Journal of Statistical Theory and Practice, 6:492–500, 2012
Copyright © Grace Scientific Publishing, LLC
ISSN: 1559-8608 print / 1559-8616 online
DOI: 10.1080/15598608.2012.695700

A Note on Near-Orthogonal Latin Hypercubes with Good Space-Filling Properties

NAM-KY NGUYEN[1] AND DENNIS K. J. LIN[2]

[1]International School and Centre for High Performance Computing, Vietnam National University, Hanoi, Vietnam
[2]Department of Statistics, Pennsylvania State University, University Park, Pennsylvania, USA

Orthogonal Latin hypercubes (OLHs) are generally inflexible with respect to run sizes and the numbers of factors, and do not guarantee desirable space-filling properties. This article presents a fast algorithm to construct near-OLHs. The constructed near-OLHs achieve near-orthogonality among columns and good space-filling properties. These designs improve those of Cioppa and Lucas (2007) and those constructed by the OA-based approach of Lin et al. (2009) with respect to both orthogonality and space-filling properties.

AMS Classification: 62K99.

Keywords: Algorithm; Computer experiments; Latin squares.

1. Introduction

Latin hypercubes (LHs) were introduced by McKay, Beckman, and Conover (1979) for computer experiments. Recently, this area of research has received a great deal of attention in the recent literature, for example, by Georgiou (2009), Lin et al. (2009), Pang et al. (2009), Sun et al. (2009; 2010), and Yang and Liu (2012). An $n \times k$ LH can be represented by a design matrix $X_{n\times k}$ with n rows (runs) and k columns (factors), each of which includes n uniformly spaced levels. An LH is called an orthogonal LH (OLH) if each pair of columns of this LH has zero correlation. Examples of OLHs can be found in Ye (1998), Steinberg and Lin (2006), and Cioppa and Lucas (2007). OLHs are generally inflexible with respect to the numbers of runs and factors and poor with respect to the space-filling property: that is, these designs do not spread points evenly throughout experimental region. The OLHs of Steinberg and Lin (2006), for example, are available for nearly $n - 1$ columns in n runs only when $n = 2^{2^m}$. So the method gives designs when $n = 16$, 256, or 65,536, but not for any intermediate sample sizes.

This paper discusses a fast algorithm for constructing near-OLHs in various sizes with good space-filling properties. The near-OLHs constructed by this algorithm will be compared with those constructed by the algorithm of Cioppa and Lucas (2007) (hereafter

Received March 21, 2011; accepted February 11, 2012.
Address correspondence to: Nam-Ky Nguyen, International School & Centre for High Performance Computing, Hanoi, Vietnam. Email: namnk@vnu.edu.vn

abbreviated as CL) and those constructed by the OA-based approach of Lin et al. (2009) (hereafter abbreviated as LMT) with respect to both properties. Before discussing this algorithm, we review two methods of constructing OLHs and near-OLHs.

2. Two Construction Methods for OLHs and Near-OLHs

2.1. *Construction of* $(2^{r+1} + 1) \times 2^r$ *OLHs*

Ye (1998) introduced a class of OLHs for $n = 2^{r+1} + 1$ rows and $k = 2r$ columns ($r = 1, 2, \ldots$) using permutation matrices. CL extended Ye's method and were able to introduce $1 + r + \binom{r}{2} - 2r$ additional orthogonal columns to Ye's OLHs. Methods independently developed by Nguyen (2008) and Sun et al. (2009) can construct OLHs with $n = 2^{r+1} + 1$ rows and $k = 2^r$ columns. In both methods, we define the matrix $T_1 = \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}$. T_r is then generated from T_{r-1} and the corresponding OLH can then be formed as $[T_r'\ 0'\ T_r]'$ where $0_{1\times 2^r}$ is a row vector of $0'$s. Details are as follows:

