

Latin Hypercubes and Space-filling Designs

C. Devon Lin and Boxin Tang

1 Introduction

This chapter discusses a general design approach to planning computer experiments. The approach seeks design points that cover the design region as uniformly as possible. Such designs are referred to as *space-filling designs*.

Because of the deterministic feature of computer models, the three fundamental design principles, *randomization*, *replication*, and *blocking*, are irrelevant in computer experiments. The true relationship between the inputs and the responses is unknown and often very complicated. Various statistical models can be built using different techniques. However, before data are collected, quite often little priori or background knowledge is available about which model would be appropriate, and designs for computer experiments should facilitate diverse modelling methods. For this purpose, a space-filling design is the best choice. If we believe the response from a computer experiment follows a stationary process, local points might not provide useful information and it is more appropriate to use designs that represent all portions of the design region. When the primary goal of experiments is to make prediction at unsampled points, space-filling designs allow us to build a predictor with better accuracy.

One most commonly used class of space-filling designs for computer experiments is Latin hypercube designs. Such designs, introduced by McKay, Beckman and Conover (1979), do not have repeated runs. Latin hypercube designs have one-dimensional uniformity in that

when projected on each dimension, each portion of the design region has a design point. However, a random Latin hypercube design may not be a good choice with respect to some useful criteria such as maximin distance and orthogonality. The maximin distance criterion, first introduced by Johnson, Moore and Ylvisaker (1990), maximizes the smallest distance between any two design points so that no two design points are too close. Therefore, a maximin distance design spreads out its points evenly over the entire design region. To further enhance the space-filling property of maximin distance designs, Morris and Mitchell (1995) considered maximin Latin hypercube designs.

Many applications involve a large number of input variables. As such, finding space-filling designs with a limited number of design points that provide a good coverage of the entire high dimensional input space is a hopeless undertaking. To break this curse of dimensionality, the approach of constructing designs that are space-filling in the low dimensional projections has been considered by several researchers. Randomized orthogonal arrays (Owen, 1992) and orthogonal array-based Latin hypercubes (Tang, 1993) are such designs. Another important approach is to construct orthogonal Latin hypercube designs. Orthogonality is directly useful when polynomial models are used to fit the data. In addition, Bingham, Sitter and Tang (2009) argued that orthogonality can be viewed as stepping stones to designs that are space-filling in low dimensional projections.

Originated as popular tools in numerical analysis, low-discrepancy nets, low-discrepancy sequences and uniform designs have also been well recognized as space-filling designs for computer experiments. These designs are chosen to achieve better uniformity in the design space based on the discrepancy criteria such as the L_p discrepancy.

An alternative approach to designs for computer experiments is to choose designs that perform well with respect to some model-dependent criteria such as the minimum integrated mean square error and the maximum entropy (Sacks et al., 1989; Shewry and Wynn, 1987). Such designs are computationally difficult to generate. They require the prior knowledge on the model and thus have limited practical use. A detailed account of model-dependent designs can be found in Santner, Williams and Notz (2003), Fang, Li and Sudjianto (2006) and the reference therein.

This chapter is organized as follows. Section 2 gives a detailed review of Latin hypercube designs, and discusses three important types of Latin hypercube designs. Section 3 describes other space-filling designs that are not Latin hypercube designs. Concluding remarks are provided in Section 4.

2 Latin Hypercube Designs

2.1 Introduction and examples

A Latin hypercube of n runs for m factors is represented by an $n \times m$ matrix, each column of which is a permutation of n equally spaced levels. For convenience, the n levels are taken to be $-(n-1)/2, -(n-3)/2, \dots, (n-3)/2, (n-1)/2$. For example, design L in Table 1 is a Latin hypercube of 5 runs for 3 factors. Given an $n \times m$ Latin hypercube $L = (l_{ij})$, a Latin hypercube design $D = (d_{ij})$ in the design space $[0, 1]^m$ is generated via

$$d_{ij} = \frac{l_{ij} + (n-1)/2 + u_{ij}}{n}, \quad i = 1, \dots, n, j = 1, \dots, m, \quad (1)$$

where u_{ij} 's are independent random numbers from $[0, 1)$. If each u_{ij} in (1) is taken to be 0.5, the resulting design D is termed "lattice sample" due to Patterson (1954). When projected on any single factor, Latin hypercube designs have exactly one point in each of the n intervals $[0, 1/n), [1/n, 2/n), \dots, [(n-1)/n, 1)$. Thus, Latin hypercube designs achieve maximum univariate stratification. For instance, design D in Table 1 is a Latin hypercube design based on the L in the table, and its pairwise plot in Figure 1 illustrates the one-dimensional uniformity of Latin hypercube designs.

Table 1. A 5×3 Latin hypercube L and a Latin hypercube design D based on L

L			D		
2	0	-2	0.9253	0.5117	0.1610
1	-2	-1	0.7621	0.1117	0.3081
-2	2	0	0.1241	0.9878	0.4473
0	-1	2	0.5744	0.3719	0.8270
-1	1	1	0.3181	0.7514	0.6916

The popularity of Latin hypercube designs was largely attributed to their theoretical justification for the variance reduction in numerical integration. Consider a deterministic function $Y = f(X)$ where f is known, X has a uniform distribution in the unit hypercube $[0, 1)^m$, and $Y \in R$. The expectation of Y ,

$$\mu = E(Y),$$

is of interest. When the expectation μ cannot be computed explicitly or its derivation is unwieldy, one can resort to approximate methods. The simplest method is to generate

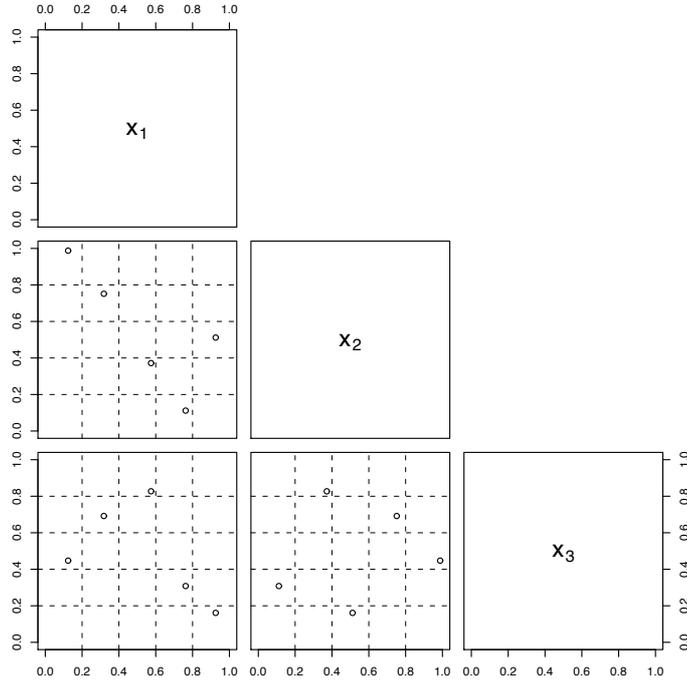


Figure 1. The pairwise plot of the Latin hypercube design D in Table 1 for the three factors x_1, x_2, x_3

X_1, \dots, X_n independently from the uniform distribution in $[0, 1]^m$ and estimate μ using

$$\hat{\mu}_{srs} = \frac{1}{n} \sum_{i=1}^n f(X_i).$$

We shall refer to this method as *simple random sampling*. McKay, Beckman and Conover (1979) suggested a sampling approach based on a Latin hypercube design with n runs X_1, \dots, X_n . Denote the estimator of μ under this Latin hypercube sampling by $\hat{\mu}_{lhs}$. McKay, Beckman and Conover (1979) established the following theorem.

Theorem 1. *If $Y = f(X)$ is monotonic in each of its input variables, then $\text{Var}(\hat{\mu}_{lhs}) \leq \text{Var}(\hat{\mu}_{srs})$.*

The proof of Theorem 1 can be found in McKay, Beckman and Conover (1979) and Sant-

ner, Williams and Notz (2003). Theorem 1 says that when the monotonicity conditions hold, Latin hypercube sampling offers improvement over simple random sampling with respect to variances of the sample mean. Theorem 2 below (Stein, 1987) provides some insights into the variances of the sample mean under simple random sampling and Latin hypercube sampling.

Theorem 2. *We have that for $X \in [0, 1]^m$,*

$$\text{Var}(\hat{\mu}_{srs}) = \frac{1}{n} \text{Var}[f(X)]$$

and

$$\text{Var}(\hat{\mu}_{ths}) = \frac{1}{n} \text{Var}[f(X)] - \frac{1}{n} \sum_{j=1}^m \text{Var}[f_j(X^j)] + o\left(\frac{1}{n}\right),$$

where X^j is the j th input of X and $f_j(X^j) = E[f(X)|X^j] - E(f(X))$.

Theorem 2 states that Latin hypercube sampling filters out the main effects by stratifying each univariate margin and thus gives smaller variance than simple random sampling. Stein (1987) also showed that the extent of the variance reduction depends on the extent to which the function f is additive. Asymptotical normality and a central limit theorem of Latin hypercube sampling were established in Stein (1987) and Owen (1992), respectively.

A random Latin hypercube design does not necessarily compare well with respect to other useful criteria such as space-filling property or orthogonality. For example, when projected on two factors, design points in a random Latin hypercube design may lie on the diagonal as in the plot of x_1 versus x_2 in Figure 1. Such two columns are also highly correlated and leave a large area in the design space unexplored. Examples of Latin hypercube designs with useful properties are maximin Latin hypercube designs, orthogonal-array based Latin hypercube designs, and orthogonal or nearly orthogonal Latin hypercube designs.

2.2 Maximin Latin hypercube designs

Maximin Latin hypercube designs are optimal Latin hypercube designs with respect to the popular maximin distance criterion introduced by Johnson, Moore and Ylvisaker (1990). The idea is to enhance space-filling property of Latin hypercube designs by using the maximin distance criterion.

Let $u = (u_1, \dots, u_m)$ and $v = (v_1, \dots, v_m)$ be two design points in the design space $\chi = [0, 1]^m$. For $t > 0$, define the inter-site distance between u and v to be

$$d(u, v) = \left\{ \sum_{j=1}^m |u_j - v_j|^t \right\}^{1/t}. \quad (2)$$

When $t = 1$ and $t = 2$, the measure in (2) becomes the rectangular and Euclidean distances, respectively. The maximin distance criterion seeks a design \mathcal{D} of n points in the design space χ that maximizes

$$\min_{\substack{u, v \in \mathcal{D} \\ u \neq v}} d(u, v),$$

where $d(u, v)$ is defined as in (2) for any given t . This criterion attempts to locate the design points such that no two points are too close to each other. The theoretical justification of the maximin distance criterion was provided by Johnson, Moore and Ylvisaker (1990) as follows. Consider the model,

$$Y(x) = \mu + Z(x), \quad (3)$$

where μ is the unknown but constant mean function, $Z(x)$ is a stationary Gaussian process with mean 0, variance σ^2 , and correlation function $R(\cdot|\theta)$. A popular choice for the correlation function is the power exponential correlation

$$R(h|\theta) = \exp\left\{ -\theta \sum_{j=1}^m |h_j|^p \right\}, \quad 0 < p \leq 2.$$

Johnson, Moore and Ylvisaker (1990) showed that as the correlation parameter θ goes to infinity, a maximin design maximizes the determinant of the correlation matrix. That is, a maximin design is asymptotically D-optimal under the model in (3) as the correlations become weak. Thus, a maximin design is also asymptotically optimal with respect to the maximum entropy criterion (Shewry and Wynn, 1987).

