COMISEF Computational Optimization Methods in Statistics, Econometrics and Finance

- Marie Curie Research and Training Network funded by the EU Commission through MRTN-CT-2006-034270 -

COMISEF WORKING PAPERS SERIES

WPS-013 20/08/2009

Optimized U-type Designs on Flexible Regions

D. K. J. Lin
C. Sharpe
P. Winker

Optimized U-type Designs on Flexible Regions*

Dennis K.J. Lin, Chris Sharpe, Peter Winker†

August 20, 2009

Abstract

The concept of a flexible region describes an infinite variety of symmetrical shapes to enclose a particular region of interest within a space. In experimental design, the properties of a function on the region of interest is analyzed based on a set of design points. The choice of design points can be made based on some discrepancy criterion. This paper investigates the generation of design points on a flexible region. It uses a recently proposed new measure of discrepancy for this purpose, the Central Composite Discrepancy. The optimization heuristic Threshold Accepting is used to generate low discrepancy U-type designs. The proposed algorithm is capable to construct optimal U-type designs under various flexible experimental regions in two or more dimensions. The illustrative results for the two dimensional case indicate that using an optimization heuristic in combination with an appropriate discrepancy measure, it is possible to produce high quality experimental designs on flexible regions.

Keywords: Central composite discrepancy; Experimental design; Flexible regions; Threshold accepting; U-type design.

*Financial support from the EU Commission through MRTN-CT-2006-034270 COMISEF and the DFG through grant No. 62201080 (Mercator Lin) is gratefully acknowledged.

†Corresponding author: Dept. of Economics, University of Giessen, Licher Str. 64, 35394 Giessen, Germany, E-mail: Peter.Winker@wirtschaft.uni-giessen.de, Phone: +49-641-99-22-640, Fax: +49-641-99-22-649.

1 Introduction

The development of a flexible region approach by Draper and Guttman (1986) built on previous research in experimental design. The central question is to understand the effect from points selecting in the experimental domain (input space) for response surface analysis, where the functional relationship between input and output is not known. It is assumed that a sub-space of the input space can be localized such that inputs from this sub-space produce an output that meets some set objective. This sub-space selected for experimental examination is coined the region of interest, represented by a set of design points.

In standard settings, simple shapes are preferred for the region of interest such as spherical, cuboidal, and triangular.[1] The cuboidal shape, typically multi-dimensional unit cubes, has received a great deal of attention, and many approaches for constructing good experimental designs have been proposed.[2] However, for many applications, a standard shape might not provide a good approximation of the region of interest. The target is to generate low discrepancy designs for a given region of interest, but it is difficult to derive appropriate designs from simple transformations. A viable solution to overcome this obstacle are designs constructed for differently shaped regions of interest.

The idea of a flexible region as proposed by Draper and Guttman (1986) can be considered as a combination of the first two of the standard shapes (spherical and cuboidal). By an adjustment of a single parameter, it becomes possible to obtain an infinite variety of intermediate symmetrical shapes. The approach is particularly useful as it offers the possibility for intuitive linguistic mapping to shape types. When applying a flexible region to a real problem Draper and Guttman (1986) point out that we do have some a priori knowledge as to what the region of interest should 'look like', which is first expressed in terms of natural language. This specification is then mapped to a specific value of the parameter determining the shape of the flexible region. This process offers some intriguing research possibilities for a tool to generate suitable flexible regions systematically.[3]

In this paper we consider U-type designs as proposed Fang (1980) and Wang and Fang (1981). In a U-type design all design points are located on a fine uniform grid over the search space. No specific assumptions are made

[1] See Draper and Lawrence (1965, 1966).

[2] See, e.g., Fang *et al.* (2006) for a recent survey.

[3] In particular, fuzzy sets and fuzzy logic (Zadeh (1965, 1987)), invented to map imprecise linguistic specifications to numeric values, would appear to be a possible candidate for this purpose.

about the functional relationship between input factors and output. Thus, the aim consists in finding a U-type design which covers the input space as uniformly as possible, e.g., by minimizing some appropriate measure of discrepancy. The resulting optimized U-type designs have the attractive feature to gather the maximum amount of information for a given number of design points without assuming any a priori knowledge.

There are several common measures for uniformity, amongst which the centered L_2-discrepancy proposed by Hickernell (1996) would appear at first glance to be a reasonable choice given that a flexible region is a symmetrical shape with its origin at the center of the region of interest. However, in common with other discrepancy measures, the discrepancy value is defined for a hypercube and cannot be directly adapted for other shape types. The central composite discrepancy (CCD) suggested by Chuang and Hung (2009) is not fixed at a constant point from which to calculate the discrepancy value, which removes the constraint on the shape type and hence makes it possible to apply it also for a flexible region. At present, there is no closed form expression for the CCD. Therefore, the calculation of the CCD for a given design is computational intensive imposing binding constraints on the dimension and size of designs to be considered in an optimization approach.

The process of generating low discrepancy designs on a flexible region is framed as an optimization problem, with the CCD the objective function to be minimized. Threshold Accepting (TA) is the heuristic technique implemented to tackle this optimization problem. It has been applied successfully to many experimental design problems (see, e.g., Fang *et al.* (2000), Winker (2001, Ch. 11), and Fang *et al.* (2005)). For some problem instances, lower bounds could be derived which could be achieved by the TA implementation (Fang *et al.* 2003).

In this paper, we describe how to produce designs of low discrepancy for different flexible region shapes and different design configurations. The results show that the TA implementation is able to derive U-type designs with low values of the CCD for different regions of interest.

The remainder of this paper is structured as follows: In Section 2 we discuss Draper and Guttman's (1986) formulation of a flexible region; in Section 3 we cover U-type design and the CCD as measure of uniformity; in Section 4 we state the algorithm TA in the context of design optimization; in Section 5 we describe the experiment set up and results; and in the final Section 6 we summarize our findings and consider the next steps for research.

