1999

Uniform designs based on Latin squares

Kai-Tai Fang
Hong Kong Baptist University

Wai Chee Shiu
Hong Kong Baptist University, wcshiu@hkbu.edu.hk

Jian-Xin Pan
Hong Kong Baptist University

Statistica Sinica **9**(1999), 905-912

UNIFORM DESIGNS BASED ON LATIN SQUARES

Kai-Tai Fang, Wai-Chee Shiu and Jian-Xin Pan

Hong Kong Baptist University

Abstract: Uniform designs (UDs) have been widely used in various fields. Traditionally they were generated by so-called good lattice point (*glp*) sets. In this paper, we propose a new approach based on cyclic Latin squares (CLS). The UDs generated have much smaller discrepancies than those proposed by Wang and Fang (1981). A threshold accepting algorithm is employed for finding the CLS with smallest L_2-discrepancy.

Key words and phrases: Experimental design, Latin squares, L_2-discrepancy, threshold accepting algorithm, uniform design, uniformity.

1. Introduction

The uniform designs (UD) proposed by Fang and Wang (Fang (1980), Wang and Fang (1981)) scatter points uniformly over the experimental domain. They have the advantage of providing a good representation of the experimental domain with fewer runs. In China, such designs have been successfully applied to many experiments in diverse fields. Computer experiments (Sacks, Welch, Mitchell and Wynn (1989)), which have attracted considerable attention in recent years, also call for "space filling" designs which fill the experimental domain in a uniform fashion (Bates, Buck, Riccomagno and Wynn (1996)).

Suppose there are s factors, and n runs are to be conducted. Without loss of generality, assume that the experimental domain is the unit cube $C^s = [0,1]^s$; then a design consists of n points in C^s. We want these points to be uniformly scattered over C^s. A measure of uniformity called *discrepancy* (defined in Section 2) is adopted, and the goal is to choose n points with the smallest discrepancy. The resulting design is called a *uniform design* (UD). Obviously it is impractical to have a complete search, and we shall restrict the search to a subset of candidate designs.

Definition 1. A U-type design of size $n \times s$, denoted by $U_{n,s} = (u_{ij})$, is an $n \times s$ ($s \leq n$) matrix with rank s such that each column is a permutation of $\{1, \ldots, n\}$. Its induced matrix is $X_{n,s} = (x_{ij})$, where $x_{ij} = (u_{ij} - 0.5)/n$, for $i = 1, \ldots, n$ and $j = 1, \ldots, s$.

There is a one-to-one correspondence between $U_{n,s}$ and $X_{n,s}$ so that they can be used interchangeably. The matrix $X_{n,s}$ can be considered as n points in

C^s. A $U_{n,s}$ with induced matrix $X_{n,s}$ having the smallest discrepancy is called a *U-uniform design*.

Two U-type designs are called equivalent if they can be obtained from each other by permuting rows and/or columns. Clearly, *any criterion of uniformity should be invariant under permutations of rows or columns of a U-type design*. Therefore, the first column of a U-uniform design can always be taken as $(1, \ldots, n)'$. There are $n! - 1$ possible permutations for the second column, $n! - 2$ choices for the third column, and so on. Even for moderate n and s, the search of the best s columns could lead to intolerable computing time.

To reduce computing cost, Wang and Fang (1981) suggested the use of good lattice point (*glp*) sets. For given n, let $\{h_1, \ldots, h_q\}$ be the set of all positive integers such that the greatest common divisor $(n, h_i) = 1$ and $h_i \leq n$, $i = 1, \ldots, q$, where q is the Euler function $\phi(n)$, i.e., $q = \phi(n) = n(1 - 1/p_1) \cdots (1 - 1/p_r)$ where $n = p_1^{t_1} \cdots p_r^{t_r}$ is the prime decomposition of n. Define $G_{n,q} = (u_{ij})$, where $u_{ij} \equiv ih_j \pmod{n}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, q$, and multiplication modulo n is modified such that $0 < u_{ij} \leq n$, i.e., $u_{ij} = n$ if $ih_j = mn$ for some integer m. A matrix of rank s formed by any s columns of $G_{n,q}$ is called a UG-type design and is denoted by $UG_{n,s}$. A UG-type design with the smallest discrepancy over all $UG_{n,s}$'s is called a *G-uniform design* and is denoted by $UG_n(n^s)$.

