CONSTRUCTION OF LATIN HYPERCUBE DESIGNS WITH TWO FACTORS

Shyamsundar Parui, Rajender Parsad[1]* and B.N. Mandal
ICAR-IASRI, Library Avenue, Pusa, New Delhi-110012

ABSTRACT

Latin hypercube designs (LHD) are widely used as space-filling designs in the field of computer experiments. Most of the available methods of construction of good space-filling LHDs in literature are based on computer algorithms. In this article, we propose a general construction method of LHDs with two factors which possess good space-filling property for small number of runs.

Keywords: Latin hypercube designs, Computer experiments, Space filling

1. Introduction

Now a days, computer experiments are becoming increasingly surrogates for many physical experiments (Santner *et al.* 2003; Fang *et al*. 2006). In many scientific and engineering research investigations, physical experimentation is often very expensive and quite time consuming. Instead of physically conducting an experiment, the main approach in computer experiments is to describe a physical system by some mathematical models and then assess the performance of the experiments using some engineering/physics laws and solve on computers through numerical methods. Because of the deterministic models are used for experiments, the output of a computer experiment is not subject to random variations, which is quite different from physical experiments (see Sacks *et al.* 1989). For example, randomization is not needed in computer experiments. In fact, it is desirable to avoid replicates, since it may create redundancy of data. While projecting the design on to a subset of factors replication may not require, because a few out of the many factors in the system usually dominate the performance of the product (known as effect sparsity principle). Thus using only these few important factors, a good model can be fitted. Different physical experiments involves different mathematical models and the true relationship between the input variables and the response variables is unknown and in most of the cases, very complicated. Various statistical models can be built using different techniques. Before data are collected, in most of the cases, little a priori knowledge may be available about

[1] rajender.parsad@icar.gov.in

underlying appropriate model. So the designs for computer experiments should provide diverse modelling methods. For this purpose, a space-filling design is the best choice in computer experiments.

Latin Hypercube Designs (LHD) introduced by McKay *et al.* (1979) with good space-filling property, are very useful in this particular situations. A Latin hypercube [$\mathbf{A=(}\ a_{ij}\ \mathbf{)}$] of n runs and k factors or dimension is represented by an $n \times k$ matrix, where each column is a uniform permutation of n equally spaced levels like $\{1, 2, 3, \ldots, n\}$ and all the columns are obtained independently.

Several researchers have contributed towards obtaining methods of construction of LHDs which provides designs with good space-filling property. Tang (1993) used orthogonal array (OA) for construction of LHD and proved that when used for integration, a sampling scheme with OA-based LHDs are more efficient than Latin hypercube sampling. Tang (1994) obtained a method of construction of maximin LHDs. Morris and Mitchell (1995) developed an algorithm to find LHDs which provide good space filling in terms of entropy and maximin distance criteria. Ye *et al*. (2000) proposed an algorithm to find symmetric LHDs by using exchange algorithm. Jin *et al.* (2005) developed an algorithm to find optimal LHDs by using Enhanced Stochastic Evolutionary (ESE) with respect to maximin distance criterion, entropy criterion (Shannon (1948)) and central L_2 discrepancy criterion (Hickernell, 1998), etc. Liefvendahl and Stocki (2005) compared the efficiency between columnwise-pairwise (CP) algorithm and genetic algorithm and concluded that columnwise-pairwise algorithm is preferred for small Latin hypercube over genetic algorithm where genetic algorithm is preferred over columnwise-pairwise in case of large LHDs. Dam *et al.* (2007) obtained maximin LHDs for run size ≤ 70. Viana *et al.* (2010) developed an algorithm based on Transitional Propagation to find optimal LHD with respect to minimization of Φp criterion (Morris and Mitchell, 1995), i.e., a criterion based on minimization of maximum distance between design points and compared with existing algorithms such as random search, genetic algorithm, enhanced stochastic evolutionary algorithm and concluded that the Φp criterion value tends to decrease as the dimension of LHD increases. Zhu *et al.* (2011) obtained an algorithm for finding maximin LHD using successive local enumeration and compared the algorithm based on successive local enumeration with the existing algorithm such as lhs design

function of MATLAB, binary coded genetic algorithm, permutation coded genetic algorithm and translation propagation algorithm and concluded that the algorithm proposed by them using successive local enumeration provides designs with better space-filling property as well as good projective property as compared to other algorithms. Pan *et al.* (2014) developed Translational Propagation and Successive Local Enumeration algorithm (TPSLE) to find optimal or near optimal LHD by combing two existing algorithms viz. (i) translational propagation algorithm of Viana *et al.,* 2010 and (ii) successive local enumeration algorithm of Zhu *et al.,* 2011 and showed that TPSLE is more efficient with respect to computational time, space-filling and projective property of the design obtained. LHDs with good space-filling properties can also be obtained using JMP version 10. JMP gives discrepancy criteria values for a generated LHD.

