NUMBER THEORETIC METHODS IN DESIGNING EXPERIMENTS

by

Ismael Suleman Talke

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Statistics

MONTANA STATE UNIVERSITY
Bozeman, Montana

June, 2012

APPROVAL

of a dissertation submitted by

Ismael Suleman Talke

This dissertation has been read by each member of the dissertation committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to The Graduate School.

Dr. John J. Borkowski

Approved for the Department of Mathematical Sciences

Dr. Kenneth L. Bowers

Approved for The Graduate School

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

Ismael Suleman Talke

June, 2012

DEDICATION

This dissertation is dedicated to my beloved mother, beautiful wife and son.

ACKNOWLEDGEMENTS

First and for most, I would like to give my sincere gratitude to my advisor, Professor John J. Borkowski, for his caring, tireless guidance, assistance, encouragement and quality supervision throughout the entire process. Thank you!

My special thanks also go to my committee members, Professors Steve Cherry, Jim Robison-Cox, Mark Greenwood and Jeffrey Banfield, for their valuable time, useful comments and suggestions that has helped improve the quality of this dissertation.

I thank everyone though not mentioned by name but who have greatly contributed to my academic success.

Finally, I would like to thank my wife for her love and encouragement during the good and difficult times.

TABLE OF CONTENTS

TABLE OF CONTENTS - CONTINUED

LIST OF TABLES

LIST OF TABLES - CONTINUED

Table Page

LIST OF TABLES - CONTINUED

Table Page

LIST OF FIGURES

Figure Page

LIST OF FIGURES - CONTINUED

Figure Page

LIST OF FIGURES - CONTINUED

Figure Page

ABSTRACT

The objective of this dissertation is to explore and contrast several methods of generating robust space-filling uniform designs (UDs) and provide tables of UDs for practitioners to use. Four methods of constructing UDs considered in this research are the good lattice point (glp) method, the cyclotomic field (CF or good point (gp)) method, the Halton method and the Hammersley method.

UD tables are provided for experimental regions in the s dimensional unit cube C^s and sphere B^s for $s = 2, 3, 4$ and for mixture UDs for the q component simplex S^q for $q = 3, 4, 5$. For each of these experimental regions the glp method was found to be the best method for generating UDs. However, the glp method is computationally expensive. To circumvent this problem two methods have been proposed to reduce the computational cost of the glp method. These are the equivalence method and the projection method. The projection method is easier to implement and is the recommended alternative especially for higher dimensions.

Two classes of measures to assess uniformity (i.e., space-filling properties of a design) used in this dissertation are new discrepancy measures and distance-based measures. The new discrepancy measures are proposed for all the aforementioned experimental regions. Specifically, the modified star discrepancy (MSTRD) is proposed for a unit cube C^s and new discrepancy measures are also proposed for two and three dimensional spherical regions B^2 and B^3. Directional systematic discrepancy (DSD) and the non-directional random discrepancy (NDRD) are proposed for the simplex S^3 for a 3 component mixture experiment. However, for mixture experiments with single component constraints (SCCs) the NDRD is the new proposed measure of discrepancy. In higher dimensions, the distance-based criteria are easier to implement and hence are the recommended alterative to discrepancy measures. Generating UDs for mixture experiments with multiple component constraints (MCCs) is complicated and, hence, a modification of the one-pass exchange algorithm of Borkowski and Piepel (2009) that generates UDs for experiments with MCCs is presented. This new method is called the one-step exchange algorithm.

INTRODUCTION

Background

Experimental design is a statistical methodology used for studying the relationship between the experimental factors on one or more responses of interest (Myers et al., 2009). Statistical experimental designs, such as full factorial designs, fractional factorial designs, response surface designs (including optimal designs, orthogonal arrays, and central composite designs) have been extensively and widely applied in different scientific areas (Zhang et al., 1998). For example, in industrial applications, experimental design has played an important role in product and process design (Pham, 2006). Industrial experiments often have many factors, and as a result, there is always a need for a method that can identify those factors that have large effects. If there are enough resources to run all combinations of the factor levels, then a full factorial design is often used. However, the availability of enough resources to run a full factorial design often does not exist. Thus, when it is impossible to run a full factorial design, a fractional factorial design is frequently the alternative design of choice.

Response surface designs are typically used to fit polynomial regression models. For fitting a second-order (or quadratic) polynomial regression model, the central composite designs (CCDs) and Box-Behnken designs (BBDs) are commonly used (Myers et al., 2009). When the goal is optimization of certain prediction properties (such as parameter estimation or the prediction variance) for a pre-specified regression model and the classical response surface designs (e.g., CCDs and BBDs) cannot be used due to design size constraints, an optimal (computer-generated) design may be appropriate (Li et al., 2004).

In general, a good experimental design should be efficient. That is, it should minimize the number of experimental runs while still providing an acceptable amount of information. For complete details of the above mentioned experimental designs see Myers et al. (2009).

The aforementioned classes of experimental designs are often based on pre-specified regression models with the experimental data analyzed using analysis of variance (ANOVA) methods. In general, the underlying response model can be written as

$$Y(\mathbf{x}) = h(\mathbf{x}) + \varepsilon \tag{1.1}$$

where $\mathbf{x} = (x_1, ..., x_s)$ is the vector of s input factors, $h(\mathbf{x})$ is the mean response at input $\mathbf{x}$, and ε is a random error term assumed to have zero mean. Thus, $h(\mathbf{x})$ is an experimental output with the random error removed, and $E[Y(\mathbf{x})] = h(\mathbf{x})$ (Fang et al. (2000) and Fang et al. (1994)).

One goal of an experimental design may be to study the relationship between the response (output variable) Y and the vector $\mathbf{x}$ of inputs (factors) (Zhang et al., 1998). For example, suppose an industrial engineer plans to study the yield (Y) of a product produced from a chemical process under different reaction conditions ($\mathbf{x}$) with the goal of gaining an understanding how the reaction conditions affect the yield (Pham, 2006). Other common goals may include the estimation of the maximum, minimum, or expected value $E(h(\mathbf{x}))$ of the response over all $\mathbf{x}$ in the experimental domain.

To optimize a process (such as the maximization of process yield), understanding the relationship between the response and the input (e.g., via modeling) has to be established. However, it is difficult, if not impossible, to determine the underlying theoretical relationship between the input ($\mathbf{x}$) and the response $Y(\mathbf{x})$. That is, in many experiments the function $h(\mathbf{x})$ in equation (1.1) is unknown. However, we can rewrite $h(\mathbf{x})$ as the sum of a linear form and the bias of the linear form. That is,

$$h(\mathbf{x}) = \beta' g(\mathbf{x}) + f(\mathbf{x})$$

where $g(\mathbf{x}) = (1, g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_{p-1}(\mathbf{x}))$ is a vector of specified functions, $\beta = (\beta_0, \beta_1, \ldots, \beta_{p-1})$ is the vector of unknown parameters, and $f(\mathbf{x}) = h(\mathbf{x}) - \beta' g(\mathbf{x})$ is the unknown bias. Thus, the model in (1.1) can be rewritten as

$$Y(\mathbf{x}) = \sum_{i=0}^{p-1} \beta_i g_i(\mathbf{x}) + f(\mathbf{x}) + \varepsilon = \beta' g(\mathbf{x}) + f(\mathbf{x}) + \varepsilon. \tag{1.2}$$

The experimenter then wants to find a linear model $\beta' g(\mathbf{x})$ that provides a suitable approximation of $h(\mathbf{x})$. Or, in other words, the bias $f(\mathbf{x})$ is negligible for all $\mathbf{x}$ in the experimental domain (Fang et al., 2000; Wiens, 1991). Assuming $f(\mathbf{x}) \approx 0$ for all $\mathbf{x}$ leads to the regression model

$$Y(\mathbf{x}) = \sum_{i=0}^{p-1} \beta_i g_i(\mathbf{x}) + \varepsilon = \beta' g(\mathbf{x}) + \varepsilon \tag{1.3}$$

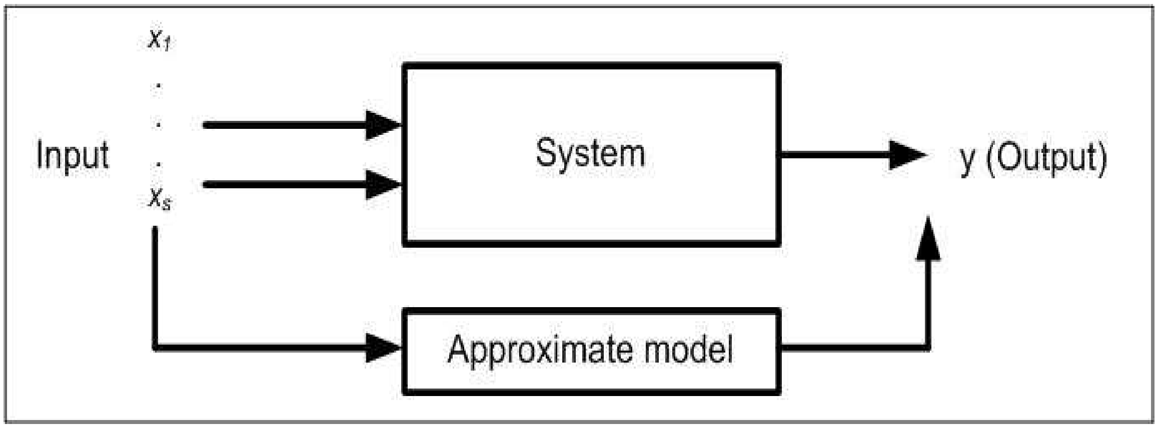
The practical use of classical experimental designs for estimating the parameters in vector β in equation (1.3) is based on application of statistical distributions that depend on model assumptions. A common set of assumptions for the linear relationship between the dependent (output) variable and functions of independent (input) variables are the independence of errors, homoscedasticity (constant variance) and the normal distribution of errors. As a result the performance of classical designs will be best when these model assumptions are met.

