International Journal of Statistics and Applied Mathematics 2024; 9(2): 27-31

International Journal of Statistics and Applied Mathematics

ISSN: 2456-1452
Maths 2024; 9(2): 27-31

www.mathsjournal.com
Received: 17-12-2023
Accepted: 28-01-2024

A Anil Kumar
ICAR-Indian Agricultural Statistics Research Institute, New Delhi, India

Sukanta Dash
ICAR-Indian Agricultural Statistics Research Institute, New Delhi, India

Baidya Nath Mandal
ICAR-Indian Agricultural Statistics Research Institute, New Delhi, India

Corresponding Author:
A Anil Kumar
ICAR-Indian Agricultural Statistics Research Institute, New Delhi, India

On construction of sliced Latin hypercube designs

A Anil Kumar, Sukanta Dash and Baidya Nath Mandal

Abstract
Latin hypercube designs (LHDs) are often used in designing complex computer models. A new special type of Latin hypercube design known as Sliced LHD (SLHD) are now a days gaining much importance in the field of computer experiments. SLHDs are a special type of LHDs which can be further partitioned into different slices and act as batches of smaller Latin hypercube designs. In this article we have proposed methods of construction for Sliced Latin hypercube designs of equal as well as unequal batch size. The whole design and each slice of SLHD can have maximum coverage of design points in one-dimension region.

Keywords: Computer experiments, Latin hypercube designs, Sliced Latin hypercube designs.

1. Introduction

Often while conducting physical experiments, there may be situations arises where, it is very difficult to conduct experiments due to time and cost effect or even sometime looks impossible thus, on those situations computer experiments are suggested. Space-filling designs are a class of designs which will serve this purpose. Furthermore, these designs are providing better prediction accuracy whenever the primary goal of investigation is making prediction at un-sampled points. Latin Hypercube designs are most commonly used space-filling designs introduced by McKay *et al*. (1979) [3]. These designs are said to have the one-dimensional space-filling property i.e. when the design points are projected onto each dimension (McKay, Beckman and Conover, 1979) [3].

Consider an experimental design with p points in d dimensions denoted as a $p \times d$ matrix, where each column $X = [X_1 X_2 \cdots X_P]^T$ represents a variable, and each row $X_i = \left[X_i^{(1)} X_i^{(2)} \cdots X_i^{(d)}\right]$, $i = 1,2,\dots,p$ represents a sample. A LHD can be constructed in such a way that each of the d dimensions is divided into p equal levels and that there is only one point (or sample) at each level. A Latin hypercube design with n runs and q factors denoted as LHD (n,q), is a $n \times q$ matrix whose columns are permutations of the column vector $(1,2,\dots,n)'$. Here, the columns of LHD represent the factors and the rows represent the runs.

Latin hypercube designs serve as popular designs for conducting any computer experiments with single computer models but sometimes it becomes necessary to run multiple computer models for the interest of collective evaluation of related computer model (Williams, Morris, and Santner 2009). Consider the following example of designing a controlled atmospheric storage system for horticultural/agricultural products to better understand the problem.

Example: Consider designing a controlled atmospheric storage system for horticultural/agricultural products, where we have to store fruits of type climacteric and non-climacteric, vegetables of type leafy and bulb and food grains of the type cereals and pulses. Each of the variants definitely have different optimum storage conditions for their short-term/long-term storage. Now, designing of storage system for each of the variants of fruits, vegetables and food grains separately, involves building a separate computer models say $\mathrm{A}_1, A_2, B_1, B_2 and\ C_1, C_2$ and estimating expected output of each of the computer model by using different LHD's. Suppose our aim is to estimate expected output of each of the 6 models, a linear combination of expected output of the variants of fruits, vegetables and food grains, and also of all of the 6 computer models. Here, the multiple models can be run in multiple slices for the estimation of the expected outputs of each of them, and a linear combination of the expected outputs of the multiple models. Now, instead of using different LHD's for different model outputs or for linear combination of the model outputs, it is desirable to use a new type of design to run computer model in batches, with each batch of input values being one slice of the design. On the one hand, Latin hypercube design, along with each slices when data from different batches (slices) must be analysed separately, then design in each slice must have the properties of Latin hypercube design. Experimental situations as mentioned above poses difficulty of searching for number of LHD's for each model and linear combinations of each variant.