Above review reveals that most of the methods for obtaining LHDs for good space-filling are mainly based on algorithms. The main problems of algorithms are, it is useful to those user who are mainly familiar with it and sometimes algorithms may be very time consuming. So, definitely there is always a need of general method of construction for LHDs which provides good space-filling values. In this article, we propose a method of construction of LHDs for two factors which can provide solution for any numbers of runs. We also compared property of space-filling values of LHDs proposed method with method given by Dam *et al.* (2007) and Latin hypercube obtained from JMP 10 software.

2. Methods of construction

In this Section, two methods are proposed for construction of LHDs for two factors. First method is for constructing LHDs with even number of runs and the second method is for odd number of runs. We will describe two methods step by step.

2.1 Method of construction for even number of runs

Let runs be $n = 2r$. Following steps would give a Latin hypercube for two factors in even runs.

Step 1: Construct a matrix (**A**) of order $r \times 2$ for $n = 2r$ number of runs with elements as 1, 2, …, $2r$. That is,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ \vdots & \vdots \\ 2r-3 & 2r-2 \\ 2r-1 & 2r \end{bmatrix}$$

Step 2: Interchange the elements of the matrix for even numbered rows, i.e., if the elements of an even numbered row are (a, b) then change it to (b, a). Rewrite the matrix as matrix ($\mathbf{A_1}$) after interchange.

Step 3: Construct another matrix (**B**) of order ($r - 2$) ×2 as shown below.

$$\mathbf{B} = \begin{bmatrix} 2 & 5 \\ 4 & 7 \\ \vdots & \vdots \\ 2r-6 & 2r-3 \\ 2r-4 & 2r-1 \end{bmatrix}$$

Step 4: Augment the matrix **B** in Step 3 with two rows, one in first position and another in last position. Elements of the first row are as (1, 3) and for last row are ($n - 2$, n). The new matrix obtained after augmenting matrix **B** with two rows is named as **C** and is given as

$$\mathbf{C} = \begin{bmatrix} \begin{bmatrix} 1 & 3 \end{bmatrix} \\ \begin{bmatrix} \mathbf{B} \end{bmatrix} \\ \begin{bmatrix} n-2 & n \end{bmatrix} \end{bmatrix}$$

Step 5: Now rearrange the elements of **C** by replacing element of 2nd column by 1st column and vice-versa of that particular row i.e. $c_{i1} \leftrightarrow c_{i2}$ for all odd numbered rows. Rewrite the matrix (**C**) as matrix ($\mathbf{C_1}$) after alteration.

Step 6: A Latin hyper cube (**D**) can be obtained by vertically joining these two matrices $\mathbf{A_1}$ and $\mathbf{C_1}$ resulted from Step 2 and Step 5, respectively. That is,

$$\mathbf{D} = \begin{bmatrix} \mathbf{A_1} \\ \mathbf{C_1} \end{bmatrix}$$

is the Latin hypercube in $n = 2r$ runs for two factors.

Example 2.1: Consider an example of 10 runs and 2 factors. Here $r = 5$. As described above in step 1, matrix (**A**) is constructed of order 5×2. In step 2, it is updated by construction of matrix ($\mathbf{A_1}$) after interchange of elements for all even numbered rows. A new matrix (**B**) of order 3×2 has been constructed in step 3. Matrix (**C**) is constructed after augmentation of the matrix **B** in step 3 with two rows, one in first position (1, 3) and another in last position (8, 10). In step 5 matrix ($\mathbf{C_1}$) is constructed after alteration of elements for all odd numbered rows. The Latin hypercube is then obtained using Step 6. These steps are given as below:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \\ 9 & 10 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 4 & 3 \\ 5 & 6 \\ 8 & 7 \\ 9 & 10 \end{bmatrix} \qquad \begin{bmatrix} 2 & 5 \\ 4 & 7 \\ 6 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 \\ 2 & 5 \\ 4 & 7 \\ 6 & 9 \\ 8 & 10 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 1 \\ 2 & 5 \\ 7 & 4 \\ 6 & 9 \\ 10 & 8 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 4 & 3 \\ 5 & 6 \\ 8 & 7 \\ 9 & 10 \\ 3 & 1 \\ 2 & 5 \\ 7 & 4 \\ 6 & 9 \\ 10 & 8 \end{bmatrix}$$