However, even if the model assumptions are met, other problems will exist when the number of factors and, thus, the number of possible combinations of factor levels, becomes large. For example, suppose there are s factors each having q levels. Then a full factorial experimental design will require n=$r \times q^s$ experimental runs, where r is the number of replications (Zhang et al., 1998). Even though a fractional factorial design (i.e., a subset of a full factorial design) can be used to reduce the number of experimental runs significantly, designs with a small number of levels are often unsatisfactory when the structure of the model is unknown (Pham, 2006). This happens because the most commonly-used fractional factorial designs are unreplicated two-level (2^{s-p}) and three-level (3^{s-p}) designs where p is an integer less than s and q^{-p} is the fraction size (Myers et al., 2009). The two-level and three-level fractional factorial designs are often special cases of orthogonal arrays (Liang et al., 2001). The aforementioned problems, that is,

1. the relationship between the mean response and the factors being unknown,
2. the case when the number of factors is large and number of combinations of factor levels is large, and
3. the imposition of distributional assumptions associated with a specified models,

limit the practical application of many classical experimental designs. When modeling under such conditions, the experimenter would like to explore the entire experimental domain. This can be accomplished by selecting a "space-filling" design whose design points are scattered uniformly over the entire experimental domain (Fang and Lin, 2003; Fang et al., 2000). Different types of space-filling designs have been used in computer experiments. Computer experiments generate observations from a computer model from a set of input variables (Sacks and Welch, 1989). Computer experiments are different from traditional physical experiments because the response is deterministic. That is, repeated observations at the same inputs yield the same

Figure 1.1: Illustration of computer experiment



response by the computer model (Santner and Notz, 2003). Thus, replication is not needed in computer experiments. Uncertainty in computer experiments comes from the fact that the relationship between the input variables and the response is unknown. Computer experiments are often used in science and engineering to describe complex physical process or systems that might be linear or non-linear equations (Fang and Lin, 2003). A description of a computer experiment is outlined in Figure 1.1 and is given in (Fang and Lin, 2003). When the process or systems are complex it is difficult to simultaneously solve the equations. For example, engineers may be interested in modeling tsunami (water waves) but they cannot perform the experiment. Instead they use a computer experiment to simulate the characteristics of the process or the system to find an approximation for the true model.

Uniform Designs (UDs) are a class of experimental designs that have been used successfully to examine the relationship between the input and output variables when the underlying function $h(.)$ in the model given in (1.1) is unknown, or equivalently, h is unknown in the model

$$Output \ = \ h(input) \ + \ random\ error.$$

Unlike the classical experimental designs that are defined in terms of combinatorial structures, UDs are based on measures of the spread of the design points over the experimental domain. UDs are often used for estimation related to a nonparametric regression model, such as the estimation of $E(h(\mathbf{x}))$ in equation (1.1). In contrast, classical experimental designs are often focused on parameter estimation in parametric regression models, such as the estimation of unknown parameters in an approximating polynomial model of $h(\mathbf{x})$ in equation (1.3) (Fang et al. (2000)).

Wiens (1991) addressed the relationship between uniform designs and model lack-of-fit. Suppose equation (1.2) is considered to be the "full" model and equation (1.3) as the "reduced" model. Note that equation (1.2) and equation (1.3) are the same if the bias $f(\mathbf{x}) = 0$ for all $\mathbf{x}$ in the experimental region. For a design with repeated observations, the usual test for model lack-of-fit can be performed to test the fit of the "reduced" model within the "full model" based on the test statistic

$$F = \frac{(n-p)S^2_{REG} - (n-c)S^2_{PE}}{(c-p)S^2_{PE}}$$

where S^2_{REG} is the least squares estimate of σ^2, and c is the number of distinct replications, p is the number of parameters, and

$$S^2_{PE} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n_i}(y_{ij} - \bar{y})^2}{(n-c)}$$

where S^2_{PE} is the Pure Error estimate.

Furthermore, suppose that the following two conditions hold:

1. $\int_{C^s} f^2(\mathbf{x})d\mathbf{x}$ is bounded from below.

2. $\int_{C^s} \beta' g(\mathbf{x})f(\mathbf{x})d\mathbf{x} = 0$.

Then, assuming equation (1.2) is true, Wiens (1991) showed that the continuous uniform design maximizes the minimum power of the lack-of-fit test.

Next, consider the case where the design may or may not have repeated observations, and suppose that the following two conditions hold:

1. $\int_{C^s} f^2(\mathbf{x})d\mathbf{x}$ is bounded from above.

2. $\int_{C^s} \beta' g(\mathbf{x})f(\mathbf{x})d\mathbf{x} = 0$.

Then, assuming equation (1.2) is true, Wiens (1991) showed that the continuous uniform design minimizes the maximum bias in the estimation of σ^2.

For both cases, these results apply to continuous designs which represent a probability measure defined over the design space. Thus, continuous designs are conceptual in nature, and not necessarily implementable in practice. For practical implementation, Wiens (1991) states that an experimenter would approximate the continuous uniform design with a discrete uniform design with points "equally spaced throughout" the design space and with equal number of replicates. As a result of these findings, one does not need to know the model to select a reasonable design. A simulation study example presented in the last chapter of this dissertation considers the presence of systematic bias caused by model lack-of-fit and shows how robust uniform designs can be.

The uniform designs (UDs) that were proposed and introduced by Fang (1980) and Wang and Fang (1981) are based on *number-theoretic methods* (NTM's). They are also known as *quasi Monte-Carlo methods*. These UDs address the stated problems concerning classical experimental designs. The NTMs are methods that represent a combination of number theory and numerical analysis (Fang et al., 1994). One appealing characteristic of NTMs in comparison to Monte-Carlo methods is that Monte-Carlo methods use random points between zero and one to approximate multiple integrals whereas the quasi Monte-Carlo methods use a more uniformly scattered set of points. Reconsider the problem of estimating $E(h(\mathbf{x}))$ over the experimental design region. Assume, as in most of the previously published studies, that the experimental design region for a s-factor experiment is the s-dimensional unit cube C^s defined by $C^s{=}[0,1]^s$. In this case,

$$E(h(\mathbf{x})) = \int_{C^s} h(\mathbf{x})\, d\mathbf{x}. \tag{1.4}$$

Because $h(\mathbf{x})$ is unknown, a closed form cannot be given for the integral in (1.4). As a result, numerical methods have to be used to approximate the integral. One estimation approach is to use Monte-Carlo methods to generate n random points in the experimental region $\mathcal{P}$ and then estimate $E(h(\mathbf{x}))$ by the following sample mean:

$$\overline{h}_n = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{P}} h(\mathbf{x}) \tag{1.5}$$

for the deterministic case when there is no random error (i.e., the observed response is $h(\mathbf{x})$), or,

$$\overline{h}_n = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{P}} Y(\mathbf{x}) \tag{1.6}$$

for the case when there is random error (i.e., $Y(\mathbf{x}) = h(\mathbf{x}) + \varepsilon$ is the observed response). For the deterministic case, the rate of convergence using the Monte-Carlo method is

$$\left|E(h(\mathbf{x})) - \overline{h}_n\right| \leq O(n^{-1/2}).$$

However, when the NTM method is used, one gets a better rate of convergence (Fang et al., 1994; Fang and Wang, 1994) which was shown to be

$$\left|E(h(\mathbf{x})) - \overline{h}_n\right| \leq O(n^{-1}(\log n)^s).$$

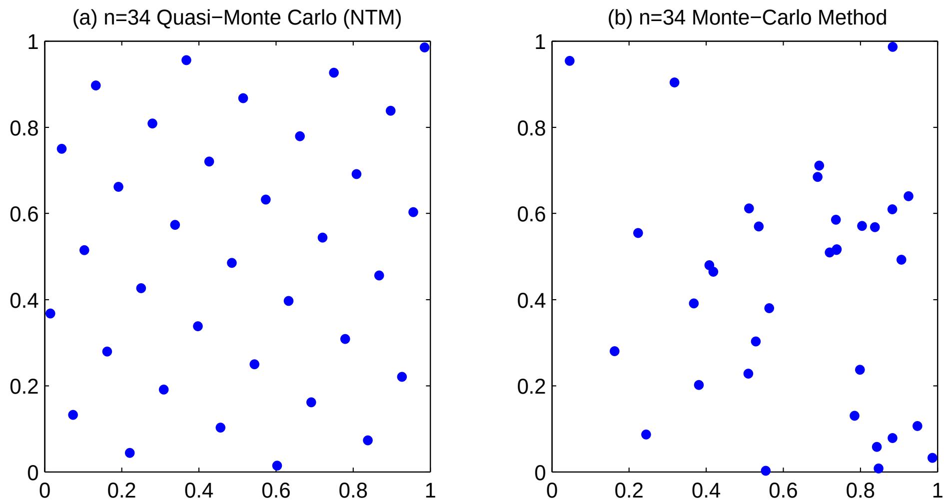
For example, consider $n = 34$ design points in C^2 (a unit square domain) generated using NTM's (quasi Monte-Carlo) as well as Monte-Carlo methods. Figure 1.2 shows that the points of this UD generated by the quasi Monte-Carlo method (Fig. 2a) provides better coverage and representation than the Monte-Carlo set of points (Fig. 2b) throughout the experimental region. To illustrate this further, consider a random vector $\mathbf{x} = (x_1, x_2)$ from the bivariate normal distribution $N(0, I_2)$ where I_2 is a 2×2 identity matrix, and the problem of interest is how to evaluate $E(h(\mathbf{x}))$ in a unit square C^2. Mathematically,

$$E(h(\mathbf{x})) = \int_{C^s} h(\mathbf{x})d\mathbf{x} = \int_0^1 \int_0^1 (2\pi)^{-1} \exp\left(\frac{-1}{2}(x_1^2 + x_2^2)\right) dx_1 dx_2. \tag{1.7}$$

See Casella and Berger (2002).

Exact evaluation of the double integral gives $E(h(\mathbf{x})) = [\Phi(1) - \Phi(0)]^2 \approx 0.1165162$ where $\Phi(x)$ is the cumulative distribution function for the standard normal distribution $N(0,1)$. The sample mean estimate $\overline{h}_n$ of $E(h(\mathbf{x}))$ from a set of n points generated by the Monte-Carlo method and from the n points ($n = 21, 34, 45$) generated by the NTMs or quasi Monte-Carlo method was computed and presented in Table 1.1.

Figure 1.2: 34 design points from NTM's and Monte-Carlo methods



The good lattice point (glp) method with generating vectors (21;1,13), (34;1,13) and (45;1,19) were used to construct the NT designs in Table 1.1. The glp method will be discussed later in Chapter 2. The absolute error or difference $|E(h(\mathbf{x})) - \overline{h}_n|$ in Table 1.1 shows that the NTM or the quasi Monte-Carlo method provides a better estimate of $E(h(\mathbf{x}))$ than the estimate resulting using monte carlo methods. This is consistent with a visual assessment of the scatter of the points in Figure 1.2. The space-filling property of NTMs is an appealing characteristic of NTMs. This is the primary reason why NTMs are considered as a basis for generating space-filling experimental designs. Specifically, NTMs can generate uniformly scattered sets of points throughout the experimental domain.

For $s = 3$ dimensions, Fang et al. (1994) presented an example for the multivariate normal distribution with 3 variables and different design sizes.