Despite the efficiency and convenience brought about by the use of *glp* sets, there are some disadvantages: 1) $q = \phi(n)$ is much smaller than n in many cases; 2) The rank of the matrix $U_n(n^q)$ is at most $(q/2) + 1$ (cf. Ding (1986)). These two shortcomings imply that a) we cannot find a G-uniform design $UG_n(n^s)$ with $s > (\phi(n)/2) + 1$, nor can we obtain any UG-type saturated designs, and b) G-uniform designs may have poor uniformity.

In this paper we shall propose another subset of U-type designs, called UL-type designs, based on cyclic Latin squares. A UL-type design can be obtained by choosing s linearly independent columns of a cyclic Latin square. A threshold accepting algorithm is proposed for finding the "best" UL-type design. Such a design is called an *L-uniform design,* and is denoted $UL_n(n^s)$.

Section 2 introduces some measures of uniformity. In Section 3 we study properties of UL-type designs that can be used to reduce the computing time. The algorithm is discussed in Section 4, along with some numerical comparisons between $UG_n(n^s)$ and $UL_n(n^s)$. Conclusions and additional discussion are given in the last section.

2. Measures of Uniformity

We denote an n-run design over C^s by $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ and let $\mathcal{P}_{n,s}$ be the set of all such designs. Let $F_n(\mathbf{x})$ be the empirical distribution function of $\mathbf{X}$, i.e., $F_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} I\{\mathbf{x}_i \leq \mathbf{x}\}$, where $I\{\cdot\}$ is the indicator function and all inequalities are understood to be componentwise. The L_p-discrepancy of $\mathbf{X}$ is

defined as $D_p(\mathbf{X}) = \left[\int_{C^s} |F_n(\mathbf{x}) - F(\mathbf{x})|^p \, d\mathbf{x}\right]^{\frac{1}{p}}$, where $F(\mathbf{x})$ is the distribution function of the uniform distribution over C^s. It is the most commonly used measure of uniformity, especially when $p = \infty$ and 2. When $p = \infty$, $D \equiv D_\infty$ is also called the discrepancy (or star-discrepancy): $D(\mathbf{X}) = \sup_{\mathbf{x}\in C^s} |F_n(\mathbf{x}) - F(\mathbf{x})|$. The calculation of the discrepancy in multidimensional situations is very cumbersome, and there had been no general algorithm available until Bundschuh and Zhu (1993) gave one for low-dimensions. In the current work we employ their algorithm for evaluating discrepancy. Warnock (1972) gave an analytic formula for calculating L_2-discrepancy,

$$D_2(\mathbf{X}) = 3^{-s} - \frac{2^{1-s}}{n}\sum_{k=1}^{n}\prod_{l=1}^{s}(1 - x_{kl}^2) + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{s}[1 - \max(x_{ki}, x_{ji})], \quad (2.1)$$

where $\mathbf{x}_k = (x_{k1}, \ldots, x_{ks})$. Obviously, the computing load of $D_2(\mathbf{X})$ is $O(n^2)$ and is much lighter than that of $D(\mathbf{X})$.

3. U-type Designs Based on Cyclic Latin Squares

Let $x_1, \ldots, x_n$ be the entries of a Latin square. Define the left shift operator L by $L(x_1, \ldots, x_n) = (x_2, x_3, \ldots, x_n, x_1)$.

Definition 2. A left cyclic Latin square (LCLS) of order n is a Latin square of order n such that $\mathbf{x}_{i+1} = L\mathbf{x}_i, i = 1, \ldots, n-1$, where $\mathbf{x}_i$ is the ith row of the square.