The problem of finding maximin designs is referred to as the maximin facility dispersion problem (Erkut, 1990) in location theory. It is closely related to the sphere packing problem in the field of discrete and computational geometry (Melissen, 1997; Conway, Sloane and Bannai, 1999). Although there is a rich literature on these two fields containing useful results for obtaining maximin designs, the problems are, however, different, as explained in Johnson, Moore and Ylvisaker (1990).

An extended definition of a maximin design was given by Morris and Mitchell (1995). Define a distance list (d_1, \dots, d_k) and an index list (J_1, \dots, J_k) respectively in the following way. The distance list contains the distinct values of inter-site distances, sorted from the smallest to the largest, and J_i in the index list is the number of pairs of design points in the design separated by the distance d_i , $i = 1, \dots, k$. Note that $1 \leq k \leq \binom{n}{2}$. A design is called a maximin design if it sequentially maximizes d_i 's and minimizes J_i 's in the order $d_1, J_1, d_2, J_2, \dots, d_k, J_k$. Morris and Mitchell (1995) further introduced a computationally efficient scalar-value criterion

$$\phi_p = \left(\sum_{i=1}^k \frac{J_i}{d_i^p} \right)^{1/p}, \quad (4)$$

where p is a positive integer. Minimizing ϕ_p with a large p results in a maximin design. Values of p are chosen heavily depending on the size of the design searched for, ranging from

5 for small designs to 20 for moderate-sized designs to 50 for large designs.

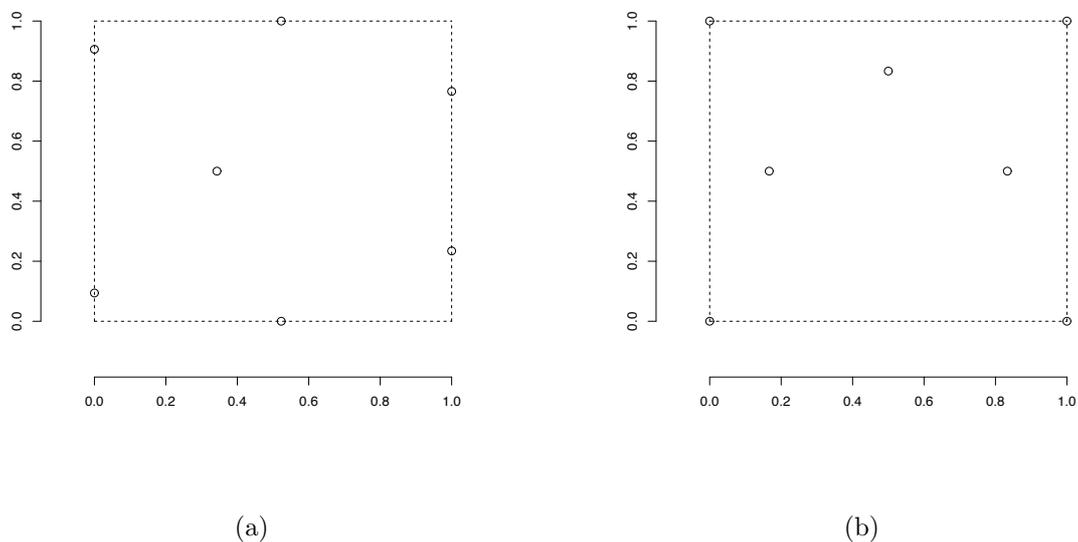


Figure 2. Maximin designs with $n = 7$ points on $[0, 1]^2$: (a) Euclidean distance; (b) rectangle distance

Maximin designs tend to locate design points toward or on the boundary. For example, Figure 2 exhibits a maximin Euclidean distance design and a maximin rectangle distance design, both of seven points. Maximin designs are likely to have clumped projections onto one-dimension. Thus, such designs may not possess desirable one-dimensional space-filling property which is guaranteed in Latin hypercube designs. To strike the balance, Morris and Mitchell (1995) examined maximin designs within Latin hypercube designs. Although this idea sounds simple, generating maximin Latin hypercube designs is a challenging task particularly for large designs, because Latin hypercube designs are a very large class of designs. The primary approach for obtaining maximin Latin hypercube designs is using the algorithms summarized in Table 2. Some available implementations of these algorithms

Table 2. Algorithms for generating maximin Latin hypercube designs

Article	Algorithm	Criterion
Morris and Mitchell (1995)	simulated annealing	$\phi_p^{(a)}$
Ye et al. (2000)	columnwise-pairwise	ϕ_p and entropy
Jin et al. (2005)	enhanced stochastic evolutionary algorithm	ϕ_p , entropy and L_2 discrepancy
Liefvendahl and Stocki (2006)	columnwise-pairwise and genetic algorithm	maximin and the Audze-Eglais function ^(b)
van Dam et al. (2007)	branch-and-bound	maximin with Euclidean distance
Grosso et al. (2009)	iterated local search heuristics	ϕ_p
Viana et al. (2010)	translational propagation	ϕ_p
Zhu et al. (2011)	successive local enumeration	maximin

(a): ϕ_p is given as in (4); (b): the Audze-Eglais function (Audze and Eglais, 1977) is $\min \sum_{i=1}^{n-1} \sum_{j=i+1}^n d(x_i, x_j)^{-2}$ for a design with n points x_1, \dots, x_n where $d(x_i, x_j)$ is defined as in (2).

include the Matlab code provided in Viana et al. (2010), the function *maximinLHS* in the R package *lhs* (Carnell, 2009), and the function *lhsdesign* in the Matlab statistics toolbox. It should be noted, however, these designs are approximate maximin Latin hypercube designs. No general method is available to construct exact maximin Latin hypercube designs of flexible run sizes except that Tang (1994) and van Dam et al. (2007) provided methods for constructing exact maximin Latin hypercube designs of certain run sizes and limited numbers of input variables. Tang (1994) constructed Latin hypercubes based on full factorial designs with a single replicate (Wu and Hamada, 2011) and showed that if the latter is a maximin design, so is the corresponding Latin hypercube. van Dam et al. (2007) constructed

two-dimensional maximin Latin hypercubes with the distance measures $t = 1$ and $t = \infty$ in (2). For the Euclidean distance $t = 2$, van Dam et al. (2007) used the branch-and-bound algorithm to find maximin Latin hypercube designs with $n \leq 70$. These designs are available from the website <http://www.spacefillingdesigns.nl>.

2.3 Orthogonal array-based Latin hypercube designs

Tang (1993) proposed orthogonal array-based Latin hypercube designs, also known as U designs, that achieve better multi-dimensional space-filling property. We first review the concept of orthogonal arrays. An s -level orthogonal array (OA) of n runs, m factors and strength r , denoted by $\text{OA}(n, m, s, r)$, is an $n \times m$ matrix of s entries, called levels, such that for any r columns all of their level combinations appear equally often. The number $\lambda = n/s^r$ is called the index of the array. The s levels are taken to be $1, 2, \dots, s$ in this chapter. By the definition of orthogonal arrays, a Latin hypercube of n runs for m factors is an $\text{OA}(n, m, n, 1)$. Table 3 presents an $\text{OA}(9, 4, 3, 2)$.

The construction of OA-based Latin hypercubes in Tang (1993) works as follows. Let A be an $\text{OA}(n, m, s, r)$. For each column of A and $k = 1, \dots, s$, replace the n/s positions with entry k by a random permutation of $(k-1)n/s + 1, (k-1)n/s + 2, \dots, kn/s$. Denote the design after the above replacement procedure by $B = (b_{ij})_{n \times m}$. In our notation, such an OA-based Latin hypercube is represented by $L = B - (n+1)/2$. An OA-based Latin hypercube design in the design space $[0, 1]^m$ can be generated via (1). In addition to the univariate maximum stratification, an $\text{OA}(n, m, s, r)$ -based Latin hypercube has the r -dimensional projection property that when projected onto any r columns, it has exactly λ

Table 3. An OA(9, 4, 3, 2) and a corresponding OA-based Latin hypercube

OA(9, 4, 3, 2)				L			
1	1	1	1	-2	-2	-4	-2
1	2	2	3	-4	0	1	2
1	3	3	2	-3	4	2	1
2	1	2	2	-1	-4	-1	-1
2	2	3	1	1	-1	4	-3
2	3	1	3	0	2	-3	4
3	1	3	3	3	-3	3	3
3	2	1	2	2	1	-2	0
3	3	2	1	4	3	0	-4

points in each of the s^r cells \mathcal{P}^r where $\mathcal{P} = \{[0, 1/s], [1/s, 2/s), \dots, [1 - 1/s, 1)\}$. Example 1 illustrates this feature of an OA(9, 4, 3, 2)-based Latin hypercube.

Example 1. Table 3 displays an OA-based Latin hypercube L based on the OA(9, 4, 3, 2) in the table. Figure 3 depicts the pairwise plot of a design associated with such a Latin hypercube. In each subplot, there is exactly one point in each of nine dot-dash line boxes.

Using the same idea of the above level replacement procedure, a generalization of OA-based Latin hypercubes using asymmetrical orthogonal arrays can be readily made. An orthogonal array is asymmetrical if the i th column has s_i levels and not all s_i 's are equal, $i = 1, \dots, m$. Asymmetrical orthogonal arrays are also referred to as mixed-level orthogonal arrays (Wang and Wu, 1991). As an illustration, Figure 4(a) displays a Latin hypercube

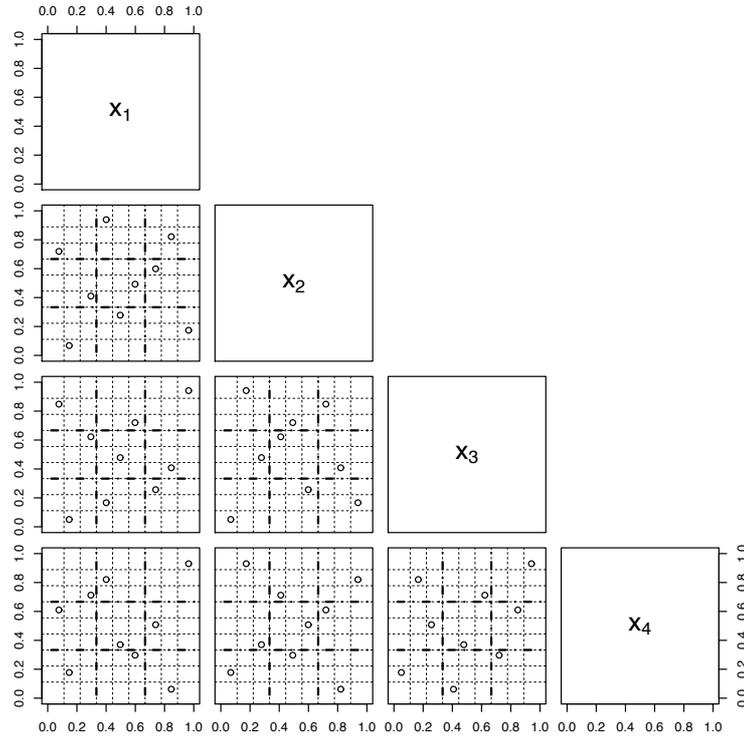


Figure 3. The pairwise plot of an OA-based Latin hypercube design based on the Latin hypercube in Table 3 for the four factors x_1, \dots, x_4

design based on an asymmetrical orthogonal array of six runs for two factors with three levels in the first factor and two levels in the second factor. Note that each of the six cells separated by dot-dash lines contains exactly one point. By contrast, two out of six such cells do not contain any point in a six-point random Latin hypercube design, as displayed in Figure 4(b).