2 Flexible Region Formulation

For any positive value of m, Draper and Guttman (1986) define a flexible region R in dimension s by the following constraint on the potential design points $x = (x_1, \ldots, x_s)$:

$$|x_1|^m + |x_2|^m +, \cdots, +|x_s|^m \leq 1. \tag{1}$$

This definition if for the hypercube $[-1, 1]^s$, while in recent work on experimental design the hypercube $C^s = [0, 1)^s$ is considered. To account for this change, the original definition is modified as follows: for a given shape parameter $m > 0$, the flexible region R_m is defined by

$$R_m = \{x \in [0, 1]^s \mid (|2(x_1 - 0.5)|^m + \ldots + |2(x_s - 0.5)|^m) \leq 1\}. \tag{2}$$

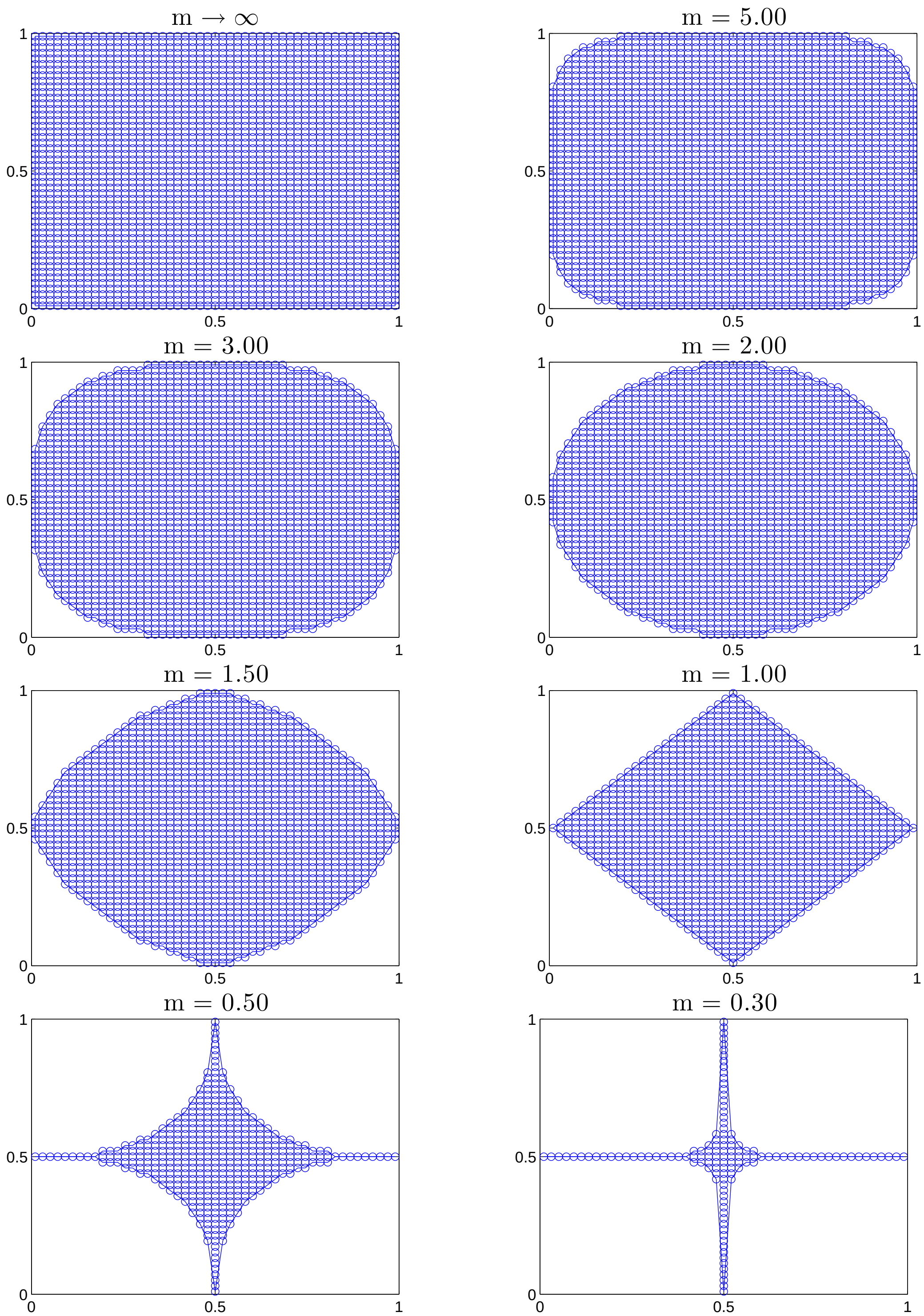
In both original and modified definition, the parameter m determines the shape of the flexible region. Figure 1 shows some examples for different values of m in the two dimensional case. Referring to the concept of U-type designs, for this figures – as in the following optimization approach – only design points lying on a grid over the unit cube are considered. Specifically, the unit cube is covered by a grid of q^2 points, where $q = 49$ for the current application.

As can be seen, when $m \to \infty$ then the entire input space is the region of interest, and when $m = 0.3$ only the very center and the extreme points extending to one edge cover the region of interest.[4]

With regard to the goal of a uniform covering of the input space by only a few design points, it is obvious from Figure 1 that using good designs for hypercubes, that corresponds to $m \to \infty$, is not a feasible approach for smaller values of m. In particular, a simple projection of a good design on C^s to R_m will contain only few points for small values of m which might not be scattered uniformly on R_m. Constructing more refined mappings and keeping the number of design points fixed while minimizing the discrepancy on R_m appears not possible. Therefore, we consider explicit construction of low discrepancy designs on flexible regions in the following based on a measure of discrepancy appropriate for each flexible region case.

[4]It might be observed that as m gets smaller the flexible region transformation is a deformation and rotation of three types of symmetrical shape, which are, in their most basic form, a square, circle, and astroid.

Figure 1: Examples of Flexible Regions



3 U-Type Design and Central Composite Discrepancy

We define a U-type design $\mathcal{P}$ as a set of n points, $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, sampled from the s-dimensional unit cube C^s on a grid with q^s points. This set of points can also be described by a $n \times s$ design matrix U, where each row corresponds to one run and each column to one factor. Thereby, the factors can take on $l = 1, \ldots, q$ different levels. The correspondence between the set of design point sets and the set of design matrices $U(n, q^s)$ is given by the transformation $\frac{2l-1}{2q}, l = 1, \ldots, q$. Obviously, the number of possible grid points grows at rate q^s. Therefore, each design will only consist of a number n of grid points which typically is much smaller than q^s.

Having to represent the input space by a subset of grid points results in the problem to select these points in an optimal way. To this end, first one has to define properties of 'good' point sets. This is done using measures of design uniformity (Fang *et al.* 2006, Ch. 3). The lower the design discrepancy value, the more uniform the design. Several discrepancy formulations have been proposed for the case of unit hypercubes (Hickernell 1996). However, these measures are not appropriate for a flexible region when m is small. In fact, aiming at low discrepancy designs based on these standard measures results in designs on the flexible region putting almost all design points on its boundary.