1. Partition T_{r-1} as $[\begin{smallmatrix} A \\ B \end{smallmatrix}]$ where $A = (a_{ij})$ and $B = (b_{ij})$ are two matrices of the same size.
2. Form matrix $A^* = (a^*_{ij})$ where $a^*_{ij} = \text{sign}(a_{ij})(|a_{ij}| + 2^{r-1})$, $i = 1, \ldots, 2^{r-2}$; $j = 1, \ldots, 2^{r-1}$ and $\text{sign}(a_{ij}) = a_{ij}/|a_{ij}|$.
3. Form matrix $B^* = (b^*_{ij})$ where $b^*_{ij} = \text{sign}(b_{ij})(|b_{ij}| + 2^{r-1})$, $i = 1, \ldots, 2^{r-2}$; $j = 1, \ldots, 2^{r-1}$.
4. Form T_r as:

$$\begin{pmatrix} A & A^* \\ B & -B^* \\ A^* & -A \\ B^* & B \end{pmatrix} \tag{1}$$

Following is the transpose 17 × 8 OLH constructed this way. It can be seen that the seven columns of the 17 × 7 OLH of CL are associated to columns 1–5 and 7–8 of this OLH.

1	2	3	4	5	6	7	8	0	−1	−2	−3	−4	−5	−6	−7	−8
2	−1	4	−3	6	−5	8	−7	0	−2	1	−4	3	−6	5	−8	7
3	−4	−1	2	7	−8	−5	6	0	−3	4	1	−2	−7	8	5	−6
4	3	−2	−1	8	7	−6	−5	0	−4	−3	2	1	−8	−7	6	5
5	6	−7	−8	−1	−2	3	4	0	−5	−6	7	8	1	2	−3	−4
6	−5	−8	7	−2	1	4	−3	0	−6	5	8	−7	2	−1	−4	3
7	−8	5	−6	−3	4	−1	2	0	−7	8	−5	6	3	−4	1	−2
8	7	6	5	−4	−3	−2	−1	0	−8	−7	−6	−5	4	3	2	1

The webpage http://designcomputing.net/olh displays the constructed OLHs for $r \leq 9$. Table 1 compares the number of orthogonal columns k of OLHs in Ye (1998), CL, and the newly obtained ones. It can be seen in this table that unlike the number of runs n in our OLHs, the run sizes n in Ye (1998) and CL increase dramatically as the number of orthogonal columns k increases. For example, to build an OLH for 32 columns, the just shown method requires only 65 runs, whereas the CL design requires 513 runs and Ye's (1998) design requires 131,073 ($= 2^{17} + 1$) runs.

Table 1
Comparing the number of orthogonal columns of OLHs in Ye (1998), CL (2007), and new OLHs

r	n	Ye (1998)	CL (2007)	New OLHs
3	17	6[†]	7	8
4	33	8[†]	11	16
5	65	10	16	32
6	129	12	22	64
7	257	14	29	128
8	513	16	37	256
9	1025	18	46	512

[†]These values have been updated to 8 and 9, respectively, in http://www.ams.sunysb.edu/~kye/olh.html

Two near-OLHs can be constructed from T_r (cf. Yang and Liu, 2012). Let $t^*_{ij} = \text{sign}(t^*_{ij})(|t_{ij}| + 1$ where t_{ij} and t^*_{ij} are the elements in the ith row and jth column of T_r and T^*_r, respectively. The first near-OLH is formed as $[T^{*\prime}_r\ 0' - T^{*\prime}_r]$ where $0_{1\times 2^r}$ is a row vector of 0's. Now let $t^*_{ij} = \text{sign}(t^*_{ij})(2|t_{ij}| + 1)$ where t_{ij} and t^*_{ij} are the elements in the ith row and jth column of T_r and T^*_r, respectively. The second near-OLH is formed as $[T^{*\prime}_r\ 1' - 1' - T^{*\prime}_r]$, where $1_{1\times 2^r}$ is a row vector of $1'$s.

2.2. OA-Based OLHs (and Near-OLHs)

Let A be an orthogonal array $\text{OA}(n^2, q, n, 2)$ with n^2 rows, q columns, n symbols, strength two, index unity, and symbols denoted by $0, \ldots, n-1$. Let B be an OLH or near-OLH with n rows and p columns. Assuming pq is even, the following operations proposed by LMT can be used to construct an OLH or near-OLH with n^2 rows and pq columns.