Qian (2012) [5] proposed a new type of LHD called Sliced Latin hypercube design (SLHD) to tackle experimental situations as given above. A SLHD is a special type of LHD that can be partitioned into slices of smaller Latin hypercube designs. Such designs have two important features, one, each slice (batch) of the design achieves maximum uniformity and coverage of design points; second, when collapsed over all slices (batches), the overall design possesses maximum stratification in any one-dimensional projection (Wang *et al*. 2017). A n run SLHD is a LHD which can be partitioned into t different slices where, each of which is also an LHD of n/t runs, after collapsing the n levels to n/t levels and then each slice can be used under one of the t different level combinations. For integers m and t, an SLHD with $n = mt$ runs, q factors and t slices, denoted by SLHD(m, t, q). Consider the following example to better understand SLHD's.

Consider $m = 3$, $t = 3$, and $q = 4$, with $n = 12$ develop a 9×4 Latin hypercube design or an SLHD (3, 4, 3) with three slices is given below.

$$D = \begin{pmatrix} X_1\ X_2\ X_3\ X_4 \\ \begin{pmatrix} 1 & 5 & 3 & 7 \\ 7 & 2 & 6 & 1 \\ 4 & 8 & 9 & 4 \end{pmatrix} \\ \begin{pmatrix} 8 & 3 & 1 & 6 \\ 2 & 9 & 4 & 9 \\ 5 & 6 & 7 & 3 \end{pmatrix} \\ \begin{pmatrix} 3 & 4 & 8 & 2 \\ 9 & 1 & 5 & 8 \\ 6 & 7 & 2 & 5 \end{pmatrix} \end{pmatrix}$$

The above given design is an SLHD (3, 3, 4), which is a special type of LHD (9, 4) and the whole design can be partitioned into 3 slices or batches of run size 3. For more details, one may refer to Lin *et al*. (2010) [2], Xiong *et al*. (2014) [6], Shan *et al*.(2015) [4], Xu *et al*. (2019) [7], Dash *et al*. (2019) [1] and Zhang *et al*. (2019) [8-9] etc.

The main purpose of this paper is to develop suitable methods of construction of designs to carryout computer experiments with multiple complex computer models to estimate expected output of each of the computer model as well as a linear combination of the expected output of the multiple computer models. Methods have been developed for construction of SLHD's of equal batch size and unequal batch size. The methods of construction developed in general in nature in reference to number of runs, number of factors and number of slices.

2. Preliminaries

An LHD with *n* runs and *m* factors is denoted as $L(n, m) = (l_1, \dots, l_m)$, where $l_j, j = 1,2, \dots, m$is the j^{th} factor, and each factor includes *n* uniformly spaced levels {1, 2, …, *n*}. When *l* is projected onto any one dimension, precisely one point falls within one of the n equally spaced
intervals of (0, 1] given by (0, 1/*n*], (1/*n*, 2/*n*], . . . , ((*n* −1)/*n*, 1].

A n run SLHD is a new type of LHD that can be partitioned into t different slices where each of which is also having the property of LHD with n/t runs, after collapsing the n levels to n/t levels. For integers m and t, an SLHD with $n = mt$ runs, q factors and t slices, denoted by SLHD(m, t, q). When number of runs are consider different in different slice then total number of runs become $n = \sum_{i=1}^{t} m_i$and can be denoted as SLHD(m_i, t, q).

3. Methodology

Sliced LHD are very much useful for different type of computer experiments especially in agriculture. Hence, by looking into the importance of these designs, there is a need to develop methods of construction to obtain these designs. Here, we propose method of construction SLHD of both equal and unequal batch/slice size in section 3.1 to 3.2 respectively.

3.1. SLHD of equal batch size

In this section, we propose a construction method to obtain SLHDs with equal runs in each slice. This method is very simple to implement and general in nature with respect to number of runs, factors and slices.