The central composite discrepancy (CCD), as proposed by Chuang and Hung (2009), measures uniformity only with regard to the flexible region, i.e., considering subsets falling within the flexible region. While the concept is similar to the centered L_2-discrepancy proposed by Hickernell (1996), it provides a useful discrepancy measure which is suitable for differently shaped regions of interest. The CCD for a set of points $\mathcal{P}$ in a region of interest R is defined by

$$CCD_p(\mathcal{P}) \;=\; \left\{ \frac{1}{v(R)} \int_R \frac{1}{2^s} \sum_{k=1}^{2^s} \left| \frac{N(R_k(x), \mathcal{P})}{n} - \frac{v(R_k(x))}{v(R)} \right|^p dx \right\}^{1/p}, \quad (3)$$

where $p > 0$ is a parameter defining the underlying norm (for our application, $p = 2$ is used, corresponding to the L_2-norm); $v(R)$ is the volume of the region R and $v(R_k(x))$ the volume of $R_k(x)$, a subregion of R defined below; n is the number of all design points, and $N(R_k(x), \mathcal{P})$ the number of design points in subregion $R_k(x)$. In contrast to the centered L_2-discrepancy, no simple analytical expression is available for CCD_p on general flexible regions. Therefore, we resort to an approximation based on the q^s grid on $[0, 1]^s$

introduced above. The volume of R is approximated by the number of these grid points falling into R and similarly for $R_k(x)$.

The principle idea of the CCD, and the reason why it can be applied to flexible regions, is that measurement is not taken from one fixed point: every point in R is considered a center point. At a given grid point we divide R into 2^s subregions. The dividing hyperplanes pass through the grid point and are parallel to one axis. For example, in the two dimensional case, R is cut into 4 subregions, divided by a horizontal and vertical line crossing at the grid point. For each resulting subregion R_k the share of design points in that subregion from all design points ($N(R_k(x), \mathcal{P})/n$) is compared with the relative volume of the subspace which is approximated by the fraction of grid points falling into $R_k(x)$ versus R. The absolute values of this differences are taken to the p-th power before adding up over all 2^s subregions. Multiplication by $\frac{1}{2^s}$ provides the average over the subregions. This calculation is repeated for all grid points in R. The resulting integral (which is again approximated by a finite sum as only points on the q^s grid are considered) provides the discrepancy measurement after normalization with $\frac{1}{v(R)}$.

An optimal U-type design $\mathcal{P}^*$ on R is obtained if the n design points from $\mathcal{P}$ are distributed in R in a way to minimize $CCD_p(n, \mathcal{P})$, or more formally,

$$\mathcal{P}^* \quad = \quad \arg\min_{P} CCD_p(\mathcal{P})\,. \tag{4}$$

4 Threshold Accepting

The local search algorithm TA, first devised by Dueck and Scheuer (1990), is an optimization technique that has been applied successfully to construct low discrepancy designs on unit cubes using centered discrepancy and several other discrepancy measures by, e.g., Fang *et al.* (2000) and Fang *et al.* (2002). Based on theoretical lower bounds for the discrepancy, Fang *et al.* (2003) and Fang *et al.* (2005) obtain uniform designs, i.e., U-type designs with the lowest possible discrepancy, for several problem instances using this methodology. It is highly flexible (non problem specific) and simple to implement and adapt. The new algorithm for the construction of low discrepancy designs on flexible regions is given by Algorithm 1. Is is adapted from the algorithm suggested by Winker (2001, Ch. 11) for the uniform design problem on the unit cube. The fact that no closed form analytical expression for CCD_p is available requires additional attention. Some of these issues will be addressed after discussing the general outline of the algorithm.

First (1:), we initialize the algorithm with a U-type design U^c on R_m, a design with points chosen from the q^s grid points falling into R_m. To

Algorithm 1 TA Algorithm for Design Optimization.

1: Initialize design $U^c \in \mathcal{U}$
2: Initialize n_R, n_S, and threshold sequence $\tau_r, r = 1, \ldots, n_R$
3: **for** $r = 1$ to n_R **do**
4:     **for** $i = 1$ to n_S **do**
5:         Generate at random a solution $U^n \in \mathcal{N}(U^c)$
6:         **if** $CCD(U^n) - CCD(U^c) < \tau_r$ **then**
7:             $U^c = U^n$
8:         **end if**
9:     **end for**
10: **end for**

this end, each column of U^c is filled with n uniform random numbers drawn from $1, \ldots, q$. Obviously, the resulting design points x (rows of U^c after transformation to $[0, 1)$) may not fall in the flexible region R_m as desired. Therefore, the initial design matrix U^c is iterated through row by row. The corresponding design point x is in the region of interest R_m if $\sum_{j=1}^{s} |x_j|^m \leq 1$. Any point that is found to lie outside the flexible region is subsequently moved until it is inside. Once all points are inside the region of interest, the objective function, the CCD value, is then calculated for this design.

The next stage of the initialization phase is to generate the threshold sequence, a critical component of the algorithm and one that controls the criteria for accepting and rejecting the design changes at each iteration i. A data-driven approach as outlined by Fang and Winker (1997) is implemented to perform this task. It takes the initial design and selects a design point element – a row and column index – at random. A level is drawn at random from $\{1, \ldots, q\}$ to replace the previous point element. The new point is tested as before to check whether it falls into R_m. If it is rejected, then the level selection process is repeated until the point is in the region of interest. The objective function value is calculated for this modified design and subtracted from the original objective function value; the absolute value of the difference is stored in a vector of length n_R. This process is repeated n_R times and the resulting vector is sorted in descending order providing the threshold sequence.

In the main body of the algorithm, from line 3:, we begin optimization of the initial design matrix. At each iteration we select a new design matrix U^n in a neighborhood of the current one ($\mathcal{N}(U^c)$) at random. The neighborhood mimics the idea of a natural ordering (Winker 2001, Ch. 11) where a neighborhood is defined by a generalization of the Hamming distance. This states that the new design U^n might differ from the previous one U^c by entries in $k \leq s$ columns. We set $k = s$, and select each column at random, i.e., we might have several changes in one column. A row from that column is selected at random and the point element is replaced. The new point element can take any level, as long as the point is inside the flexible region – if it is outside we repeat the replacement procedure until a feasible neighbor is found, in exactly the same way as done in the threshold sequence generation.

