 1]^q$, denoted by F. For $i = 1, \ldots, t$, $j = 1, \ldots, s$, define $\mu_{ij} = E[f_{ij}(\mathbf{x})]$. For $i_1, i_2 = 1, \ldots, t$, $j_1, j_2 = 1, \ldots, s$, define $\text{Cov}_{\,i_1 j_1 i_2 j_2} = \text{Cov}\,[f_{i_1 j_1}(\mathbf{x}), f_{i_2 j_2}(\mathbf{x})]$, which is $\sigma^2_{i_1 j_1} = \text{Var}\,[f_{i_1 j_1}(\mathbf{x})]$ if $i_1 = i_2$ and $j_1 = j_2$. The mean value μ_{ij} can be estimated by running f_{ij} at m selected input values. For $0 \le \lambda_{ij} \le 1$, $i = 1, \ldots, t$, $j = 1, \ldots, s$, the following linear combinations are of interest in practice:

$$\mu_{i\bullet} = \sum_j \lambda_{ij}\mu_{ij},$$

$$\mu_{\bullet j} = \sum_i \lambda_{ij}\mu_{ij},$$

$$\mu = \sum_i \sum_j \lambda_{ij}\mu_{ij}.$$

In collectively evaluating t different computer models each with s variants, $\mu_{i\bullet}$ gives the weighted mean output of the ith model across all of its variants, $\mu_{\bullet j}$ gives the weighted mean output of all t models for their jth variant, and μ is the weighted grand mean.

Consider the following schemes to achieve this goal.

Definition 1. Let m, s, t, and p be strictly positive integers with $n = mp = mst$.

(i) Let IID denote a scheme that takes an independent and identically distributed sample of m runs for each f_{ij}, with the p samples generated independently.

(ii) Let LH denote the scheme that obtains p independent ordinary LHDs of m runs, each of which is associated with one f_{ij}.

(iii) Let SLH denote the scheme that produces an $n \times q$ SLHD with p slices by using the method in Qian (2012), where each slice is a smaller LHD with m runs and randomly assigned to one f_{ij}.

(iv) Let S-ROW denote the scheme that independently produces t SLHDs of size $ms \times q$, each of which has s slices, by using the method in Qian (2012). For $i = 1, \ldots, t$, $j = 1, \ldots, s$, assign the jth slice of the ith SLHD to f_{ij}.

(v) Let S-COL denote the scheme that independently produces s SLHDs of size $mt \times q$, each of which has t slices, by using the method in Qian (2012). For $i = 1, \ldots, t$, $j = 1, \ldots, s$, assign the ith slice of the jth SLHD to f_{ij}.

(vi) Let GS-ROW denote the scheme that produces an $n \times q$ GSLHD of two layers by using the method in Xie et al. (2014), where its first layer contains s slices and its second layer contains t slices, $s_1 = s$ and $s_2 = t$ in their notation. For $i = 1, \ldots, t$, and $j = 1, \ldots, s$, the sub-design at the jth slice of the first layer and the ith slice of the second layer is an LHD with m runs and is assigned to f_{ij}.

(vii) Let GS-COL denote the scheme that produces an $n \times q$ GSLHD of two layers by using the method in Xie et al. (2014), where its first layer contains t slices and its second layer contains s slices, i.e., $s_1 = t$ and $s_2 = s$ in their notation. For $i = 1, \ldots, t$, and $j = 1, \ldots, s$, the sub-design at the ith slice of the first layer and the jth slice of the second layer is an LHD with m runs and is assigned to f_{ij}.

(viii) Let BSLH denote the scheme that produces a BSLHD (m, t, s, q). For $i = 1, \ldots, t$, and $j = 1, \ldots, s$, its element design $\mathbf{D}_{ij}$ is assigned to f_{ij}.

Expectations, variances, and covariances under the above schemes (in the same order) are denoted by the subscripts IID, LH, SLH, S-ROW, S-COL, GS-ROW, GS-COL and BSLH, respectively. For any of these schemes, let $\mathbf{D}_{ij}$ denote its design set for f_{ij}, and $\mathbf{x}$ denote one row of $\mathbf{D}$. For $i = 1, \ldots, t$, $j = 1, \ldots, s$, we have the following estimators

$$\begin{aligned} \hat{\mu}_{ij} &= m^{-1} \sum_{\mathbf{x} \in \mathbf{D}_{ij}} f_{ij}(\mathbf{x}), \\ \hat{\mu}_{i\bullet} &= \sum_{j} \lambda_{ij} \hat{\mu}_{ij}, \\ \hat{\mu}_{\bullet j} &= \sum_{i} \lambda_{ij} \hat{\mu}_{ij}, \\ \hat{\mu} &= \sum_{i} \sum_{j} \lambda_{ij} \hat{\mu}_{ij}. \end{aligned} \tag{3.1}$$

For later development, we describe the ANOVA decomposition of integrable functions on $(0, 1]^q$ (Owen (1992); Loh (1996)). With F the uniform measure on $(0, 1]^q$, let $dF = \prod_{k=1}^{q} dF^k$ and $dF^{-k} = \prod_{i \neq k} dF^i$. If $f\colon R^q \to R$ is a measurable function of $\mathbf{x} = (x^1, \ldots, x^q)$ and $E\{[f(\mathbf{x})]^2\}$ is well defined and finite, then f can be decomposed as

$$f(\mathbf{x}) = \mu + \sum_{k=1}^{q} f^{-k}(x^k) + r(\mathbf{x}),$$

where $\mu = \int f(\mathbf{x})\, dF$ is the grand mean and the functional main effect of x^k is

$$f^{-k}(x^k) = \int [f(\mathbf{x}) - \mu]\, dF^{-k}, \quad \text{for } k = 1, \ldots, q. \tag{3.2}$$

