A New Approach to Construction of Nearly Uniform Designs

Changxing Ma
Division of Biostatistics, University of Florida
Gainesville, FL 32611, U.S.A., Email: cxma@ufl.edu
and
Kai-Tai Fang*
Department of Mathematics, Hong Kong Baptist University
Kowloon Tong, Hong Kong, CHINA, Email: Ktfang@hkbu.edu.hk

The uniform design is one of space filling designs and has been widely used in computer and industrial experiments. Many methods for construction of uniform designs or nearly uniform designs, such as the *glp* method, optimization method etc. have been proposed. A nearly uniform design is a design with low-discrepancy, where the discrepancy is a measure of uniformity. Various discrepancies have been suggested. To find a uniform design for n runs and s factors under a given discrepancy is a NP hard problem in the sense of computation complexity when $n \to \infty$ and $s > 1$. In this paper we propose a new method, called the cutting method, for construction of nearly uniform designs. It shows that the computation load of the new method is light and designs obtained by the new approach have better uniformity.

Keywords: Complexity of computation, computer experiments, discrepancy, uniform design.

1. Introduction

Computer experiments have been used in various fields of science and engineering to describe complicated physical phenomena which is governed by a number of equations, or by several softwares. Here relationships between the input variables and the output in a system have no analytic expressions. For studying properties of the system one wishes to find an approximate model that is much simpler than the true one and has an analytic expression. Suppose that the output y is determined by

$$y = g(x_1, \cdots, x_s) = g(\mathbf{x}), \quad \mathbf{x} = (x_1, \cdots, x_s)' \in T, \tag{1.1}$$

*The corresponding author. This work was partially supported by a Hong Kong RGC grant: RGC/HKBU 2044/02P and Statistics Research and Consultancy Centre, Hong Kong Baptist University

where the function g has no analytic formula and T is the input space. Based on a training data set $\{(\mathbf{x}_k, y_k) = (x_{k1}, \cdots, x_{ks}, y_k), k = 1, \cdots, n\}$ we wish to find an approximate model

$$y = \hat{g}(x_1, \cdots, x_s) = \hat{g}(\mathbf{x}), \quad \mathbf{x} \in T, \tag{1.2}$$

which is close to the real one. Therefore, we need

(a) a space filling design; and

(b) various modelling techniques.

A comprehensive review on computer experiments can refer to Sacks et al.[16], Bates et al.[1] and Koehler and Owen[13]. In this paper we focus on construction of space filling designs.

The uniform design, proposed by Fang and Wang in 1980 (see [2] and [18]), is one kind of space filling designs and has been widely used in computer experiments. Let n denote the number of runs, s the number of input variables and T the input space in a system where one wants to implement computer experiments. A uniform design seeks n points, denoted by $\mathcal{P}_n = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, to be uniformly scattered on T. Thus one needs a measure of uniformity and a way of finding a uniform design for given (n, s, T). Let D be a measure of uniformity such that the lower D-value the set $\mathcal{P}_n$ has, the more uniform on the space T the set $\mathcal{P}_n$ is. In Section 2 we shall introduce some measures of uniformity: star discrepancy, symmetrical discrepancy and centered L_2-discrepancy. When we say 'discrepancy' in this paper, it means that we shall choose one of these discrepancies. A uniform design $\mathcal{P}_n^*$ on T minimizes D-value on all designs of n points on T. In most experiments the input space T is a rectangle $[\mathbf{a}, \mathbf{b}] = [a_1, b_1] \times \cdots \times [a_s, b_s]$. Without loss of any generality the input space can be assumed to be a unit cube $C^s = [0, 1]^s$, if a linear transformation is considered.

To search a uniform design is a NP hard problem in the sense of computation complexity. Therefore, for reducing the computation complexity some structure of experimental points has to be considered. In the literature the so-called U-type design has been widely used for construction of uniform designs. There is a unique UD $\{\frac{1}{2n}, \frac{3}{2n}, \cdots, \frac{2n-1}{2n}\}$ for the case of one input variable (i.e. $s = 1$) under the star discrepancy, the symmetrical discrepancy and the centered L_2-discrepancy (see [9], [14] and [5] for the details). Choosing n equi-distance points $\{\frac{1}{2n}, \frac{3}{2n}, \cdots, \frac{2n-1}{2n}\}$, or equivalently $\{1, 2 \cdots, n\}$ for each input variable, a set of lattice points formed by these marginal points has n^s points and a lattice design is a subset of these n^s points.