After the changes are applied, the objective function $CCD(U^n)$ is calculated for the new design U^n. Finally (6:), the new objective function value is subtracted from the objective function of the previous design matrix and the result compared against the current threshold value τ_r. The new design is deterministically adopted if this result is lower than the current threshold value.

At each step r the next value in the threshold sequence is taken, having the effect of tightening the condition on accepting a design that is worse, by some degree, than the previous design. Thereby, it is expected that the final design after the n_R times n_S iterations of the algorithm should have a small CCD value, and hence design points are distributed uniformly over the flexible region. Whether we are able to find an optimal design, one with lowest CCD value among all feasible designs, will be discussed in the results section.

The algorithm of the method to calculate the CCD along with a simple working example is presented in Appendix A. This is achieved by exploiting the symmetry of the flexible region in concert with manipulation of simple data structures in the two dimensional case. Its real worth is in its potential for extension into higher dimensions. Useful as any development in this direction may be, research effort ought to be concentrated on finding a general closed form solution to calculate the CCD in a flexible region, potentially allowing even to derive lower bounds.

5 Experiment Description and Results

5.1 Experimental Setup

In order to analyze the performance of the TA implementation for generating low discrepancy U-type designs on flexible regions, a set of experiments is run. In particular, different flexible region shapes (corresponding to different values of the parameter m) are considered as well as different numbers of design points n. Although the proposed algorithm is capable to construct optimized designs in higher dimensions, we use $d = 2$ for illustration. Finally, the TA algorithm is run with different values for the total number of iterations, i.e., $n_R \times n_S$. For each of these problem instances, the best design obtained during the run of the TA algorithm is recorded as well as the corresponding CCD value.

All experiments are implemented using Matlab 2008a, with the objective function code written using Matlab's interface to a C compiler. To use compiled code for calculating the objective function results in a relevant speed up reducing execution time by approximately a third. The algorithm is executed on an Intel 2 Core Quad, clocking at 2.83 GHz, with 8 GB of RAM, and with Windows XP Professional x64 Edition, 2003 operating system. Taking the most intensive experiment, execution time was approximately 26 hours for 20 design configurations, with 10 separate runs for each configuration to take account for the stochastic component of TA and at each run a design optimized over $n_R \times n_S = 10^6$ iterations.[5]

For all designs $U(n, q^s)$ in the experiment we set the dimensions $s = 2$, and the number of levels $q = 31$. A smaller number of levels would make the optimization process a trivial task, especially for flexible regions with $m < 1$. The level needs to be of fine enough resolution to well cover smaller regions (for small values of m) and allow the optimization process the opportunity to be effective in lowering the discrepancy once points are moved into the flexible region. Basically, for higher values of q, there are more points to choose from. Having an odd number of levels increases the space available for regions of interest for values $m < 1$.

The number of runs (design points), n, is one of the two varying factors in the experiment. We select four values for n, $\{5,7,9,11\}$. The parameter characterizing the shape of the flexible region, m, is the other varying factor. We select five values for m, $\{9999,2,1,0.5,0.3\}$, which represents a general sample of flexible region shape types. Note that due to the constraint on U-type designs on the q^s grid, the first value results in a covering of all grid points for the given design parameters.

In addition to discussing the stochastic properties of TA applied to design problems, Winker (2006) provides some suggestions as to how to best concentrate computational resources in relation to the rate of convergence towards a global optimum. The algorithm is run with a total of I iterations ($I = n_r \times n_S$, threshold reductions $\times$ design change iterations), and $\mathcal{R}$ replications, the number of times we restart an individual experiment with different random design to start with and different random numbers for the selection of neighbors. Thus, total computational resources used for one problem instance are $C = I \times \mathcal{R}$. If the design matrix is small and the number of levels few, then it is suggested that it is possible to find an optimum design for $\mathcal{R} = \frac{C}{I}$, where $I \geq 5000$ might be sufficient, although most of the instances considered here might be classified as medium sized problem

[5]All code and any other experiment details are available on request.

instances, owing to the number of levels. Of course, the specific shape of the flexible region might possibly alter the problem type classification, e.g., for small m, the actual number of feasible grid points becomes rather small. These subtleties are not explored here. In general, it is not clear how to select I and $\mathcal{R}$ for given C in order to increase the possibility to find good or even optimal designs. The suggestion, in light of the resources at our disposal, is to select a large I and $10 \leq \mathcal{R} \leq 20$ as suggested by Winker (2001, p. 129ff). We select $\mathcal{R} = 10$. The computational load for a single evaluation of the objective function CCD directly influences the upper limit of I. We select $I \in \{100\,000, 1\,000\,000\}$, so we are able to see if the empirical results conform to convergence theory, i.e., the quality of the results improves with increasing I.

5.2 Results

The results in Tables 1 and 2 are for the two values of I. Given that we run 10 replications for each problem instance, the results obtained can be interpreted as an empirical distribution. To provide relevant distributional information along the guidelines suggested by Gilli and Winker (2007), the best and worse objective function values obtained by optimization are reported, along with the empirical mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$, and the frequency at which the best objective function was obtained.

To corroborate the quality of these designs, Figure 2 shows the distribution of points in the flexible region for the designs corresponding to the best discrepancy values reported in Table 2.

It can be seen that the points are evenly distributed over the design space, exploiting the space available as best possible, most noticeably when the flexible region is contracted. Therefore, it appears that using the CCD as a measure of uniformity is suitable for design optimization on flexible regions.

It might be worth mentioning that, e.g., the designs obtained for $m = 0.5$ and five design points in Tables 1 and 2 exhibit the same discrepancy, but have different design patterns as shown in Figure 3. Obviously, the designs can be obtained from one another by a straight 90 degree rotation of points about the center of the region of interest. This is an expected outcome as the CCD meets a key requirement for discrepancy measures, namely to be invariant under rotation of the coordinates.