Compared with orthogonal arrays, OA-based Latin hypercubes are more favorable for computer experiments. When projected onto lower dimensions, the design points in orthogonal arrays often have replicates. This is undesirable at the early stage of deterministic computer experiments when only a few factors are deemed to be important. In addition,

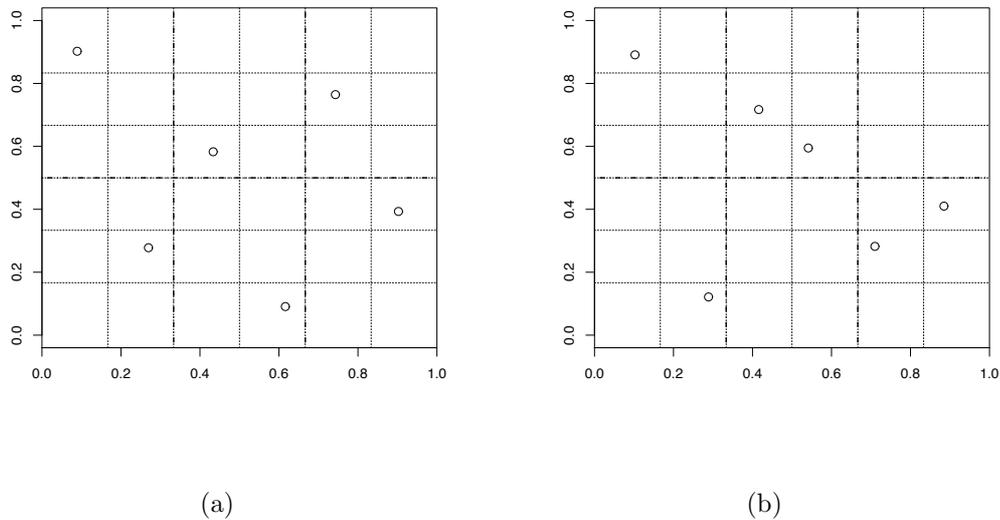


Figure 4. Plots of six-points Latin hypercube designs: (a) a Latin hypercube design based on an asymmetrical orthogonal array; (b) a random Latin hypercube design

OA-based Latin hypercubes are more space-filling than orthogonal arrays in that the former tend to place points both in the interior and on the boundary of the design space while the latter is more likely to have design points on the boundary.

The construction of OA-based Latin hypercubes depends on the existence of orthogonal arrays. For the existence results of orthogonal arrays, see, for example, Hedayat, Sloane and Stufken (1999) and Mukerjee and Wu (2006). A library of orthogonal arrays is freely available on the N.J.A. Sloane website and the MktEx macro using the software SAS (Kuhfeld, 2009). It should be noted that for certain given run sizes and the number of factors, orthogonal arrays of different numbers of levels and different strength may be used. For instance, an $OA(16, 5, 4, 2)$, an $OA(16, 5, 2, 3)$ and an $OA(16, 5, 2, 2)$ all produce OA-based Latin hypercubes of 16 runs for 5 factors. However, orthogonal arrays of higher levels or higher strength

are preferable because they lead to designs with better projection space-filling property.

Given an orthogonal array, the construction can produce many OA-based Latin hypercubes because of the randomness in the replacement of the levels in each column of an orthogonal array. This leads to the problem of choosing a preferable OA-based Latin hypercube. Tang (1994) provided a way to obtain OA-based Latin hypercubes based on single-replicated full factorial designs and showed that if the underlying orthogonal array is optimal with respect to the maximin distance criterion, so is the corresponding OA-based Latin hypercube. Leary, Bhaskar and Keane (2003) considered searching for optimal OA-based Latin hypercubes that minimize

$$\sum_{i=1}^n \sum_{j \neq i} \frac{1}{d_{ij}^2},$$

where d_{ij} is the Euclidean distance, defined as in (2) with $t = 2$, between the i th and j th design points. The optimization was performed via the simulated annealing algorithm (Morris and Mitchell, 1995) and the columnwise-pairwise algorithm (Li and Wu, 1997).

A random Latin hypercube design stratifies each univariate margin simultaneously and thus achieves a variance reduction compared with simple random sampling. Tang (1993) showed that Latin hypercube designs based on orthogonal arrays of strength two and higher result in more variance reduction than random Latin hypercube designs do. This is quantified in the theorem below. Following the notations in Section 2.1, consider estimating the mean μ of a known function $Y = f(X)$ using a design with n points X_1, \dots, X_n .

Theorem 3. *Suppose that f is continuous on $[0, 1]^m$. Let $\bar{Y} = n^{-1} \sum_{i=1}^n f(X_i)$. If the n*

design points X_1, \dots, X_n form an OA-based Latin hypercube design, we have that

$$\text{Var}(\bar{Y}) = \frac{1}{n} \text{Var}[f(X)] - \frac{1}{n} \sum_{j=1}^m \text{Var}[f_j(X^j)] - \frac{1}{n} \sum_{i < j}^m \text{Var}[f_{ij}(X^i, X^j)] + o\left(\frac{1}{n}\right),$$

where X^j is the j th input of X , $f_j(X^j) = E[f(X)|X^j] - \mu$, and $f_{ij}(X^i, X^j) = E[f(X)|X^i, X^j] - \mu - f_i(X^i) - f_j(X^j)$.

The detailed proof of Theorem 3 is given in Tang (1993). To understand the importance of the theorem, we can write

$$f(X) = \mu + \sum_{j=1}^m f_j(X^j) + \sum_{i < j}^m f_{ij}(X^i, X^j) + r(X),$$

where the terms on the right side of the equation are uncorrelated with each other. Thus, the variance decomposition of the function f is

$$\text{Var}[f(X)] = \sum_{j=1}^m \text{Var}(f_j) + \sum_{i < j}^m \text{Var}(f_{ij}) + \text{Var}[r(X)].$$

Consequently, with the result in Theorem 3, the variance of \bar{Y} associated with OA-based Latin hypercube designs is $n^{-1}\text{Var}[r(X)] + o(n^{-1})$, implying that if the variance due to three-factor or higher order interactions is low, so is the variance of \bar{Y} under the sampling using OA-based Latin hypercube designs.

We conclude this section by mentioning that randomized orthogonal arrays proposed by Owen (1992) also enjoy space-filling property in the low-dimensional projections. Results similar to Theorem 3 are given in Owen (1992). For the definition and the sampling property of randomized orthogonal arrays, we refer readers to Owen (1992).

2.4 Orthogonal and nearly orthogonal Latin hypercube designs

This section discusses the properties and constructions of orthogonal and nearly orthogonal Latin hypercube designs. Such designs are Latin hypercube designs of which any two columns have zero or small correlations. Orthogonal Latin hypercube designs are directly useful in fitting data using polynomial models because they allow uncorrelated estimates of linear main effects. Another rationale of using orthogonal or nearly orthogonal Latin hypercube designs is that they may not be space-filling, but space-filling designs should be orthogonal or nearly orthogonal. Thus orthogonality or near orthogonality may be viewed as stepping stone to space-filling designs. Other justifications are given in Iman and Conover (1982), Owen (1994), Tang (1998), Joseph and Hung (2008), among others.

Extensive research has been carried out on the construction of orthogonal or nearly orthogonal Latin hypercube designs. Ye (1998) initiated this line of research and constructed orthogonal Latin hypercubes with the run sizes of the form $n = 2^k$ or $2^k + 1$ and the number of factors $m = 2k - 2$ where $k \geq 2$. Ye's construction was extended by Cioppa and Lucas (2007) to obtain more columns for the given run sizes. Steinberg and Lin (2006) provided orthogonal Latin hypercubes of the run sizes $n = 2^{2^k}$ by rotating groups of factors in a two-level 2^{2^k} -run regular fractional factorial designs. This idea was generalized by Pang, Liu and Lin (2009) who constructed orthogonal Latin hypercubes of p^{2^k} runs and up to $(p^{2^k} - 1)/(p - 1)$ factors by rotating groups of factors in a p -level p^{2^k} -run regular factorial designs, where p is a prime. Lin (2008) obtained orthogonal Latin hypercube designs of small run sizes ($n \leq 20$) through an algorithm that adds columns sequentially to an existing design. More recently, various methods have been proposed to construct orthogonal Latin

hypercubes of more flexible run sizes and with large factor-to-run ratios. We here review constructions in Lin, Mukerjee and Tang (2009), Sun, Liu and Lin (2009), and Lin et al. (2010). For other methods, see Georgiou (2009), Sun, Liu and Lin (2010), and Yang and Liu (2012).

We give some useful notations and background. A vector $A = (a_1, \dots, a_n)$ is said to be balanced if its distinct values have equal frequency. For an $n_1 \times m_1$ matrix A and an $n_2 \times m_2$ matrix B , their Kronecker product $A \otimes B$ is the $(n_1 n_2) \times (m_1 m_2)$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m_1}B \\ a_{21}B & a_{22}B & \dots & a_{2m_1}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_1 1}B & a_{n_1 2}B & \dots & a_{n_1 m_1}B \end{bmatrix}$$

with $a_{ij}B$ itself being an $n_2 \times m_2$ matrix. For an $n \times m$ design or matrix $D = (d_{ij})$, define its correlation matrix to be

$$R(D) = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mm} \end{pmatrix}, \quad (5)$$

where

$$r_{ij} = \frac{\sum_{k=1}^n (d_{ki} - \bar{d}_i)(d_{kj} - \bar{d}_j)}{\sqrt{\sum_k (d_{ki} - \bar{d}_i)^2 \sum_k (d_{kj} - \bar{d}_j)^2}} \quad (6)$$

represents the correlation between the i th and j th columns of D , $\bar{d}_i = n^{-1} \sum_{k=1}^n d_{ki}$ and $\bar{d}_j = n^{-1} \sum_{k=1}^n d_{kj}$. A design or matrix D is column-orthogonal if R in (5) is an identity matrix. Furthermore, a design or matrix $D = (d_{ij})$ is orthogonal if it is balanced and

column-orthogonal. To assess near orthogonality of design D , Bingham, Sitter and Tang (2009) introduced two measures, the maximum correlation $\rho_M(D) = \max_{i,j} |r_{ij}|$ and the average squared correlation $\rho_{ave}^2(D) = \sum_{i < j} r_{ij}^2 / [(m(m-1)/2)]$, where r_{ij} is defined as in (6). Smaller values of $\rho_M(D)$ and $\rho_{ave}^2(D)$ imply near orthogonality. Obviously, if $\rho_M(D) = 0$ or $\rho_{ave}^2(D) = 0$, then an orthogonal design is obtained. For a concise presentation, we use $OLH(n, m)$ to denote an orthogonal Latin hypercube of n runs for m factors. Lin et al. (2010) established the following theorem on the existence of orthogonal Latin hypercubes. Note that the run size n in an orthogonal Latin hypercube cannot be two or three.