The residual term, $r(\mathbf{x})$, contains all possible functional interaction effects of the function $f(\mathbf{x})$. Explicit forms of interaction effects are omitted here due to irrelevance in the later development, but can be found in Owen (1992). For $k = 1, \ldots, q$, $\int f^{-k}\, dF^k = 0$ and $\int r(\mathbf{x})\, dF^{-k} = 0$. Define

$$\gamma_z(a,b) = \begin{cases} 1, & \text{if } \lceil \frac{a}{z} \rceil = \lceil \frac{b}{z} \rceil, \\ 0, & \text{if } \lceil \frac{a}{z} \rceil \neq \lceil \frac{b}{z} \rceil, \end{cases} \tag{3.3}$$

where a, b and z are positive real numbers. We present some results on the probability mass function of the proposed BSPV under the condition that $t = cs$, $c = 1, 2, \ldots$

Lemma 1. *For m, t, s in Definition 1 and $t = cs$, $c = 1, 2, \ldots$, let $\pi(m,t,s) = \{\pi(1), \ldots, \pi(n)\}$ be a BSPV. For any two elements $\pi(l_1)$ and $\pi(l_2)$ from $\pi(m,t,s)$, $l_1 \neq l_2$, $l_1, l_2 = 1, \ldots, n$, they each belongs to an element block in $\mathbf{S}$. Assume $\pi(l_1) \in \mathbf{S}_{i_1 j_1}$ and $\pi(l_2) \in \mathbf{S}_{i_2 j_2}$, $i_1, i_2 = 1, \ldots, t$; $j_1, j_2 = 1, \ldots, s$. Based on the relationships between i_1 and i_2, j_1 and j_2, define the groups*

$$\begin{aligned} B_1 &= \{(\pi(l_1), \pi(l_2)) \mid i_1 = i_2 \text{ and } j_1 = j_2\}, \\ B_2 &= \{(\pi(l_1), \pi(l_2)) \mid i_1 = i_2 \text{ and } j_1 \neq j_2\}, \\ B_3 &= \{(\pi(l_1), \pi(l_2)) \mid i_1 \neq i_2 \text{ and } j_1 = j_2\}, \\ B_4 &= \{(\pi(l_1), \pi(l_2)) \mid i_1 \neq i_2 \text{ and } j_1 \neq j_2\}. \end{aligned}$$

For any $u, v \in \mathbf{Z}_n$, $u \neq v$, we have that

(1) *For $l = 1, \ldots, n$, the probability mass function of $\pi(l)$ is*

$$\Pr\{\pi(l) = u\} = \frac{1}{n}.$$

(2) *For any l_1, l_2, we have the following cases for the joint probability mass function*

(i) *if $(\pi(l_1), \pi(l_2)) \in B_1$*

$$\Pr\{\pi(l_1) = u, \pi(l_2) = v\} = n^{-2} + \frac{\gamma_n(u,v) - m\gamma_p(u,v)}{n^2(m-1)};$$

(ii) if $(\pi(l_1), \pi(l_2)) \in B_2$

$$\Pr\{\pi(l_1) = u, \pi(l_2) = v\} = n^{-2} + \frac{\gamma_p(u,v) - s\gamma_t(u,v)}{n^2(s-1)};$$

(iii) if $(\pi(l_1), \pi(l_2)) \in B_3$

$$\Pr\{\pi(l_1) = u, \pi(l_2) = v\} = n^{-2} + \frac{\gamma_p(u,v) - t\gamma_s(u,v)}{n^2(t-1)};$$

(iv) if $(\pi(l_1), \pi(l_2)) \in B_4$

$$\begin{aligned}\Pr\{\pi(l_1) &= u, \pi(l_2) = v\} \\ &= n^{-2} + \frac{-\gamma_p(u,v) + s\gamma_t(u,v) + t\gamma_s(u,v) - p\gamma_1(u,v)}{n^2(s-1)(t-1)}.\end{aligned}$$

For $\mathbf{D}_{ij}$, $\mathbf{D}_{i\bullet}$, and $\mathbf{D}_{\bullet j}$ defined earlier, we have the following result.

Lemma 2. *Let m, s, t be strictly positive integers with $n = mst$ and $t = cs$, $c = 1, 2, \ldots$ Consider a BSLHD denoted as $\mathbf{D}$. We have that*

(i) *each $\mathbf{D}_{ij}$, $i = 1, \ldots, t$, $j = 1, \ldots, s$ is statistically equivalent to an $m \times q$ standard LHD;*

(ii) *each $\mathbf{D}_{i\bullet}$, $i = 1, \ldots, t$, is statistically equivalent to an $ms \times q$ SLHD with s slices each containing m runs;*

(iii) *each $\mathbf{D}_{\bullet j}$, $j = 1, \ldots, s$, is statistically equivalent to an $mt \times q$ SLHD with t slices each containing m runs.*

Let $\mathbf{x}_l = (x_l^1, \ldots, x_l^q)$ denote the lth row of $\mathbf{D}$, $l = 1, \ldots, n$.

Lemma 3. *For $l = 1, \ldots, n$, the marginal distribution of $\mathbf{x}_l$ is uniform on $(0, 1]^q$.*

Lemma 4. *For strictly positive integers m, t, s, q with $n = mts$ and $t = cs$, $c = 1, 2, \ldots$, let $\mathbf{x}_{l_1}$ and $\mathbf{x}_{l_2}$ be the l_1th and l_2th rows of a BSLHD (m, t, s, q) denoted as $\mathbf{D}$, $l_1 \neq l_2$, $l_1, l_2 = 1, \ldots, n$. Assume $\mathbf{x}_{l_1} \in \mathbf{D}_{i_1 j_1}$ and $\mathbf{x}_{l_2} \in \mathbf{D}_{i_2 j_2}$, $i_1, i_2 = 1, \ldots, t$ and $j_1, j_2 = 1, \ldots, s$. The joint density function of $\mathbf{x}_{l_1}$ and $\mathbf{x}_{l_2}$ is as follows.*