The statistical distribution of the discrepancy for optimized designs, as discussed by Winker (2006), are assumed to have a left truncated distribution. Furthermore, as I increases, $\hat{\mu}$ should converge to this lower truncation point, while $\hat{\sigma^2}$ decreases to zero. The ten fold increase in I in Table 2, as predicted, produces lower discrepancy values for $m > 0.5$, where it is less

Table 1: CCD Values for Optimized Designs $U(n, 31^2)$ with $I = 10^5$ Iterations, Different Values of n and m

n	Best	Worst	Mean	Std. dev.	Freq. of Best
			$m \to \infty$		
5	6.0681e-003	7.1194e-003	6.4929e-003	4.0000e-004	1/10
7	3.6402e-003	4.6933e-003	3.9936e-003	3.2609e-004	1/10
9	2.3389e-003	3.1732e-003	2.6469e-003	2.5964e-004	1/10
11	1.8123e-003	2.1523e-003	2.0140e-003	1.3008e-004	1/10
			$m = 2$		
5	8.0134e-003	9.6182e-003	8.6211e-003	5.3071e-004	1/10
7	4.6813e-003	5.6177e-003	5.1166e-003	3.0959e-004	1/10
9	3.0772e-003	4.7956e-003	3.4864e-003	5.5378e-004	1/10
11	2.1632e-003	2.8732e-003	2.4491e-003	2.0259e-004	1/10
			$m = 1$		
5	10.7050e-003	10.7472e-003	10.7261e-003	2.2235e-005	4/10
7	5.7358e-003	6.1671e-003	5.7849e-003	1.3447e-004	1/10
9	3.5994e-003	3.9127e-003	3.7410e-003	1.0789e-004	1/10
11	2.4599e-003	3.2180e-003	2.6714e-003	2.1715e-004	1/10
			$m = 0.5$		
5	9.2659e-003	9.4682e-003	9.3266e-003	9.7756e-005	7/10
7	4.8260e-003	4.8260e-003	4.8260e-003	0	10/10
9	3.3279e-003	3.5391e-003	3.4238e-003	6.1905e-005	1/10
11	2.6070e-003	2.7642e-003	2.6936e-003	5.0525e-005	1/10
			$m = 0.3$		
5	8.9374e-003	8.9374e-003	8.9374e-003	0	10/10
7	2.9999e-003	3.0804e-003	3.0080e-003	2.5451e-005	9/10
9	3.3042e-003	4.2841e-003	3.5982e-003	3.4669e-004	5/10
11	1.2396e-003	1.3358e-003	1.2589e-003	4.0559e-005	8/10

Table 2: CCD Values for Optimized Designs $U(n, 31^2)$ with $I = 10^6$ Iterations, Different Values of n and m

n	Best	Worst	Mean	Std. dev.	Freq. of Best
$m \to \infty$					
n	Best	Worst	$\hat{\mu}$	$\hat{\sigma}$	Freq of Best
5	5.7737e-003	6.9729e-003	6.1026e-003	4.2891e-004	1/10
7	3.2194e-003	3.7075e-003	3.4968e-003	1.6254e-004	1/10
9	2.1007e-003	3.8121e-003	2.4168e-003	5.0041e-004	1/10
11	1.5385e-003	2.0494e-003	1.6968e-003	1.4887e-004	1/10
$m = 2$					
5	7.7919e-003	10.9979e-003	8.4362e-003	9.9699e-004	2/10
7	4.3638e-003	4.9518e-003	4.5883e-003	1.9771e-004	1/10
9	2.7309e-003	4.3870e-003	3.0636e-003	5.4863e-004	1/10
11	1.9115e-003	2.2086e-003	2.0380e-003	9.7840e-005	1/10
$m = 1$					
5	10.7050e-003	10.7050e-003	10.7050e-003	0	10/10
7	5.7358e-003	5.7499e-003	5.7390e-003	4.2796e-006	4/10
9	3.4662e-003	3.7450e-003	3.5827e-003	9.9585e-005	2/10
11	2.3577e-003	2.5776e-003	2.4564e-003	6.9362e-005	1/10
$m = 0.5$					
5	9.2659e-003	9.2659e-003	9.2659e-003	0	10/10
7	4.8260e-003	4.9250e-003	4.8359e-003	3.1300e-005	9/10
9	3.3031e-003	3.4597e-003	3.3998e-003	6.1503e-005	3/10
11	2.6070e-003	2.7216e-003	2.6541e-003	4.7005e-005	2/10
$m = 0.3$					
5	8.9374e-003	8.9374e-003	8.9374e-003	0	10/10
7	2.9999e-003	2.9999e-003	2.9999e-003	0	10/10
9	3.3042e-003	3.3042e-003	3.3042e-003	0	10/10
11	1.2396e-003	1.3358e-003	1.2493e-003	3.0420e-005	9/10

Figure 2: Optimized Designs $U(n, 31^2)$ for $I = 10^6$ Iterations, Different Values of n and m