	$\overline{h}_n$	$\overline{h}_n$	$\lvert E(h(\mathbf{x})) - \overline{h}_n \rvert$	$\lvert E(h(\mathbf{x})) - \overline{h}_n \rvert$
n	NTMs	MCMs	NTMs	MCMs
21	0.1169	0.1063	3.6102e-004	0.0102
34	0.1168	0.1133	2.7367e-004	0.0033
45	0.1167	0.1126	1.5766e-004	0.0039

Table 1.1: Comparison between the Monte-Carlo (MCMs) and Quasi Monte-Carlo (NTMs) Methods for different n size

An upper bound for the mean difference between the true ($E(h(\mathbf{x}))$) and approximated ($\overline{h_n}$), is given by the Koksma-Hlawaka (K-H) inequality

$$\left| E(h(\mathbf{x})) - \overline{h}_n \right| \leq D(\mathcal{P})\mathcal{V}(h) \tag{1.8}$$

where $D(\mathcal{P})$ is the measure of how uniformly (evenly) the design points ($\mathcal{P}$) are scattered. Specifically, it is a discrepancy measure of the design points in the experimental region which will be discussed in detail later in the next chapters. $\mathcal{V}(h)$ is a measure of variation of h that is used as an indicator of the regularity (bounded) condition of the function h on the experimental region $[0,1]^s$ (Fang et al., 2006, 2000; Krommer and UeberHuber, 1998; Liang et al., 2001; Niederreiter, 1992). For example, consider a univariate case where the experimental region is a line segment [0,1] and h is a function, $h : [0,1] \to \Re$. If one considers the following n subintervals $[x_{i-1}, x_i], i = 1, 2, ..., n$ in [0,1], defined by $C = \{x_i : x_0 = 0 < x_1 < ... < x_{n-1} < x_n = 1\}$ then

$$\Delta\mathcal{V}(h, C) = \sum_{i=1}^{n} |h(x_i) - h(x_{i-1})|,$$

is the variation of h for this set of subintervals (Krommer and UeberHuber, 1998). The variation of h, $\mathcal{V}(h)$, is the supremum of all such sums of differences of the function across all partitions C of $[0, 1]$. That is,

$$\mathcal{V}(h) = \sup_{C}\{\Delta\mathcal{V}(h, C)\}.$$

The extension of the variation function $\mathcal{V}(h)$, to a multivariate case, and detailed discussions are presented in (Krommer and UeberHuber, 1998) and Niederreiter (1992). One indication that the UDs can lead to good estimates of $E(h(\mathbf{x}))$ for a large class of $h(\mathbf{x})$ is because $\mathcal{V}(h)$ remains unchanged against the change of h (Fang et al., 2000).

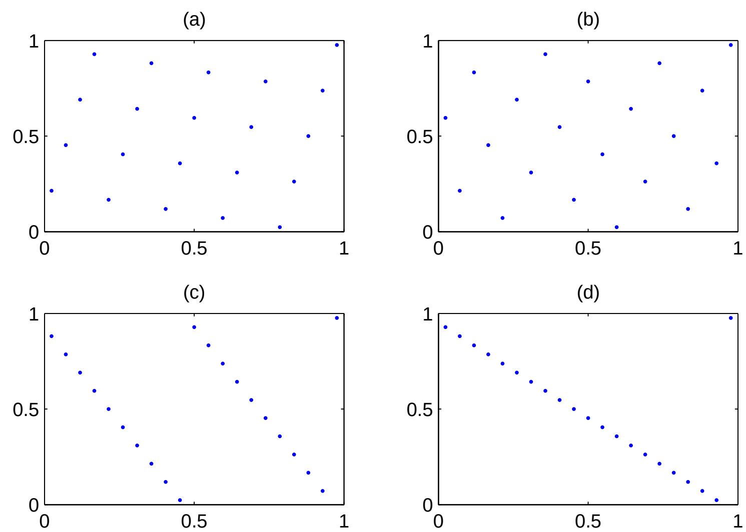
The development of UDs was motivated by the high demand and need from engineers that started more than 3 decades ago (Liang et al., 2001). In China during that period, an industrial problem was proposed where there were 6 factors each requiring at least 12 levels. Experimentation was expensive and the number of experimental runs was restricted to be at most 50. Under this restriction, a classical fractional factorial design was not possible because the number of factor levels was too large. However, when a UD was applied to the problem, a satisfactory result was achieved using only 31 experimental runs with each factor having 31 levels (Liang et al., 2001).

Another example is a complex quantitative risk assessment of a mixture of toxic chemicals (Fang and Wang, 1994). The experiment had 6 factors each having 17 levels. Therefore, the total number of potential factorial combinations would be $17^6 = 24,137,569$ which was impractically large for any lab to handle. An orthogonal array could be used, but it would require at least $17^2 = 289$ experimental runs which was still considered too large. Thus, a UD with 17 experimental runs provided a practical solution for this problem (Fang and Wang, 1994). Since then, UDs have been used extensively in China in many fields including agriculture, the textile industry, military science, chemistry and chemical engineering (Liang et al., 2001). Extensive reviews of UDs are given in (Fang and Wang, 1994; Fang and Lin, 2003; Liang et al., 2001).

Uniform experimental designs are one type of space-filling designs. A good space-filling design is one in which the design points are uniformly scattered throughout the experimental region (Borkowski and Piepel, 2009). Historically, a set of points was said to be uniformly scattered if it had a small discrepancy (Fang and Wang, 1994). More specifically, when the given experimental domain is the unit cube $C^s = [0,1]^s$ and the design has n experimental runs, the goal when assessed by a discrepancy measure is to find a design $\mathcal{P}_n = (X_1, ..., X_n)$ where $X_i \in C^s$ such that the deviations between the true and estimated approximation model is as "small" as possible for all $X \in C^s$. Thus, the goal is to make the discrepancy measure as small as possible. Or, in other words, when $\overline{h}_n$ and $E(h(\mathbf{x}))$ are very close, then the discrepancy measure is small. Generally, this occurs when the set of design points $\mathcal{P}_n$ is 'uniform' or 'uniformly scattered' in the design region (Fang et al., 2000; Fang and Wang, 1994; Fang et al., 2006). It is worth noting that a small discrepancy only indicates uniformly scattered points in a spatial sense, and should not be interpreted that the set of design points is uniformly distributed in the usual statistical meaning of a uniform probability distribution. Space-filling designs are ideally suitable for deterministic computer models because the design points are nearly evenly or uniformly spread throughout the entire experimental region (Myers et al., 2009; Santner and Notz, 2003). Various discrepancy measures exist to address the degree of space-filling uniformity. Examples of discrepancy measures include the star L_2 discrepancy, centered discrepancy and wrap-around discrepancy measures of uniformity (Hickernell, 1998a,b; Fang et al., 2006).

A further clarification of a good space-filling representation in the experimental domain is illustrated in Figure 1.3. Figure 1.3 shows the design points from four

Figure 1.3: Space-filling for $n = 21$ design points



designs generated by the 'good lattice point' method from four different generating vectors. Each design contains 21 points scattered in an experimental region that is the unit square C^2. In Figures 1.3 (a) and (b), the points are uniformly scattered throughout the experimental region unlike Figures (c) and (d) which contains large regions without any design points. The points are not uniformly scattered in Figures (c) and (d), and thus represent poor designs. Designs (a) and (b), however, are good space-filling designs because the sampled points are more representative of the entire experimental region due to the uniformity of the scatter. In the subsequent sections, methods for choosing the best design using the good lattice point method which are based on quantifying the degree of uniformity using different measures of uniformity will be presented. For example, the designs in Figure 1.3 (a) and (b) are clearly better than (c) and (d). However, to decide which is design in (a) and (b) is better, a measure of discrepancy (lack of uniformity) is needed to allow direct comparison of designs.

Advantages Of Uniform Designs (UDs)

When using a classical experimental design, such as a 2^k factorial or central composite design (CCD), it is assumed that the underlying model form is known but with unknown model parameters. In this case, a design is chosen that will produce parameter estimates having high precision (Fang and Lin, 2003). In comparison, the focus of a UD is not on fitting a specific model, but to collect data at design points that are representative of the experimental domain. This will be particularly valuable when the experimenter is unsure of the structure of the underlying response surface model. In addition to estimating $E(h(\mathbf{x}))$ without knowing the functional form of $h(\mathbf{x})$, with UDs the researcher has the flexibility of fitting a wide variety of possible models due to the space-filling properties of the experimental data that is collected. Good space-filling designs are also desirable for data analysis methods in fitting nonparametric response surface models.

A UD can also be considered as a special type of a highly fractionated factorial design as it reduces the number of factorial runs significantly while still providing useful information about the relationship between the input variables and the measured output response (Fang et al., 2000). A UD allows the largest possible number of levels for each factor among all experimental designs for a fixed number n of design points. Thus, a UD is robust to potential model misspecification because it can still prove useful even when the exact form of the model is not known (Zhang et al., 1998; Li et al., 2004; Liang et al., 2001). However, unlike most fractional factorial designs, UDs usually are not orthogonal which may create some difficulty in data analysis (Li et al., 2004). This is because a non-orthogonal design might have collinearity problems which affects parameter estimation.

For industrial experiments when model fitting is required, the following steps are necessary to conduct a successful experiment with a uniform design (Pham, 2006). Note that most of these steps are common to all industrial experiments.

1. Identify the aim of the experiment, and identify the response to examine.
2. Specify the factors and the domain for each factor.
3. Specify the maximum number of experimental runs.
4. Because uniform designs allow for the maximum number of levels to be studied, choose a sufficiently large number of factor levels.
5. Choose a uniform design.
6. Perform an experiment using the design chosen in step 5.
7. Perform the statistical modeling and analysis. This includes performing diagnostic tests for the fitted models to fulfill the aim specified in step 1.
8. If possible, conduct additional experimental runs to verify and validate the modeling and analysis results and, thereby, achieve the aim of the experiment.

Spherical Design Regions

In Response Surface Methodology (RSM), a 2-dimensional spherical design region is defined to be the 2-dimensional ball of radius $\sqrt{2}$. That is,

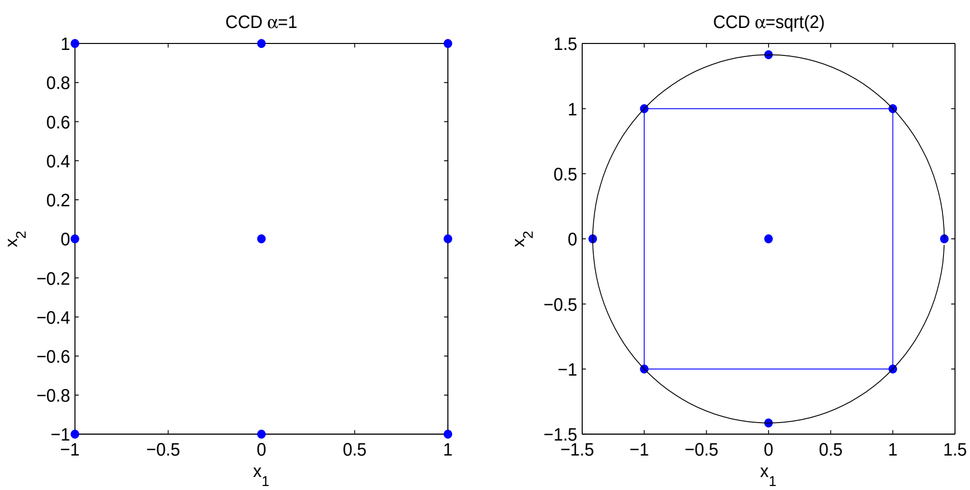
$$B^2(\sqrt{2}) = \left\{(x_1, x_2) : x_1^2 + x_2^2 \leq 2\right\}.$$

We can generalize this into k-dimensional ball notation as $B^k(\rho)$ to represent a ball scaled to a radius ρ. Thus, in RSM, the k-dimensional spherical design region can be denoted $B^k(\sqrt{k})$. This may lead one to ask "What is the advantage of working with a spherical experimental region?".