Definition 1 *A U-type design, denoted by $U(n, n^s)$, is a $n \times s$ matrix and each column is a permutation of $\{1, 2, \cdots, n\}$. The set $\mathcal{U}(n, n^s)$ is the class of all $U(n, n^s)$'s.*

Let $U = (u_{kj})$ be a U-type design $U(n, n^s)$ and let $\mathbf{x}_k = (x_{k1}, \cdots, x_{ks})$, where

$$x_{kj} = \frac{2u_{kj} - 1}{2n}, \ k = 1, \cdots, n; j = 1, \cdots, s.$$

Then the set $\mathcal{P}_u = \{\mathbf{x}_k . k = 1, \cdots, n\}$ is a lattice design on C^s and is called the *induced design* of U. Let D be a measure of uniformity on C^s. We define the D-value of U as $D(U) = D(\mathcal{P}_u)$.

Definition 2 *A U-type design that minimizes D-value on $\mathcal{U}(n, n^s)$ is called a uniform design (UD) under the measure of uniformity D and is denoted by $U_n(n^s)$*

For given (n, s) to find a uniform design $U_n(n^s)$ is still a NP hard problem when n and s increase. Therefore, we need some efficient methods of finding a UD or a nearly uniform design (NUD), a design with lower D-value in a certain sense. Wang and Fang [18] employed the good lattice point method in quasi-Monte Carlo methods and found NUDs with $n \leq 31, s \leq 15$. Fang et al.[7] proposed a construction method based on Latin squares. Fang [10] suggested a way for constructing NUDs via orthogonal designs. Winker and Fang [20] and Fang, Ma and Winker [5] employed the threshold accepting method, a powerful optimization method, to find UDs/NUDs that can be downloaded in the world web site: http://www.math.hkbu.edu.hk/UniformDesign/ (UD-web for short).

However, the number of runs and the number of factors of the UDs in the UD-web and in the literature are still small, i.e., $n \leq 50$ and $s \leq 15$. It is not enough for computer experiments. Alternatively, Fang and Qin [6] proposed a way to construct NUDs with a large number of runs by collapsing two uniform designs. By this way we can easily obtained a lot of NUDs, but the uniformity of designs by their method are not good enough. Therefore, we need more efficient ways to generating NUDs with a large number of runs.

In this paper we propose a new method, called the **cutting method**, for construction of NUDs. The paper is organized as follows. Three measures of uniformity are introduced in Section 2. Section 3 gives idea of the cutting method and its algorithm. Section 4 shows that the cutting method has good performance in the sense of computation complexity and uniformity of the resulting NUDs. The final section gives concluding remarks.

2. Measures of Uniformity

Let $\mathcal{P} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ be a set of n points in the s-dimensional unit cube $C^s = [0, 1)^s$, where $\mathbf{x}_k = (x_{k1}, \cdots, x_{ks})$. Many different measures of uniformity of $\mathcal{P}$ have been defined (cf. Fang and Wang [9] and Hickernell [11]). A reasonable measure should be invariant under reordering the runs and relabeling the factors. The most popular measure of uniformity in quasi-Monte Carlo methods is the *star L_p-discrepancy*. Denote the empirical distribution of $\mathcal{P}$ by $F_{\mathcal{P}}(\mathbf{x})$ where

$$F_{\mathcal{P}}(\mathbf{x}) = F_{\mathcal{P}}(x_1, \cdots, x_s) = \frac{1}{n} \sum_{k=1}^{n} I(x_{k1} \leq x_1, \cdots, x_{ks} \leq x_s), \tag{2.1}$$

where $I(A)$ is the indicator function of A, i.e., $I(A) = 1$ if A is true, otherwise $I(A) = 0$. Let $F(\mathbf{x})$ denote the uniform distribution on C^s. The star L_p-discrepancy is defined by $||F_{\mathcal{P}}(\mathbf{x}) - F(\mathbf{x})||_p$, where $||\cdot||_p$ is the L_p-norm. When $p \to \infty$, the limiting measure is called the *star discrepancy* which has the expression

$$D(\mathcal{P}) = \max_{\mathbf{x} \in C^s} |F_{\mathcal{P}}(\mathbf{x}) - F(\mathbf{x})|. \tag{2.2}$$

The star discrepancy is just the Kolmogorov-Smirnov statistic in the goodness-of-fit test. The star discrepancy has been widely used in quasi-Monte Carlo methods, statistics as well as in construction of UDs (see Winker and Fang [20]). The star discrepancy is not easy to be calculated (See Winker and Fang [19]) and is not invariant under coordinate rotation, i.e. the origin plays a special role among all 2^s corner points of C^s. A natural idea is that all the 2^s corner points play the same role for measuring uniformity of the set $\mathcal{P}$. By a suitable rotation of coordinates each corner point of C^s would be the origin, and we shall have 2^s star discrepancies, whose average is called *symmetrical discrepancy*, denoted by $SD(\mathcal{P})$. The symmetrical discrepancy was proposed by Ma [14] who obtained a number of good properties of the symmetrical discrepancy. Alternatively, Hickernell [11] defined the *centered L_2-discrepancy*, denoted by $CD(\mathcal{P})$, that is invariant under coordinate rotation, and derived a computational formula for $CD(\mathcal{P})$ as follows