(i) *If $i_1 = i_2$ and $j_1 = j_2$*

$$p(\mathbf{x}_{l_1}, \mathbf{x}_{l_2}) = \prod_{k=1}^{q} \left\{ 1 + \frac{\delta_1(x_{l_1}^k, x_{l_2}^k) - m\delta_m(x_{l_1}^k, x_{l_2}^k)}{m-1} \right\}.$$

(ii) *If $i_1 = i_2$ and $j_1 \neq j_2$*

$$p(\mathbf{x}_{l_1}, \mathbf{x}_{l_2}) = \prod_{k=1}^{q} \left\{ 1 + \frac{\delta_m(x_{l_1}^k, x_{l_2}^k) - s\delta_r(x_{l_1}^k, x_{l_2}^k)}{s-1} \right\}.$$

(iii) *If $i_1 \neq i_2$ and $j_1 = j_2$*

$$p(\mathbf{x}_{l_1}, \mathbf{x}_{l_2}) = \prod_{k=1}^{q} \left\{ 1 + \frac{\delta_m(x_{l_1}^k, x_{l_2}^k) - t\delta_h(x_{l_1}^k, x_{l_2}^k)}{t-1} \right\}.$$

(iv) *If $i_1 \neq i_2$ and $j_1 \neq j_2$*

$$p(\mathbf{x}_{l_1}, \mathbf{x}_{l_2}) = \prod_{k=1}^{q} \left\{ 1 + \frac{-\delta_m(x_{l_1}^k, x_{l_2}^k) + s\delta_r(x_{l_1}^k, x_{l_2}^k) + t\delta_h(x_{l_1}^k, x_{l_2}^k) - p\delta_n(x_{l_1}^k, x_{l_2}^k)}{(s-1)(t-1)} \right\},$$

where $x_{l_1}^k, x_{l_2}^k \in (0,1]$ and the δ function is defined as

$$\delta_c(y,z) = \gamma_{1/c}(y,z), \tag{3.4}$$

where γ is defined in (3.3).

The unbiasedness of the estimators in (3.1) has been proven by Lemma 3. The following theorem discusses their variances under a monotonicity assumption.

Theorem 1. *For strictly positive integers t and s with $t = cs$, $c = 1, 2, \ldots$, suppose that $f_{ij}(\mathbf{x})$ is monotonic in each argument of $\mathbf{x} = (x^1, \ldots, x^q)$, $i = 1, \ldots, t$, $j = 1, \ldots, s$, and any pair of functions $f_{i_1 j_1}$ and $f_{i_2 j_2}$, is jointly increasing or decreasing in each argument of $\mathbf{x}$, $i_1, i_2 = 1, \ldots, t$ and $j_1, j_2 = 1, \ldots, s$. For the design schemes in Definition 1, we have the following results for the estimators in* (3.1).

(i) *For $i = 1, \ldots, t$, $j = 1, \ldots, s$,*

$$\begin{aligned} \mathrm{Var}_{BSLH}(\hat{\mu}_{ij}) &= \mathrm{Var}_{GS-ROW}(\hat{\mu}_{ij}) = \mathrm{Var}_{GS-COL}(\hat{\mu}_{ij}) = \mathrm{Var}_{S-ROW}(\hat{\mu}_{ij}) \\ &= \mathrm{Var}_{S-COL}(\hat{\mu}_{ij}) = \mathrm{Var}_{SLH}(\hat{\mu}_{ij}) = \mathrm{Var}_{LH}(\hat{\mu}_{ij}) \le \mathrm{Var}_{IID}(\hat{\mu}_{ij}). \end{aligned}$$

(ii) *For $i = 1, \ldots, t$,*

$$\begin{aligned} \mathrm{Var}_{BSLH}(\hat{\mu}_{i\bullet}) &= \mathrm{Var}_{GS-ROW}(\hat{\mu}_{i\bullet}) \\ &= \mathrm{Var}_{S-ROW}(\hat{\mu}_{i\bullet}) \le \mathrm{Var}_{LH}(\hat{\mu}_{i\bullet}) \le \mathrm{Var}_{IID}(\hat{\mu}_{i\bullet}). \end{aligned}$$

(iii) *For $j = 1, \ldots, s$,*

$$\begin{aligned} \mathrm{Var}_{BSLH}(\hat{\mu}_{\bullet j}) &= \mathrm{Var}_{GS-COL}(\hat{\mu}_{\bullet j}) \\ &= \mathrm{Var}_{S-COL}(\hat{\mu}_{\bullet j}) \le \mathrm{Var}_{LH}(\hat{\mu}_{\bullet j}) \le \mathrm{Var}_{IID}(\hat{\mu}_{\bullet j}). \end{aligned}$$

Theorem 1 shows that the proposed design is consistently the best in achieving variance reduction among all the designs. Note that the monotonicity assumption does not hold for computer experiments in many cases. The next theorem gives a more general result by dropping this assumption.

Theorem 2. *Suppose that $E\{[f_{ij}(\mathbf{x})]^2\}$, $i = 1, \ldots, t$, $j = 1, \ldots, s$, are all well-defined and finite. Let f_{ij}^{-k} be the functional main effect for the variable x^k of $\mathbf{x} = (x^1, \ldots, x^q)$ in the ANOVA decomposition of f_{ij} in* (3.2). *Let m, t, s be positive integers with $n = mts$ and $t = cs$, $c = 1, 2, \ldots$. Then, as $n \to \infty$ with t and s fixed, we have that*

(i) *For* $i = 1, \ldots, t$, $j = 1, \ldots, s$, *and* $\hat{\mu}_{ij}$ *defined in* (3.1),

$$\text{Var}_{BSLH}(\hat{\mu}_{ij}) = \frac{1}{m}\sigma_{ij}^2 - \frac{1}{m}\sum_{k=1}^{q}\int_0^1 \{f_{ij}^{-k}(x)\}^2\, dx + o(m^{-1}).$$