1. Form A_j from A by replacing symbols 0, 1, . . . of A by the first, second, . . . elements of column j of $B(j = 1, \ldots, p)$.
2. Partition $[A_1, \ldots, A_p]$ as $[A^*_1, \ldots, A^*_{\frac{1}{2}pq}]$, where each A^*_k has two columns $\left(k = 1, \ldots, \frac{1}{2}pq\right)$.
3. Let $V = \begin{bmatrix} 1 & -\text{n} \\ \text{n} & 1 \end{bmatrix}$. Form $M = [M_1, \ldots, M_{\frac{1}{2}pq})$, where $M_k = A^*V$.

LMT proved, among other things, (i) M (of order $n^2 \times pq$) is an OLH if B is an OLH; (ii) the maximum absolute correlation $r_{\max}$ among columns of M is the same as that of B; and (iii) the determinant of the correlation matrix among columns of M raised to the power $1/(pq)$ equals the one of B raised to to the power $1/p$.

The following 16 × 10 OLH was constructed using the preceding operations with A as an OA (16, 5, 4, 2) and $B = \begin{bmatrix} 1 & 3 & -1 & -3 \\ 3 & -1 & -3 & 1 \end{bmatrix}'$:

5	−3	5	−3	13	−1	15	−9	15	−9
13	−1	15	−9	15	−9	−5	3	−5	3
−3	−5	−5	3	11	7	−15	9	−15	9
−11	−7	−15	9	9	15	5	−3	5	−3
7	−11	−1	−13	−7	11	−1	−13	1	13
15	−9	−11	−7	−5	3	11	7	−11	−7
−1	−13	1	13	−1	−13	1	13	−1	−13
−9	−15	11	7	−3	−5	−11	−7	11	7
3	5	−13	1	−9	−15	−9	−15	−3	−5
11	7	−7	11	−11	−7	3	5	9	15
−5	3	13	−1	−15	9	9	15	3	5
−13	1	7	−11	−13	1	−3	−5	−9	−15
1	13	9	15	3	5	7	−11	−13	1
9	15	3	5	1	13	−13	1	7	−11
−7	11	−9	−15	5	−3	−7	11	13	−1
−15	9	−3	−5	7	−11	13	−1	−7	11

As an OA$(n^2, n+1, n, 2)$ exists when n is prime or prime power, if we take B as OLHs of size 5×2, 7×3, 8×4, 9×5, 11×7, and 13×6 and A as an associated OA, we will be able to derive OLHs of sizes 25×12, 49×24, 64×36, 81×50, 121×84, and 169×84.

Similarly, if we take B as near-OLHs of sizes 7×5, 8×6, 9×7, 11×9, and 13×12 and A as an associated OA, we will be able to derive near-OLHs of sizes 4×40, 64×54, 81×70, 121×108, and 169×168.

3. A General Near-OLH Algorithm

The previous section shows a method of constructing OLHs (and near-OLHs). Although the OLHs are orthogonal, they do not carry spacing-filling properties (cf. CL). This section describes a general algorithm for the construction of LHs that are near-orthogonal and have better space-filling properties. This algorithm is an example of the *exchange algorithm*. Example of this type of algorithm can be found in Nguyen (1996) and Nguyen and Lin (2011).