$$\begin{aligned}(CD(\mathcal{P}))^2 &= \left(\frac{13}{12}\right)^s - \frac{2}{n}\sum_{k=1}^{n}\prod_{j=1}^{s}(1+\frac{1}{2}|x_{kj}-0.5|-\frac{1}{2}|x_{kj}-0.5|^2) \\ &\quad +\frac{1}{n^2}\sum_{k=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{s}\left[1+\frac{1}{2}|x_{ki}-0.5|+\frac{1}{2}|x_{ji}-0.5|-\frac{1}{2}|x_{ki}-x_{ji}|\right]. \qquad (2.3)\end{aligned}$$

The above three discrepancies have been used in our study. For saving the space we demonstration the method and examples only under the centered L_2-discrepancy.

3. The Cutting Method

It is known that the good lattice point (*glp*) method in quasi-Monte Carlo methods has been appreciated by many authors due its economic computation and good performance (see, for example, Hua and Wang [12] and Shaw [17]). In particular, the *glp* method with a power generator has the lowest computation complexity among various methods in quasi-Monte Carlo methods and a good performance in the sense of uniformity. Suppose that we want to have a uniform design $U_n(n^s)$. Let U_p be a uniform design $U_p(p^s)$, where $n < p$ and p is a prime, and let $\mathcal{P}_u$ be its induced design. Let D be a proper subset of C^s and $\mathcal{P}$ be the set of points of $\mathcal{P}_u$ fell on D. Then points in $\mathcal{P}$ are uniformly scattered on D from the theory of quasi-Monte Carlo methods. This is the key idea of the cutting method. The cutting method proposed in this paper chooses a suitable rectangle in C^s such that there are exact n points of $\mathcal{P}_u$ falling in this rectangle. These n points will form a NUD $U_n(n^s)$ by some linear transformations. Let us introduce the *glp* method and the cutting method .

(A) **The good lattice point method with a power generator.**

For given positive integer pair (p, s), we can generate a NUD $U_p(p^s)$ by the following steps:

Step 1. Find the candidate set of positive integers

$$\mathcal{A}_{p,s} = \{a : a < p, \quad g.c.d.(a^j, p) = 1, \ j = 1, \cdots, s\},$$

where $g.c.d.(e, f)$ is the great common divisor of e and f.

Step 2. For each $a \in \mathcal{A}_{p,s}$, construct a U-type design $U^a = (u^a_{kj})$ as follows:

$$u^a_{kj} = ka^{j-1} \pmod p, k = 1, \cdots, p; \ j = 1, \cdots, s,$$

where the multiplication operation modulo p is modified such that the resulted integer is between 1 and p.

Step 3. Minimize $D(U^a)$ on $\mathcal{A}_{p,s}$, i.e., find a $a_* \in \mathcal{A}_{p,s}$ such that

$$D(U^{a_*}) = \min_{a \in \mathcal{A}_{p,s}} D(U^a).$$

Then the design U^{a_*} is a NUD $U_p(p^s)$.

The cardinality of $\mathcal{A}_{p,s}$, denoted by $|\mathcal{A}_{p,s}|$, is determined by $\phi(\phi(p))$ if $s = p - 1$, where

$$\phi(n) = \{h : h \text{ is an positive integer}, h < n, g.c.d.(n, h) = 1\}$$

is the Euler function. When $s < p-1$, $|\mathcal{A}_{p,s}|$ falls in $[\phi(\phi(p)), p-1]$. $|\mathcal{A}_{p,s}|$ decreases as s increases. For each positive integer n there is a unique prime decomposition $n = p_1^{r_1} \cdots p_t^{r_t}$, where $p_1, \cdots, p_t$ are different primes and $r_1, \cdots, r_t$ are positive integers. Then the Euler function $\phi(n) = n(1-\frac{1}{p_1}) \cdots (1-\frac{1}{p_t})$. When n is a prime, it is easy to see $\phi(n) = n - 1$. For example, $\phi(31) = 30$ as 31 is a prime. The prime decomposition of 30 is 30=2*3*5 and $\phi(30) = 30(1-\frac{1}{2})(1-\frac{1}{3})(1-\frac{1}{5}) = 8$. So the cardinality of $\mathcal{A}_{31,5}$ is in $[\phi(\phi(31)), 31-1] = [8, 30]$. The numerical calculation shows that the cardinality of $\mathcal{A}_{31,5}$ equals to 26. In step 3, we need to compare only 26 U-type design candidates if we choose $p = 31$. Many NUDs were generated by the above method, for example, Fang [2] and Fang and Ma [4].