Steps: **I (A)** **II (A_1)** **III (B)** **IV (C)** **V (C_1)** **VI**

2.2 Method of construction for two factors and odd number of runs

This method utilizes the specific pattern of treatment arrangement in a standard Latin square. The steps of construction are as given below.

Step 1: Denote the levels of runs alphabetically in natural order by considering standard form of Latin square.

Step 2: From i^{th} row select $\left[(n+1)-i\right]^{th}$ alphabets where n is the number of rows present in a Latin square.

Step 3: Latin hypercube can be obtained by writing row and column number of selected alphabet for a given row. As the geometric position of alphabets in Latin square is two dimensional, therefore, this arrangement produces a Latin hypercube in two factors.

Example 2.2: Consider constructing a Latin hypercube of 5 runs and two factors. From step 1, a 5×5 Latin square should be taken where all the levels of runs are assigned like (A, B, C, D, and E). A 5×5 Latin square in standard form is

$$\begin{bmatrix} A & B & C & D & E \\ B & C & D & E & A \\ C & D & E & A & B \\ D & E & A & B & C \\ E & A & B & C & D \end{bmatrix}$$

Using step 2 alphabet selection should be done in some specific pattern as given bellow.

Selected row (ith)	Selected alphabets $\left[(n+1)-i\right]^{th}$
1	5
2	4
3	3
4	2
5	1

From Latin Square in step 1, in step 3 alphabets should be selected from each row as enlisted bellow

$$\begin{bmatrix} A & B & C & D & E \\ B & C & D & E & A \\ C & D & E & A & B \\ D & E & A & B & C \\ E & A & B & C & D \end{bmatrix}$$

Finally the Latin hypercube is obtained from Step 3 for 5 runs and 2 factors and it is given as

$$\mathbf{D} = \begin{bmatrix} 1 & 5 \\ 2 & 3 \\ 3 & 1 \\ 4 & 4 \\ 5 & 2 \end{bmatrix}$$

This method can easily be applied to find a Latin hypercube for any odd numbers of runs with two factors.

3. Space Filling Criteria and Comparison of space-filling values

For comparison purpose, we study three space-filling criteria namely entropy criterion, Φ_p criterion and central L_2 discrepancy criterion. Shannon (1948) introduced entropy criterion as a measure of 'amount of information' available from a design. Later Koehler and Owen (1996) modified it to more simplified and analytical form for obtaining maximum entropy LHD ($D_{n\times k}$) by minimizing following expression $-\log|\mathbf{R}|$, where $\mathbf{R}=\left(R_{ij}\right)$ is the correlation matrix with

$$R_{ij} = \sigma^2 \exp\left(-\theta \sum_{l=1}^{k} \left| s_{il} - t_{jl} \right|^q \right) \qquad (3.1)$$

where s_{ij} and t_{jl} are two design points $\forall\ 1 \le i,\ j \le n$ and $1 \le q \le 2$, k is number of columns, θ is a constant and σ^2 is generally assumed to be 1. We use $q = 1$ for computation of entropy values.

Morris and Mitchell (1995) introduced the concept of maximin distance criterion by maximizing the minimum intrinsic distances between design points i.e. *max min* $\delta\left(X_i - X_j\right)$ for all $i \neq j$, where X_i and X_j are corresponding i^{th} and j^{th} run of a LHDfor finding optimal LHDs with good space filling properties. Computational form of Φ_p value is given below.

$$\Phi_p = \left[\sum_{i=1}^{s} J_i \delta_i^{-p}\right]^{\frac{1}{p}} \tag{3.2}$$

where J be the index i.e. ($J_1, J_2, ..., J_S$) is the number of time ($\delta_1,\delta_2,\ldots,\delta_S$) distances occur and $s \leq {}^nC_2$, p being any integer. We use, p = 15 (following Joseph and Hung, 2008) for computation Φ_p Criterion.