To begin addressing this question, consider central composite designs (CCDs) with $n = 9$ points having $\alpha = 1$ and $\alpha = \sqrt{2}$. In a CCD, the choice of α depends on the experimental region. For $\alpha = 1$ and $\alpha = \sqrt{2}$, the CCD experimental regions are the square C^2 (cuboidal in 2 dimensions) and the circle $B^2(\sqrt{2})$ (spherical in 2 dimensions), respectively. The variable settings of the coded levels for the 9 points in a CCD with $\alpha = 1$ or $\alpha = \sqrt{2}$ are (x_1, x_2)=$(\pm1, \pm1)$,$(\pm\alpha, 0)$,$(0, \pm\alpha)$, and $(0, 0)$.

The graphical display of the two experimental regions and the 9 points in each CCD are shown in Figure 1.4.

Figure 1.4: Central composite design with $\alpha = 1$ and $\alpha = \sqrt{2}$ for $n = 9$



The choice of α depends on the region of operability and the region of interest. This is because for scientific reasons the region of operability and interest are not the same. For example, changing $\alpha = 1$ to $\alpha = \sqrt{2}$ moves the axial points $(\pm1, 0)$ and $(0, \pm1)$ in the square (C^2) experimental region to the points on the circle shown in

Figure 1.4 yielding a spherical $B^2(\sqrt{2})$ (i.e., circular) experimental region. This can happen when there is interest in predicting outside the cube, and this also improves the prediction performance of the cubical region (C^2).

In regression modeling it is common to use the coded variables rather than the actual variables. For the CCD with $\alpha = 1$ displayed in Figure 1.4, the coded variables are:

$$x_{i1} = \frac{\xi_{i1} - \left(\frac{\max(\xi_{i1}) + \min(\xi_{i1})}{2}\right)}{\left(\frac{\max(\xi_{i1}) - \min(\xi_{i1})}{2}\right)} \quad \text{and} \quad x_{i2} = \frac{\xi_{i2} - \left(\frac{\max(\xi_{i2}) + \min(\xi_{i2})}{2}\right)}{\left(\frac{\max(\xi_{i2}) - \min(\xi_{i2})}{2}\right)}$$

for $i = 1, 2, ..., 9$ where ξ_{i1} and ξ_{i2} are the actual uncoded levels of the variables corresponding to the coded variables x_{i1} and x_{i2}, respectively (Myers et al., 2009). The maximum and minimum values of ξ_{i1} and ξ_{i2} are set by the experimenter.

Reconsider the $n = 9$ point CCDs with $\alpha = 1$ and $\alpha = \sqrt{2}$, and suppose we fit the second order model given by

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon = X\beta + \epsilon,$$

where the response vector is Y, the model matrix is X, the vector of regression coefficients is β, and the vector of error terms is ϵ. More explicitly,

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_9 \end{bmatrix}, \ X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{11}^2 & x_{12}^2 & x_{11}x_{12} \\ 1 & x_{21} & x_{22} & x_{21}^2 & x_{22}^2 & x_{21}x_{22} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_{91} & x_{92} & x_{91}^2 & x_{92}^2 & x_{91}x_{92} \end{bmatrix}, \ \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{11} \\ \beta_{22} \\ \beta_{12} \end{bmatrix} \text{ and } \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_9 \end{bmatrix}.$$

Then, the variance-covariance matrix of the estimated parameters is given by, $Var(b) = \sigma^2 (X'X)^{-1}$, where $b = [b_0 \ b_1 \ b_2 \ b_{11} \ b_{22} \ b_{12}]'$ is the vector of estimated parameters.

Because σ^2 is unknown, it will be estimated from the experimental data. The estimate of σ^2 is $\frac{SSE}{n-p}$, which is the mean squared error, where p is the number of parameters and SSE is the error sum of squares from the least squares regression. In this case, the estimate of σ^2 is model dependent. However, looking at the diagonal elements of the $(X'X)^{-1}$ matrix for the fitted second order model of the above example for $n = 9$ point CCDs indicates which CCD has the minimum variance for the estimated parameters. The $(X'X)^{-1}$ matrix for the estimated parameter coefficients with $\alpha = 1$ and $\alpha = \sqrt{2}$, respectively are given below:

$$\text{For the CCD with } \alpha = 1, \ (X'X)^{-1} = \begin{bmatrix} \frac{5}{9} & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 \\ 0 & \frac{1}{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6} & 0 & 0 & 0 \\ -\frac{1}{3} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ -\frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \text{ and}$$

$$\text{for the CCD with } \alpha = \sqrt{2}, \ (X'X)^{-1} = \begin{bmatrix} 1 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{8} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{8} & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & \frac{3437}{10000} & \frac{2187}{10000} & 0 \\ -\frac{1}{2} & 0 & 0 & \frac{2187}{10000} & \frac{3437}{10000} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}.$$

The diagonal elements of $(X'X)^{-1}$ for the CCD with $\alpha = \sqrt{2}$ are smaller for all the parameter estimates except for b_0. Moreover, two design criteria were calculated: (i) the determinant optimal (D-optimal) criterion defined by $|(X'X)^{-1}|$ and (ii) integrated average prediction variance (IV) which is the average of the scaled prediction variance. That is,

$$IV = (1/A) \int_{\chi} V(x)dx$$

where $V(x) = N x^{(2)} \left(X'X\right)^{-1} x^{(2)\prime}$, $x^{(2)} = [1, x_1, x_2, x_1^2, x_2^2, x_1x_2]$, χ is the experimental region of interest, and A is the volume of the experimental region χ (Borkowski (2003) and Myers et al. (2009)). For the $\alpha = 1$ CCD, $\chi = [-1, 1]^2 = \{(x_1, x_2) : -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1\}$ with $A = 4$. For the $\alpha = \sqrt{2}$ CCD, $\chi = \{(x_1, x_2) : x_1^2 + x_2^2 \leq 2\}$ with $A = 2\pi$. The computed D-criterion and IV criterion values are given in Table 1.2.

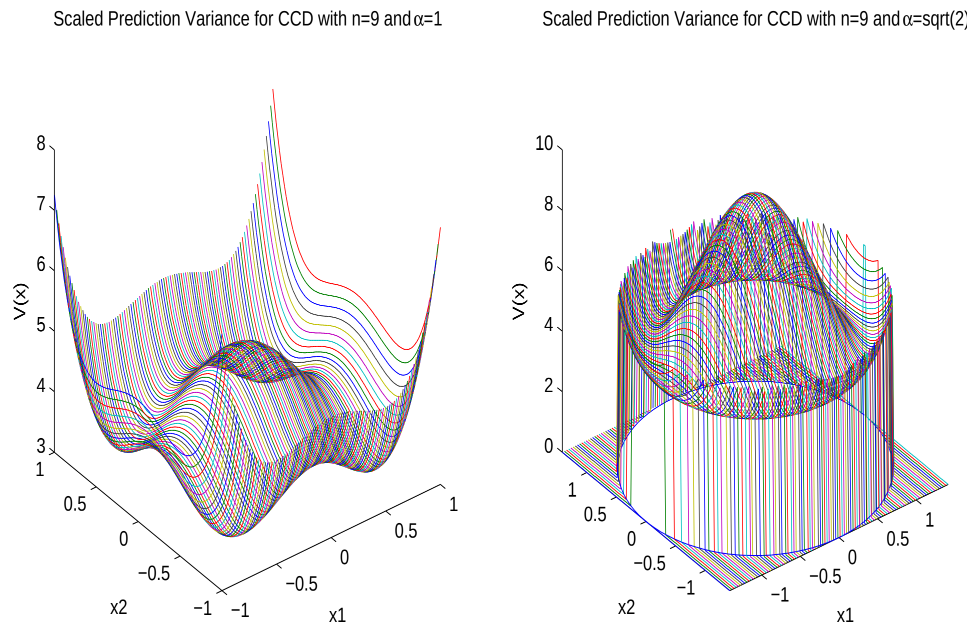
For α	D-optimal	IV
1	1.929e-004	**4.05**
$\sqrt{2}$	**3.0518e-005**	5.25

Table 1.2: D-optimal and IV prediction variance criteria for CCDs with $n = 9$ design points

The scaled prediction variance (SPV) function $V(x)$ for the CCDs with $\alpha = 1$ and $\alpha = \sqrt{2}$ for $n = 9$ are displayed graphically in Figure 1.5. The spherical region as shown in Figure 1.5 clearly shows the scaled prediction variance (SPV) values in $B^2(\sqrt{2})$ outside the square C^2 are much smaller than the SPV values for the CCDs inside C^2. That is, working on the region of operability increases the prediction performance outside the region of interest. However, the SPV is larger with $n = 9$ points for a CCD with $\alpha = \sqrt{2}$ than for a CCD with $\alpha = 1$ for all combinations of x_1 and x_2 levels in C^2 which is contained in $B^2(\sqrt{2})$. Specifically at $(0, 0)$, the scaled prediction variances are 5 and 9 for the $\alpha = 1$ and $\alpha = \sqrt{2}$ CCD respectively. The D-optimal criterion for the spherical region is better than it is for the cuboidal region (Table 1.2) and this is inline with what was observed for the variance-covariance matrices.

Another advantage for the spherical experimental region is the ability of certain designs to fit higher order models than possible in the cuboidal region. That is, sup-

Figure 1.5: 3D scaled prediction variance plot from CCDs with $\alpha = 1$ and $\alpha = \sqrt{2}$ for n=9



pose the design experimental region is restricted to be the square and the researcher wants to fit second order regression model with one cubic term added. That is, to fit

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \beta_{111} x_1^3 + \epsilon.$$

If the CCD with $\alpha = 1$ is run, the researcher will find that this model cannot be fit because the model matrix X is less than full column rank. If a spherical experimental region is possible, the CCD with $\sqrt{2}$ can be used to fit the second order regression model with a cubic term added. However, the set of models fit by any CCD is limited due to the limited number of factor levels. As a solution for this problem, a UD with 9 levels can be used. The equally spaced coded variables for the n=9 point uniform design from the best glp (good lattice point) method (9; 1,4) generator is given below:

x_1	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	1.00
x_2	-0.25	0.75	0.50	0.50	-0.75	0.25	-1.00	0.00	1.00

Unlike the CCD, the UD allows the maximum number of experimental levels for a given design size and a second-order model plus a cubic term model as well as many other models can be fit.