(ii) *For* $i = 1, \ldots, t$, $j = 1, \ldots, s$, *and* $\hat{\mu}_{i\bullet}$, $\hat{\mu}_{\bullet j}$ *defined in* (3.1),

$$\text{Var}_{BSLH}(\hat{\mu}_{i\bullet}) = \frac{1}{m}\sum_{j=1}^{s}\lambda_{ij}^2\sigma_{ij}^2 - \frac{1}{m}\sum_{j=1}^{s}\sum_{k=1}^{q}\lambda_{ij}^2\int_0^1 \{f_{ij}^{-k}(x)\}^2\, dx + o(r^{-1}),$$

$$\text{Var}_{BSLH}(\hat{\mu}_{\bullet j}) = \frac{1}{m}\sum_{i=1}^{t}\lambda_{ij}^2\sigma_{ij}^2 - \frac{1}{m}\sum_{i=1}^{t}\sum_{k=1}^{q}\lambda_{ij}^2\int_0^1 \{f_{ij}^{-k}(x)\}^2\, dx + o(h^{-1}).$$

(iii) *For* $\hat{\mu}$ *defined in* (3.1),

$$\text{Var}_{BSLH}(\hat{\mu}) = \frac{1}{m}\sum_{i=1}^{t}\sum_{j=1}^{s}\lambda_{ij}^2\sigma_{ij}^2 - \frac{1}{m}\sum_{i=1}^{t}\sum_{j=1}^{s}\sum_{k=1}^{q}\lambda_{ij}^2\int_0^1 \{f_{ij}^{-k}(x)\}^2\, dx + o(n^{-1}).$$

Remark 1. In Theorem 2, the results in cases (i) and (ii) follow directly from Stein (1987), Loh (1996), and Qian (2012) due to Lemma 2: $\mathbf{D}_{ij}$'s are statistically equivalent to ordinary LHDs, $\mathbf{D}_{i\bullet}$'s and $\mathbf{D}_{\bullet j}$'s are statistically equivalent to SLHDs. In case (iii), if the f_{ij}'s are the same with $f_{ij} = f$ and $\lambda_{ij} = 1/p$, we have $\sigma_{ij}^2 = \sigma^2$ and the variance reduces to

$$\text{Var}_{BSLH}(\hat{\mu}) = \frac{1}{n}\sigma^2 - \frac{1}{n}\sum_{k=1}^{q}\int_0^1 \{f^{-k}(x)\}^2\, dx + o(n^{-1}),$$

which is similar to that of an ordinary LHD of n runs as given in Stein (1987) and Loh (1996). This suggests that BSLHD, as a whole design, achieves the similar degree of variance reduction as ordinary LHD, in addition to its desirable structural properties.

4. Numerical Examples

In this section, two numerical examples are used to compare the properties of the BSLHD with other design schemes. In addition to the designs of Definition 1, a design scheme based on a Sobol sequence (Sobol (1967)) is also used. The Sobol sequence is one of the most popular quasi Monte Carlo methods that generate low-discrepancy point sets intended for numerical integration. In this design scheme (coded as SS), each f_{ij} is evaluated based on the first m points consecutively taken from a scrambled Sobol sequence.

Table 3. RMSEs of the nine design schemes for Example 1, $m = 5$.

		IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
	$\hat{\mu}_{11}$	0.298	0.145	0.066	0.066	0.066	0.067	0.066	0.066	0.067
$t = 2$	$\hat{\mu}_{1\bullet}$	0.105	0.028	0.024	0.020	0.012	0.024	0.012	0.022	0.012
$s = 2$	$\hat{\mu}_{\bullet 1}$	0.105	0.051	0.023	0.019	0.023	0.012	0.022	0.012	0.012
	$\hat{\mu}$	0.150	0.020	0.033	0.008	0.017	0.017	0.008	0.008	0.008
	$\hat{\mu}_{11}$	0.298	0.145	0.067	0.067	0.067	0.067	0.066	0.067	0.067
$t = 3$	$\hat{\mu}_{1\bullet}$	0.070	0.018	0.016	0.014	0.008	0.016	0.008	0.015	0.008
$s = 2$	$\hat{\mu}_{\bullet 1}$	0.087	0.044	0.019	0.015	0.019	0.006	0.018	0.006	0.006
	$\hat{\mu}$	0.120	0.018	0.028	0.005	0.014	0.009	0.005	0.005	0.005

Table 4. RMSEs of the nine design schemes for Example 1, $m = 10$.

		IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
	$\hat{\mu}_{11}$	0.212	0.055	0.023	0.024	0.024	0.024	0.024	0.023	0.024
$t = 2$	$\hat{\mu}_{1\bullet}$	0.074	0.010	0.008	0.007	0.004	0.008	0.004	0.008	0.004
$s = 2$	$\hat{\mu}_{\bullet 1}$	0.075	0.021	0.008	0.007	0.008	0.004	0.008	0.004	0.004
	$\hat{\mu}$	0.104	0.007	0.012	0.003	0.006	0.006	0.003	0.003	0.003
	$\hat{\mu}_{11}$	0.209	0.056	0.023	0.024	0.023	0.024	0.024	0.023	0.023
$t = 3$	$\hat{\mu}_{1\bullet}$	0.049	0.007	0.006	0.005	0.003	0.006	0.003	0.005	0.003
$s = 2$	$\hat{\mu}_{\bullet 1}$	0.061	0.020	0.007	0.005	0.007	0.002	0.006	0.002	0.002
	$\hat{\mu}$	0.085	0.006	0.010	0.002	0.005	0.003	0.002	0.002	0.002

Example 1. Consider a function with five-dimensional input:

$$f(\mathbf{x}) = \sum_{i=1}^{5} x_i^2,$$

where $\mathbf{x}$ is uniform on $(0, 1]^5$.