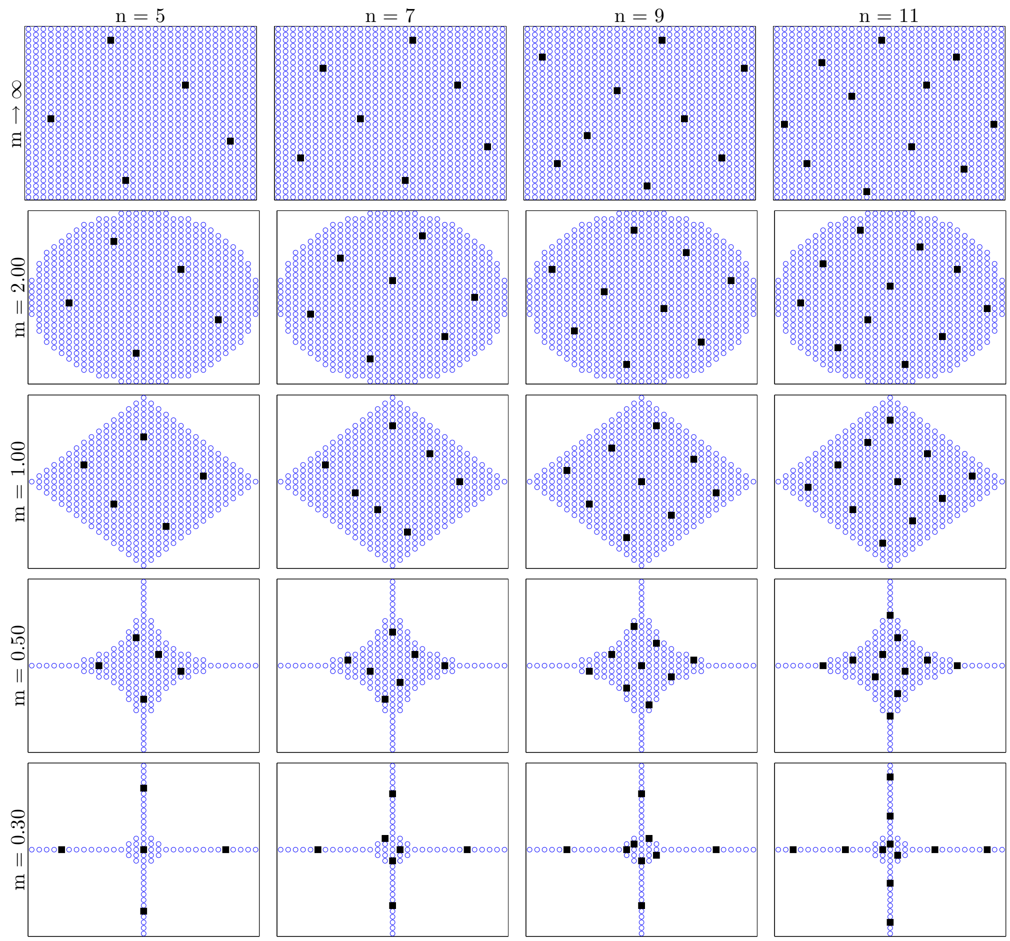
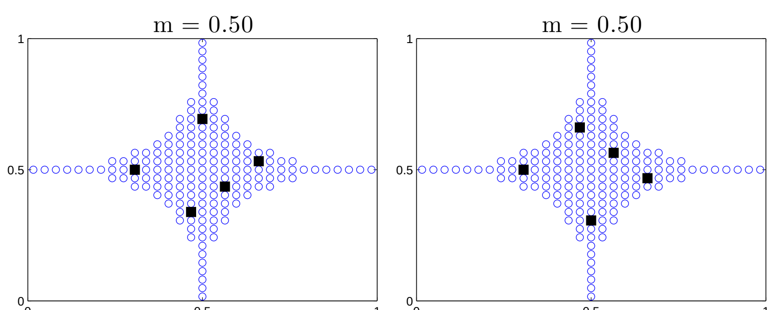
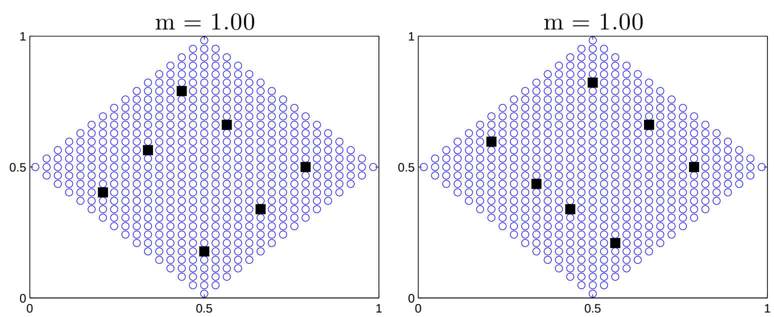


Figure 3: Equivalent Optimized Designs $U(5, 31^2)$



likely to consistently find the global minimum. The mean, $\hat{\mu}$, moves closer to the lowest value found, and the variance $\hat{\sigma^2}$ is reduced. In the case $m = 1$, $U(5, 31^2)$, at row 9, the best value is found in all 10 runs, which suggests – although does not prove – that this is the global minimum with an optimal design. Indeed, as $m < 1$, the frequency of finding the best value increases, and at $m = 0.3$, we have all but the last result always finding the best value. For larger flexible regions, i.e., $m > 1$, we cannot state how close our values of CCD are to the global minimum as no lower bounds are known. Figure 4 shows the difficulty in design optimization for these larger regions. Designs can be produced of the same discrepancy value but where the design pattern is quite different – not just a straight rotation of points.[6] Simply we have more points to play with, but we are still able to produce low discrepancy designs.

Figure 4: Optimized Designs $U(7, 31^2)$ with Equal CCD Value



5.3 Evaluation of Results

So far the results presented could demonstrate that the TA implementation is able to produce designs of low discrepancy on flexible regions. However, given that no lower bounds are available for the CCD, it cannot be proven that these designs are actually optimum designs. We provide some additional evidence on the quality of the results based on a comparison with randomly generated U-type designs on the flexible region. In passing note that considering the projections of uniform designs for the unit cube on the flexible region would not represent a reasonable benchmark as the actual number of points falling into the flexible region is hard to control for. Thus, we construct

[6]In fact, in this case, each design point is a reflection of another with the origin of the axis of reflection at the center of the flexible region.

$I = 10^5$ random designs on the flexible region. The empirical distribution of the CCD values for these randomly generated designs provides a benchmark for evaluating the optimized designs. Some statistics of the empirical distribution are presented in Table 3 in Appendix B.

It can be seen that random designs have a much worse statistical profile than optimized designs in all cases: the CCD values are substantially higher in mean ($\hat{\mu}$) and exhibit a larger standard deviation ($\hat{\sigma}$). Furthermore, it is worth mentioning, that not only the mean results for the optimized designs are always better than the 1%-quantile of the randomly generated designs, but that the same finding even holds for the worst outcomes of the optimization procedure even with only $I = 10^5$ iterations. This serves to reinforce the point that optimization is absolutely worth the additional time and effort required to obtain better designs.

6 Conclusion

In this paper we analyze the construction of designs for flexible regions, as devised by Draper and Guttman (1986), that enclose some region of interest within a larger design space. We select points on a grid from the region of interest, resulting in U-type designs. A TA implementation is used to optimize these U-type designs with regard the CCD, the discrepancy measure applied,is well suited for flexible regions. In the empirical application we consider several design configurations and flexible region shapes. The results indicate that using the TA heuristic, it is possible to produce optimized, low discrepancy designs that distribute design points uniformly over the region of interest. In some cases, in particular for small flexible regions, it might be speculated that the resulting designs are in fact already optimal designs. The same results are found for most design configurations considered.

There are two obvious paths that can be taken for the next course of research on flexible regions. The first is to run experiments and produce results for higher design dimensions, although until now the absence of an analytical formula for the CCD imposes constraints on what can be achieved due to the computational load for calculating CCD for a given design. However, this constraint might be relaxed either by obtaining an analytical formula for the CCD or by making use of efficient updating rules when calculating the CCD value for a slightly modified design in the course of the optimization procedure. The other path is to apply the method to a real meta modeling problem, such as the example described in the original paper by Draper and Guttman (1986) in order to demonstrate to what extent optimized U-type designs on flexible regions might improve the results.

Appendix

A Calculating the CCD

The procedure to calculate the CCD is summarized in Algorithm 2.

Algorithm 2 CCD Calculation Pseudo Code.