Objectives Of The Study

Although UDs have been used for over 3 decades, the UD tables given in most books are seriously insufficient because they do not provide practitioners with experimental designs within a practical range for the number of design points. For instance, the UD table in the book by Fang and Wang (1994) contains a UD table with $n = 987$ and $n = 75,025$. Thus, each design variable would have $n = 987$ or $n = 75,025$ levels. The authors seem to be focused on the mathematical properties of UDs and less so on the practical economical application of UDs to save an experimenter from having to run a large number of experimental runs.

The research in this dissertation addresses these issues. The specific objectives of this study are to:

1. Provide detailed tables of uniform designs for a feasible number of experimental runs in the unit cube C^k for $k = 2, 3, 4$ design variables. Specifically, provide designs with $n \leq 50$ points.

2. Extend UD generation to other experimental domains which have not yet been adequately studied in the statistical literature. Specifically:

    (a) Extend UD generation to the sphere for $k = 2, 3$, and 4 variables.

(b) Extend UD generation to the simplex for mixture experiments having 3,4 and 5 mixture components.

(c) Extend UD generation to subspaces of the simplex for mixture experiments.

3. Provide detailed tables of uniform designs for a feasible number of experimental runs in the sphere and the simplex.

4. Develop new discrepancy measures for different regions.

5. Provide experimenters with algorithms for generating designs with design sizes that are beyond the scope provided in this research.

METHODS OF CONSTRUCTING GOOD DESIGN POINTS

Number Theoretic Methods (NTMs) have a number of statistical applications, such as numerical evaluation of probabilities and moments of continuous multivariate distributions in different experimental regions such as cube, ball, sphere, and simplex (Wang and Fang, 1990). Fang (1980) and Wang and Fang (1981) used NTMs in the development of uniform designs (UDs) with the main goal of obtaining a set of points that are known as a number theoretic net (NT-net). Specifically, suppose there are s factors with the unit cube experimental domain C^s. An NT-net is defined as a 'representative set of points' on C^s. For a set of n points to be representative, the aim is to select a set of n design points $\mathcal{P}_n = (X_1, ..., X_n) \subset C^s$ that are 'scattered evenly' in C^s. That is, the aim is to choose the n design points with smallest discrepancy Fang et al. (1999) where discrepancy is a numerical measure of scatter.

An n-point UD with s factors each having q levels is denoted by $U_n(q^s)$. However, when the number of factor levels differ such that s_1 factors have q_1 levels and s_2 factors have q_2 levels, then the UD with n runs is denoted by $U_n(q_1^{s_1} \times q_2^{s_2})$. The notation can be generalized to three or more factor levels. A number of authors have proposed different methods of constructing UDs. Fang and Wang (1994) and Hua and Wang (1981) reviewed different construction methods. As part of this dissertation research, for a given number of factors s and number of design points n on the unit cube C^s, four different methods of constructing UDs will be used and are presented in the following section.

Good Lattice Points Method

The good lattice point (glp) method is a very efficient quasi Monte-Carlo method that was proposed by Korobov(1959) and later adopted by many authors such as Hua and Wang (1981) and Fang and Wang (1994). The quasi Monte-Carlo method can be used as an evaluation method for multiple integrals. This method consists of averaging function values calculated over a well-distributed low discrepancy set of points in the region of integration. Thus, it differs from the ordinary Monte-Carlo method that evaluates multiple integrals based on averaging function values over a randomly-selected set of points in the region of integration (Tony, 1972).

Assume the number of design points n is given and the unit cube $C^s = [0,1]^s$ is the experimental region. Let $glp_{(n,s)}$ denote a n-point glp design having s factors and let $gcd(a,b)$ be the greatest common divisor of integers a and b.

The glp method for C^s can be summarized in the following steps:

1. The first step in the glp method is to find the candidate set of positive integers (called generators). That is, find the ordered set of positive integers $H_n = \{h_1, h_2, ..., h_k\}$ such that $1 \leq h_i < h_j < n$, for $i < j$, and $gcd(n, h_i) = 1$ for $i = 1, ..., k$ where $k = \phi(n)$ is determined by Euler's function. The values of Euler's function equals the maximum possible number of h_i's for given n, and is defined as

$$\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)...\left(1 - \frac{1}{p_r}\right),$$

   where $n = p_1^{t_1} p_2^{t_2} ... p_r^{t_r}$ is the prime number decomposition of n (Fang et al., 2006). Table 2.1 gives the number of distinct $h_i^{,}s$ for $9 \leq n \leq 50$ design points.

2. From vector H_n, generate an $n \times k$ matrix $\mathbf{U}_k = (u_{ij})$ where $u_{ij} = ih_j \mod (n)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. If $u_{ij} = ih_j \mod (n) = 0$, then set $u_{ij} =$

n	$\phi(n)$	n	$\phi(n)$
9	6	30	8
10	4	31	30
11	10	32	16
12	4	33	20
13	12	34	16
14	6	35	24
15	8	36	12
16	8	37	36
17	16	38	18
18	6	39	24
19	18	40	16
20	8	41	40
21	12	42	12
22	10	43	42
23	22	44	20
24	8	45	24
25	20	46	22
26	12	47	46
27	18	48	16
28	12	49	42
29	28	50	20

Table 2.1: Number of possible h_i's from Euler's function $\phi(n)$

n. Note that each column of $\mathbf{U}_k$ is a permutation of $(1, 2, \ldots, n)'$ because $gcd(n, h_j) = 1$ for each j.

3. Form the $n \times k$ dimensional design matrix $\mathbf{X}_k$ from $\mathbf{U}_k$ where the i^{th} row of $\mathbf{X}_k = (x_{i1}, x_{i2}, \ldots, x_{ik})$ and $x_{ij} = \frac{(2u_{ij}-1)}{2n}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. Each of the k-columns of $\mathbf{X}_k$ is a permutation of $(1/2n, 3/2n, \ldots, (2n-1)/2n)'$.

4. Each subset of s columns of $\mathbf{X}_k$ forms a k-factor n-point UD. The glp method considers all designs formed by all subsets of s columns of $\mathbf{X}_k$. Fang and Wang (1994) showed that, due to equivalence of subsets of generators, the first column of $\mathbf{U}_k = (1, 2, \ldots, n)'$ (and, hence the first column of $\mathbf{X}_k$) should always be

selected to reduce the number of UDs to generate. Thus, it suffices to consider the first column of $\mathbf{X}_k$ and all subsets of $s-1$ columns from the remaining $k-1$ columns of $\mathbf{X}_k$.

The final step would be to determine which of the UDs in this set has the best space-filling properties. Different measures of the uniformity of scatter of the points will be considered for determining which UD is best. For example, consider $n = 10$ and $s = 2$ in $C^2 = [0,1]^2$. The prime number decomposition of $n = 10 = 2^1 \times 5^1$. Thus, the Euler function with $p_1 = 2$ and $p_2 = 5$ yields $k = \phi(10) = 4$ with candidate generating vector $H_n = (1,3,7,9)$. The results of the modular arithmetic (mod 10) would yield:

$$\mathbf{U}_4 = \begin{bmatrix} 1 & 3 & 7 & 9 \\ 2 & 6 & 4 & 8 \\ 3 & 9 & 1 & 7 \\ 4 & 2 & 8 & 6 \\ 5 & 5 & 5 & 5 \\ 6 & 8 & 2 & 4 \\ 7 & 1 & 9 & 3 \\ 8 & 4 & 6 & 2 \\ 9 & 7 & 3 & 1 \\ 10 & 10 & 10 & 10 \end{bmatrix} \qquad \mathbf{X}_4 = \begin{bmatrix} 1/20 & 5/20 & 13/20 & 17/20 \\ 3/20 & 11/20 & 7/20 & 15/20 \\ 5/20 & 17/20 & 1/20 & 13/20 \\ 7/20 & 3/20 & 15/20 & 11/20 \\ 9/20 & 9/20 & 9/20 & 9/20 \\ 11/20 & 15/20 & 3/20 & 7/20 \\ 13/20 & 1/20 & 17/20 & 5/20 \\ 15/20 & 7/20 & 11/20 & 3/20 \\ 17/20 & 13/20 & 5/20 & 1/20 \\ 19/20 & 19/20 & 19/20 & 19/20 \end{bmatrix}$$

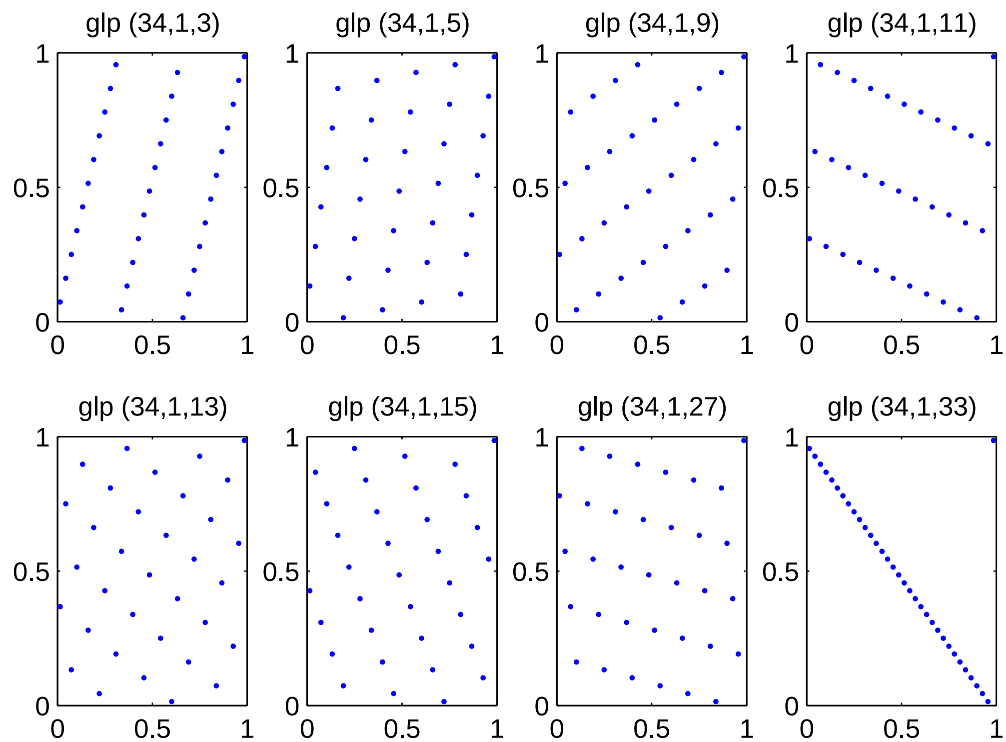
A two-factor UD would then be any 10×2 matrix formed from column 1 and one of the other three columns of $\mathbf{X}_4$. Thus, only $\binom{\phi(n)-1}{s-1} = \binom{3}{1} = 3$ candidate designs need to be considered because $h_1 = 1$ should always be included as stated in step 4. The final step would be to determine which of the 10-row, 2-column UDs in these set of candidates optimizes a chosen measure of the uniformity of scatter of its points. For additional details on the construction of UDs using the glp method, see (Fang and Wang, 1994; Zhang et al., 1998; Fang and Lin, 2003; Fang et al., 2006).