Without loss of generality, let the ith and uth row of $X_{n\times k}$ be two vectors of the form $(i \quad \mathbf{i}')$ and $(u \quad \mathbf{u}')$, where i and u are the first elements of row i and u, and $\mathbf{i}'$ and $\mathbf{u}'$ are two $1 \times (k-1)$ row vectors. It can be shown that the effect on $X'X$ obtained by swapping the two elements i of row i and u of row u is to add to it the matrix $-(i \quad \mathbf{i}')'(i \quad \mathbf{i}') - (u \quad \mathbf{u}')'(u \quad \mathbf{u}') + (u \quad \mathbf{i}')'(u \quad \mathbf{i}') + (i \quad \mathbf{u}')'(i \quad \mathbf{u}')$ or

$$\left(\begin{array}{c|c} 0 & -(u-i)(\mathbf{u}'-\mathbf{i}') \\ \hline -(u-i)(\mathbf{u}-\mathbf{i}) & \mathbf{0}_{k-1} \end{array}\right) \tag{2}$$

where $\mathbf{0}_{k-1}$ is the $(k-1) \times (k-1)$ matrix of $0'$s.

The algorithm for constructing near-OLH designs using the preceding matrix results has two basic steps:

1. Construct a *starting* design by setting all elements of row i as $i - 1 - (n-1)/2$ for odd n, and $2(i-1) - (n-1)$ for even n. Randomly order the elements in each column of

the design and form its corresponding $X'X$ matrix. Then calculate f, the sum of squares of the elements above the diagonal elements of $X'X$.

2. For columns j of X ($j = 1, \ldots k$), repeat searching a pair of elements in this column such that the swap of these two elements results in the biggest reduction in f. If the search is successful, update f, X, and $X'X$ using (1). If f cannot be reduced further, go to the next column. This step is repeated until $f = 0$ or f cannot be reduced by any further swaps.

Remarks

(i) To calculate the change in f and update f in step 2, only the nonzero elements of the vectors $-(u - i)(\mathbf{u}' - \mathbf{i}')$ will affect the changes (either increase or decrease) of the corresponding elements of $X'X$.

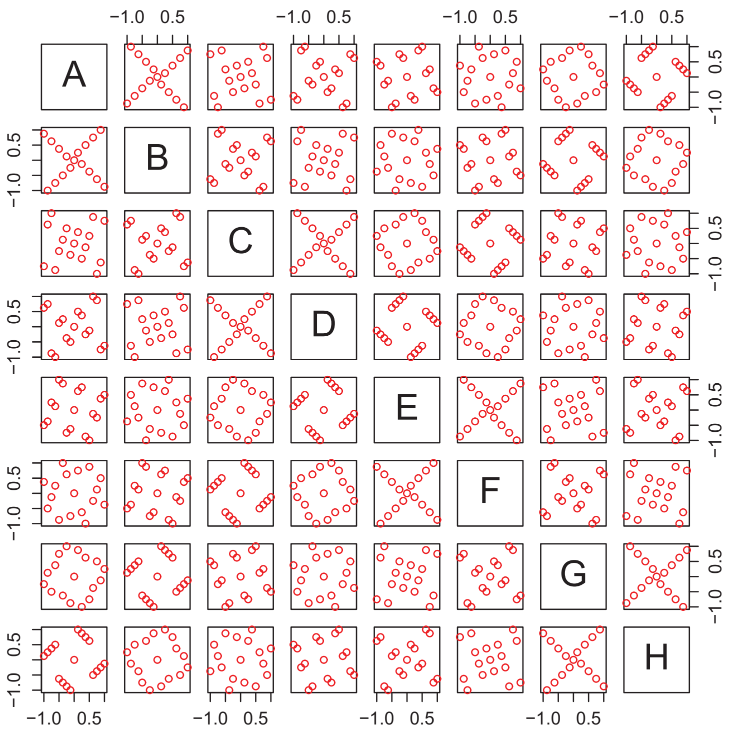
(ii) Steps 1 and 2 of the preceding algorithm constitute one complete try. Several tries are recommended to construct a design. Obviously, if the criterion is orthogonality, the try that results in the smallest $r_{\max}$ will be chosen; if the criterion is for space-filling, the try that results in the desirable *Mm* distance and/or ML_2 measure will be chosen. The following example shows the key steps in constructing a 5×3 near-OLH. Step 1 consists of (a) and (b) and step 1 consists of (c) and (d). In (b) f becomes 57. Then the second elements in the second and third rows of (b) are interchanged and (b) becomes (c) and f becomes 21. Finally, the third elements in the third and fourth rows of (c) are interchanged and (c) becomes (d) and f becomes 2.