This function is treated as a computer code. We run the computer code in batch mode with fixed batch size m. For integers s and t, let $p = st$ denote the total number of batches tested. Let μ be the true function mean, and $\hat{\mu}_{ij}$ be the sample average from one batch with its corresponding design set denoted as $\mathbf{D}_{ij}$, $i = 1, \ldots, t$, $j = 1, \ldots, s$. Here we use the same function for all batches, $f_{ij}(\mathbf{x}) = f(\mathbf{x})$, with $\lambda_{ij} = \lambda = 1/p$. For each design scheme in Definition 1 plus the Sobol sequence based design, we are interested in estimating $\hat{\mu}_{11}$, $\hat{\mu}_{1\bullet}$, $\hat{\mu}_{\bullet 1}$ and $\hat{\mu}$ as defined in (3.1).

We set up two scenarios to test all design schemes in estimating the four quantities. In the first scenario, $s = t = 2$; in the second scenario, $s = 2$ and $t = 3$. For each design scheme in each scenario, we computed $\hat{\mu}_{11}$, $\hat{\mu}_{1\bullet}$, $\hat{\mu}_{\bullet 1}$ and $\hat{\mu}$ 10,000 times for $m = 5, 10, 20, 40$. Tables 3$-$6 present the root mean squared errors (RMSE) of $\hat{\mu}_{11}$, $\hat{\mu}_{1\bullet}$, $\hat{\mu}_{\bullet 1}$ and $\hat{\mu}$ based on 10,000 estimates.

The four tables indicate that BSLH achieves the smallest variances in all scenarios, while the other eight design schemes are weak in estimating certain

Table 5. RMSEs ($\times 10^{-2}$) of the nine design schemes for Example 1, $m = 20$.

		IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
	$\hat{\mu}_{11}$	15.01	1.99	0.83	0.83	0.83	0.84	0.83	0.83	0.84
$t = 2$	$\hat{\mu}_{1\bullet}$	5.27	0.36	0.30	0.24	0.15	0.30	0.15	0.28	0.15
$s = 2$	$\hat{\mu}_{\bullet 1}$	5.35	0.64	0.29	0.24	0.29	0.15	0.28	0.15	0.15
	$\hat{\mu}$	7.51	0.25	0.42	0.10	0.21	0.21	0.10	0.10	0.10
	$\hat{\mu}_{11}$	15.11	2.01	0.84	0.84	0.83	0.83	0.82	0.85	0.83
$t = 3$	$\hat{\mu}_{1\bullet}$	3.52	0.23	0.20	0.18	0.10	0.20	0.10	0.19	0.10
$s = 2$	$\hat{\mu}_{\bullet 1}$	4.29	0.61	0.24	0.19	0.24	0.08	0.23	0.08	0.08
	$\hat{\mu}$	6.12	0.22	0.34	0.06	0.17	0.11	0.06	0.06	0.06

Table 6. RMSEs ($\times 10^{-2}$) of the nine design schemes for Example 1, $m = 32$.

		IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
	$\hat{\mu}_{11}$	11.80	0.41	0.41	0.42	0.41	0.41	0.41	0.41	0.41
$t = 2$	$\hat{\mu}_{1\bullet}$	4.16	0.07	0.15	0.12	0.07	0.15	0.07	0.14	0.07
$s = 2$	$\hat{\mu}_{\bullet 1}$	4.17	0.16	0.15	0.12	0.15	0.07	0.14	0.07	0.07
	$\hat{\mu}$	5.88	0.05	0.21	0.05	0.10	0.10	0.05	0.05	0.05
	$\hat{\mu}_{11}$	11.68	0.41	0.41	0.41	0.41	0.41	0.42	0.41	0.41
$t = 3$	$\hat{\mu}_{1\bullet}$	2.75	0.05	0.10	0.09	0.05	0.10	0.05	0.09	0.05
$s = 2$	$\hat{\mu}_{\bullet 1}$	3.38	0.13	0.12	0.09	0.12	0.04	0.11	0.04	0.04
	$\hat{\mu}$	4.82	0.05	0.17	0.03	0.08	0.06	0.03	0.03	0.03

quantities. Except for SS, the following observations can be summarized: IID produces the worst results in all situations, all other schemes achieve a similar level of variance reduction in estimating $\hat{\mu}_{11}$; S-ROW, GS-ROW, and BSLH achieve significantly smaller RMSEs in estimating $\hat{\mu}_{1\bullet}$; S-COL, GS-COL, and BSLH achieve significantly smaller RMSEs in estimating $\hat{\mu}_{\bullet 1}$; SLH, GS-ROW, GS-COL, and BSLH achieve significantly smaller RMSEs in estimating $\hat{\mu}$. The results from SS are mixed and depend on the value m. Although SS significantly outperforms IID in all cases, no advantage is seen in comparison with LHD-based methods when $m = 5, 10, 20$. When $m = 32$, though still outperformed by BSLH in all scenarios, SS shows substantial improvement in its performance. This phenomenon is also seen in many other m values we have tried, not shown here. In general, the performance of SS is competitive only when m is some power of 2, where the point set constitutes a scrambled net. On the other hand, LHD-based designs are quite consistent with all m values, showing their flexibility in terms of the number of design runs.

The above numerical results are consistent with the statistical properties of the different design schemes. Except for IID and SS, all other designs perform similarly at the $\hat{\mu}_{11}$ level as they are all equivalents of LH. In estimating $\hat{\mu}_{1\bullet}$, $\hat{\mu}_{\bullet 1}$, and $\hat{\mu}$, where combinations of $\hat{\mu}_{ij}$ are involved, the performance depends on whether a particular design can achieve a LHD structure for the combination. It can be seen that the proposed design is the most versatile. Regarding the two scenarios, the superior results of BSLH in $t = s = 2$ were shown in the previous section. The BSLH scheme also outperformed all other methods for the unproven case, t is not a multiple of s, when $t = 3$ and $s = 2$.