This criterion was introduced by Hickernell (1998) which measures the difference between empirical cumulative distribution function of a design and the uniform cumulative distribution function. Optimal LHD can be obtained by minimizing the following mathematical form

$$\begin{aligned}\left(CL_2\right)^2 = &\left(\frac{13}{12}\right)^k - \frac{2}{n}\sum_{i=1}^{n}\prod_{l=1}^{k}\left(1+\frac{1}{2}\left|d_{il}-0.5\right|-\frac{1}{2}\left|d_{il}-0.5\right|^2\right)+ \\ &\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\prod_{l=1}^{k}\left(1+\frac{1}{2}\left|d_{il}-0.5\right|+\frac{1}{2}\left|d_{jl}-0.5\right|-\frac{1}{2}\left|d_{il}-d_{jl}\right|\right)\end{aligned} \tag{3.3}$$

where d_{ij} are design points, k be the number of factors and n is the number of runs in a LHD.

The minimum of all these three criteria values ensures good space-filling LHDs. The property of space-filling values of LHDs for two factors from proposed method (denoted as PM) has been compared with LHDs available at http://www.spacefillingdesigns.nl by Dam *et al.* (2007) (denoted as DM) and LHDs obtained from JMP 10 software (denoted as JMP). The results are presented in Table 1.

Table 1: Comparison of Space filling values of Latin hypercubes

Design (runs ×factors)	Method	Space filling values			Design (runs ×factors)	Method	Space filling values		
		Φ_p value	CL_2 value	Entropy value			Φ_p value	CL_2 value	Entropy value
3×2	PM	0.5001	0.2826	0.1553	15×2	PM	0.3954	0.0725	79.5574
	DM	0.5236	0.3118	0.3093		DM	0.2461	0.0604	41.123
	JMP	0.5001	0.2826	0.1553		JMP	0.2574	0.0504	39.465
4×2	PM	0.3658	0.1954	0.4588	16×2	PM	0.3675	0.1093	66.7751
	DM	0.3658	0.1954	0.4588		DM	0.2447	0.0513	47.1583
	JMP	0.3658	0.1954	0.4588		JMP	0.2417	0.1592	29.8116
5×2	PM	0.3713	0.1633	1.7735	17×2	PM	0.3992	0.0688	113.143
	DM	0.3713	0.1633	1.7735		DM	0.2116	0.0435	51.48
	JMP	0.3713	0.1633	1.7735		JMP	0.2430	0.0497	54.0342
6×2	PM	0.3660	0.1357	3.4094	18×2	PM	0.3678	0.109	89.0611
	DM	0.3658	0.1276	3.5149		DM	0.2124	0.046	60.0988
	JMP	0.3593	0.1273	3.0436		JMP	0.2383	0.0419	60.882
7×2	PM	0.3712	0.1194	6.8841	19×2	PM	0.4026	0.066	153.549
	DM	0.2950	0.1072	4.6385		DM	0.2111	0.043	67.5299
	JMP	0.2950	0.1072	4.6385		JMP	0.2359	0.0403	67.9792
8×2	PM	0.3663	0.1185	9.3637	20×2	PM	0.3681	0.1088	114.121
	DM	0.2981	0.1036	7.8401		DM	0.2092	0.0384	75.1667
	JMP	0.2966	0.0963	7.6143		JMP	0.2267	0.0393	82.8645
9×2	PM	0.3795	0.0981	16.6201	21×2	PM	0.4056	0.0628	201.024
	DM	0.2950	0.0838	11.2208		DM	0.2100	0.0365	85.5824
	JMP	0.2950	0.0838	11.2208		JMP	0.2282	0.0361	90.0027
10×2	PM	0.3666	0.1129	18.6982	22×2	PM	0.3684	0.1086	142.506
	DM	0.2966	0.0811	15.7698		DM	0.2036	0.0351	92.5326
	JMP	0.2966	0.0811	15.7698		JMP	0.2231	0.034	98.6109
11×2	PM	0.3859	0.0856	31.6612	23×2	PM	0.4083	0.0623	255.792
	DM	0.2936	0.0728	20.0012		DM	0.4108	0.0386	105.266
	JMP	0.2828	0.0677	18.5916		JMP	0.2037	0.0322	109.399
12×2	PM	0.3669	0.1109	31.5279	24×2	PM	0.3687	0.1084	173.659
	DM	0.2414	0.0606	22.5212		DM	0.2117	0.043	118.644
	JMP	0.2829	0.0643	23.4594		JMP	0.2108	0.0335	120.679