Theorem 4. *There exists an orthogonal Latin hypercube of $n \geq 4$ runs with more than one factor if and only if $n \neq 4k + 2$ where k is an integer.*

2.4.1 A method based on an orthogonal array and a small orthogonal Latin hypercube

Lin, Mukerjee and Tang (2009) constructed a large orthogonal or nearly orthogonal Latin hypercube by coupling an orthogonal array of index unity with a small orthogonal or nearly orthogonal Latin hypercube. Let $B = (b_{ij})$ be an $n \times m$ Latin hypercube. Then the elements in every column of B add up to zero while the sum of squares of these elements is $n(n^2 - 1)/12$. Thus the correlation matrix defined as in (5) is

$$R(B) = \left\{ \frac{1}{12} n(n^2 - 1) \right\}^{-1} B^T B. \quad (7)$$

Let A be an orthogonal array $OA(n^2, 2f, n, 2)$ defined as in Section 2.3. The construction proposed by Lin, Mukerjee and Tang (2009) proceeds as follows.

Step I: For $1 \leq j \leq m$, obtain an $n^2 \times (2f)$ matrix A_j from A by replacing the symbols $1, 2, \dots, n$ in the latter by $b_{1j}, b_{2j}, \dots, b_{nj}$ respectively, and then partition A_j as $A_j = [A_{j1}, \dots, A_{jf}]$, where each of A_{j1}, \dots, A_{jf} has two columns.

Step II: For $1 \leq j \leq m$, obtain the $n^2 \times (2f)$ matrix $L_j = [A_{j1}V, \dots, A_{jf}V]$, where

$$V = \begin{bmatrix} 1 & -n \\ n & 1 \end{bmatrix}.$$

Step III: Obtain the matrix $L = [L_1, \dots, L_m]$, of order $N \times q$, where $N = n^2$ and $q = 2mf$.

For $m = 1$, the above construction is equivalent to that in Steinberg and Lin (2006) and Pang, Liu and Lin (2009). However, we have $m \geq 2$ when n is not equal to 3 or $4k+2$ for any non-negative integer k , as in Theorem 4. Thus, the above method provides orthogonal or nearly orthogonal Latin hypercubes with an appreciably larger number of factors as compared to those two methods. Theorem 5 shows the structure and orthogonality of the matrix L constructed by the above procedure.

Theorem 5. *For the matrix L constructed above, we have*

(i) *the matrix L is a Latin hypercube, and*

(ii) *the correlation matrix of L is given by*

$$R(L) = R(B) \otimes I_{2f},$$

where $R(B)$, given in (7), is the correlation matrix of B , I_{2f} is the identity matrix of order $2f$ and \otimes denotes Kronecker product.

Corollary 6. *If B is an orthogonal Latin hypercube, then so is L . The maximum correlation and average squared correlation of L are given by*

$$\rho_M(L) = \rho_M(B) \quad \text{and} \quad \rho_{ave}^2(L) = \frac{m-1}{2mf-1} \rho_{ave}^2(B).$$

Corollary 6 reveals that the large Latin hypercube L inherits the exact or near orthogonality of the small Latin hypercube B . As a result, the effort for searching for large orthogonal or nearly orthogonal Latin hypercube can be focused on finding small orthogonal or nearly orthogonal Latin hypercubes which are easier to obtain via some general efficient robust optimization algorithms such as simulated annealing and genetic algorithms, by using ρ_{ave}^2 or ρ_M . Example 2 below illustrates the actual construction of some orthogonal Latin hypercubes using the method of Lin, Mukerjee and Tang (2009).

Example 2. Consider constructing orthogonal Latin hypercubes of n^2 runs where n is a prime or prime power for which an $OA(n^2, n+1, n, 2)$ exists (Hedayat, Sloane and Stufken, 1999). For instance, consider $n = 5, 7, 8, 9, 11$. Now if we take B to be an $OLH(5, 2)$, an $OLH(7, 3)$, an $OLH(8, 4)$, an $OLH(9, 5)$, or an $OLH(11, 7)$, as displayed in Table 4 and take A respectively to be an $OA(25, 6, 5, 2)$, an $OA(49, 8, 7, 2)$, an $OA(64, 8, 8, 2)$, an $OA(81, 10, 9, 2)$, or an $OA(121, 12, 11, 2)$, then the construction of Lin, Mukerjee and Tang (2009) provides an $OLH(25, 12)$, an $OLH(49, 24)$, an $OLH(64, 32)$, an $OLH(81, 50)$, or an $OLH(121, 84)$, respectively.

2.4.2 A recursive method

Sun, Liu and Lin (2009) introduced second-order orthogonal Latin hypercubes defined as follows. For a design D with columns d_1, \dots, d_m , let \tilde{D} be the $n \times [m(m+1)/2]$ matrix

Table 4. Orthogonal Latin hypercubes OLH(5, 2) OLH(7, 3), OLH(8, 4), OLH(9, 5) and OLH(11, 7)

OLH(5, 2)	OLH(7, 3)	OLH(8, 4)									
1 -2	-3 3 2	0.5	-1.5	3.5	2.5						
2 1	-2 0 -3	1.5	0.5	2.5	-3.5						
0 0	-1 -2 -1	2.5	-3.5	-1.5	-0.5						
-1 2	0 -3 1	3.5	2.5	-0.5	1.5						
-2 -1	1 -1 3	-3.5	-2.5	0.5	-1.5						
	2 1 -2	-2.5	3.5	1.5	0.5						
	3 2 0	-1.5	-0.5	-2.5	3.5						
		-0.5	2.5	-3.5	-2.5						
OLH(9, 5)					OLH(11, 7)						
-4	-2	0	-3	3	-5	-4	-5	-5	-3	0	0
-3	4	2	1	-2	-4	2	-1	3	4	5	4
-2	-3	-4	-1	-3	-3	-2	4	5	-4	-2	-1
-1	3	-2	3	4	-2	3	-3	4	1	-4	-2
0	-4	4	4	0	-1	4	2	-4	3	2	-4
1	2	-1	0	-4	0	-5	5	-2	5	-3	2
2	0	3	-2	-1	1	5	3	-3	-5	-1	5
3	1	1	-4	2	2	-1	1	1	-2	3	-5
4	-1	-3	2	1	3	0	0	-1	0	1	-3
					4	1	-4	0	2	-5	1
					5	-3	-2	2	-1	4	3

whose columns consist of all possible products $d_i \odot d_j$, where \odot denotes the Hadamard product of vectors, $i = 1, \dots, m$, $j = 1, \dots, m$ and $i \leq j$. Define the correlation matrix between D and \tilde{D} to be

$$R(D, \tilde{D}) = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1q} \\ r_{21} & r_{22} & \cdots & r_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mq} \end{pmatrix}, \quad (8)$$

where $q = m(m+1)/2$ and r_{ij} is the correlation between the i th column of D and the j th column of \tilde{D} . A second-order orthogonal Latin hypercube D has the properties; (a) $R(D)$ in (5) is an identity matrix, and (b) $R(D, \tilde{D})$ in (8) is a zero matrix. The main effects of such a design are orthogonal to each other as well as to all the quadratic effects and the two-factor interactions.

Sun, Liu and Lin (2009) proposed the following procedure for constructing second-order orthogonal Latin hypercubes of $2^{c+1} + 1$ runs in 2^c factors for any integer $c \geq 1$. Throughout this section, let X^* represent the matrix obtained by switching the signs in the top half of the matrix X with an even number of rows.

Step I: For $c = 1$, let

$$S_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad T_1 = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}.$$

Step II: For an integer $c \geq 2$, define

$$S_c = \begin{pmatrix} S_{c-1} & -S_{c-1}^* \\ S_{c-1} & S_{c-1}^* \end{pmatrix} \quad \text{and} \quad T_c = \begin{pmatrix} T_{c-1} & -(T_{c-1}^* + 2^{c-1}S_{c-1}^*) \\ T_{c-1} + 2^{c-1}S_{c-1} & T_{c-1}^* \end{pmatrix}. \quad (9)$$

Step III: Obtain a $(2^{c+1} + 1) \times 2^c$ Latin hypercube L_c as

$$L_c = \begin{pmatrix} T_c \\ 0_{2^c} \\ -T_c \end{pmatrix}, \quad (10)$$

where 0_{2^c} denotes a zero row vector of length 2^c .

The following theorem captures the structure of L_c constructed above.

Theorem 7. *The L_c in (10) is a second-order orthogonal Latin hypercube.*

Theorem 7 can be shown by induction with the aid of the following lemma which also follows by induction.

Lemma 1. *(i) For any two square matrices A and B with the same even number of rows, we have $A^{*T}B^* = A^TB$. (ii) For the S_c and T_c defined in (9), we have $S_c^T S_c = S_c^{*T} S_c^* = 2^c I_{2^c}$, $S_c^T T_c + T_c^T S_c = (2^{2c} + 2^c) I_{2^c}$ and $S_c^T T_c^{*T} - T_c^T S_c^{*T} = 0$.*

Sun, Liu and Lin (2009) further constructed second-order orthogonal Latin hypercubes of 2^{c+1} runs in 2^c factors by modifying Step III in the above procedure. That is, in Step III, let $H_c = T_c - S_c/2$ and obtain L_c as

$$L_c = \begin{pmatrix} H_c \\ -H_c \end{pmatrix}.$$

Example 3. A second-order orthogonal Latin hypercube of 17 runs for 8 factors constructed

using the procedure in Sun, Liu and Lin (2009) is given by

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & -1 & -4 & 3 & 6 & -5 & -8 & 7 \\ 3 & 4 & -1 & -2 & -7 & -8 & 5 & 6 \\ 4 & -3 & 2 & -1 & -8 & 7 & -6 & 5 \\ 5 & 6 & 7 & 8 & -1 & -2 & -3 & -4 \\ 6 & -5 & -8 & 7 & -2 & 1 & 4 & -3 \\ 7 & 8 & -5 & -6 & 3 & 4 & -1 & -2 \\ 8 & -7 & 6 & -5 & 4 & -3 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -4 & -5 & -6 & -7 & -8 \\ -2 & 1 & 4 & -3 & -6 & 5 & 8 & -7 \\ -3 & -4 & 1 & 2 & 7 & 8 & -5 & -6 \\ -4 & 3 & -2 & 1 & 8 & -7 & 6 & -5 \\ -5 & -6 & -7 & -8 & 1 & 2 & 3 & 4 \\ -6 & 5 & 8 & -7 & 2 & -1 & -4 & 3 \\ -7 & -8 & 5 & 6 & -3 & -4 & 1 & 2 \\ -8 & 7 & -6 & 5 & -4 & 3 & -2 & 1 \end{pmatrix}.$$

2.4.3 Based on small orthogonal designs and small orthogonal Latin hypercubes

Let $A = (a_{ij})$ be an $n_1 \times m_1$ matrix with entries $a_{ij} = \pm 1$, $B = (b_{ij})$ be an $n_2 \times m_2$ Latin hypercube, $C = (c_{ij})$ be an $n_1 \times m_1$ Latin hypercube, and $D = (d_{ij})$ be an $n_2 \times m_2$ matrix with entries $d_{ij} = \pm 1$. Let γ be a real number. Lin et al. (2010) considered constructing designs via

$$L = A \otimes B + \gamma C \otimes D. \quad (11)$$

The resulting design L in (11) has $n = n_1 n_2$ runs and $m = m_1 m_2$ factors.