The *glp* method has some disadvantage, for example, $\phi(p)$ may be much less than p and $|\mathcal{A}_{p,s}|$ is smaller, when p is not a prime. Some modifications of the *glp* method have be raised. Let U_p be a NUD $U_p(p^s)$ generated by the above method, where p is a prime. Deleting the last row from this design, the remaining $(p-1)$ points form a NUD $U_{p-1}((p-1)^s)$. Fang and Li [3] showed that many NUDs obtained by this way have lower discrepancy than the corresponding NUDs generated by directly using *glp* method, as $p-1$ is an even number and $\phi(p-1) < (p-1)/2$. Ma [15] suggested to shift points generated by the *glp* method.

(B) **The cutting method and its algorithm**

Let $U_p(p^s)$ be a UD/NUD and $\mathcal{P}_u$ be its induced design on C^s and R be a rectangle in C^s. Let $\mathcal{P}$ be the set of points of $\mathcal{P}_u$ fell on R. Then points in $\mathcal{P}$ are uniformly scattered on R from the theory of quasi-Monte Carlo methods. Let us first see an example.

Example 1 A UD $U_{30}(30^2)$ was obtained by Fang, Ma and Winker [5] as follows

$$U_{30} = \left(\begin{array}{cccccccccccccccc} 24 & 23 & 1 & 9 & 5 & 11 & 19 & 6 & 21 & 3 & 12 & 15 & 20 & 18 & 17 \\ 25 & 6 & 12 & 18 & 16 & 7 & 4 & 9 & 11 & 3 & 14 & 20 & 30 & 15 & 24 \\ 26 & 7 & 4 & 28 & 27 & 25 & 13 & 14 & 29 & 22 & 8 & 2 & 16 & 30 & 10 \\ 2 & 29 & 21 & 13 & 28 & 17 & 27 & 1 & 8 & 19 & 5 & 26 & 10 & 22 & 23 \end{array} \right)'.$$

Its reduced design on C^2 is plotted in Figure 1(a). If we wish to have a NUD $U_{10}(10^2)$ from this U_{30}, we can choose ten successive points according to the first coordinate (Figure 1(b)) or according to the second coordinate (Figure 1(c)). For each coordinate we can wrap the square such that position 0 and position 1 to be at the same location. By this wrapping consideration, ten successive points can be separated in two rectangles: some points are near to 0 while others are near to 1 (Figure 1(d)). There are 60=30*2 such subsets of 10 points. The ten points in each cutting are uniformly scattered in the related (wrapped) rectangle. By a linear transformation the ten points in each cutting can be transformed into a unit square and the transfered points are uniformly scattered on the unit square. So we have 60 sets of 10 points, or 60 designs of 10 runs. Suppose that the centered L_2-discrepancy (CD) is chosen as measure of uniformity. Finally, we choose one design with smallest CD-value among these 60 designs. This design is a NUD $U_{10}(10^2)$ with CD=0.0543 and is given by

$$CU_{10} = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 9 & 1 & 7 & 3 & 8 & 4 & 10 & 2 & 6 \end{array} \right].$$

Its reduced design is plotted in Figure 2(a). Fang, Ma and Winker [5] found a uniform design $U_{10}(10^2)$ with CD=0.0543 by the threshold accepting algorithm as follows

$$U_{10} = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 9 & 1 & 7 & 3 & 10 & 4 & 6 & 2 & 8 \end{array} \right].$$

Its induced design is plotted in Figure 2(b). Note that CU_{10} obtained by the cutting method is a uniform design $U_{10}(10^2)$, as it has the same CD-value as U_{10}. If you directly employ the *glp* method, a NUD can be found

$$U_{glp,10} = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 3 & 6 & 9 & 2 & 5 & 8 & 1 & 4 & 7 & 10 \end{array} \right].$$

Its $CD = 0.0614$ is larger than the CD-value of CU_{10} and U_{10} and its uniformity is worse than that of CD_{10} and U_{10}.