$$
\begin{array}{ccc}
-2 & -2 & -2 \\
-1 & -1 & -1 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
2 & 2 & 2 \\
\multicolumn{3}{c}{\text{(a)}}
\end{array}
\qquad
\begin{array}{ccc}
-2 & -1 & 0 \\
-1 & 2 & -2 \\
0 & 1 & 1 \\
1 & -2 & 2 \\
2 & 0 & -1 \\
\multicolumn{3}{c}{\text{(b)}}
\end{array}
\qquad
\begin{array}{ccc}
-2 & -1 & 0 \\
-1 & 1 & -2 \\
0 & 2 & 1 \\
1 & -2 & 2 \\
2 & 0 & -1 \\
\multicolumn{3}{c}{\text{(c)}}
\end{array}
\qquad
\begin{array}{ccc}
-2 & -1 & 0 \\
-1 & 1 & -2 \\
0 & 2 & 2 \\
1 & -2 & 1 \\
2 & 0 & -1 \\
\multicolumn{3}{c}{\text{(d)}}
\end{array}
$$

Figure 1 displays the two-dimensional (2-D) graphs of the variables of an 17×8 OLH displayed in the previous section and of a near-OLH of the same size constructed by the algorithm in this section using the *Mm* distance criterion (second graph). The variables in these two graphs have been scaled to range from -1 to $+1$. This figure confirms the fact that OLHs and near-OLHs of Yang and Liu (2010) may not perform well with respect to the space-filling property.

Table 2 compares near-OLHs of CL and the newly obtained designs in terms of criteria for orthogonality and space-filling. The first orthogonality measure is $r_{\max} = \max(|r_{ij}|)$, where r_{ij} is the correlation between columns i and j of the LH. The second orthogonality measure used in CL is the condition number cond($X'X$) $= \psi_1/\psi_k$, where ψ_1 and ψ_k are the largest and smallest eigenvalues of $X'X$. As a benchmark, cond $(X'X) = 1$ is most ideal.

For the space-filling properties, we consider (i) the Euclidean maximin (*Mm*) distance and (ii) the modified L_2 (ML_2) discrepancy. The Euclidean maximin (*Mm*) distance is defined as the shortest distance among all the $\binom{n}{2}$ pairwise Euclidean distances of the n design points, calculated after the design is scaled to the domain $[-1,1]^k$. A large minimum distance is desirable. *Mm* distance has been used by Johnson et al. (1990), Morris



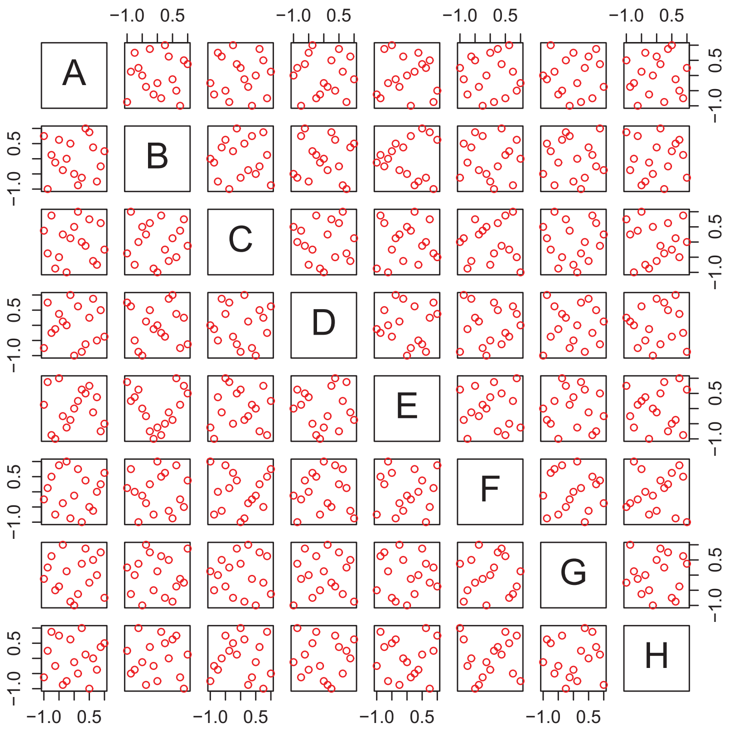


Figure 1. Two 2-D graphs of the variables of an OLH and of a near-OLH (color figure available online).