Theorem 1 *Let D be an LHD*(n, q) *such that* $n = m * t$. *Where t is number of slice and m is number of run in each slice. Then there always a SLHD (m,t,q) exist by taking the entries of slice as* $D_i = (a_{jk})$, where a_{jk} is j^{th} row and k^{th} column element of D_i. $i = 1,2, \dots, t$, $j = 1,2, \dots, m$ *and* $k = 1,2, \dots, q$. Now, $\forall\ a_{jk} \epsilon D_i$, let, $a_{jk}/t = \lceil a_{jk} \rceil$. Where $\lceil a_{jk} \rceil$ is a greatest integer function no less than a_{jk} then, $D_i' = (\lceil a_{jk} \rceil)$ is an LHD (*m*, *q*).

Consider the following design

$$D = \left[D_1{'}, D_2{'}, \cdots, D_t{'}\right]'$$

Where

$$D_1 = \begin{bmatrix} 1 & 2 & 3 & \cdots & q \\ 1+t & 2+t & 3+t & \cdots & q+t \\ 1+2t & 2+2t & 3+2t & \cdots & q+2t \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1+(m-1)t & 2+(m-1)t & 3+(m-1)t & \cdots & q+(m-1)t \end{bmatrix}$$

Here, D_2 to D_t are the developed from the design D_1 such that each column of D_1 come exactly once in each column of Design D. i.e. design D_2 content any of the q-1 column of D_1 except the first column as it is in first column. After obtaining the design D, entries within a column of each slice can be permuted. If any entry of the design found to be more than *n* then value will be taking of mod *n*. This permutation of entries with in a column in a slice has been practiced to get a good coverage of design points in the region. Following is an example to illustrate the proposed method of construction of SLHD of equal batch size.

Example2: Suppose we are interested in SLHD (3,4,6), i.e. *m*=3, *t*=4 and *q*=6.

Here, $D = (D_1', D_2', D_3', D_4')'$

$$D_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 6 & 7 & 8 & 9 & 10 \\ 9 & 10 & 11 & 12 & 1 & 2 \end{bmatrix}$$

Note: The 3rd row of 4th and 5th column coming out 1 and 2 by taking mod *n* (=12) of 13 and 14.
Similarly D_2, D_3 and D_4 can easily be constructed and the final design after permutation of entries within a column of a particular slice will be

		X_1	X_2	X_3	X_4	X_5	X_5
D =	t_1	1	6	7	4	5	10
		5	10	3	8	1	6
		9	2	11	12	9	2
	t_2	10	7	8	5	2	11
		6	11	4	1	6	7
		2	3	12	9	10	3
	t_3	3	8	9	2	7	12
		7	12	5	6	11	8
		11	4	1	10	3	4
	t_4	4	5	6	11	8	1
		8	1	2	3	12	9

Here, Design D_1, D_2, D_3, and D_4 are considered as four different slice/batch denoted as t_1, t_2, t_3 and t_4. Now $\forall\, a_{jk} \epsilon\, t_i$, let, $a_{jk}/t = \lceil a_{jk} \rceil$. Where $\lceil a_{jk} \rceil$ is a greatest integer function no less than a_{jk} then, $t_i' = (\lceil a_{jk} \rceil)$ is an LHD (m, q). Here, the design is partitioned into 4 slices say t_1, t_2, t_3 and t_4 and each of the 4 slices is again an LHD (3,6), after collapsing the n *(12)* levels to n/t *((12/4) =3)* levels. Let each of the slice be denoted as $t_i = (a_{jk})$, where a_{jk} is j^{th} row and k^{th} column element of t_i. $i = 1,2, \dots, t$, $j = 1,2, \dots, m$ and $k = 1,2, \dots, q$. Now, $\forall\, a_{jk} \epsilon\, t_i$, let, $a_{jk}/t = \lceil a_{jk} \rceil$. Where $\lceil a_{jk} \rceil$ is greatest integer no less than a_{jk} then, $t_i' = (\lceil a_{jk} \rceil)$ is an LHD (m, q). Therefore, in the above example of SLHD (3,4,6), slice t_1 is an *LHD* (3,6). Now one can see that, $t_1 = \begin{bmatrix} 1\ 6\ 7\ 4\ 5\ 10 \\ 5\ 10\ 3\ 8\ 1\ 6 \\ 9\ 2\ 11\ 12\ 9\ 2 \end{bmatrix}$ dividing each element of t_1 by t, $\{t_1 = (a_{jk}/t)\}$ gives $\begin{pmatrix} \lceil 0.25\rceil\lceil 1.5\rceil\lceil 1.75\rceil\ \lceil 1\rceil\lceil 1.25\rceil\lceil 2.5\rceil \\ \lceil 1.25\rceil\lceil 2.5\rceil\lceil 0.75\rceil\lceil 2\rceil\lceil 0.25\rceil\lceil 1.5\rceil \\ \lceil 2.25\rceil\lceil 0.5\rceil\lceil 2.75\rceil\lceil 3\rceil\lceil 2.25\rceil\lceil 0.5\rceil \end{pmatrix}$