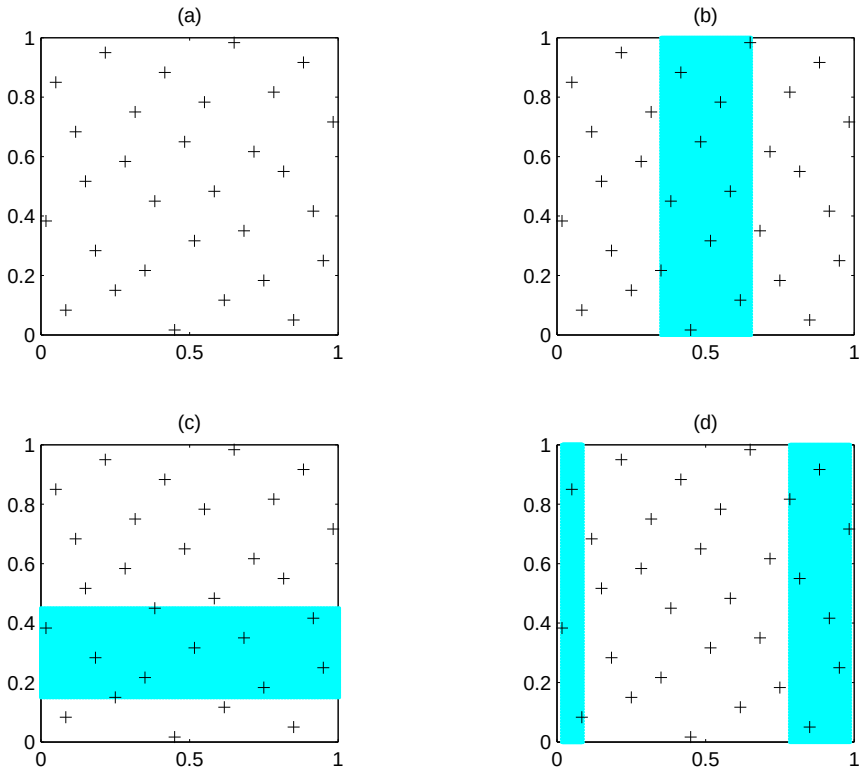

Figure 1: The cutting method

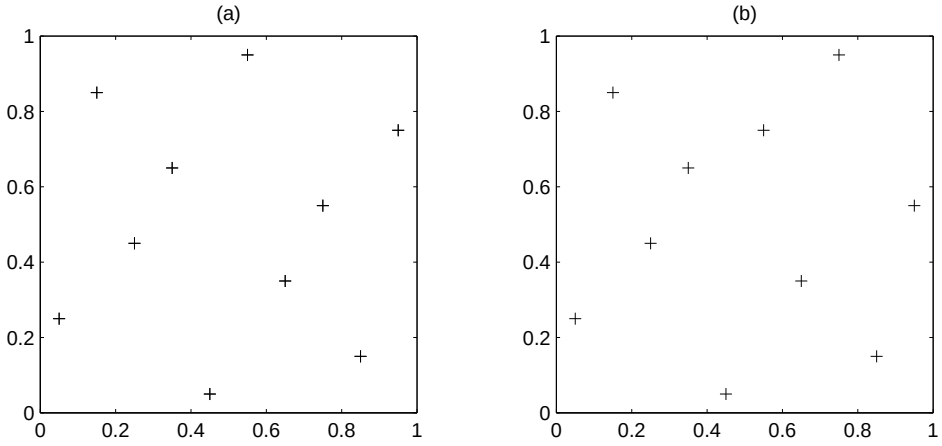

Figure 2: Plots of The Induced Designs of Two $U_{10}(10^2)'s$

From the above example, it shows that the cutting method has a good performance in construction of UDs/NUDs. Now, we give the algorithm of the cutting method.

Step 1. For given (n, s), find a NUD $U_p(p^s)$, where $p >> n$ and p or $p+1$ is a prime, and calculate its induced design $\mathcal{P}_u = \{\mathbf{c}_1, \cdots, \mathbf{c}_p\}$. The design $U_p(p^s)$ or $\mathcal{P}_u$ is called *initial design.*

Step 2. Denoted by $\mathbf{C} = (\mathbf{c}_{ij})$. For $l = 1, \cdots, s$ reorder rows of $\mathbf{C}$ by sorting the jth column of $\mathbf{C}$ and denoted the reordered matrix by $\mathbf{C}^{(l)} = (c_{kj}^{(l)})$.

Step 3. For $m = 1, \cdots, p$, let $\mathbf{C}^{(l,m)} = (c_{kj}^{(l,m)})$, where

$$c_{kj}^{(l,m)} = \begin{cases} c^{(l)}_{k+m-n-1\ j}, & m > n, k = 1, \cdots, n, \\ c^{(l)}_{k\ j}, & m \le n, k = 1, \cdots, m-1, \quad j = 1, \cdots, s. \\ c^{(l)}_{k+p-n\ j}, & m \le n, k = m, \cdots, n, \end{cases}$$

Step 4. Relabel elements of the jth column of $\mathbf{C}^{(l,m)}$ by $1, 2, \cdots, n$ according to magnitude of these elements. The resulted matrix becomes a U-type design $U(n, n^s)$ and is denoted by $U^{(l,m)}$. We have ps such U-type designs.