Table 2
Comparisons of CL's near-OLHs and new ones in terms of orthogonality and space-filling properties

n	k	Near-OLH	$r_{\max}$[†]	cond $(X'X)$	Mm distance[‡]	ML_2[†]
33	9	CL	0.0230	1.100	1.512	0.229
		New	0.007	1.025	1.5143	0.239
33	11	CL	0.0234	1.123	1.758	0.73
		New	0.0023	1.034	1.774	0.726
65	16	CL	0.0219	1.103	2.035	4.46
		New	0.0018	1.011	2.062	4.353
129	22	CL	0.0015	1.036	2.265	37.8
		New	0.0006	1.004	2.318	34.75

[†]The smaller, the better.
[‡]The larger, the better.

and Mitchell (1995), and CL. The other space-filling measure is the modified L_2 (ML_2) discrepancy, defined as

$$ML_2 = \left(\frac{4}{3}\right)^k - \frac{2^{1-k}}{n}\sum_{d=1}^{n}\prod_{i=1}^{k}(3 - x_{di}^2 + \frac{1}{n^2}\sum_{d=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{k}\{2 - \max(x_{di}, x_{ji})\} \qquad (3)$$

calculated after the design is scaled to the domain $[0, 1]^k$. ML_2 has been used by Hickernell (1998), Fang et al. (2000), and CL.

It can be seen in Table 2 that with the exception of the 33 $\times$ 9 near-OLH, all our near-OLHs are superior to those of CL with respect to the orthogonality property and the space-filling measures. Our 33 $\times$ 9 near-OLH, however, can only slightly improve the corresponding CL near-OLH with respect to the *Mm* distance but not with the ML_2 measure. Overall, all near-OLHs of ours have far smaller $r_{\max}$'s than the corresponding than the corresponding CL near-OLHs.

Each design listed in Table 2 is the result of 10,000 tries. The computer time varies for each near-OLH constructed. It is about 0.01 seconds per try for the 33 $\times$ 9 near-OLH and 2 seconds per try for the 129 $\times$ 22 near-OLH on a 2.6-GHz $\times$ 2 laptop.

Table 3 compares the near-OLHs constructed by the LMT approach and new ones in terms of orthogonality and space-filling properties. In this table, the two orthogonality measures used are the $r_{\max}$ and $|R|^{1/m}$, where R is the correlation matrix among columns of the LH. Note that the OLHs will have $|R|^{1/m} = 1$ as R becomes an identity matrix. The two space-filling properties *Mm* distance and *ML*2 have been explained in the previous paragraphs. As can be seen in this table, the new designs are better than the ones constructed by the LMT approach with respect to all listed measures.

With the exception of the 169 $\times$ 168 near-OLH, which is the result of just 10 tries, each design listed in Table 2 is the result of 100 tries. While it takes about 2 seconds per try to construct the 49 $\times$ 40 near-OLH in Table 3, it takes almost an hour per try to construct the 169 $\times$ 168 one in this table.

Table 3
Comparisons of near-OLHs constructed by the LMT approach and new ones in terms of orthogonality and space-filling properties

n	k	Near-OLH	$r_{\max}$[†]	$\lvert R\rvert^{1/m}$	Mm distance[‡]	$\ln(ML_2)$[†]
49	40	LMT	0.0357	0.9987	3.975	12.25
		New	0.0163	0.9998	4.410	12.01
64	54	LMT	0.0238	0.9994	4.747	17.54
		New	0.0063	0.9999	5.129	17.49
81	70	LMT	0.0333	0.9990	5.600	23.85
		New	0.0086	0.9999	6.266	23.72
121	108	LMT	0.0364	0.9985	5.920	34.46
		New	0.0029	$\simeq$1	7.98	34.46
169	168	LMT	0.0385	0.9974	10.30	34.46
		New	0.0021	$\simeq$1	10.66	34.46

[†]The smaller, the better.
[‡]The larger, the better.