When the number of design points is a prime p and the dimensionality of the experimental region s are both large, Fang and Wang (1994) and Fang et al. (2006) described the power generator method using primitive roots to significantly reduce the extremely large number of candidate generating vectors. That is, they showed how to use $(1, h_2, ..., h_s) = (1, a, a^2, ..., a^{s-1})(mod(p))$, where $1 < a < p$, $a^i \neq a^j(mod(p))$, $1 \leq i < j < p$. The number of primitive roots modulo n was shown to be $\phi(\phi(n))$, where $\phi(n)$ is Euler's function. However, by limiting the search to designs generated by this method, generating vectors that produce the best UDs might be excluded. Figure 2.1 contains plots of eight sets of $n = 34$ points that were generated using the good lattice point (glp) method having generating vectors $(1, h_i)$ with $h_i = 3, 5, 9, 11, 13, 15, 27, 33$, respectively. We have only 9 distinct generating vectors although the Euler function in Table 2.1 indicates 16 distinct $h_i^{'}s$ as generators. The reason for that is that some of the generators are simply a rotation of the other. In this dissertation, in addition to the proposed methods of selecting the best UDs, we are also proposing methods of reducing the number of distinct generating vectors significantly which is what happened in this example. Details for the proposed methods will be presented in Chapter 4. As can be seen from the plots, the NT-nets with generators (34;1,13) and (34;1,5) appear to have more uniformly scattered points than the other NT-nets.

Good Points Method

Hua and Wang (1981) and Fang and Wang (1994) presented different methods of generating points that can be applied to the generation of a set of space-filling design points. The set of n design points in C^s obtained by a good point method have

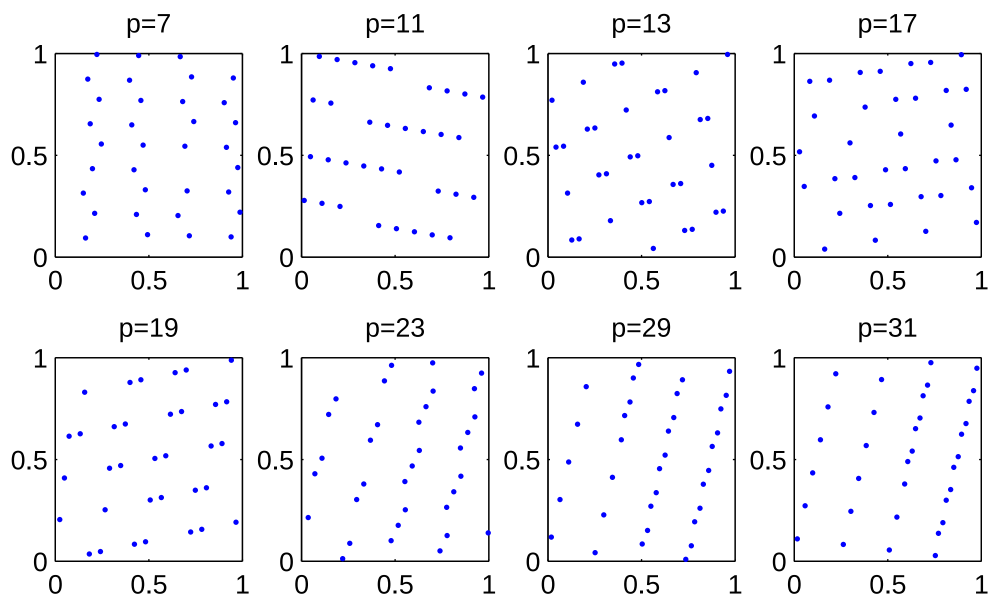
Figure 2.1: $n = 34$ design points from glp method in C^2



the form $\{\lfloor k\gamma_1 \rfloor, ..., \lfloor k\gamma_s \rfloor\}$ for $k = 1, \ldots, n$ where $\gamma = (\gamma_1, ..., \gamma_s)$ is the generating point and $\lfloor \quad \rfloor$ represents the function that returns the fractional part of $k\gamma_i$ for $i = 1, \ldots, s$. The good points generated by these methods are called *good point* or *gp* sets. The square root sequence method, the power of the $(s+1)^{th}$ root of a prime method and cyclotomic field methods generate good point sets. As an example of these good point methods, the gp set generated by the cyclotomic field (CF) method will be presented. The CF method is defined as follows: Let p be a prime number such that $p \geq 2s+3$, and let $\gamma = (\gamma_1, ..., \gamma_s) = (2\cos(\frac{2\pi}{p}), ..., 2\cos(\frac{2\pi s}{p}))$. The good point set using the CF method is generated by $\gamma(k) = (\lfloor k\gamma_1 \rfloor, ..., \lfloor k\gamma_s \rfloor)$ for $k = 1, \ldots, n$. Therefore, construction of a space-filling gp set using the CF method requires finding the vector γ that generates points having good space-filling properties. For example for $n = 12$ and $s = 2$, a candidate prime p has to be $\geq 2s + 3$, i.e., $7 \leq p \leq 12$. Thus, $p = 7$ and $p = 11$ are the only candidate primes. Using $p = 11$, the gp set of 12 design points from the CF method is given below:

$$\begin{bmatrix} 0.6825 & 0.8308 \\ 0.3650 & 0.6616 \\ 0.0475 & 0.4924 \\ 0.7300 & 0.3232 \\ 0.4125 & 0.1540 \\ 0.0950 & 0.9848 \\ 0.7775 & 0.8156 \\ 0.4600 & 0.6464 \\ 0.1425 & 0.4772 \\ 0.8250 & 0.3080 \\ 0.5075 & 0.1388 \\ 0.1900 & 0.9696 \end{bmatrix}$$

Figure 2.2: $n = 34$ design points from CF(gp) method in C^2



For the $s = 2$ factors, the 34 good points generated using the CF method are given in Figure 2.2 for the eight prime generators. It appears that the 34 design points generated for the best designs using the glp method (Figure 2.1) are more uniformly scattered than those obtained from the CF method. Fang and Wang (1994) recommended using the gp set found by the CF method when the number of experimental factors (s) is large.

The Halton (H) Set

The Halton or H method is another method of constructing a low discrepancy set of design points and is used as a building block for the Hammersley method that also provides a set of design points with low discrepancy. The detailed mathematical formulation of the H method is available in Fang and Wang (1994) and Hua and Wang (1981). The H method uses the following facts about positive integers and rational numbers in the interval [0,1]:

- Any positive integer k can be uniquely expressed using a prime base $p \geq 2$ as

  $$k = k_0 + k_1 p + k_2 p^2 + \ldots + k_M p^M \qquad 0 \leq k_i \leq p-1$$

  where $p^M \leq k < p^{M+1}$.

- Any rational number c in the interval [0,1] can be represented uniquely by

  $$c = c_0 p^{-1} + c_1 p^{-2} + \ldots + c_M p^{-M-1} \qquad 0 \leq c_i \leq p-1.$$

- There exists 1-1 correspondence between the positive integers and the rational numbers in (0,1). Specifically, for any integer $k \geq 1$, let $y_p(k) = k_0 p^{-1} + k_1 p^{-2} +$

$... + k_M p^{-M-1}$. Then, $y_p(k) \in (0, 1)$ and is known as the radical inverse of k base p.

If $p = 2$, $y_2(0)$, $y_2(1)$, $y_2(2)$,..., is called the Van der Corput sequence. If $\{p_1, p_2, ..., p_s\}$ is a set of s distinct prime numbers, then the H-set proposed by Halton is the set of points x_k such that

$$x_k \;=\; (y_{p_1}(k), \ldots, y_{p_s}(k)) \quad \text{for } k = 1, \ldots, n.$$

That is, the dimension of the Halton or H-set is simply the generalization of the Van der Corput sequence using the first s prime numbers as the base. The H-set of points on C^s when $s > 2$ was proposed by Halton (1960). Halton has also proved mathematically as in Hua and Wang (1981) that the set formed by the first n points has discrepancy

$$\mathcal{D}(n) \leq \prod_{i=1}^{s} n^{-1} \left(\frac{p_i \ln(p_i n)}{\ln(p_i)} \right) = O(n^{-1}(\ln(n))^s)$$

where $(\mathcal{D}(n))$ is the discrepancy. For example when $n = 15$, $s = 2$, and using $p_1 = 2$ and $p_2 = 3$, the 15 point H-set design is given by

$$\begin{bmatrix} 0.5000 & 0.3333 \\ 0.2500 & 0.6667 \\ 0.7500 & 0.1111 \\ 0.1250 & 0.4444 \\ 0.6250 & 0.7778 \\ 0.3750 & 0.2222 \\ 0.8750 & 0.5556 \\ 0.0625 & 0.8889 \\ 0.5625 & 0.0370 \\ 0.3125 & 0.3704 \\ 0.8125 & 0.7037 \\ 0.1875 & 0.1481 \\ 0.6875 & 0.4815 \\ 0.4375 & 0.8148 \\ 0.9375 & 0.2593 \end{bmatrix}$$

The computational burden is high when the number of primes (p_i's) and s are large. Thus, the H-set is suitably used when s is small (Fang and Wang, 1994).