Step 5. For a given measure of uniformity D, compare ps designs $U^{(l,m)}$ obtained in the previous step and choose one with the smallest D-value. That one is a NUD $U_n(n^s)$.

4. Performance of The Cutting Method

The cutting methods proposed in the previous section has several advantages in computation complexity and uniformity of designs obtained by this method.

(A) **Complexity of the computation for finding an initial design**

Let $\mathcal{P}_p$ be a sequence of point sets generated by some quasi-Monte Carlo method, where p is a prime and $\mathcal{P}_p$ has p points in C^s. The best convergence rate of the star discrepancy $D(\mathcal{P}_p)$ or symmetrical discrepancy $SD(\mathcal{P}_p)$ is $O(p^{-1}(\log p)^{s-1})$ as $p \to \infty$. If this sequence is generated by *glp* method, the best convergency rate becomes $O(p^{-1}(\log p)^s)$ that is slight lower than $O(p^{-1}(\log p)^{s-1})$. Furthermore, if the sequence is constructed by the *glp* method with a power generator, the best rate reduces to $O(p^{-1}(\log p)^s \log\log p)$ that is slight lower than $O(p^{-1}(\log p)^s)$ (see Fang and Wang [9]). This fact ensures that the initial design $\mathcal{P}_p$ has a good uniformity. To find such an initial design we need to compare only m designs of p runs and s factors, where $m \in [\phi(\phi(p)), p-1]$.

(B) **Comparisons between the cutting method and *glp* method**

We have done some comparisons among the *glp* method, optimization with TA algorithm and cutting method, like in Example 1. For illustration we give two examples below.

Example 2. Construction of NUDs for $n = 4, \cdots, 29$ and $s = 2, 3, 4, 5$

We compare the following three methods for construction of the above designs. (1) By the use of the TA algorithm for search UDs $U_n(n^s)$ (Cf. Fang, Ma and Winker [5]) required in this example, all NUDs can be

downloaded from the world web site http://www.math.hkbu.edu.hk/UniformDesign/. (2) We directly use the *glp* method for generating nearly uniform designs. The latter can be found in Fang and Ma [4]. (3) For applying the cutting method, the initial designs $U_{30}(30^s)$ for $s = 2, 3, 4, 5$ are chosen from Fang, Ma and Winker [5] and can be downloaded in the UD web.

Figure 3 give comparisons of centered L_2-discrepancy between designs generated by the above three methods. Obviously, the designs obtained by TA algorithm has the lowest CD-value. Designs obtained by the cutting methods have better uniformity than designs directly generated by the *glp* method. The CD-value of designs obtained by the cutting methods are very close or equal to the corresponding design obtained by the TA algorithm. Furthermore, all NUDs ($n = 4, \cdots, 29, s = 2, 3, 4, 5$) are generated from only four initial UDs. This fact shows that the cutting method needs a very limited computing time for constructing many NUDs.

Example 3. Construction of NUDs for $n = 100$ and $s = 5$

The initial design is constructed by the *glp* method with a power generator. We choose a prime $p = 151$ and find the best $a = 117$ in the sense of the centered L_2-discrepancy. The generator is $(1, 117, 117^2, 117^3, 117^4)$ (mod 151)=$(1, 117, 99, 107, 137)$. Now we have a NUD $U_{151}(151^5)$ as an initial design. There are $151*5 = 755$ candidate designs by applying steps 2-4 of the cutting method. Among these candidates the design listed in Table 1 has the lowest $CD^2 = 0.0012$. If we directly use the *glp* method with a power generator, the generator $(1, 63, 63^2, 63^3, 63^4)$ (mod 101)= $(1, 63, 30, 72, 92)$ is the best one and related NUD has $CD^2 = 0.0013$. If we use the TA algorithm, the search computing time is much longer than the time by the use of the cutting method in this example.

5. Concluding Remarks

In this paper the cutting method for generating uniform designs and nearly uniform designs is proposed. For given (n, s) the cutting method need to compare m designs of p runs and s factors, where $m \in [\phi(\phi(p)), p-1]$ and $p >> n$, and compare ps designs of n runs and s factors. We have show that the designs obtained by the cutting method have better uniformity than those directly generated by the *glp* method. The cutting method is good for construction of NUDs with n being large.

In the above three examples the centered L_2-discrepancy is used as measure of uniformity. If we employ the star discrepancy or the symmetrical discrepancy in these cases, the cutting method also show its advantages mentioned before. This indicates that performance of the cutting method does not depend on specific measure of uniformity.