Example 2. Consider four similar functions from Qian (2012):

$$
\begin{aligned}
f_{11}(\mathbf{x}) &= \log\left(\frac{1}{\sqrt{x_1}} + \frac{1}{\sqrt{x_2}}\right), \\
f_{12}(\mathbf{x}) &= \log\left(\frac{0.98}{\sqrt{x_1}} + \frac{0.95}{\sqrt{x_2}}\right), \\
f_{21}(\mathbf{x}) &= \log\left(\frac{1.02}{\sqrt{x_1}} + \frac{1.02}{\sqrt{x_2}}\right), \\
f_{22}(\mathbf{x}) &= \log\left(\frac{1}{\sqrt{x_1}} + \frac{1.03}{\sqrt{x_2}}\right).
\end{aligned}
$$

Suppose that they can be categorized in two ways: f_{11} and f_{12} belong to one group, f_{21} and f_{22} belong to the other group; f_{11} and f_{21} belong to one group, f_{12} and f_{22} belong to the other group. For example, consider a physical system with a categorical variable of two levels, L_1 and L_2, that might be two operating modes of the system. Assume two computer models based on different algorithms, A and B, are available for simulating this system. Functions f_{11} and f_{12} might be level L_1 implementations of algorithms A and B, respectively; f_{21} and f_{22} might be L_2 implementations of algorithms A and B, respectively. Then the four cases can be grouped either by the level of the categorical variable or by the algorithm they are built on.

Let μ_{11}, μ_{12}, μ_{21}, μ_{22} denote the mean of f_{11}, f_{12}, f_{21}, f_{22}, respectively. We are interested in the mean of each function, as well as $\mu = (\mu_{11} + \mu_{12} + \mu_{21} + \mu_{22})/4$, $\mu_{1\bullet} = (\mu_{11} + \mu_{12})/2$, $\mu_{\bullet 1} = (\mu_{11} + \mu_{21})/2$, $\mu_{2\bullet} = (\mu_{21} + \mu_{22})/2$, and $\mu_{\bullet 2} = (\mu_{12} + \mu_{22})/2$. Here μ is the grand mean of all functions; $\mu_{1\bullet}$ and $\mu_{2\bullet}$ are the means of level L_1 and L_2 implementations, respectively; $\mu_{\bullet 1}$ and $\mu_{\bullet 2}$ are the means of algorithm A and B models, respectively. With the same setup as in the previous example, we compare all the schemes in Definition 1 for estimating these parameters, plus the Sobol sequence based design. As $\mu_{1\bullet}$ and $\mu_{2\bullet}$ / $\mu_{\bullet 1}$ and $\mu_{\bullet 2}$ are similar in evaluating the design schemes, Tables 7−10 $\hat{\mu}_{11}$, $\hat{\mu}_{1\bullet}$, $\hat{\mu}_{\bullet 1}$ and $\hat{\mu}$. The conclusion is similar from that in Example 1. The proposed design works consistently well in estimating all parameters, while other schemes have their respective weaknesses in estimating one or more parameters. The performance's dependence upon the m value is again quite evident in this example for SS. It even slightly outperforms the BSLH in some cases when $m = 32$.

Table 7. RMSEs of the nine design schemes for Example 2, $m = 5$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{11}$	0.193	0.130	0.113	0.113	0.112	0.109	0.112	0.113	0.112
$\hat{\mu}_{1\bullet}$	0.069	0.036	0.040	0.035	0.030	0.040	0.031	0.037	0.030
$\hat{\mu}_{\bullet 1}$	0.069	0.047	0.040	0.035	0.039	0.031	0.037	0.031	0.031
$\hat{\mu}$	0.097	0.039	0.057	0.033	0.043	0.044	0.033	0.034	0.033

Table 8. RMSEs of the nine design schemes for Example 2, $m = 10$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{11}$	0.138	0.072	0.062	0.063	0.062	0.061	0.061	0.061	0.060
$\hat{\mu}_{1\bullet}$	0.048	0.019	0.022	0.020	0.017	0.021	0.017	0.020	0.017
$\hat{\mu}_{\bullet 1}$	0.049	0.027	0.022	0.019	0.022	0.016	0.020	0.017	0.017
$\hat{\mu}$	0.068	0.020	0.031	0.019	0.024	0.023	0.018	0.018	0.018

Table 9. RMSEs ($\times 10^{-2}$) of the nine design schemes for Example 2, $m = 20$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{11}$	9.77	3.82	3.30	3.26	3.31	3.29	3.32	3.35	3.31
$\hat{\mu}_{1\bullet}$	3.43	1.03	1.18	1.03	0.91	1.16	0.91	1.11	0.90
$\hat{\mu}_{\bullet 1}$	3.45	1.24	1.15	1.04	1.18	0.92	1.11	0.92	0.93
$\hat{\mu}$	4.80	1.08	1.65	1.04	1.31	1.28	1.02	1.03	1.03

Table 10. RMSEs ($\times 10^{-2}$) of the nine design schemes for Example 2, $m = 32$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{11}$	7.69	2.04	2.21	2.21	2.21	2.25	2.23	2.21	2.22
$\hat{\mu}_{1\bullet}$	2.74	0.53	0.79	0.70	0.62	0.79	0.62	0.73	0.62
$\hat{\mu}_{\bullet 1}$	2.72	0.78	0.78	0.71	0.78	0.62	0.74	0.61	0.62
$\hat{\mu}$	3.82	0.54	1.10	0.72	0.87	0.88	0.71	0.71	0.72

So far, we have used equal weights λ_{ij}'s for μ_{ij}'s in all estimations. It can be desirable to use different weights. For example, in multi-model ensemble analysis for climate prediction, Krishnamurti et al. (1999) proposed a superensemble approach where each constituent model is weighted by its performance to obtain a weighted ensemble average that shows major improvement in overall performance. With the same experiment setup, results in Tables 11−14 illustrate their performances using $\lambda_{11} = 0.4$, $\lambda_{12} = 0.3$, $\lambda_{21} = 0.2$, and $\lambda_{22} = 0.1$. As can be seen, similar conclusions as those from the equal weights scenario can be drawn.