and the resulting $t_i'\{t_i' = (\lceil a_{jk} \rceil)\}$ will be, $t_1' = \begin{pmatrix} 1\ 2\ 2\ 1\ 2\ 3 \\ 2\ 3\ 1\ 2\ 1\ 2 \\ 3\ 1\ 3\ 3\ 3\ 1 \end{pmatrix}$ which is an $LHD\ (3,6)$. Similarly, D_2 , D_3 and D_4will be an *LHD* (3,6).

3.2. SLHD of unequal batch size

Sliced LHD with equal batch size are often use in computer model. However, some experimental situation, run size of each slices may be different. Hence, in this section we have proposed a new method of construction for obtaining SLHD with unequal batch size. The method is developed for number of slice (t) is two and three.

Theorem 2 *Let* L_1^* *be an* $LHD(m_1, q)$ *and* L_2^* *be an* $LHD(m_2, q)$ *such that* $m_2 = m_1 + 1$. *Where t is number of slice and* m_1 *and* m_2 *are number of run in first and second slice. Then there always a SLHD* $((m_1, m_2), t, q)$ *exist by taking the entries of slice as* $D_i = (a_{jk})$, where a_{jk} is j^{th} row and k^{th} column element of t_i. $i = 1,2$, $j = 1,2, \dots, mi$ and $k = 1,2, \dots, q$. Now, $\forall\, a_{jk} \epsilon\, D_i$, let, $a_{jk}/t = \lceil a_{jk} \rceil$. Where $\lceil a_{jk} \rceil$ is a greatest integer function no less than a_{jk} then, $D_i' = (\lceil a_{jk} \rceil)$ is an LHD (m_i, *q*).
Consider the following design

$D = (D_1{}', D_2{}')'$

Where the entries of first column of D_1 is $(t, 2t, \cdots, m_1 t)$and the entries of the first column of D_2 is $(1, 1+t, 1+2t, \cdots, 1+m_1 t)$. Rest of the q-1 columns of D_1 and D_2 are the permutation of entries of first column of design D_1 and D_2 respectively. Then design D will be always a SL HD *((m₁,m₂),t,q)*

Example3: Suppose we are interested in SLHD ((2,3),2,2), i.e. m_1=2 m_2=3, t=2 and q=2.

$D = (D_1{}', D_2{}')'$

First column of D_1= (2,4) and D_2 = (1,3,5)

Hence design D=$\begin{bmatrix} 2 & 4 & 2 \\ 4 & 2 & 4 \\ \cdots & \cdots & \\ 1 & 3 & 5 \\ 3 & 1 & 5 \\ 5 & 3 & 1 \end{bmatrix}$ is clearly a SLHD ((2,3),2,3)

Consider each slice of the designs D_i denoted as t_i. Here, the design is partitioned into 2 slices say t_1, and t_2 and each of the 2 slices is again an LHD (2,3) and LHD (3,3), after collapsing the n *(5)* levels to one at 2 level and other at 3 level. Let each of the slice be denoted as $t_i = (a_{jk})$, where a_{jk} is j^{th} row and k^{th} column element of t_i. $i = 1,2$, $j_1 = 1,2, \dots, m_1, j_2 = 1,2 \ \dots, m_2 \ and \ k = 1,2, \dots, q$. Now, $\forall\ a_{jk} \in b_i$, let, $a_{jk}/t = \lceil a_{jk} \rceil$. Where $\lceil a_{jk} \rceil$ is greatest integer no less than a_{jk} then, $t_i' = (\lceil a_{jk} \rceil)$ is an LHD (m_1, q) and LHD(m_2,q) Therefore, in the above example of SLHD ((2,3),2,3), slice t_2 is an LHD (3,3).