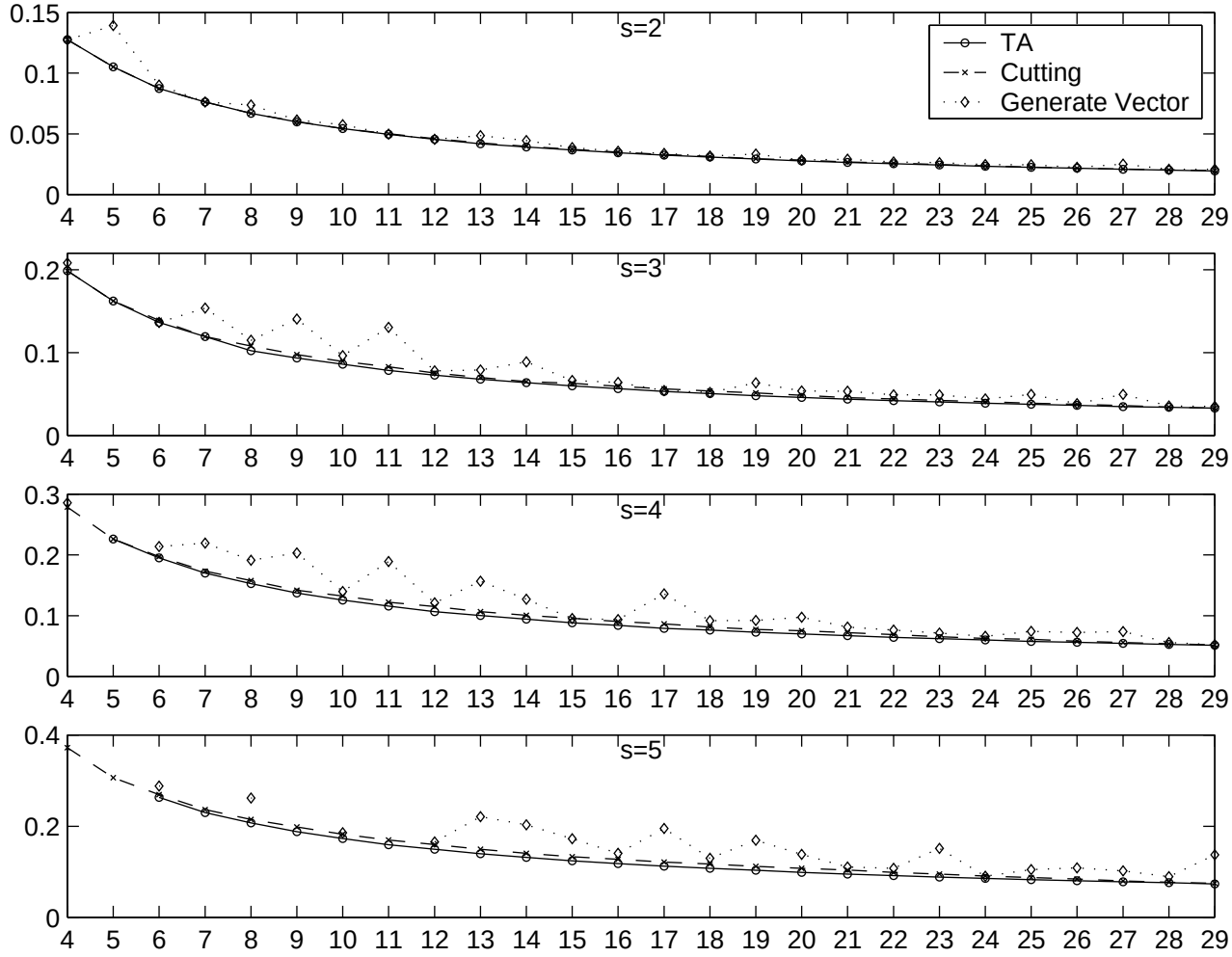

Figure 3: CD-value of designs generated by the *glp*, TA and cutting methods

References

[1] Bates, R. A., R.J. Buck, E. Riccomagno, and H.P. Wynn (1996). Experimental design and observation for large systems. *J. R. Statist. Soc. B.* **58**, 77–94.

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

[3] Fang, K. T. and J. K. Li (1995). Some new results on uniform design, *Chinese Science Bulletin*, **40**, 268-272.

[4] Fang, K.T. and Ma, C.X. (2001), *Orthogonal and Uniform Experimental Designs*, Science Press, Beijing.

[5] Fang, K.T., C.X. Ma and P. Winker (2001). Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform design, *Math. Computation.*, **71**, 275–296.

[6] Fang, K.T. and H. Qin (2002), A note on construction of nearly uniform designs with large number of runs, *Statist. & Prob. Letters*, in press.