Let $\mathcal{P}_n = \{(x_1, \ldots, x_n) : (x_1, \ldots, x_n)$ is a permutation of $\{1, \ldots, n\}\}$. An LCLS is uniquely determined by its first row. The LCLS with the first row $\alpha \in \mathcal{P}_n$ is denoted by $LC(\alpha)$. Any s columns of an LCLS form a U-type design, and is called a *UL-type design.*

Definition 3. Let $i_1, \ldots, i_s$ be s integers such that $1 \le i_1 < \cdots < i_s \le n + i_1 - 1$. The sequence $\{i_2 - i_1, i_3 - i_2, \ldots, i_s - i_{s-1}, i_1 - i_s\}$ (mod n) is called a difference sequence, where the modulo operation is modified such that the sequence takes values in $\{1, \ldots, n\}$.

Note that if $\{d_1, \ldots, d_s\}$ is a difference sequence, then $\sum_{i=1}^{s} d_i = n$. For $\mathbf{K} = LC(\boldsymbol{\alpha})$, let $\mathbf{K}(i_1, \ldots, i_s)$ be the submatrix consisting of the $i_1th, \ldots, i_sth$ columns of $\mathbf{K}$. Then we have the following results, useful for reducing computing time in Section 4.

Theorem 1. *If the indices of two submatrices* $\mathbf{K}(i_1, \ldots, i_s)$ *and* $\mathbf{K}(j_1, \ldots, j_s)$ *of* $\mathbf{K} = LC(\boldsymbol{\alpha})$, $\boldsymbol{\alpha} \in \mathcal{P}_n$, *have the same difference sequence, then these two UL-type designs are equivalent.*

Theorem 2. *If* $\mathbf{A}$ *and* $\mathbf{B}$ *are two* $n \times s$ *submatrices of* $\mathbf{K} = LC(\boldsymbol{\alpha})$, $\boldsymbol{\alpha} \in \mathcal{P}_n$, *with difference sequences* $\{d_1, \ldots, d_s\}$ *and* $\{d_2, \ldots, d_s, d_1\}$, *respectively. Then* $\mathbf{A}$ *and* $\mathbf{B}$ *are equivalent.*

The proofs are omitted. More details can be found in our technical report.

4. L-Uniform Designs, Threshold Accepting Algorithm and Numerical Comparisons

To construct the "best" UL-type design, we use a two stage procedure. We first choose an $\boldsymbol{\alpha}^* \in \mathcal{P}_n$ such that

$$D_2(LC(\boldsymbol{\alpha}^*)) = \min_{\boldsymbol{\alpha} \in \mathcal{P}_n} D_2(LC(\boldsymbol{\alpha})), \tag{4.1}$$

then for $\mathbf{K} = LC(\boldsymbol{\alpha}^*)$, we find $(i_1^*, \ldots, i_s^*)$ such that

$$D(\mathbf{K}(i_1^*, \ldots, i_s^*)) = \min_{1 \le i_1 < \cdots < i_s \le n} D(\mathbf{K}(i_1, \ldots, i_s)).$$

The design $\mathbf{K}(i_1^*, \ldots, i_s^*)$ is called an L-uniform design.

The L_2-discrepancy is used in the first stage since it is easier to calculate (see (2.1)). To search for the best $\boldsymbol{\alpha}$, we propose to use the threshold accepting (TA) algorithm introduced by Dueck and Scheues (1990).

To minimize a certain function over a finite set $\mathcal{X}$, the TA algorithm starts with an arbitrary element in $\mathcal{X}$. At each iteration an element is chosen randomly from a predefined neighborhood of the current solution and the value of the objective function is calculated for the new element. If it is not much worse than the old value, the new element is accepted as the current solution. The TA algorithm terminates when the increment of the objective function is more than a given positive threshold value. Since the threshold value is positive, unwanted local minima which might trap our search can be left again. The thresholds are changed throughout the procedure so that a local minimum, with a good chance of being the global minimum, can be reached.