4. Concluding Remarks

Orthogonality is known to be important for the linear model, where the unknown parameters can be estimated efficiently and independently (uncorrelated). On the other hand, the space-filling properties are important for model robustness. All existing designs seem to be optimal in one way, but could be poor from another. The near-OLHs constructed by the algorithm in the previous section keep the balance—they are good in both orthogonality (i.e., with very small $r_{\max}$) and space-filling properties, though they may not be optimal in a single dimension. This algorithm could also produce small OLHs (OLHs for seven or less factors). As mentioned in section 2, certain large OLHs or near-OLHs can be constructed from smaller ones and the latter can be easily constructed by our algorithm.

A special feature of our algorithm is that it can augment existing LH with additional columns that are orthogonal or near-orthogonal to the existing columns. For example, it is found that the column $(1, -3, 2, -1, -5, 3, -4, 8, -2, 7, -8, -6, 6, -7, 4, 0, 5)'$ is orthogonal to all columns of the CL 17×7 OLH.

All designs in Tables 2 and 3 of this article are available from the first author. LHD (http://designcomputing.net/gendex/lhd), the program used to generate all near-OLH designs in this articles is a module of the first author's Gendex DOE toolkit.

Acknowledgment

The authors are grateful to the two referees for helpful comments and corrections of the first draft.

References

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

Fang, K. T., C. Ma, and P. Winker. 2000. Centered L_2 discrepancy of random sampling and Latin hypercube design, and construction of uniform designs. *Math. Computation*, 71, 275–296.

Georgiou, S. D. 2009. Orthogonal Latin hypercube designs from generalized orthogonal designs. *J. Stat. Plan. Inference*, 139, 1530–1540.

Hickernell, F. J. 1998. A generealized discrepancy and quadrature error bound. *Math. Computation*, 67, 299–322.

Johnson, M., L. Moore, and D. Ylvisaker. 1990. Minimax and maximin distance designs. *J. Stat. Plan. Inference*, 26, 131–148.

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

McKay, M. D., R. J. Beckman, and W. J. Conover. 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 T. J. Mitchell. 1995. Exploratory designs for computer experiments. *J. Stat. Plan. Inference*, 43, 381–402.

Nguyen, N.-K. 1996. An algorithmic approach to constructing supersaturated designs. *Technometrics*, 38, 69–73.

Nguyen, N.-K. 2008. A new class of orthogonal Latin hypercubes. Special volume in honour of Aloke Dey. *Stat. Appl.*, 6, 119–123.

Nguyen, N.-K., and D. K. J. Lin. 2011. A Note on small composite designs for sequential experimentation. *J. Stat. Theory Pract.*, 5, 109–117.

Pang, F., M. Q. Liu, and D. K. J. Lin. 2009. A construction method for orthogonal Latin hypercube designs with prime power levels. *Stat. Sin.*, 19, 1721–1728.

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

Sun, F., M.Q. Liu, and D. K. J. Lin. 2009. Construction of orthogonal Latin hypercube designs. *Biometrika*, 96, 971–974.

Sun, F., M.Q. Liu, and D. K. J. Lin. 2010. Second-order orthogonal Latin hypercube designs with flexible run sizes. *J. Stat. Plan. Inference*, 140, 3235–3242.

Yang, J. Y., and M. Q. Liu. 2012. Construction of orthogonal and near orthogonal Latin hypercubes from orthogonal designs. *Stat. Sin.*, 22, 433–442.

Ye, K. Q. 1998. Orthogonal Latin hypercubes and their application in computer experiments. *J. Am. Stat. Assoc.*, 93, 1430–1439.