Table 1: Table $U_{100}(100^5)$

1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	77	66	70	86	34	29	9	84	27	67	58	18	68	54
2	65	93	24	67	35	7	75	54	13	68	36	84	39	40
3	42	59	95	53	36	84	40	26	99	69	13	49	11	26
4	20	24	66	39	37	62	6	97	85	70	91	15	81	12
5	97	90	37	25	38	49	33	50	66	71	69	81	52	98
6	75	56	8	11	39	26	99	21	52	72	46	46	23	84
7	53	21	79	97	40	4	65	93	38	73	33	74	77	65
8	30	87	49	83	41	81	30	63	24	74	10	39	47	51
9	17	14	4	64	42	59	96	34	10	75	88	5	19	37
10	94	80	74	50	43	37	62	6	96	76	66	71	90	23
11	72	45	45	36	44	14	27	76	82	77	43	36	60	9
12	50	11	16	22	45	1	55	30	63	78	21	2	32	95
13	27	77	87	8	46	78	20	1	49	79	98	68	3	81
14	5	42	58	94	47	56	86	72	35	80	85	95	56	62
15	82	8	29	80	48	34	52	42	21	81	63	61	27	48
16	70	35	83	61	49	11	17	14	7	82	40	26	99	34
17	47	1	53	47	50	89	83	85	93	83	18	92	69	20
18	24	67	25	33	51	67	48	55	79	84	95	58	40	6
19	2	32	96	19	52	54	76	10	60	85	73	23	12	92
20	79	98	67	5	53	31	41	80	46	86	51	89	82	78
21	57	64	38	91	54	8	7	51	32	87	60	51	65	73
22	35	29	9	77	55	86	73	22	18	88	38	16	36	59
23	44	91	92	72	56	64	38	94	4	89	15	82	7	45
24	22	57	62	58	57	41	4	64	90	90	92	47	78	31
25	99	22	33	44	58	19	70	35	76	91	71	13	48	17
26	76	88	5	30	59	28	31	18	71	92	48	79	20	3
27	55	54	75	16	60	6	97	89	57	93	25	44	91	89
28	32	19	46	2	61	83	63	59	43	94	3	10	61	75
29	9	85	17	88	62	61	28	31	29	95	12	72	44	70
30	87	50	88	74	63	39	94	2	15	96	90	37	15	56
31	96	12	71	69	64	16	60	73	1	97	68	3	86	42
32	74	78	41	55	65	93	25	43	87	98	45	69	57	28
33	52	43	13	41	66	80	53	98	68	99	23	34	28	14
										100	100	100	100	100

[7] Fang, K.T., W.C. Shiu and J. X. Pan (1999), Uniform designs based on Latin squares, *Statistica Sinica*, **9**, 905–912.

[8] Fang, K.-T. and P. Winker (1998). Uniformity and orthogonality, Technical Report MATH-175, Hong Kong Baptist University.

[9] Fang, K.T. and Wang, Y. (1994). *Number-theoretic Methods in Statistics*, Chapman and Hall, London.

[10] Fang, Y. (1995). Relationships between uniform design and orthogonal design, *The 3rd International Chinese Statistical Association Statistical Conference,* Beijing.

[11] Hickernell, F.J. (1998). A generalized discrepancy and quadrature error bound, *Math. Comp.*, **67**, 299–322.

[12] Hua, L.K. and Y. Wang (1981). *Applications of Number Theory to Numerical Analysis*, Springer and Science Press, Berlin and Beijing.

[13] Koehler, J.R. and A.B. Owen (1996). Computer experiments, in *Handbook of Statistics, Vol. 13*, Eds by S. Ghosh and C.R.Rao, Elsevier Science B.V., Amsterdam, 261-308.

[14] Ma, C.X. (1997a). A new criterion of uniformity — symmetrical discrepancy, *J. Nankai Univ.*,**30**(1), 30–37.

[15] Ma, C.X. (1997b). Construction of uniform designs using symmetrical discrepancy, *Application of statistics and management*, 166–169.

[16] Sacks, J. W.J. Welch, T.J. Mitchell, and H.P. Wynn (1989). Design and analysis of computer experiments (with discussion), *Statist. Sciences*, **4**, 409–435.

[17] Shaw, J.E.H. (1988). A quasi-random approach to integration in Bayesian statistics, *Annals of Statistics*, **16**, 895–914.

[18] Wang, Y. and K.T. Fang (1981). A note on uniform distribution and experimental design, *KeXue TongBao*, **26**, 485–489.

[19] Winker, P. and K.T. Fang (1997). Application of Threshold accepting to the evaluation of the discrepancy of a set of points, *SIAM Numer. Analysis*, **34**, 2038-2042.

[20] Winker, P. and K.T. Fang (1998). Optimal U-type design, in *Monte Carlo and Quasi-Monte Carlo Methods 1996*, eds. by H. Niederreiter, P. Zinterhof and P. Hellekalek, Springer, 436–448.