In our application, $\mathcal{X}$ is $\mathcal{P}_n$. In fact, since $LC(\boldsymbol{\alpha})$ and $LC(L^m\boldsymbol{\alpha})$ are obviously equivalent for $m = 1, \ldots, n-1$, the number of possible $\boldsymbol{\alpha}$ that need to be considered is reduced to $(n-1)!$. Let $\boldsymbol{\alpha} = (a_1, \ldots, a_n) \in \mathcal{X}$. A neighborhood $N(\boldsymbol{\alpha})$ is defined as the set of permutations that exchange two components of $\boldsymbol{\alpha}$. Therefore, there are $n(n-1)/2$ permutations in $N(\boldsymbol{\alpha})$. A part of the final results are given in Table 1. The CPU time was from 0.001 second to 368.25 seconds for $n = 4 \sim 50$ with the program run on a PC486. It shows the power of the TA algorithm. Figure 1 presents a plot of $\log(D_2(LC(\boldsymbol{\alpha})))$ against n. The curve is linearly decreasing. It is known that

$$E([D_2(\mathbf{X})]^2) = \frac{1}{n}\left(\frac{1}{2^s} - \frac{1}{3^s}\right) \tag{4.2}$$

if the rows of $\mathbf{X}$ are i.i.d. random vectors from the uniform distribution over C^s. It is easy to prove that $D_2(LC(\boldsymbol{\alpha}))$ is much lower than the corresponding expected value of D_2 in (4.2).

Table 1. The "best" $\boldsymbol{\alpha}$ by using TA algorithm.

n	Best First Row of LCLS	L_2-Discrepancy
4	1 2 3 4	7.767E-02
5	1 5 3 4 2	4.965E-02
6	1 3 5 6 2 4	3.178E-02
7	1 3 2 7 4 5 6	1.922E-02
8	1 5 2 7 8 4 3 6	1.170E-02
9	1 8 9 4 2 5 6 3 7	7.043E-03
10	1 7 3 10 4 6 9 8 5 2	4.209E-03
11	1 4 6 8 9 11 5 3 10 2 7	2.500E-03
12	1 7 10 4 5 12 9 11 2 3 6 8	1.478E-03
13	1 12 4 6 5 7 9 10 3 2 8 13 11	8.707E-04
14	1 8 10 4 11 12 9 13 3 2 6 14 5 7	5.116E-04
15	1 7 12 6 9 10 13 8 15 5 3 2 4 14 11	3.000E-04
16	1 9 8 2 16 6 7 12 15 11 14 10 5 3 4 13	1.756E-04
17	1 10 2 5 14 6 7 9 12 11 3 4 16 8 13 17 15	1.027E-04
18	1 7 17 18 13 5 14 12 3 16 6 8 9 11 4 10 2 15	6.005E-05
19	1 19 18 13 9 6 11 5 16 15 3 14 4 12 10 8 2 17 7	3.508E-05
20	1 16 15 18 20 5 6 8 2 13 4 10 17 11 3 12 14 7 9 19	2.049E-05
21	1 21 19 12 11 17 13 6 10 8 2 3 5 14 15 20 4 9 16 7 18	1.197E-05
22	1 16 14 8 17 15 7 10 21 2 4 5 13 19 6 9 3 12 11 18 20 22	6.989E-06
23	1 5 14 6 19 11 17 7 8 12 2 21 9 10 4 16 20 18 3 23 22 13 15	4.086E-06
24	1 16 24 5 21 13 6 9 14 22 2 23 17 12 20 19 15 10 7 11 18 4 8 3	2.386E-06
25	1 20 3 25 12 22 23 2 9 5 11 8 15 13 4 24 17 14 7 6 21 10 19 18 16	1.395E-06
26	1 3 21 13 17 25 8 6 19 23 15 14 9 7 22 18 12 26 20 16 5 11 4 10 2 24	8.160E-07
27	1 9 8 4 3 18 26 10 20 23 25 22 14 19 2 11 27 21 13 6 16 12 7 15 24 5 17	4.776E-07
28	1 25 6 16 13 17 20 26 18 11 21 28 23 19 4 10 24 3 15 9 5 8 14 12 7 22 27	2.787E-07
29	1 12 27 5 20 4 17 2 15 6 14 25 21 28 29 10 22 16 13 8 25 9 11 23 18 7 26 19 3	1.639E-07
30	1 22 20 27 17 29 18 8 15 19 23 14 11 9 26 21 16 13 6 10 4 30 7 24 25 5 3 2 12 28	9.610E-08
31	1 7 28 29 9 13 6 19 2 15 25 10 4 23 11 20 21 18 12 26 5 30 14 16 31 8 24 3 27 17 22	5.639E-08
32	1 18 10 28 17 14 2 16 24 30 9 11 32 5 19 12 6 29 20 31 25 26 23 13 22 3 4 21 15 27 8 7	3.312E-08