5. Conclusion

We have proposed a new type of design called BSLHDs for computer experiments. It extends the idea of the standard SLHDs and features a special bi-directional slicing structure. The proposed construction procedure of BSLHDs is

Table 11. RMSEs of the nine design schemes for Example 2, unequal weights, $m = 5$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{1\bullet}$	0.193	0.130	0.113	0.113	0.112	0.109	0.112	0.113	0.112
$\hat{\mu}_{\bullet 1}$	0.097	0.050	0.056	0.050	0.044	0.056	0.044	0.053	0.044
$\hat{\mu}$	0.087	0.059	0.051	0.046	0.050	0.040	0.047	0.042	0.042

Table 12. RMSEs of the nine design schemes for Example 2, unequal weights, $m = 10$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{1\bullet}$	0.138	0.072	0.062	0.063	0.062	0.061	0.061	0.061	0.060
$\hat{\mu}_{\bullet 1}$	0.069	0.027	0.031	0.028	0.024	0.030	0.024	0.028	0.024
$\hat{\mu}$	0.062	0.033	0.028	0.026	0.027	0.022	0.026	0.022	0.022

Table 13. RMSEs ($\times 10^{-2}$) of the nine design schemes for Example 2, unequal weights, $m = 20$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{1\bullet}$	9.77	3.82	3.30	3.26	3.31	3.29	3.32	3.35	3.31
$\hat{\mu}_{\bullet 1}$	4.85	1.46	1.66	1.46	1.30	1.64	1.31	1.58	1.30
$\hat{\mu}$	4.36	1.59	1.46	1.34	1.49	1.23	1.42	1.24	1.23

Table 14. RMSEs ($\times 10^{-2}$) of the nine design schemes for Example 2, unequal weights, $m = 32$.

	IID	SS	LH	SLH	S-ROW	S-COL	GS-ROW	GS-COL	BSLH
$\hat{\mu}_{1\bullet}$	7.69	2.04	2.21	2.21	2.21	2.25	2.23	2.21	2.22
$\hat{\mu}_{\bullet 1}$	3.87	0.77	1.12	1.00	0.89	1.12	0.89	1.04	0.89
$\hat{\mu}$	3.44	0.98	0.99	0.91	0.99	0.84	0.96	0.82	0.83

general. When t is a multiple of s, we have proved the equivalence between some substructures of a BSLHD with ordinary LHDs and SLHDs, and hence their statistical properties. By comparing with some existing design schemes, desirable statistical properties of the proposed design have been further demonstrated through the numerical examples. In cases when t is not a multiple of s, although the design's statistical property has not been proved, numerical examples have shown a similar performance on variance reduction as in the $t = cs$ case.

The special structure of BSLHD makes it suitable for applications include, but not limited to batch-running computer experiments, design and cross- validation for computer models with qualitative factor(s), and collective evaluation of multiple computer models (multi-model ensemble analysis). The experimental design for computer experiments involving discrete ingredients is still at its early stage of development. However, we are starting to see growing needs for this type of design due to the increasingly complex settings of computer experiments for practical needs. We believe this trend will continue.

The proposed BSLHD has a similar genesis as the GSLHD developed by Xie et al. (2014), but with different properties. The GSLHD is able to provide multiple layers for the slicing structure, including the ordinary LHD (zero layer) and SLHD (one layer) as its special cases. Following their definition, the BSLHD has a two layer slicing structure (a two layer GSLHD is called *doubly* SLHD in their paper). However, in obtaining the next layer slice $\mathbf{D}_{ij}$'s from slice $\mathbf{D}_{i\bullet}$'s, their algorithm is independently implemented across i, while our algorithm correlates the $\mathbf{D}_{ij}$'s both within and across i. This achieves the crucial property that $\mathbf{D}_{\bullet j} = \bigcup_{i=1}^{t} \mathbf{D}_{ij}$ is also a SLHD with some $\mathbf{D}_{ij}$'s as its next layer slices. $\mathbf{D}_{i\bullet}$'s and $\mathbf{D}_{\bullet j}$'s play interchangeable roles in a BSLHD, as illustrated in Section 3 as well as the numerical examples. While in their doubly SLHD, $\mathbf{D}_{\bullet j} = \bigcup_{i=1}^{t} \mathbf{D}_{ij}$ has no such desirable property.

A possible future work is to extend the BSLHD structure to more than two layers, in a similar spirit as with GSLHD. However, our initial attempts have shown that the extension to more layers is not trivial, certain restrictions on the design parameters may need to be imposed.

Supplementary Materials

This paper has a supplementary document that covers the proofs of Lemma 1−4 and Theorem 1−2.

Acknowledgements

The authors thank the Editor, an associate editor, and two anonymous referees for their helpful comments that led to improvement in this article. Qiang Zhou's work is supported by the National Natural Science Foundation of China under Grant No. 11301441. Tian Jin's work is supported by the Graduate Innovation Project (CXJJ-2011-443) and Outstanding PhD Fund from Shanghai University of Finance and Economics. Peter Qian is supported by ARO Grant W911NF-15-1-0156.

Appendix: Detailed Procedures for Step 1

The NPT $\mathbf{T}$ contains exactly s elements in each row (not subrow) and each column. Define a $t \times t$ matrix $\mathbf{G}$ such that the value of its ith row and jth column cell, g_{ij}, equals the number of elements in the ith row (not subrow) and jth column cell of $\mathbf{T}$ (equals zero if the corresponding cell in $\mathbf{T}$ has no element).