$t_2 = \begin{bmatrix} 1 & 5 & 3 \\ 3 & 1 & 5 \\ 5 & 3 & 1 \end{bmatrix}$ dividing each element of t_2 by t, $\{t_2 = (a_{jk}/t)\}$ gives $\begin{pmatrix} \lceil 0.5 \rceil \lceil 2.5 \rceil \lceil 1.5 \rceil \\ \lceil 1.5 \rceil 0.5 \lceil 2.5 \rceil \\ \lceil 2.5 \rceil \lceil 1.5 \rceil \lceil 0.5 \rceil \end{pmatrix}$ and the resulting design will be, $\begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}$ which is an $LHD\ (3,3)$. Similarly, t_1 will be an LHD (2,3).

Theorem 3 *Let L_1^* be an $LHD(m_1, q)$, L_2^* be an $LHD(m_2, q)$ and L_3^* be an $LHD(m_3, q)$ such that m2=m1+1 and m3=2m1. Where t is number of slice and m₁, m₂ and m₃ are number of run in first, second and third slice respectively. Then there always a SLHD ((m₁,m₂,m₃),t,q) exist by taking the entries of slice as $t_i = (a_{jk})$* , where a_{jk} is j^{th} row and k^{th} column element of t_i. $i = 1,2,3$, $j = 1,2, \dots, m_i \ and \ k = 1,2, \dots, q$. Now, $\forall\ a_{jk} \in D_i$, let, $a_{jk}/t = \lceil a_{jk} \rceil$. Where $\lceil a_{jk} \rceil$ is a greatest integer function no less than a_{jk} then, $D_i' = (\lceil a_{jk} \rceil)$ is an LHD (m_i, q).
Consider the following design

$D = [D_1{}', D_2{}', D_3{}']'$

Where the entries of first column of D_1 is $(t, 2(t+1)-1, \cdots, m_1(t+1)-1)$ and the entries of the first column of D_2 is $(1, 2(t+1)-3, 3(t+1)-3, \cdots, (m_1+1)(t+1)-3)$ and entries of the first column of D_3 is $(2,4,6, \cdots, 4m_1)$. Rest of the *q*-1 columns of D_1, D_2 and D_3 are the permutation of entries of first column of design D_1, D_2 and D_3 respectively. Then the design D is always a SLHD *((m₁,m₂,m₃),t,q).*

Example 4: Suppose we are interested in SLHD ((4,5,8),3,3), i.e. m_1=4 m_2=5, m_3=8, t=3 and q=3.

$D = [D_1{}', D_2{}', D_3{}']'$

Here, first column of D_1 is (3,7, 11, 15)
First column of D_2 is (1,5,9,13,17)
First column of D_3 is (2,4,6,8,10,12,14,16)
Rest of the column of each slice will be the permutation of entries of first column of D_1, D_2, and D_3.
Hence the final design is

$$D = \begin{bmatrix} 3\ 11\ 7 \\ 7\ 15\ 3 \\ 11\ 7\ 15 \\ 15\ 3\ 11 \\ \dots\ \dots \\ 1\ 9\ 17 \\ 5\ 17\ 1 \\ 9\ 13\ 5 \\ 13\ 1\ 9 \\ 17\ 5\ 13 \\ \dots\ \dots \\ 2\ 14\ 12 \\ 4\ 12\ 10 \\ 6\ 10\ 14 \\ 8\ 16\ 6 \\ 10\ 8\ 4 \\ 12\ 2\ 6 \\ 14\ 4\ 2 \\ 16\ 6\ 16 \end{bmatrix}$$ is clearly a SLHD ((4,5,8),3,3)