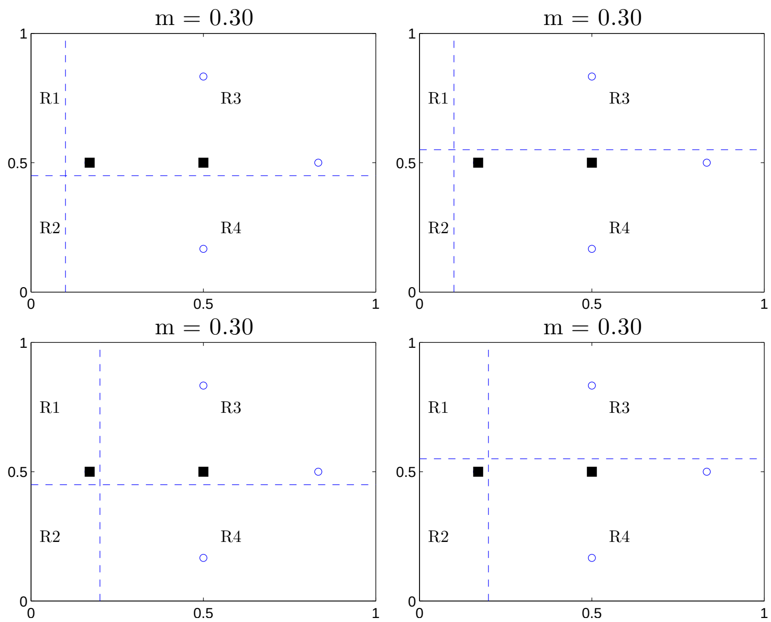
```
 1: initialize the grid point and design volume matrices for subregions
 2: while flexible region not traversed do
 3:     if we are not on a slope of the border, advance to the next grid point row
 4:     for grid points in a row do
 5:         if we are not on the slope of the border, update the grid point and design volume
            matrices
 6:         calculate the discrepancy contribution at each point in the row
 7:         sum the discrepancy contributions for all points in the row
 8:     end for
 9:     sum the total discrepancy contributions for rows
10: end while
11: calculate and return the CCD
```

We shall demonstrate the workings of the algorithm using as example a design $U(2, 3^2)$ and $m = 0.3$, as shown in Figure 5.

The top left hand diagram shows the position of the subregions at the start of a flexible region traversal, where the horizontal and vertical lines mark the border lines between regions and the labels denote region partitions. The design points are represented as black squares and the other grid points by circles. The traversal of the flexible region area is done by row and then column. With exception of a square flexible region, the number of moves left to right will be slightly greater than the number or columns, to account for edges in the slope of a border.

There are several data structures that hold information on a flexible region to calculate the CCD contributions made at each grid point (1:). These are now described. A flexible region profile matrix contains the grid point volume profile and its border coordinates, the column and row numbering of which is taken relative to the square enclosing the flexible region. For the example it is {1,2,1; 2,1,3; 3,2,1}. Taking the first row of the border coordinate matrix, it tells us that the start of the border of the flexible region is at column 1, row 2, and contains in total 1 grid point in the row of that column; and so on and so forth for all the other rows in the matrix. A design point profile is a vector containing the number of design points in each column; another vector contains the row coordinate of each individual design point, and is grouped by column and sorted into ascending order. The design point volume profile is {1, 1, 0}, and the row reference is {1,1}. A grid point volume state matrix

Figure 5: CCD Calculation Example, $U(2, 3^2)$



contains state data for all the regions, and is updated dynamically during the traversal (5:). For the design example its initial state is, {0,0,0; 1,3,1}. The first column represents Regions 1 and 2, and the other column Regions 3 and 4. Similarly, there is a design point state matrix. Its initial state is {0, 0, 0; 0,2,0}.[7]

At each column of the flexible region, we need to detect whether there is a difference in grid points between two successive columns. If this is the case it indicates a slope – a series of edge steps – and determines if the vertical border is advanced and the state of the volume matrices require updating. The number of moves of the horizontal border is from 0 to the number of grid point rows in a column. On the first move, at the left extreme of the flexible region, no changes are made to any of the volume state matrices. A reference index for the volume data structures represents the horizontal border partition for the regions. This index is calculated on each column move by taking the midpoint of the levels and subtracting it from, (grid point rows in a column +1)/2. Therefore, the initial reference index starts at row 2, and is incremented on each row move. The grid point region volumes

[7]As can be seen here, we 'know' we have 2 design points on the second row, but we also need the row coordinate to get the correct index to update the design point volume state matrix.

are, $v(R1) = v(R2) = 0$, $v(R3) = 4$ and $v(R4) = 1$. The design point region volumes are $N(R1) = v(R2) = 0, N(R3) = 2$, and $N(R4) = 0$. The CCD contribution for all regions at a given grid point in a row is calculated by (6:),

$$\frac{1}{2^s} * \sum_{k=1}^{2^s} \left| \frac{\text{grid points in } R_k}{\text{total design points}} - \frac{\text{design points in } R_k}{\text{total grid points}} \right|^2,$$

which is accumulated for all rows in that column and then at every column for the combined row contributions (9:).

The horizontal border line is moved up one row, as shown in the top right hand diagram of Figure 5; the reference index for the volume data structures is incremented to point at row 3. The grid points volumes are $v(R1) = v(R2) = 0, v(R3) = 1$, and $v(R4) = 4$.[8] The design point volumes are $N(R1) = N(R2) = 0$, $N(R3) = 0$, and $N(R4) = 2$.

The vertical border is moved past the first grid point column, and the horizontal border line moved to below the border of the flexible region, as shown in the bottom left diagram of Figure 5. As a slope is detected (the extremes of the flexible region are special cases), this indicates that grid point volume state needs updating every time the vertical border is moved past a grid point. The state updating procedure for the design point volume matrix depends on whether there are any design points in that column, and, if so, whether the design point row coordinate is the same as the reference index, the row value. The grid point volume state update is achieved by subtracting 1 from the grid point volume profile matrix for columns representing regions R3 and R4 at the reference index, and then adding 1 to the columns representing Regions R1 and R2 at the same reference index. The new state is thus, {1,2,1; 0,1,0}. This is the mechanism by which we adjust the volumes for the horizontal partition of the regions. As this process is done dynamically, we only have accurate state data for two horizontal regions, R2 and R4. However, the total grid points in all rows at either side of the vertical border can be obtained from the flexible region profile matrix. We subtract the R2 and R4 values from the vertical region grid point values to obtain the volumes for the remaining regions.