13×2	PM	0.3911	0.0778	52.5075	25×2	PM	0.4108	0.0611	317.908
	DM	0.2424	0.0609	28.6118		DM	0.2124	0.0368	128.085
	JMP	0.2588	0.057	27.8658		JMP	0.2027	0.0316	129.79
14×2	PM	0.3672	0.1099	47.7204					
	DM	0.2446	0.0562	33.2194					
	JMP	0.2396	0.0572	33.7296					

For run size 3 with two factors LHDs from the proposed method has better space-filling than LHDs of Dam *et al.* (2007) and as efficient as those given by JMP. For run sizes 4 and 5 with two factors, proposed method is as efficient as other two. So it can be said that the proposed method is providing good LHDs in case of small number of runs. For medium run sizes, Latin hypercubes obtained from the proposed method shows slightly more space-filling values in comparison to other two but they do provide reasonable good space filling. It is also interesting to note here that the LHD for 4 runs obtainable from all the three methods of construction (including the proposed method) is orthogonal. Further for LHDs with odd number of runs obtainable from the proposed method of construction, the value of performance measure of correlation proposed by Owen(1994) is fixed as 0.5 irrespective of number of runs and also provides good space filling.

4. Conclusion

In literature, theoretical methods of construction of LHDs with good space filling properties are very rare. In this paper, two methods of construction of LHDs with good space filling properties are presented for two factors. LHDs for 3, 4 and 5 runs constructed using the proposed methods has same space filling values for different space filling criteria as LHDs from JMP 10 and even better than Dam *et al.* (2007) for 3 runs. For small number of runs, space-filling values of LHDs from proposed methods are good. SAs the method is general in nature and can be used for obtaining LHD for two factor in any number of runs.

Acknowledgements: This research work is part of M.Sc. thesis submitted by first author to PG School, ICAR-Indian Agricultural Research Institute, New Delhi. The research fellowship received by first author is duly acknowledged. Authors are also grateful to the reviewer for useful comments that has helped in improving the presentation of results.

References

Dam, E. R., Husslage, B., den Hertog, D. and Melissen, H. 2007. Maximin Latin hypercube designs in two dimensions. *Operations Research,* 55: 158–169.

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

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

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

Jin, R., Chen, W. and Sudjianto, A. 2005. An efficient algorithm for constructing optimal design of computer experiments. *J. Statist. Plann. Inf.,* 134: 268–287.

Koehler, J.R., Owen, A.B. 1996. Computer experiments. in: Ghosh, S., Rao, C.R. (Eds.), *Handbook of Statistics*. Elsevier Science, New York, 261 –308.

Liefvendahl, M. and Stocki, R. 2005. A study on algorithms for optimization of L[illegible] hypercubes. *J. Statist. Plann Inf.,*136**:** 3231 – 3247.

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: 239–245.

Morris, M. D. and Mitchell, T. J. 1995. Exploratory designs for computer experiments. *J. Statist. Plann. Inf.,* 43: 381–402.

Owen, A.B. 1994. A central limit theorem for Latin hypercube sampling. *J. Roy. Statist. Soc.* B 54: 541-551.

Pan, G., Ye, P., and Wang P. 2014. A novel Latin hypercube algorithm via translational Propagation. *The Scientific World Journal*, 1-15.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. 1989. Design and analysis of computer experiments. *Statistical Science* 4: 409–423.

Santner, T. J., B. J. Williams, W. I. Notz. 2003. The Design and Analysis of Computer Experiments. Springer Series in Statistics. Springer-Verlag, New York

Shannon, C. E. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27: 379–423.

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

Tang, B. 1994. A theorem for selecting OA-based Latin hypercubes using a distance criterion. *Commun. Statist: Theory and Meth.*, 23: 2047–2058.

Viana, F. A. C., Venter, G. and Balabanov, V. 2010. An algorithm for fast optimal Latin hypercube design of experiments. *International Journal for Numerical Methods in Engineering*, 82: 135–156.

Ye, K. Q., Li, W. and Sudjianto, A. 2000. Algorithmic construction of optimal symmetric Latin hypercube designs. *J. Statist. Plann. Inf.*, 90:145–159.

Zhu, H., Liu, L., Long, T. and Peng, L. 2011. A novel algorithm of maximin Latin hypercube design using successive local enumeration. *Engineering Optimization,* 1: 1–14.