Consider each slice of the designs D_i denoted as t_i. Here, the design is partitioned into 3 slices say t_1, t_2 and t_3 hence, each of the 3 slices is again an LHD (4,3) LHD (5,3) and LHD (8,3) after collapsing the n *(17)* levels to one at 4 level and other two are at 5 and 8 level respectively. Let each of the slice be denoted as $t_i = \left(a_{jk}\right)$, where a_{jk} is j^{th} row and k^{th} column element of t_i. $i = 1,2,3$, $j_1 = 1,2,\dots,m_1, j_2 = 1,2\ \dots,m_2, j_3 = 1,2\ \dots,m_3\ and\ k = 1,2,\dots,q$. Now, $\forall\ a_{jk}\ \epsilon\ t_i$, let, $a_{j_1k}/(t+1) = \lceil a_{j_1k} \rceil$, $a_{j_2k}/(t+1) = \lceil a_{j_2k} \rceil$ and $a_{j^3k}/(t-1) = \lceil a_{j_3k} \rceil$ Where $\left\lceil a_{j_ik} \right\rceil$ is greatest integer no less than a_{j_ik} then, $t_i' = \left(\lceil a_{jk} \rceil\right)$ is an LHD (m_1, q), LHD(m_2,q) and LHD(m_3,q). Therefore, in the above example of SLHD ((4,5,8),3,3), slice t_2 is an LHD (5,3). One can see that,

$t_2 = \begin{bmatrix} 1\ 9\ 17 \\ 5\ 17\ 1 \\ 9\ 13\ 5 \\ 13\ 1\ 9 \\ 17\ 5\ 13 \end{bmatrix}$ dividing each element of t_2 by t+1 $\{t_2 = \left(a_{jk}/(t+1)\right)\}$ gives $\begin{pmatrix} \lceil 0.25 \rceil \lceil 2.25 \rceil \lceil 4.25 \rceil \\ \lceil 1.25 \rceil 4.25 \lceil 0.25 \rceil \\ \lceil 2.25 \rceil \lceil 3.25 \rceil \lceil 1.25 \rceil \\ \lceil 3.25 \rceil \lceil 0.25 \rceil \lceil 2.25 \rceil \\ \lceil 4.25 \rceil \lceil 1.25 \rceil \lceil 3.25 \rceil \end{pmatrix}$ and the resulting design will be, $\begin{pmatrix} 1\ 3\ 5 \\ 2\ 5\ 1 \\ 3\ 4\ 2 \\ 4\ 1\ 3 \\ 5\ 2\ 4 \end{pmatrix}$ which is an LHD $(5, 3)$. Similarly, t_1 and t_3 will be an LHD (4,3) and LHD (8,3) respectively.

4. Discussion

SLHD's are very much useful for different type of computer experiments especially in agriculture. Hence, by looking into the importance of these designs, there is a need to develop methods of construction to obtain these designs. In this article we have proposed three general methods of constructing Sliced Latin hypercube designs for equal batch size and unequal batch size. The proposed method for constructing equal batch size Sliced Latin hypercube designs is general in nature for any number of slices and are capable of accommodating flexible number of runs and factors in each slice. In case of unequal batch size SLHD's, methods are developed for up to three slices. The permutation of numbers in a slice are done in such a way that the design obtained are having good space filling property.

5. References

1. Dash S, Mandal BN, Parsad R. On the construction of nested orthogonal Latin hypercube designs. Metrika - International Journal for Theoretical and Applied Statistics. 2019;83:347-353.
2. Lin CD, Bingham D, Sitter RR, Tang B. A new and flexible method for constructing designs for computer experiments. The Annals of Statistics. 2010;38(3):1460–1477.
3. McKay MD, Beckman RJ, Conover WJ. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. 1979;21(2):239–245.
4. Shan B, William RM, William AB. Optimal sliced Latin hypercube designs. Technometrics. 2015;57(4):479-487.
5. Qian P. Sliced Latin hypercube designs. Journal of the American Statistical Association. 2012;107:393–399.
6. Xie H, Xiong S, Qian PZ, Wu CJ. General sliced Latin hypercube designs. Statistica Sinica. 2014, 1239-1256.
7. Xu J, He X, Duan X, Wang Z. Sliced Latin hypercube designs with arbitrary run sizes. arXiv preprint arXiv:1905.02721. 2019.
8. Zhang J, Xu J, Wang Z. A flexible construction for sliced Latin hypercube designs. Journal of Physics: Conference Series. 2019;1168(2).
9. Zhang J, Xu J, Jia K, Yin Y, Wang Z. Optimal Sliced Latin Hypercube Designs with Slices of Arbitrary Run Sizes. Mathematics. 2019;7(9):854