Theorem 8. *Let $\gamma = n_2$. Then design L in (11) is an orthogonal Latin hypercube if*

(i) A and D are column-orthogonal matrices of ± 1 ,

(ii) B and C are orthogonal Latin hypercubes,

(iii) at least one of the two, $A^T C = 0$ and $B^T D = 0$, is true, and

(iv) at least one of the following two conditions is true:

(a) A and C satisfy that for any i , if p and p' are such that $c_{pi} = -c_{p'i}$, then $a_{pi} = a_{p'i}$;

(b) B and D satisfy that for any j , if q and q' are such that $b_{qj} = -b_{q'j}$, then $d_{qj} = d_{q'j}$.

Theorem 8 tells us how to make L to be an orthogonal Latin hypercube. Condition (iv) in Theorem 8 is needed for L to be a Latin hypercube. To make L orthogonal, orthogonality of A , B , C and D and condition (iii) are necessary. Choices for A and D include Hadamard matrices and orthogonal arrays with levels ± 1 . A Hadamard matrix is a square orthogonal matrix of ± 1 . Because of the orthogonality of A and D , n_1 and n_2 must be equal to two or multiples of four. Theorem 8 requires designs B and C to be orthogonal Latin hypercubes. All known orthogonal Latin hypercubes of run sizes that are two or multiples of four can be used. As a result, Theorem 8 can be used to construct a vast number of orthogonal Latin hypercubes of $n = 8k$ runs. Example 4 illustrates the use of Theorem 8.

Example 4. Consider constructing an orthogonal Latin hypercube of 32 runs. Let $A = (1, 1)^T$, B be an 16×12 orthogonal Latin hypercube in Table 5, $C = (1/2, -1/2)^T$, and D be a matrix obtained by taking any 12 columns from a Hadamard matrix of order 16. By Theorem 8, L in (11) with the chosen A, B, C, D constitutes a 32×12 orthogonal Latin hypercube.

When $n_1 = n_2$, a stronger result than Theorem 8 can be established.

Table 5. A 16×12 orthogonal Latin hypercube from Steinberg and Lin (2006)

$$B = \frac{1}{2} \begin{pmatrix} -15 & 5 & 9 & -3 & 7 & 11 & -11 & 7 & -9 & 3 & -15 & 5 \\ -13 & 1 & 1 & 13 & -7 & -11 & 11 & -7 & -1 & -13 & -13 & 1 \\ -11 & 7 & -7 & -11 & 13 & -1 & -1 & -13 & 9 & -3 & 15 & -5 \\ -9 & 3 & -15 & 5 & -13 & 1 & 1 & 13 & 1 & 13 & 13 & -1 \\ -7 & -11 & 11 & -7 & 11 & -7 & 7 & 11 & 5 & 15 & -3 & -9 \\ -5 & -15 & 3 & 9 & -11 & 7 & -7 & -11 & 13 & -1 & -1 & -13 \\ -3 & -9 & -5 & -15 & 1 & 13 & 13 & -1 & -5 & -15 & 3 & 9 \\ -1 & -13 & -13 & 1 & -1 & -13 & -13 & 1 & -13 & 1 & 1 & 13 \\ 1 & 13 & 13 & -1 & -9 & 3 & -15 & 5 & 11 & -7 & 7 & 11 \\ 3 & 9 & 5 & 15 & 9 & -3 & 15 & -5 & 3 & 9 & 5 & 15 \\ 5 & 15 & -3 & -9 & -3 & -9 & -5 & -15 & -11 & 7 & -7 & -11 \\ 7 & 11 & -11 & 7 & 3 & 9 & 5 & 15 & -3 & -9 & -5 & -15 \\ 9 & -3 & 15 & -5 & -5 & -15 & 3 & 9 & -7 & -11 & 11 & -7 \\ 11 & -7 & 7 & 11 & 5 & 15 & -3 & -9 & -15 & 5 & 9 & -3 \\ 13 & -1 & -1 & -13 & -15 & 5 & 9 & -3 & 7 & 11 & -11 & 7 \\ 15 & -5 & -9 & 3 & 15 & -5 & -9 & 3 & 15 & -5 & -9 & 3 \end{pmatrix}$$

Proposition 1. *If $n_1 = n_2 = n_0$ and A, B, C, D and γ are chosen according to Theorem 8, then design (L, U) is an orthogonal Latin hypercube with $2m_1m_2$ factors, where L is as in Theorem 8 and $U = -n_0A \otimes B + C \otimes D$.*

Example 5. Consider constructing orthogonal Latin hypercubes of 64 runs. Let $n_1 = n_2 =$

8. If we take

$$A = D = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \text{ and } B = C = \frac{1}{2} \begin{pmatrix} 1 & -3 & 7 & 5 \\ 3 & 1 & 5 & -7 \\ 5 & -7 & -3 & -1 \\ 7 & 5 & -1 & 3 \\ -1 & 3 & -7 & -5 \\ -3 & -1 & -5 & 7 \\ -5 & 7 & 3 & 1 \\ -7 & -5 & 1 & -3 \end{pmatrix},$$

then design (L, U) in Proposition 1 is a 64×32 orthogonal Latin hypercube.

Theorem 9. *Suppose that an $OLH(n, m)$ is available, where n is a multiple of 4 such that a Hadamard matrix of order n exists. Then the following orthogonal Latin hypercubes, an $OLH(2n, m)$, an $OLH(4n, 2m)$, an $OLH(8n, 4m)$ and an $OLH(16n, 8m)$, can all be constructed.*

We provide a sketch of the proof for Theorem 9. The proof provides the actual construction of these orthogonal Latin hypercubes. The theorem results from an application of Theorem 8 and the use of orthogonal designs in Table 6. Note that each of the four matrices in Table 6 can be written as $(X^T, -X^T)^T$. Let $A = (S^T, S^T)^T$ with S obtained from X by letting $x_i = 1$ for all i 's, and C be an orthogonal Latin hypercube derived from a matrix in Table 6 by letting $x_i = (2i - 1)/2$ for $i = 1, \dots, n/2$. Now we choose B to be the given $OLH(n, m)$ and D be the matrix obtained by taking any m columns from a Hadamard matrix order n . Such matrices A, B, C , and D meet conditions (i), (ii), (iii) and (iv) in Theorem 8.

Thus, Theorem 9 follows by Theorem 8.

Table 6. Four orthogonal designs

n														
2	4		8				16							
x_1	x_1	x_2	x_1	$-x_2$	x_4	x_3	x_1	$-x_2$	$-x_4$	$-x_3$	$-x_8$	x_7	x_5	x_6
$-x_1$	x_2	$-x_1$	x_2	x_1	x_3	$-x_4$	x_2	x_1	$-x_3$	x_4	$-x_7$	$-x_8$	$-x_6$	x_5
	$-x_1$	$-x_2$	x_3	$-x_4$	$-x_2$	$-x_1$	x_3	$-x_4$	x_2	x_1	$-x_6$	$-x_5$	x_7	$-x_8$
	$-x_2$	x_1	x_4	x_3	$-x_1$	x_2	x_4	x_3	x_1	$-x_2$	$-x_5$	x_6	$-x_8$	$-x_7$
			$-x_1$	x_2	$-x_4$	$-x_3$	x_5	$-x_6$	$-x_8$	x_7	x_4	x_3	$-x_1$	$-x_2$
			$-x_2$	$-x_1$	$-x_3$	x_4	x_6	x_5	$-x_7$	$-x_8$	x_3	$-x_4$	x_2	$-x_1$
			$-x_3$	x_4	x_2	x_1	x_7	$-x_8$	x_6	$-x_5$	x_2	$-x_1$	$-x_3$	x_4
			$-x_4$	$-x_3$	x_1	$-x_2$	x_8	x_7	x_5	x_6	x_1	x_2	x_4	x_3
							$-x_1$	x_2	x_4	x_3	x_8	$-x_7$	$-x_5$	$-x_6$
							$-x_2$	$-x_1$	x_3	$-x_4$	x_7	x_8	x_6	$-x_5$
							$-x_3$	x_4	$-x_2$	$-x_1$	x_6	x_5	$-x_7$	x_8
							$-x_4$	$-x_3$	$-x_1$	x_2	x_5	$-x_6$	x_8	x_7
							$-x_5$	x_6	x_8	$-x_7$	$-x_4$	$-x_3$	x_1	x_2
							$-x_6$	$-x_5$	x_7	x_8	$-x_3$	x_4	$-x_2$	x_1
							$-x_7$	x_8	$-x_6$	x_5	$-x_2$	x_1	x_3	$-x_4$
							$-x_8$	$-x_7$	$-x_5$	$-x_6$	$-x_1$	$-x_2$	$-x_4$	$-x_3$

Theorem 9 is a very powerful result. By repeatedly applying Theorem 9, one can obtain many infinite series of orthogonal Latin hypercubes. For example, starting with an OLH(12, 6), we can obtain an OLH(192, 48), which can be used in turn to construct an OLH(768, 96) and so on. For another example, an OLH(256, 248) in Steinberg and Lin

(2006) can be used to construct an OLH(1024, 496), an OLH(4096, 1984) and so on.

Another result from Lin et al. (2010) shows how the method in (11) can be used to construct nearly orthogonal Latin hypercubes.

Proposition 2. *Suppose that A, B, C, D and γ in (11) are chosen so that design L in (11) is a Latin hypercube. In addition, we assume that A and D are orthogonal and that at least one of the two, $A^T C = 0$ and $B^T D = 0$, holds true. We then have that*

$$(i) \rho_{ave}^2(L) = w_1 \rho_{ave}^2(B) + w_2 \rho_{ave}^2(C), \text{ and}$$

$$(ii) \rho_M(L) = \text{Max}\{w_3 \rho_M(B), w_4 \rho_M(C)\},$$

where w_1, w_2, w_3 and w_4 are given by $w_1 = (m_2 - 1)(n_2^2 - 1)^2 / [(m_1 m_2 - 1)(n^2 - 1)^2]$, $w_2 = n_2^4 (m_1 - 1)(n_1^2 - 1)^2 / [(m_1 m_2 - 1)(n^2 - 1)^2]$, $w_3 = (n_2^2 - 1) / (n^2 - 1)$ and $w_4 = n_2^2 (n_1^2 - 1) / (n^2 - 1)$.