In the second stage, Theorems 1 and 2 play an important role in reducing the computational load. L-uniform designs for $n \leq 50$ and $s \leq 7$ are obtained. We omit the detailed results here. Comparing the discrepancy of the G- and the L-uniform designs we conclude that:

1) When $s = 2$, the G-uniform design beats the L-uniform design for most n.
2) When $s > 2$ the L-uniform design beats the G-uniform design for many n.

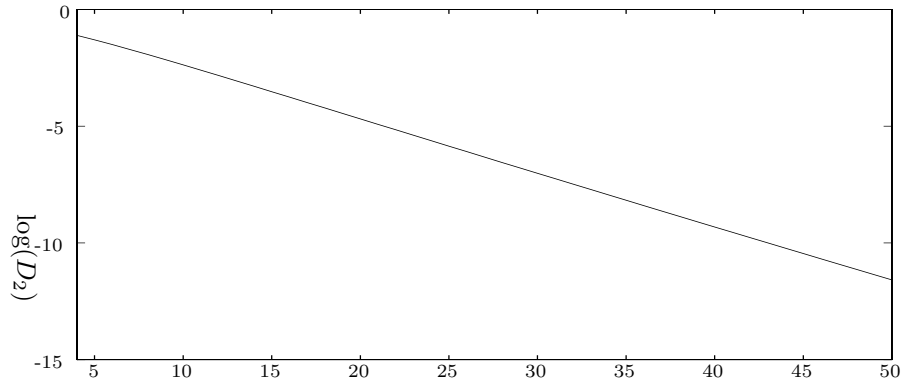

Figure 1

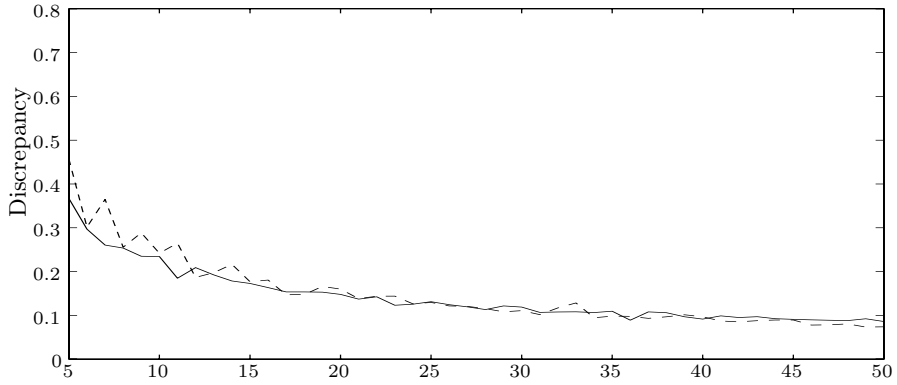

Figure 2($s = 3$)

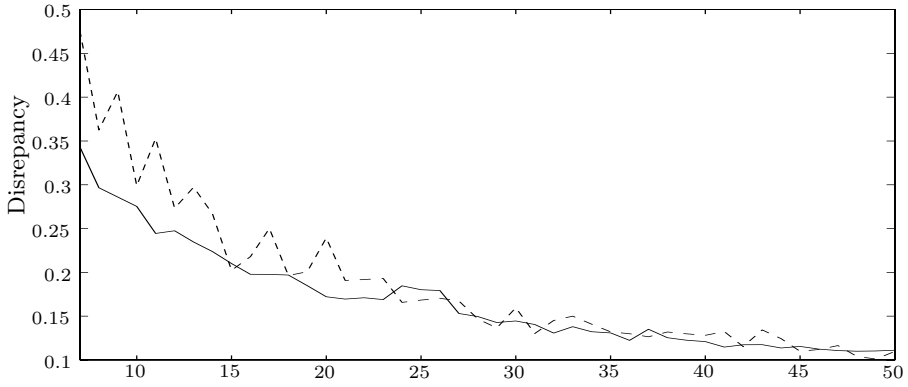

Figure 3($s = 4$)