The Hammersley Method

The Hammersley method is a simple modification of the H-set method. Let $s \geq 2$ and $p_1, ..., p_{s-1}$ be $s - 1$ distinct prime numbers. The Hammersley set is given by

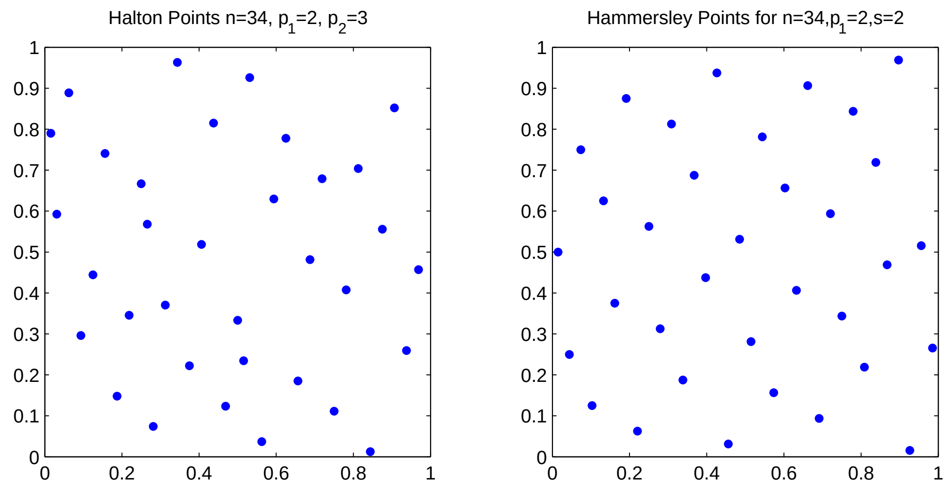
$$x_k \;=\; \left(\frac{2k-1}{2n}, y_{p_1}(k), \ldots, y_{p_{s-1}}(k)\right) \quad for\ k = 1, 2, ..., n$$

and the discrepancy is

$$\mathcal{D}(n) \leq \prod_{i=1}^{s-1} n^{-1} \left(\frac{p_i \ln(p_i n)}{\ln(p_i)}\right)$$

which is lower than the discrepancy for the H-set (Hammersley, 1964).

Figure 2.3: $n = 34$ design points from Halton and Hammersleymethods in C^2



The Hammersley set for the example with $n = 15$ points using $p_1 = 2$ is given by

$$\begin{bmatrix} 0.0333 & 0.5000 \\ 0.1000 & 0.2500 \\ 0.1667 & 0.7500 \\ 0.2333 & 0.1250 \\ 0.3000 & 0.6250 \\ 0.3667 & 0.3750 \\ 0.4333 & 0.8750 \\ 0.5000 & 0.0625 \\ 0.5667 & 0.5625 \\ 0.6333 & 0.3125 \\ 0.7000 & 0.8125 \\ 0.7667 & 0.1875 \\ 0.8333 & 0.6875 \\ 0.9000 & 0.4375 \\ 0.9667 & 0.9375 \end{bmatrix}$$

The good point sets with $n = 34$ generated from Halton and Hammersley are presented in Figure 2.3. It can be seen from Figure 2.3 that the design points generated using the Hammersley method are more uniformly scattered than those from the Halton method, and this is inline with the mathematical measure of discrepancy that states the Hammersley set has lower discrepancy than the Halton set.

Other Methods

There are several other methods of generating uniformly scattered points. These include the square root sequence method and the Haber sequence method (Fang and Wang, 1994). The square root sequence method works by taking the fractional parts of the scaled square roots of the first s primes. That is,

$$\gamma(k) \;=\; (\lfloor k\sqrt{p_1}\rfloor, \ldots, \lfloor k\sqrt{p_s}\rfloor) \quad for\ k = 1, \ldots, n$$

where $p_1, p_2, \ldots\ , p_s$ are the first s primes. Measures of uniformity, such as discrepancy, are much better for the square root sequence method than for the Haber sequence method. The Haber method, (results not presented here) produces designs that have worst measures of uniformity in comparison to the other methods used in this study. A Haber sequence is defined as

$$y_k \;=\; \left(\frac{k(k+1)}{2}\sqrt{p_1}, ..., \frac{k(k+1)}{2}\sqrt{p_s}\right)(\mathtt{mod1}) \text{ for } \mathtt{k} = \mathtt{1}, ..., \mathtt{n}$$

where $p_1, p_2..., p_s$ are the first s primes.

MEASURES OF UNIFORMITY

Traditionally the aim of a uniform design is to provide design points that are uniformly scattered throughout the experimental region C^s. In the previous sections, methods of constructing uniformly scattered good space filling designs have been presented and discussed. However, those methods on their own do not provide a quantifiable measure of uniformity. Thus, the degree of uniformity has to be measured and quantified so that one can directly compare different designs. Different measures of uniformity have been adopted by different authors. For example, Hickernell (1998a,b) modified the star L_2 discrepancy measure and obtained the star L_2 centered discrepancy and the star L_2 wrap-around discrepancy measures of uniformity. Hickernell also provided analytic formulas for computing both of these discrepancy measures. Fang and Wang (1994) also used the star discrepancy in measuring uniformity while Borkowski and Piepel (2009) used distance-based measures of uniformity for highly constrained mixture designs.

In this study two commonly-used classes of measures of uniformity will be studied. These are the discrepancy measures used to study number theoretic methods (NTMs) (Fang and Wang, 1994) and the metric distance approach proposed by Borkowski and Piepel (2009) for highly constrained mixture designs. Finally, modifications of the discrepancy measure used by Fang and Wang (1994) will be proposed and studied. Both the discrepancy measures and metric distance approach methods proposed in this dissertation will also be modified for applications to experimental regions other than the unit cube C^s.

A New Discrepancy Measure

Let $F(\mathbf{x})$ be the multivariate uniform distribution of s independent uniform $U(0,1)$ random variable on the unit cube experimental region $C^s = [0,1]^s$, and let $F_n(\mathbf{x})$ be the empirical distribution of the design points $\mathcal{P} = \{\mathbf{x}_1, \dots \mathbf{x}_n\}$:

$$F_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} I\{x_{i1} \leq x_1,\ x_{i2} \leq x_2\ , \dots,\ x_{is} \leq x_s\},$$

where $\mathbf{x} = (x_1, \dots, x_s)$, $\mathbf{x}_i = (x_{i1}, \dots, x_{is})$ for $i = 1, \dots, n$, and $I(A) = 1$ if A is true, and is zero otherwise. Thus, $F_n(\mathbf{x})$ equals the proportion of points in $\mathcal{P}$ that lie in the hyperrectangle formed by the origin and point $\mathbf{x}$. The star L_p-discrepancy is defined as

$$D_p(\mathcal{P}) = \left\{\int_{C^s} |F_n(\mathbf{x}) - F(\mathbf{x})|^p d\mathbf{x}\right\}^{\frac{1}{p}}. \tag{3.1}$$

The star discrepancy is defined as the following limit:

$$D_\infty(\mathcal{P}) \ = \ \lim_{p\to\infty}\left\{\int_{C^s} |F_n(\mathbf{x}) - F(\mathbf{x})|^p d_{\mathbf{x}}\right\}^{\frac{1}{p}} \ = \ \sup_{x \epsilon C^s} |F_n(\mathbf{x}) - F(\mathbf{x})|.$$

The star L_2 centered discrepancy and the star L_2 wrap-around discrepancy measures described by Hickernell (1998a,b) are special cases of the star L_p-discrepancy given in (3.1). Hickernell provided analytic formulas for these two discrepancies that are easy to compute. However, because they are modifications of the L_p-discrepancy, both can yield biased underestimates of $D_p(\mathcal{P}_n)$.

The application of discrepancy measures to number theoretic methods for generating designs will now be discussed. For any vector $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_s) \in C^s$, let $N(\gamma, \mathcal{P}) = nF_n(\gamma) =$ the number of design points that satisfy $I\{x_{i1} \leq \gamma_1, x_{i2} \leq \gamma_2, \dots, x_{is} \leq \gamma_s\} = 1$. Let the hyperrectangle formed by γ and the origin be denoted

$[0, \gamma]$. The star discrepancy measure for a set of design points $\mathcal{P}$ is given by

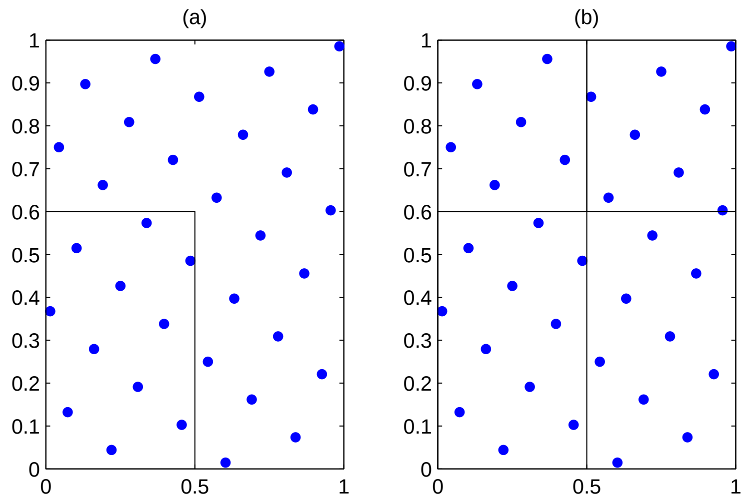
$$D(n, \mathcal{P}) = \sup_{\gamma \in C^s} \left| \frac{N(\gamma, \mathcal{P})}{n} - \mathcal{V}([0, \gamma]) \right|, \tag{3.2}$$

where $\mathcal{V}([0,\gamma]) = \prod_{i=1}^{s} \gamma_i$ denotes the volume of $[0,\gamma]$. In other words, for $D(n,\mathcal{P})$, we are comparing the proportion of points $\frac{N(\gamma,\mathcal{P})}{n}$ that lie in $[0,\gamma]$ to the ratio of the volumes $\frac{\mathcal{V}([0,\gamma])}{\mathcal{V}(C^s)}$. Note that $\frac{\mathcal{V}([0,\gamma])}{\mathcal{V}(C^s)} = \mathcal{V}([0,\gamma])$ because $\mathcal{V}(C^s) = 1$. Then, $D(n,\mathcal{P})$ is simply the supremum of the absolute difference over all possible hyperrectangles $[0,\gamma]$ for all $\gamma \in C^s$. If $D(n,\mathcal{P})$ is small then the proportion of points within each hyperrectangle $[0,\gamma]$ is nearly proportional to the volume of the hyperrectangle indicating good uniformity of scatter. Or, in other terms, it is a good space-filling design. Conversely, a large discrepancy measure indicates a poor space-filling design. $D(n,\mathcal{P})$ in (3.2) is equivalent to the Kolmogorov Smirnov statistic in goodness-of-fit testing. For an overview of discrepancy measures, see (Fang and Wang, 1994; Fang et al., 2000, 2006).

Figure 3.1(a) geometrically illustrates how the star discrepancy $D(n,\mathcal{P})$ measures uniformity. Consider the two (inner and outer) rectangles displayed in Figure 3.1(a). There are 34 design points in the outer rectangle corresponding to $C^2 = [0,1]^2$ and 11 design points in $[0,\gamma]$ in the inner rectangle of width 0.5 and height 0.6 (i.e., $\gamma = (0.5, 0.6)$). Thus, $N(\gamma,\mathcal{P})/n = 11/34 = .3235$ = the proportion of points that fall in the rectangle $[0,\gamma]$. Then $|0.3235 - \mathcal{V}([0,\gamma])| = |0.3235 - 0.3000| = 0.0235$ is the discrepancy for this particular γ. Therefore, to find the star discrepancy (STRD) $D(n,\mathcal{P})$, we must consider these absolute deviations over all $\gamma \in C^2$, and then take the maximum.