Proposition 2 can be shown by the definition of ρ_M and ρ_{ave}^2 . Proposition 2 says that if B and C are nearly orthogonal, the resulting Latin hypercube L is also nearly orthogonal. An example, illustrating the use of this result, is considered below.

Example 6. Let $A = (1, 1)^T$, $C = (1/2, -1/2)^T$, and $\gamma = 16$. Choose a 16×15 nearly orthogonal Latin hypercube $B = B_0/2$ where B_0 is displayed in Table 7, and B has $\rho_{ave}^2(B) = 0.0003$ and $\rho_M(B) = 0.0765$. Taking any 15 columns of a Hadamard matrix of order 16 to be D and then applying (11), we obtain a Latin hypercube L of 32 runs and 15 factors. As $\rho_{ave}^2(C) = \rho_M(C) = 0$, we have $\rho_{ave}^2(L) = (n_2^2 - 1)^2 \rho_{ave}^2(B) / (n^2 - 1)^2 = 0.00002$ and $\rho_M(L) = (n_2^2 - 1) \rho_M(B) / (n^2 - 1) = 0.0191$.

Table 7. Design matrix of B_0 in Example 6

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

2.4.4 Existence of orthogonal Latin hypercubes

An important problem in the study of orthogonal Latin hypercubes is to determine the maximum number m^* of columns in an orthogonal Latin hypercube of a given run size n . Theorem 4 tells us that $m^* = 1$ if n is 3 or $n = 4k + 2$ for any non-negative integer k and $m^* \geq 2$ otherwise. Lin et al. (2010) obtained a stronger result.

Proposition 3. *The maximum number m^* of factors for an orthogonal Latin hypercube of $n = 16k + j$ runs has a lower bound given below:*

(i) $m^* \geq 6$ for all $n = 16k + j$ where $k \geq 1$ and $j \neq 2, 6, 10, 14$;

(ii) $m^* \geq 7$ for $n = 16k + 11$ where $k \geq 0$;

(iii) $m^* \geq 12$ for $n = 16k, 16k + 1$ where $k \geq 2$;

(iv) $m^* \geq 24$ for $n = 32k, 32k + 1$ where $k \geq 2$;

(v) $m^* \geq 48$ for $n = 64k, 64k + 1$ where $k \geq 2$.

We summarize the results on the maximum number m^* in $\text{OLH}(n, m^*)$ provided by all existing approaches. Table 8 lists the maximum number m^* in $\text{OLH}(n, m^*)$ for $n \leq 24$. These values except the case $n = 16$ were obtained by Lin (2008) through an algorithm. For $n = 16$, Steinberg and Lin (2006) obtained an orthogonal Latin hypercube with 12 columns. Table 9 reports the maximum number m^* in $\text{OLH}(n, m^*)$ for $24 < n \leq 256$ as well as the corresponding approach for achieving m^* . Note that the m^* values in Tables 8 and 9 are in fact the best lower bounds on the true m^* values.

Table 8. The maximum number m^* of factors in $\text{OLH}(n, m^*)$ for $n \leq 24$

n	4	5	7	8	9	11	12	13	15	16	17	19	20	21	23	24
m^*	2	2	3	4	5	7	6	6	6	12 ^(a)	6	6	6	6	6	6

(a): Steinberg and Lin (2006)

Table 9. The maximum number m^* of factors in $\text{OLH}(n, m^*)$

n	m^*	Reference	n	m^*	Reference
25	12	Lin, Mukerjee and Tang (2009)	144	24	Lin et al. (2010)
32	16	Sun, Lin and Lin (2009)	145	12	Lin et al. (2010)
33	16	Sun, Lin and Lin (2009)	160	24	Lin et al. (2010)
48	12	Lin et al. (2010)	161	24	Lin et al. (2010)
49	24	Lin, Mukerjee and Tang (2009)	169	84	Lin, Mukerjee and Tang (2009)
64	32	Sun, Lin and Lin (2009)	176	12	Lin et al. (2010)
65	32	Sun, Lin and Lin (2009)	177	12	Lin et al. (2010)
80	12	Lin et al. (2010)	192	48	Lin et al. (2010)
81	50	Lin, Mukerjee and Tang (2009)	193	48	Lin et al. (2010)
96	24	Lin et al. (2010)	208	12	Lin et al. (2010)
97	24	Lin et al. (2010)	209	12	Lin et al. (2010)
112	12	Lin et al. (2010)	224	24	Lin et al. (2010)
113	12	Lin et al. (2010)	225	24	Lin et al. (2010)
121	84	Lin, Mukerjee and Tang (2009)	240	12	Lin et al. (2010)
128	64	Sun, Lin and Lin (2009)	241	12	Lin et al. (2010)
129	64	Sun, Lin and Lin (2009)	256	248	Steinberg and Lin (2006)

3 Other Space-filling Designs

3.1 Orthogonal designs with many levels

Bingham, Sitter and Tang (2009) proposed the use of orthogonal and nearly orthogonal designs for computer experiments. The key motivation is that designs with many levels are desirable but it is absolutely unnecessary to require the number of levels equal to the run sizes. They introduced a rich class of orthogonal designs, including two-level orthogonal designs and orthogonal Latin hypercubes as special cases.

Consider designs of n runs for m factors each of s levels, where $2 \leq s \leq n$. Without loss of generality, the s levels are chosen to be $-(s-1)/2, \dots, (s-1)/2$. Such a design is denoted by $D(n, s^m)$ and can be represented by an $n \times m$ matrix $D = (d_{ij})$ with entries from the above set of s levels. A Latin hypercube of n runs for m factors is a $D(n, s^m)$ with $n = s$.

Let $A = (a_{ij})$ be an $n_1 \times m_1$ matrix with entries $a_{ij} = \pm 1$ and D_0 be a $D(n_2, s^{m_2})$. Bingham, Sitter and Tang (2009) constructed design

$$D = A \otimes D_0. \tag{12}$$

Proposition 4. *Let A be column-orthogonal. Then design D in (12) is orthogonal if and only if D_0 is orthogonal.*

Proposition 4 provides a powerful way to construct a rich class of orthogonal designs for computer experiments. Example 7 illustrates the power of this method.

Example 7. Let D_0 be the OLH(16, 12) constructed by Steinberg and Lin (2006). The construction method in (12) gives a series of orthogonal designs of $16k$ runs for $12k$ factors, where k is an integer such that a Hadamard matrix of order k exists.

Higher order orthogonality and near orthogonality of D in (12) are also discussed in Bingham, Sitter and Tang (2009). When projected onto some pairs of columns, design points in D in (12) lie on the diagonals. To eliminate this undesirable pattern, Bingham, Sitter and Tang (2009) provides a generalization of the method in (12) to construct designs with better projection properties. The generalization works as follows.

Let D_j be a $D(n_2, s^{m_2})$, for each $j = 1, \dots, m_1$. They considered the following generalization

$$D = (a_{ij}D_j) = \begin{bmatrix} a_{11}D_1 & a_{12}D_2 & \dots & a_{1m_1}D_{m_1} \\ a_{21}D_1 & a_{22}D_2 & \dots & a_{2m_1}D_{m_1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_11}D_1 & a_{n_12}D_2 & \dots & a_{n_1m_1}D_{m_1} \end{bmatrix}. \quad (13)$$

The following results study the orthogonality of design D in (13).

Proposition 5. *Let A be column-orthogonal. We have that*

- (i) $\rho_M = \max\{\rho_M(D_1), \dots, \rho_M(D_{m_1})\}$,
- (ii) $\rho_{ave}^2(D) = w\{\rho_{ave}^2(D_1) + \dots, \rho_{ave}^2(D_{m_1})\}/m_1$, where $w = (m_2 - 1)/(m_1m_2 - 1)$, and
- (iii) D in (13) is orthogonal if and only if D_1, \dots, D_{m_1} are all orthogonal.

This line of research was further pursued by Georgiou (2010).

3.2 Low-discrepancy sequences and uniform designs

This section reviews space-filling designs originated from the field of numerical analysis. Many problems in various fields such as quantum physics and computational finance require

calculating definite integrals of a function over a multi-dimensional unit cube. It is very common that the function may be so complicated that the integral cannot be obtained analytically and precisely. Many numerical methods of approximating the integral have been developed.

Suppose we wish to compute an integral of a function f in an s -dimensional unit cube $\chi = [0, 1]^s$,

$$I(f) = \int_{\chi} f(x)dx. \quad (14)$$

Note that respecting the common notations, we use s to denote the number of input factors in this section. Let $\mathcal{P} = (x_1, \dots, x_n)$ be a set of n points in χ . One approximation to $I(f)$ in (14) is

$$\hat{I}(f, \mathcal{P}) = \frac{1}{n} \sum_{i=1}^n f(x_i).$$

The bound of the integration error is given by Koksma-Hlawka inequality,

$$\left| I(f) - \hat{I}(f, \mathcal{P}) \right| \leq V(f)D^*(\mathcal{P}), \quad (15)$$

where $V(f)$ is the variation of f in the sense of Hardy and Krause and $D^*(\mathcal{P})$ is the star discrepancy of the n points \mathcal{P} (Weyl, 1916). Motivated by the fact that the Koksma-Hlawka bound in (15) is proportional to the star discrepancy of the points, different methods for generating points in χ with as small a star discrepancy as possible have been proposed. Such methods are referred to as quasi-Monte Carlo methods (Niederreiter, 1992).

For each $x = (x_1, \dots, x_s)$ in χ , let $[0, x)$ denote the interval $[0, x_1) \times \dots \times [0, x_s)$, $N(\mathcal{P}, [0, x))$ denote the number of points of \mathcal{P} falling in $[0, x)$, and $\text{Vol}(J)$ be the volume of

interval J . The *star discrepancy* $D^*(\mathcal{P})$ of \mathcal{P} is defined by

$$D^*(\mathcal{P}) = \max_{x \in \chi} \left| \frac{N(\mathcal{P}, [0, x))}{n} - \text{Vol}([0, x)) \right|. \quad (16)$$

A sequence S of points in χ is called a *low-discrepancy sequence* if its first n points have

$$D^*(\mathcal{P}) = O(n^{-1}(\log n)^s), \quad \text{for all } n \geq 2.$$

By contrast, if the set \mathcal{P} is chosen by the Monte Carlo method, that is, x_1, \dots, x_n are independent random samples from the uniform distribution, then $D^*(\mathcal{P}) = O(n^{-1/2})$.

Construction of low-discrepancy sequences is a very active research area in the study of quasi-Monte Carlo methods. There are many constructions available, such as the good lattice point method, the good point method, Halton sequences, Faure sequences, (t, m, s) -nets, and (t, s) -sequences. Here we provide a brief review of (t, s) -sequences and related uniform designs. For a comprehensive treatment of low-discrepancy sequences, see Niederreiter (1992).