Figures 2 and 3 compare the two kinds of designs for $s = 3$ and 4, where the solid line corresponds to the discrepancy of the L-uniform design obtained by our algorithm, and the dashed line corresponds to the discrepancy of the G-uniform design. Let D_g be the discrepancy of the G-uniform design and D_l the discrepancy of the L-uniform design for given n and s. The relative improvement in discrepancy is given by $I_1 = (D_g - D_l)/D_g$ or $I_2 = (D_g - D_l)/D_l$. Our results show that I_1 and I_2 range from 0.01% to 60% for $s \geq 3$, if $D_g > D_l$. For example, for $n = 7$ and $s = 3$, $D_g = 0.3721, D_l = 0.2606$, $I_1 = 29.97\%$ and $I_2 = 39.98\%$. For $n = 13$ and $s = 7$, $D_g = 0.4992, D_l = 0.3145$, $I_1 = 37\%$ and $I_2 = 58.73\%$.

5. Conclusion and Further Discussion

In many cases, the proposed L-uniform designs can significantly improve the uniformity of the corresponding G-uniform designs. The TA algorithm is powerful in finding the "best" LCLS. Note that an LCLS whose first row is given in Table 1 has full rank. Therefore, we always have n linearly independent columns for constructing an L-uniform design while there are fewer choices for G-uniform designs (at most $q/2+1$ columns where $q = \phi(n)$ (cf. Section 1)). Perhaps this is why an L-uniform design can improve upon G-uniform designs.

For further study, we may consider the following problems. 1) The convergence rate of the discrepancy of L-uniform designs as n increases. 2) Hickernell (1995) gave a detailed comparison of the Monte Carlo method, random orthogonal arrays, *glp* sets, and Faure points and he listed advantages and disadvantages of the *glp* sets in multidimensional quadrature. L-uniform designs are candidates for future comparisons.

Acknowledgement

This work was partially supported by a Hong Kong UGC Grant and Statistics Research and Consultancy Centre of HKBU. The authors are grateful for the valuable comments from the Associate Editor and two anonymous referees. We thank J. T. Zhang, Cori L. K. Tsoi, Vincent M. W. Chin and Clara S. H. Ma for their programming work.

References

Bates, R., Buck, R., Riccomagno, E. and Wynn, H. P. (1996). Experimental design and observation for large systems. *J. Roy. Statist. Soc. B.* **58**, 77-79.

Bundschuh, P. and Zhu, Y. C. (1993). A method for exact calculation of the discrepancy of low-dimensional finite point sets(I). Abhandlungen aus dem Math. Seminar der Univ. Hamburg, Bd., 63.

Ding, Y. (1986). An exploration of optimality of uniform design. *Chinese J. Appl. Prob. and Statist.* **2**, 153-160.

Dueck, G. and Scheuer, T. (1990). Threshold accepting: a general purpose algorithm appearing superior to simulated annealing. *J. Comput. Phys.* **90**, 161-75.

Fang, K. T. (1980). The uniform design: application of number-theoretic methods in experimental design. *Acta Math. Appl. Sinica* **3**, 363-372.

Hickernell, F. J. (1995). A comparison of random and quasirandom sampling methods for multidimensional quadrature. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (Edited by H. Niederreiter and P. J.-S. Shiue), 213-227. Springer-Verlag, Berlin.

Sacks, J., Welch, W., Mitchell, T. and Wynn, H. (1989). Design and analysis of computer experiments (with discussion). *Statist. Sci.* **4**, 409-435.

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. In *Applications of Number Theory to Numerical Analysis* (Edited by S. K. Zaremba), 319-343. Academic Press, New York.

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong and Chinese Academy of Sciences, Beijing 100080.

E-mail: KTFANG@HKBU.EDU.HK

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

E-mail: WCSHIU@HKBU.EDU.HK

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

E-mail: JX.PAN@BBSRC.AC.UK

(Received March 1995; accepted September 1998)