For example, for the table illustrated in Table 2, its corresponding $\mathbf{G}$ matrix is

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Such a $\mathbf{G}$ matrix has row and column sums all equal to s. It is equivalent, by a multiplicative factor s, to a rational doubly stochastic matrix with common denominator s. A doubly stochastic matrix is a square matrix with nonnegative real numbers whose row and column sums all equal to 1. By the Birkhoff–von Neumann theorem (Birkhoff (1946); von Neumann (1953)), any doubly stochastic matrix can be represented as a convex combination of permutation matrices. A direct corollary of this theorem guarantees that any such $\mathbf{G}$ matrix can be represented as a sum of s permutation matrices (Asratian, Denley, and Häggkvist (1998)). A permutation matrix is a binary square matrix with exactly one entry 1 in each row and each column. For the matrix $\mathbf{G}$ in the example, we have a representation

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \\ &= \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Once such a representation is found, a group of t numbers, $\mathbf{q}_k$, from $\mathbf{G}$'s corresponding $\mathbf{T}$ table can be extracted based on each $\mathbf{P}_k$, $k = 1, \ldots, s$. Specifically, for each $\mathbf{P}_k$, if its ith row and jth column element is '1', $i, j = 1, \ldots, t$, a number in the ith row (not subrow) and jth column cell of the $\mathbf{T}$ table is *removed* and assigned to the jth position of $\mathbf{q}_k$. If there is more than one number in the chosen cell, randomly pick one. After this has been completed for all $\mathbf{P}_k$'s, each number in $\mathbf{T}$ has been chosen once and only once. We obtain $\mathbf{q}_1, \ldots, \mathbf{q}_s$, each of which is a $t \times 1$ vector. Then the $s \times t$ matrix $\mathbf{Q}$ is obtained by letting $\mathbf{Q}(k, :) = \mathbf{q}_k^T$, $k = 1, \ldots, s$.

Taking $\mathbf{T}$ and $\mathbf{G}$ in the previous example, corresponding to $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$, we might obtain

$$\mathbf{q}_1^T = (5, 8, 1, 11), \quad \mathbf{q}_2^T = (2, 9, 10, 6), \quad \mathbf{q}_3^T = (12, 3, 7, 4),$$

$$\mathbf{Q} = \begin{bmatrix} 5 & 8 & 1 & 11 \\ 2 & 9 & 10 & 6 \\ 12 & 3 & 7 & 4 \end{bmatrix}.$$

It can be easily verified that for any $j = 1, \ldots, 3$, $\lceil \mathbf{Q}(j, :)/3 \rceil$ is a permutation on $\mathbf{Z}_4$, and for any $i = 1, \ldots, 4$, $\lceil \mathbf{Q}(:, i)/4 \rceil$ is a permutation on $\mathbf{Z}_3$.

While the existence of such representation of a $\mathbf{G}$ matrix is guaranteed by the Birkhoff–von Neumann theorem, it does not give a recipe for the decomposition. First of all, we point out that any constructive proof of the Birkhoff–von Neumann theorem may be used as an algorithm to decompose a $\mathbf{G}$ matrix (e.g., Bapat and Ragnavan (1997)). Nonetheless, here we introduce an algorithm that is easy to implement:

Step (a) Generate all $t!$ possible permutations on $\mathbf{Z}_t$, denote the ith permutation as $\pi_i = \{\pi_i(1), \pi_i(2), \ldots, \pi_i(t)\}$, $i = 1, \ldots, t!$.

Step (b) For each π_i, calculate $c_i = \prod_{j=1}^{t} \mathbf{G}_{j,\pi_i(j)}$, where $\mathbf{G}_{j,\pi_i(j)}$ is the jth row and $\pi_i(j)$th column element in $\mathbf{G}$; identify all π_i's with its corresponding c_i being nonzero;

Step (c) Randomly choose a π_i from those identified in Step (b) with nonzero c_i, define a permutation matrix $\mathbf{P}$, whose jth row and $\pi_i(j)$th column is 1.

Let $\mathbf{P}_1 = \mathbf{P}$, and $\mathbf{G}_{(1)} = \mathbf{G} - \mathbf{P}_1$. Since $\mathbf{G}_{(1)}$ still has equal row and column sums, similar procedures as in Steps (a)$-$(c) can be applied to $\mathbf{G}_{(1)}$ as well to obtain $\mathbf{P}_2$. Repeat this approach $s-1$ times and we obtain $\mathbf{G} = \mathbf{P}_1 + \cdots + \mathbf{P}_s$, where $\mathbf{P}_s = \mathbf{G}_{(s-1)}$.

References

Asratian, A. S., Denley, T. M. J. and Häggkvist, R. (1998). *Bipartite Graphs and Their Applications.* Cambridge University Press, Cambridge.

Bapat, R. B. and Ragnavan, T. E. S. (1997). *Nonnegative Matrices and Applications.* Cambridge University Press, Cambridge.

Birkhoff, G. (1946). Three observations on linear algebra. *Univ. Nac. Tucumán. Rev. Ser. A.* **5**, 147-151.

Fang, K. T., Li, R. and Sudjianto, A. (2005). *Design and Modeling for Computer Experiments.* Chapman & Hall/CRC Press, New York.

Krishnamurti, T. N., Kishtawal, C. M., LaRow, T., Bachiochi, D., Zhang, Z., Williford, E. C., Gadgil, S. and Surendran, S. (1999). Improved weather and seasonal climate forecasts from multimodel superensemble. *Science* **285**, 1548-1550.

Loh, W. L. (1996). On Latin hypercube sampling. *Ann. Statist.* **24**, 2058-2080.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239-45.

Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration and visualization. *Statist. Sinica* **2**, 439-452.

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

Santner, T. J., Williams, B. J. and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Sobol, I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics* **7**, 86-112.

Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics* **29**, 143-151.

von Neumann, J. (1953). A certain zero-sum two-person game equivalent to an optimal assignment problem. *Ann. Math. Studies* **28**, 5-12.

Xie, H., Xiong, S., Qian, P. Z. G. and Wu, C. F. J. (2014). General sliced Latin hypercube designs. *Statist. Sinica* **24**, 1239-1256.

Department of Systems Engineering and Engineering Management, City University of Hong Kong.

E-mail: q.zhou@cityu.edu.hk

School of Statistics and Management, Shanghai University of Finance and Economics.

E-mail: jintian82@gmail.com

Department of Statistics, University of Wisconsin-Madison.

E-mail: peterq@stat.wisc.edu

Department of Statistics, University of Wisconsin-Madison.

E-mail: szhou@engr.wisc.edu

(Received July 2014; accepted June 2015)