3.2.1 (t, m, s) -nets and (t, s) -sequences

The definition of (t, m, s) -nets and (t, s) -sequences requires a concept of elementary intervals.

An *elementary interval* in base b is an interval E in $[0, 1)^s$ of the form

$$E = \prod_{i=1}^s \left[\frac{a_i}{b^{d_i}}, \frac{a_i + 1}{b^{d_i}} \right) \quad (17)$$

with integers a_i and d_i satisfying $d_i \geq 0$ and $0 \leq a_i < b^{d_i}$. For $i = 1, \dots, s$, the i th axis of an elementary interval has length b^{-d_i} and thus an elementary interval has a volume $b^{-\sum_{i=1}^s d_i}$.

For integers $b \geq 2$ and $0 \leq t \leq m$, a (t, m, s) -net in base b is a set of b^m points in $[0, 1)^s$ such that every elementary interval in base b of volume b^{t-m} contains exactly b^t points.

Note that for given values of b , m and s , a smaller value of t leads to a smaller elementary interval, and thus a set of points with better uniformity. Consequently, a smaller value of t in (t, m, s) -nets in base b is preferred.

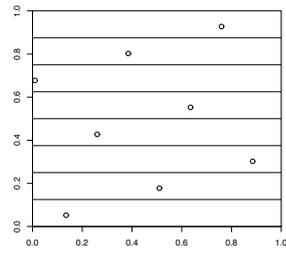
An infinite sequence of points $\{x_n\}$ in $[0, 1]^s$ is a (t, s) -sequence in base b if for all $k \geq 0$ and $m > t$, the finite sequence $x_{kb^{m+1}}, \dots, x_{(k+1)b^m}$ forms a (t, m, s) -net in base b . Example 8 illustrates both concepts.

Example 8. Consider a $(0, 2)$ -sequence in base 2. Its first 2^3 points form a $(0, 3, 2)$ -net in base 2 and are displayed in Figure 5. There are four types of elementary intervals in base 2 of volume 2^{-3} with (d_1, d_2) 's in (17) being $(0, 3)$, $(3, 0)$, $(1, 2)$, and $(2, 1)$. Figures 5(a) - 5(d) show a $(0, 3, 2)$ -net in base 2 when elementary intervals are given by $(d_1, d_2) = (0, 3)$, $(d_1, d_2) = (3, 0)$, $(d_1, d_2) = (1, 2)$, and $(d_1, d_2) = (2, 1)$, respectively. Note that in every elementary interval of the form

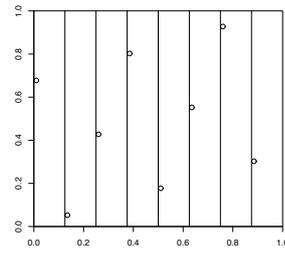
$$\left[\frac{a_1}{2^{d_1}}, \frac{(a_1 + 1)}{2^{d_1}} \right) \times \left[\frac{a_2}{2^{d_2}}, \frac{(a_2 + 1)}{2^{d_2}} \right), \quad 0 \leq a_i < 2^{d_i}, \quad i = 1, 2,$$

there is exactly one point. Now consider adding another 8 points to the $(0, 3, 2)$ -net in base 2. The newly added 8 points form a $(0, 3, 2)$ -net in base 2. Moreover, the resulting 16 points is a $(0, 4, 2)$ -net in base 2. Analogous to Figure 5, Figure 6 exhibits the $(0, 4, 2)$ -net in base 2 when elementary intervals are given by all (d_1, d_2) 's that satisfy $d_1 + d_2 = m = 4$.

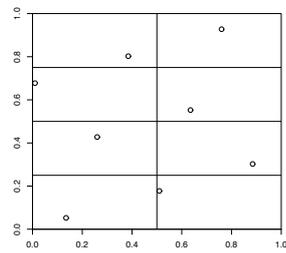
A general theory of (t, m, s) -nets and (t, s) -sequences was developed by Niederreiter (1987). Some special cases of (t, s) -sequences are as follows. Sobol' sequences (Sobol', 1967) are (t, s) -sequences in base 2. Faure sequences (Faure, 1982) are $(0, s)$ -sequences in base q where q is a prime with $s \leq q$. Niederreiter sequences (Niederreiter, 1987) are $(0, s)$



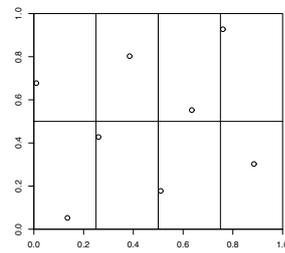
(a) $d_1 = 0, d_2 = 3$



(b) $d_1 = 3, d_2 = 0$

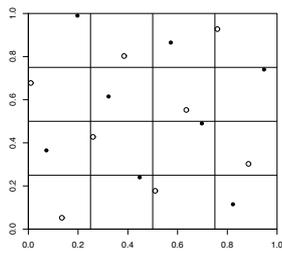


(c) $d_1 = 1, d_2 = 2$

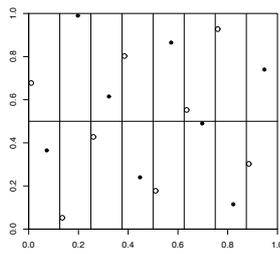


(d) $d_1 = 2, d_2 = 1$

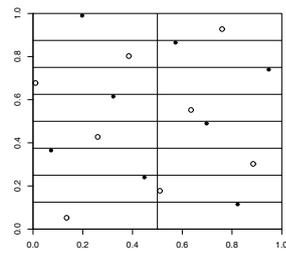
Figure 5. A $(0, 3, 2)$ -net in base 2 seen using four types of elementary intervals



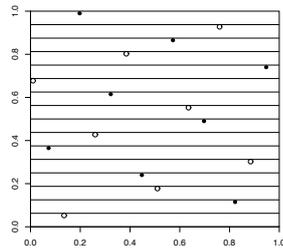
(a) $d_1 = 2, d_2 = 2$



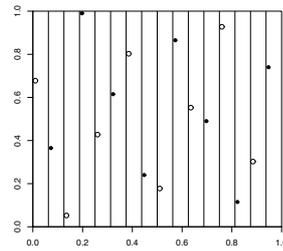
(b) $d_1 = 3, d_2 = 1$



(c) $d_1 = 1, d_2 = 3$



(d) $d_1 = 4, d_2 = 0$



(e) $d_1 = 0, d_2 = 4$

Figure 6. A $(0, 4, 2)$ -net in base 2 seen using five types of elementary intervals, the first and second 8 points are represented by \circ and \bullet

sequence in base q where q is a prime or a prime power with $s \leq q$. Niederreiter-Xing sequences (Niederreiter and Xing, 1996) are (t, s) -sequences in base q for some certain t where q is a prime or a prime power with $s > q$. For constructions of all these sequences, we refer the readers to Niederreiter (2008). Results on existing (t, s) -sequences are available at <http://mint.sbg.ac.at>.

Related to designs for computer experiments, an important problem in the study of low-discrepancy sequences is their actual space-filling property. Wang and Sloan (2008) investigated the uniformity of Sobol' sequences in high dimensions. The empirical examination indicates that Sobol' sequences have no better two and higher dimensional projection uniformity on average than Latin hypercube designs and random designs for moderate values of run sizes.

3.2.2 Uniform designs

Motivated by the Koksma-Hlawka inequality in (15), Fang and Wang (Fang, 1980; Wang and Fang, 1981) introduced uniform designs, and by their definition, a uniform design is a set of design points with the smallest discrepancy among all possible designs of the same run size. One natural choice of discrepancy is the star discrepancy in (16). More generally, one can use the L_p discrepancy,

$$D_p(\mathcal{P}) = \left[\int_{\chi} \left| \frac{N(\mathcal{P}, x)}{n} - \text{Vol}([0, x]) \right|^p dx \right]^{1/p},$$

where $N(\mathcal{P}, x)$ and $\text{Vol}([0, x])$ are defined as in (16). Two special cases of the L_p discrepancy are the L_∞ discrepancy, which is the star discrepancy, and the L_2 discrepancy. The L_∞ discrepancy is difficult to compute but the L_2 discrepancy is much easier to calculate

numerically because of a simple formula given by Warnock (1972),

$$D_2(\mathcal{P}) = 2^{-s} - \frac{2^{1-s}}{n} \sum_{i=1}^n \prod_{k=1}^s (1 - x_{ik}^2) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \prod_{k=1}^s [1 - \max(x_{ik}, x_{jk})],$$

where x_{ik} is the setting of the k th factor in the i th run, $i = 1, \dots, n$ and $k = 1, \dots, s$.

The L_p discrepancy aims to achieve uniformity in the whole design space. Designs with the smallest L_p discrepancy do not necessarily give good projection uniformity in low dimensions. Hickernell (1998) proposed three new measures of uniformity, the symmetric L_2 discrepancy (SL_2), the centered L_2 discrepancy (CL_2), and the modified L_2 discrepancy (ML_2). They are all defined through

$$D_{mod}(\mathcal{P}) = \sum_{u \neq \emptyset} \int_{\chi_u} \left| \frac{N(\mathcal{P}_u, J_{x_u})}{n} - \text{Vol}(J_{x_u}) \right|^2 du, \quad (18)$$

where \emptyset represents the empty set, u is a non-empty subset of the set $\{1, \dots, s\}$, $|u|$ denotes the cardinality of u , χ_u is the $|u|$ -dimensional unit cube involving the coordinates in u , \mathcal{P}_u is the projection of \mathcal{P} on χ_u , J_x is a rectangle uniquely determined by x , and J_{x_u} is the projection of J_x on χ_u . The symmetric L_2 discrepancy chooses J_x such that it is invariant if x_{ik} is placed by $1 - x_{ik}$, $i = 1, \dots, n$ and $k = 1, \dots, s$. The symmetric L_2 discrepancy has the formula

$$(SL_2(\mathcal{P}))^2 = \left(\frac{4}{3}\right)^s - \frac{2}{n} \sum_{i=1}^n \prod_{k=1}^s (1 + 2x_{ik} - 2x_{ik}^2) + \frac{2^s}{n^2} \sum_{i=1}^n \sum_{j=1}^n \prod_{k=1}^s [1 - |x_{ik} - x_{jk}|].$$

The center L_2 discrepancy chooses J_x such that it is invariant under the reflections of \mathcal{P} around any hyperplane with the k th coordinate being 0.5. Divide the unit cube χ into 2^s cells. Each cell contains one vertex of χ . Let $a = (a_1, \dots, a_s)$ be the vertex corresponding to the cell in which x falls. The centered L_2 discrepancy takes J_x in (18) to be

$$\{d = (d_1, \dots, d_s) \in \chi | \min(a_j, x_j) \leq d_j < \max(a_j, x_j), j = 1, \dots, s\}.$$

The formula for the centered L_2 discrepancy is given by

$$\begin{aligned} (CL_2(\mathcal{P}))^2 &= \left(\frac{13}{12}\right)^2 - \frac{2}{n} \sum_{i=1}^n \prod_{k=1}^s \left(1 + \frac{1}{2}|x_{ik} - 0.5| - \frac{1}{2}|x_{ik} - 0.5|^2\right) \\ &\quad + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \prod_{k=1}^s \left(1 + \frac{1}{2}|x_{ik} - 0.5| + \frac{1}{2}|x_{jk} - 0.5| - \frac{1}{2}|x_{ik} - x_{jk}|\right). \end{aligned}$$

The modified L_2 discrepancy takes $J_x = [0, x)$ and has the formula

$$(ML_2(\mathcal{P}))^2 = \left(\frac{4}{3}\right)^s - \frac{2^{1-s}}{n} \sum_{i=1}^n \prod_{k=1}^s (3 - x_{ik}^2) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \prod_{k=1}^s [2 - \max(x_{ik}, x_{jk})].$$

For other discrepancy measures such as the wrap-around discrepancy, see Fang, Lin and Sudjianto (2006).