Next, from the design profile vector it is detected that a design point lies somewhere in this column. The row coordinate vector is referenced and its value compared against the row counter. This counter is initialized at current row coordinate of the flexible region border, obtained from the flexible region

[8] In R1 and R2 we can see there is no change as the vertical border is still at the far extreme. In R3 and R4 there is a swap in volume values, as you would expect with the symmetry of grid point distribution in a flexible Region.

profile matrix. The row counter is incremented as the vertical border is moved through the row of grid points. In this instance, as the row counter matches the value in row coordinates vector, we use the row value to perform the subtracting and addition in the design point matrix at the correct index. An index to reference the row coordinates vector is incremented until there is either a change in the coordinate value (no change means that there are multiple design points on that grid point), or there are no more design points in the row. The new state of the design point volume matrix is {0,1,0; 0,1,0}. The grid point region volumes are $v(R1) = 1$, $v(R2) = 0$, $v(R3) = 3$, and $v(R4) = 1$. The design point volume region volumes are $N(R1) = 1$, $N(R2) = 0$, $N(R3) = 1$, and $N(R4) = 0$.

The horizontal border is moved up a row, as shown in the diagram at the bottom right hand corner of Figure 5. The grid point volumes become $v(R1) = 1$, $v(R2) = 0$, $v(R3) = 3$, and $v(R4) = 1$. The design point region volumes are $N(R1) = 1$, $N(R2) = 0$, $N(R3) = 1$, and $N(R4) = 0$. There is no change in state for any of the volume state matrices.

The whole process is repeated until the right extreme of the flexible region is reached and the CCD calculation is completed.

B Results for Random Designs

Table 3: CCD Values of Random Designs $U(n, 31^2)$, Different Values of n and m

n	1%-Quantile	5%-Quantile	Mean	Std. dev.
$m \to \infty$				
5	10.4719e-003	12.9712e-003	28.8015e-003	13.9609e-003
7	7.2431e-003	9.0378e-003	20.6056e-003	10.3125e-003
9	5.4937e-003	6.9260e-003	16.0290e-003	8.1655e-003
11	4.4725e-003	5.6412e-003	13.1449e-003	6.7456e-003
$m = 2$				
5	15.2915e-003	19.0478e-003	44.8859e-003	23.1688e-003
7	10.5268e-003	13.3224e-003	32.2371e-003	17.1538e-003
9	8.1290e-003	10.3040e-003	25.1070e-003	13.5587e-003
11	6.5368e-003	8.4048e-003	20.5922e-003	11.1532e-003
$m = 1$				
5	20.1259e-003	25.4062e-003	62.8825e-003	33.7933e-003
7	13.9074e-003	17.9494e-003	45.3668e-003	25.0180e-003
9	10.7174e-003	13.8306e-003	35.3250e-003	19.7503e-003
11	8.7272e-003	11.3008e-003	29.0611e-003	16.3076e-003
$m = 0.5$				
5	17.8230e-003	23.0175e-003	63.5459e-003	36.3084e-003
7	12.2780e-003	16.1212e-003	45.6648e-003	26.7830e-003
9	9.4691e-003	12.7493e-003	35.8907e-003	21.2384e-003
11	7.8787e-003	10.4734e-003	29.5056e-003	17.5753e-003
$m = 0.3$				
5	14.8372-003	17.0365e-003	60.8469e-003	38.4940e-003
7	7.9344e-003	11.6205e-003	43.5809e-003	28.4498e-003
9	6.3830e-003	8.4889e-003	33.9289e-003	22.5533e-003
11	4.9523e-003	7.3624e-003	27.8753e-003	18.7548e-003

References

Chuang, S.C. and Y.C. Hung (2009). Uniform design over general input domains with applications to target region estimation in computer experiments. *Computational Statistics and Data Analysis* p. forthcoming.

Draper, N.R. and I. Guttman (1986). Response surface designs in flexible regions. *Journal of the American Statistical Association* **81**, 1089–1094.

Draper, N.R. and W.E. Lawrence (1965). Designs which minimise model inadequacies; cuboidal regions of interest. *Biometrika* **52**, 111–118.

Draper, N.R. and W.E. Lawrence (1966). The use of second-order 'spherical' and 'cuboidal' designs in the wrong regions. *Biometrika* **53**, 496–599.

Dueck, G. and T. Scheuer (1990). Threshold accepting. a general purpose optimization algorithm superior to simulated annealing. *Journal of Computational Physics* **90**, 161–175.

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

Fang, K.-T. and P. Winker (1997). Application of threshold accepting to the evaluation of the discrepancy of a set of points. *SIAM Journal on Numerical Analysis* **34**(5), 2028–2042.

Fang, K.-T., C.-X. Ma and P. Winker (2002). Centered l_2-discrepancy of random sampling and latin hypercube design, and construction of uniform designs. *Math. Comput.* **71**, 275–296.

Fang, K.-T., D. Maringer, Y. Tang and P. Winker (2005). Lower bounds and stochastic optimization algorithms for uniform designs with three or four levels. *Mathematics of Computation* **75**(254), 859–878.

Fang, K.-T., D.K.J. Lin, P. Winker and Y. Zhang (2000). Uniform design: Theory and application. *Technometrics* **42**, 237–248.

Fang, K.-T., R. Li and A. Sudjianto (2006). *Design and Modeling for Computer Experiments*. Chapman & Hall/CRC. Boca Raton, FL.

Fang, K.-T., X. Lu and P. Winker (2003). Lower bounds for centered and wrap-around l2-discrepancy and construction of uniform designs by threshold accepting. *Journal of Complexity* **19**, 692–711.

Gilli, M. and P. Winker (2007). Editorial - 2nd special issue on applications of optimization heuristics to estimation and modelling problems. *Computational Statistics and Data Analysis* **52**(1), 2–3.

Hickernell, F.J. (1996). A generalized discrepancy and quadrature error bound. *Mathematics of Computation* **67**, 299–322.

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

Winker, P. (2001). *Optimisation Heuristics in Econometrics: Applications of Threshold Accepting*. Wiley. Chichester.

Winker, P. (2006). The stochastics of Threshold Accepting: Analysis of an application to the uniform design problem. In: *COMPSTAT 2006, Proceedings in Computational Statistics* (A. Rizzi and M. Vichi, Eds.). pp. 495–503. Physica. Heidelberg.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control* **8**, 338–353.

Zadeh, L.A. (1987). *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*. John Wiley and Sons. New York.