Finding uniform designs based on a discrepancy criterion is an optimization problem. However, searching uniform designs in the entire unit cube is computationally prohibitive for large designs. Instead, uniform designs are found within a class of designs called *U-type designs*. Suppose that each of the s factors in an experiment has q levels, $\{1, \dots, q\}$. A U-type design, denoted by $U(n; q^s)$, is an $n \times s$ matrix in which the q levels in each column appear equally often. Table 10 displays a $U(6; 3^2)$ and a $U(6, 6^2)$. For $q = n$, uniform designs based on U-type designs are constructed by several methods such as the good lattice method, the Latin square method, the expanding orthogonal array method and the cutting method. For general values of q , optimization algorithms have been considered, such as simulated annealing, genetic algorithm, and threshold accepting. Tables of uniform designs are available on the website <http://www.math.hkbu.edu.hk/UniformDesign>. For more detailed reviews of theory and applications of uniform designs, see Fang et al. (2000) and Fang and Lin (2003).

Table 10. Uniform designs $U(6; 3^2)$ and $U(6; 6^2)$

$U(6; 3^2)$	$U(6; 6^2)$
1 1	1 3
2 2	2 5
3 3	3 1
1 3	4 6
2 1	5 2
3 2	6 4

4 Concluding Remarks

Space-filling designs are those with their design points uniformly scattered in the design space. They are the best choices when the underlying relationship between the inputs and responses is very complex. Latin hypercube designs are a popular class of space-filling designs with guaranteed one-dimensional uniformity. To enhance multi-dimensional uniformity and overcome the curse of dimensionality, maximin Latin hypercubes, orthogonal array-based Latin hypercubes, and orthogonal and nearly orthogonal Latin hypercubes, are considered, and their properties and constructions are reviewed. Other space-filling designs such as orthogonal designs, low-discrepancy nets and sequences and uniform designs are also discussed. Due to limited space we do not give applications of space-filling designs to computer experiments. Interested readers can refer to Santner, Williams and Notz (2003) and Fang, Li and Sudjianto (2006).

References

- Audze, P. and Eglais, V. (1977). New approach to planning out of experiments. *Problems of Dynamics and Strength* **35**, 104–107.
- Bingham, D., Sitter, R. R. and Tang, B. (2009). Orthogonal and nearly orthogonal designs for computer experiments. *Biometrika* **96**, 51.
- Carnell, R. (2009). Package lhs in R .
- Cioppa, T. M. and Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* **49**, 45–55.
- Conway, J. H., Sloane, N. J. A. and Bannai, E. (1999). *Sphere Packings, Lattices, and Groups*, volume 290. Springer Verlag.
- Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research* **46**, 48–60.
- Fang, K. T. (1980). The uniform design: application of number-theoretic methods in experimental design. *Acta Mathematicae Applicatae Sinica* **3**, 363–372.
- Fang, K. T., Li, R. and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*, volume 6. CRC Press.
- Fang, K. T. and Lin, D. K. J. (2003). Uniform experimental designs and their applications in industry. *Handbook of Statistics* **22**, 131–170.

- Fang, K. T., Lin, D. K. J., Winker, P. and Zhang, Y. (2000). Uniform design: theory and application. *Technometrics* **42**, 237–248.
- Faure, H. (1982). Discrépance de suites associées a un systeme de numération (en dimension s). *Acta Arith* **41**, 337–351.
- Georgiou, S. D. (2009). Orthogonal Latin hypercube designs from generalized orthogonal designs. *Journal of Statistical Planning and Inference* **139**, 1530–1540.
- Georgiou, S. D. (2010). Orthogonal designs for computer experiments. *Journal of Statistical Planning and Inference* **141**, 1519–1525.
- Grosso, A., Jamali, A. and Locatelli, M. (2009). Finding maximin Latin hypercube designs by iterated local search heuristics. *European Journal of Operational Research* **197**, 541–547.
- Hedayat, A., Sloane, N. J. A. and Stufken, J. (1999). *Orthogonal Arrays: Theory and Applications*. Springer Verlag.
- Hickernell, F. J. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of Computation* **67**, 299–322.
- Iman, R. L. and Conover, W. J. (1982). Distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics - Simulation and Computation* **11**, 311–334.
- Jin, R., Chen, W. and Sudjianto, A. (2005). An efficient algorithm for constructing optimal

- design of computer experiments. *Journal of Statistical Planning and Inference* **134**, 268–287.
- Johnson, M. E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26**, 131–148.
- Joseph, V. R. and Hung, Y. (2008). Orthogonal-maximin Latin hypercube designs. *Statistica Sinica* **18**, 171–186.
- Kuhfeld, W. F. (2009). Orthogonal arrays. *Website courtesy of SAS Institute* .
- Leary, S., Bhaskar, A. and Keane, A. (2003). Optimal orthogonal-array-based Latin hypercubes. *Journal of Applied Statistics* **30**, 585–598.
- Li, W. W. and Wu, C. F. J. (1997). Columnwise-pairwise algorithms with applications to the construction of supersaturated designs. *Technometrics* **39**, 171–179.
- Liefvendahl, M. and Stocki, R. (2006). A study on algorithms for optimization of Latin hypercubes. *Journal of Statistical Planning and Inference* **136**, 3231–3247.
- Lin, C. D. (2008). *New Development in Designs for Computer Experiments and Physical Experiments*. Ph.D. thesis, Simon Fraser University.
- Lin, C. D., Bingham, D., Sitter, R. R. and Tang, B. (2010). A new and flexible method for constructing designs for computer experiments. *The Annals of Statistics* **38**, 1460–1477.
- Lin, C. D., Mukerjee, R. and Tang, B. (2009). Construction of orthogonal and nearly orthogonal Latin hypercubes. *Biometrika* **96**, 243.

- 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.
- Melissen, J. B. M. (1987). *Packing and Covering with Circles*. Ph.D. thesis, Utrecht University, The Netherlands.
- Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computer experiments. *Journal of Statistical Planning and Inference* **43**, 381–402.
- Mukerjee, R. and Wu, C. F. (2006). *A Modern Theory of Factorial Designs*. Springer Verlag.
- Niederreiter, H. (1987). Point sets and sequences with small discrepancy. *Monatshefte für Mathematik* **104**, 273–337.
- Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM CBMS-NSF Regional Conference Series in Applied Mathematics. Philadelphia.
- Niederreiter, H. (2008). Nets,(t, s)-sequences, and codes. *Monte Carlo and Quasi-Monte Carlo Methods 2006* **1**, 83–100.
- Niederreiter, H. and Xing, C. (1996). Low-discrepancy sequences and global function fields with many rational places. *Finite Fields and Their Applications* **2**, 241–273.
- Owen, A. B. (1992). A central limit theorem for Latin hypercube sampling. *Journal of the Royal Statistical Society. Series B (Methodological)* **54**, 541–551.
- Owen, A. B. (1994). Controlling correlations in Latin hypercube samples. *Journal of the American Statistical Association* **89**, 1517–1522.

- Pang, F., Liu, M. Q. and Lin, D. K. J. (2009). A construction method for orthogonal Latin hypercube designs with prime power levels. *Statistica Sinica* **19**, 1721–1728.
- Patterson, H. D. (1954). The errors of lattice sampling. *Journal of the Royal Statistical Society. Series B (Methodological)* **16**, 140–149.
- 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., Williams, B. J. and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer Verlag.
- Shewry, M. C. and Wynn, H. P. (1987). Maximum entropy sampling. *Journal of Applied Statistics* **14**, 165–170.
- Sobol', I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* **7**, 784–802.
- Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics* **29**, 143–151.
- Steinberg, D. M. and Lin, D. K. J. (2006). A construction method for orthogonal Latin hypercube designs. *Biometrika* **93**, 279.
- Sun, F., Liu, M. Q. and Lin, D. K. J. (2009). Construction of orthogonal Latin hypercube designs. *Biometrika* **96**, 971.
- Sun, F., Liu, M. Q. and Lin, D. K. J. (2010). Construction of orthogonal Latin hypercube

- designs with flexible run sizes. *Journal of Statistical Planning and Inference* **140**, 3236–3242.
- Tang, B. (1993). Orthogonal array-based Latin hypercubes. *Journal of the American Statistical Association* **88**, 1392–1397.
- Tang, B. (1994). A theorem for selecting OA-based Latin hypercubes using a distance criterion. *Communications in Statistics - Theory and Methods* **23**, 2047–2058.
- Tang, B. (1998). Selecting Latin hypercubes using correlation criteria. *Statistica Sinica* **8**, 965–978.
- van Dam, E. R., Husslage, B., den Hertog, D. and Melissen, H. (2007). Maximin Latin hypercube designs in two dimensions. *Operations Research* **55**, 158–169.
- 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.
- Wang, J. C. and Wu, C. F. J. (1991). An approach to the construction of asymmetrical orthogonal arrays. *Journal of the American Statistical Association* **86**, 450–456.
- Wang, X. and Sloan, I. H. (2008). Low discrepancy sequences in high dimensions: How well are their projections distributed? *Journal of Computational and Applied Mathematics* **213**, 366–386.
- Wang, Y. and Fang, K. T. (1981). A note on uniform distribution and experimental design. *KeXue TongBao* **26**, 485–489.

- Warnock, T. T. (1972). Computational investigations of low-discrepancy point sets. *Applications of Number Theory to Numerical Analysis* , 319–343.
- Weyl, H. (1916). Über die gleichverteilung von zahlen mod. eins. *Mathematische Annalen* **77**, 313–352.
- Wu, C. F. J. and Hamada, M. S. (2011). *Experiments: Planning, Analysis, and Optimization*. Wiley Series in Probability and Statistics. John Wiley & Sons.
- Yang, J. and Liu, M. Q. (2012). Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs. *Statistica Sinica* , In Press.
- Ye, K. Q. (1998). Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association* **93**, 1430–1439.
- Ye, K. Q., Li, W. and Sudjianto, A. (2000). Algorithmic construction of optimal symmetric Latin hypercube designs. *Journal of Statistical Planning and Inference* **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.