In this dissertation, however, instead of working with only the hyperrectangles $[0,\gamma]$ for γ in C^s , we propose to partition the experimental region C^s into 2^s hyper-

Figure 3.1: Geometric illustration of regions used to calculate (a) the star discrepancy criteria and (b) modified star discrepancy for a 34 point design



rectangles formed by γ and each of the 2^s vertices of C^s, and then take the maximum of the 2^s discrepancies associated with this set of 2^s hyperrectangles. Notationally, let R_j be the jth hyperrectangle formed by γ and vertex $\mathbf{v}_j$ ($j = 1, \ldots, 2^s$) of C^s. Let $N(\gamma, \mathcal{P}, R_j)$ be the number of design points in R_j and $\mathcal{V}(R_j)$ be the volume of R_j. Then define:

$$D(n, \mathcal{P}, \gamma) = \max_{j=1,\ldots,2^s} \left| \frac{N(\gamma, \mathcal{P}, R_j)}{n} - \mathcal{V}(R_j) \right|.$$

The proposed modified star discrepancy (MSTRD) is now defined as

$$MSTRD(n, \mathcal{P}) = \sup_{\gamma \in C^s} D(n, \mathcal{P}, \gamma). \tag{3.3}$$

For example, for $s = 2$, C^2 is divided into 4 rectangles for a given γ, and is illustrated in Figure 3.1(b). The discrepancy $D(34, \mathcal{P}, \gamma)$ values for the bottom left ($j = 1$), bottom right ($j = 2$), top left ($j = 3$), and top right ($j = 4$) rectangles are, respectively, $|\frac{11}{34} - 0.5 \times 0.6|$=0.0235, $|\frac{9}{34} - 0.5 \times 0.6|$=0.0353, $|\frac{6}{34} - 0.4 \times 0.5|$=0.0235, and $|\frac{8}{34} - 0.4 \times 0.5|$=0.0353. We then take the maximum = 0.0353, which is the numerical measure of modified star discrepancy for $\gamma = (0.5, 0.6)$. Therefore, to find the modified star discrepancy $MSTRD(n, \mathcal{P})$, we must consider taking the maximum over all $\gamma \in C^2$. Note that the STRD $\leq$ MSTRD. In this example STRD=.0235 which is smaller than MSTRD=0.0353 (or, $\approx$ 67% smaller). Thus, using STRD to compare the uniformity of scatter for competing designs can potentially lead to the selection of an inferior design. The MSTRD also provides an improvement to the STRD as a discrepancy measure because it assesses uniformity of scatter for more subspaces of C^s.

The concept of dividing the experimental region into 2^s hyper-rectangles appears, at first glance, to be similar to the approach used by Hickernell in the star L_2 centered discrepancy. His approach, however, is restricted to using only the center point $\gamma = (0.5, \ldots, 0.5)$ (i.e., a single γ) to form the hyper-rectangles. In this study, we remove this single center point restriction and permit the use of random evaluation points to form the hyper-rectangles. The MSTRD approach, like that of Hickernell's method, is invariant under $90^o, 180^o$ and 270^o rotations.

Determining exact or near-exact MSTRD (or STRD) values can be computationally expensive because we are optimizing over all $\gamma \in C^s$. Thus, as the dimension of the experimental region C^s increases, the computational cost for calculating a measure of discrepancy increases. In this dissertation, estimation of MSTRD and STRD values is restricted to low $(s = 2, 3)$ dimensional unit cubes.

Because the computational demands for calculating a measure of discrepancy increases with dimension, the distance-based measures used in Borkowski and Piepel (2009) are recommended alternatives to the discrepancy-based measures.

The Distance Metric Approach

Johnson et al. (1990) proposed minimax and maximin designs which are space-filling designs that are dependent on distance measures (Koehler and Owen, 1996). In business, the maximin criterion can be described as a plan for allocating shops that would ensure no 2 shops are very close to each other within a fixed spatial region. Conversely, the minimax criterion would ensure that remotely located customers have one or more shops to choose from within a reasonable distance (Johnson et al., 1990) . In short, maximin can be considered as maximizing the minimum gain (profit) where as minimax as minimizing the maximum loss. Recently, Borkowski and Piepel (2009) have proposed and used an alternate version of the maximin distance criterion and two other criteria for selecting the best space-filling designs for highly-constrained mixture designs. They can be classified as distance-based criteria. Thus, in addition to the STRD and MSTRD discrepancy measures, the distance-based criteria for design selection proposed by Borkowski and Piepel (2009) will be used in this research. The details of the distance-based approach are now presented.

Let $\mathbf{x}_1, ..., \mathbf{x}_N$ be any set of N points in C^s which form the rows of an $N \times s$ design matrix $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & . & . & . & , x_{1s} \\ x_{21} & x_{22} & . & . & . & , x_{2s} \\ . & . & . & . & . & , . \\ . & . & . & . & . & , . \\ . & . & . & . & . & , . \\ x_{N1} & x_{N2} & . & . & . & , x_{Ns} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ . \\ . \\ . \\ \mathbf{x}_N \end{bmatrix}$$

Then the distance between a point $\mathbf{x}$ in C^s and matrix $\mathbf{X}$ is defined as

$$d(\mathbf{x}, \mathbf{X}) \;=\; \min_j ||\mathbf{x} - \mathbf{x}_j||_2 = \min_j \sqrt[2]{(\mathbf{x} - \mathbf{x}_j)(\mathbf{x} - \mathbf{x}_j)'}$$

for $j = 1, 2, ..., N$ (Borkowski and Piepel, 2009). That is, the N distances between a point $\mathbf{x}$ in C^S and each of the N points in $\mathbf{X}$ are calculated, and then the minimum of these N distances is retained. That is, $d(\mathbf{x}, X)$ is the distance between a point $\mathbf{x}$ in C^s and the nearest design point in $\mathbf{X}$. The following selection criteria are measures of uniformity for space-filing designs that were used by Borkowski and Piepel (2009):

1. The Root Mean Squared Distance ($RMSD(\mathbf{X})$):

   $$RMSD(\mathbf{X}) = \sqrt[2]{E[(d(\mathbf{x}, \mathbf{X})^2]}$$

   is the square root of the expected value of the squared distance.

2. The Average Distance ($AD(\mathbf{X})$):

   $$AD(\mathbf{X}) = E[d(\mathbf{x}, \mathbf{X})]$$

   is the expected value of the distance.

3. The Maximum distance ($MD(\mathbf{X})$):

   $$MD(\mathbf{X}) = \max_{\mathbf{x}}(d(\mathbf{x}, \mathbf{X}))$$

   is the maximum distance between any design point $\mathbf{x}$ in C^s and the point of $\mathbf{X}$ closest to $\mathbf{x}$.

The $AD(\mathbf{X})$ and $RMSD(\mathbf{X})$ criteria are based on expectations, and thus require integration over the experimental region to determine their exact values. Because it is difficult to calculate the exact $RSMD(\mathbf{X})$ and $AD(\mathbf{X})$ values, sample mean approximations were used in Borkowski and Piepel (2009). That is, for a large random sample of N points $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ in C^s, the approximations are

$$\begin{aligned} rmsd(\mathbf{X}) &= \sqrt[2]{\frac{\sum_{l=1}^{N}(d(\mathbf{x}_l, \mathbf{X}))^2}{N}} \\ ad(\mathbf{X}) &= \frac{\sum_{l=1}^{N} d(\mathbf{x}_l, \mathbf{X})}{N} \\ md(\mathbf{X}) &= \max_{l=1,\ldots,N}(d(\mathbf{x}_l, \mathbf{X})). \end{aligned}$$

A good space-filling design will have small $rmsd(\mathbf{X})$, $ad(\mathbf{X})$ and $md(\mathbf{X})$ values because small values indicate the points in C^s are close to the design $\mathbf{X}$ (i.e., there are no large volume subregions in C^s that do not contain any design points from $\mathbf{X}$). Conversely, large values of these criteria indicate that a design is poor with respect to its space-filling properties.

Therefore, the design selection criteria considered in this dissertation are the $rmsd(\mathbf{X})$, $ad(\mathbf{X})$ and $md(\mathbf{X})$ distance-based criteria plus the STRD and MSTRD discrepancy criteria discussed. The hope is to find a design that is superior for multiple criteria and performs well across all criteria.

Figures 2.1, 2.2, and 2.3 show graphically how the design points generated by the glp, CF(gp) and Halton and Hammersley methods respectively may be scattered. The plots, however, do not identify which design has the most uniformly scattered set of points across methods and criteria. Table 3.1 summarizes the numerical measures for each of the five measures of uniformity of scatter for each generating method.

Measures of Uniformity						
Methods	Generator	*rmsd*	*ad*	*md*	STRD	MSTRD
glp	(34,1;3)	0.1098	0.0949	0.3235	0.1113	0.1113
	(34,1;5)	0.0773	0.0707	0.2145	0.0867	0.0867
	(34,1;9)	0.0811	0.0737	0.2286	0.0660	0.0667
	(34,1;11)	0.1025	0.0900	0.2890	0.0841	0.0936
	(34,1;13)	**0.0743**	**0.0691**	**0.1753**	0.0531	**0.0531**
	(34,1;15)	0.0789	0.0724	0.1853	**0.0525**	0.0578
	(34,1;27)	0.0757	0.0700	0.1781	0.0635	0.0694
	(34,1;33)	0.2479	0.2065	0.6667	0.2467	0.2467
CF(gp)	p=7	0.0857	0.0769	0.2199	0.1934	0.2175
	$p = 11$	0.0936	0.0826	0.2755	0.1113	0.1430
	$p = 13$	0.0887	0.0810	0.2293	**0.0900**	**0.0900**
	$p = 17$	**0.0798**	**0.0725**	**0.1971**	0.1164	0.1361
	$p = 19$	0.0867	0.0791	0.2262	0.0945	0.1090
	$p = 23$	0.0899	0.0808	0.2688	0.1242	0.1477
	$p = 29$	0.0904	0.0814	0.2443	0.1414	0.1682
	$p = 31$	0.0892	0.0802	0.2459	0.1727	0.2053
Halton	$p_1 = 2$ and $p_2 = 3$	0.0835	0.0753	**0.2186**	0.1071	0.1971
Hammersley	$p_1 = 2$	**0.0780**	**0.0715**	0.2230	**0.0666**	**0.0666**

Table 3.1: Measures of Uniformity from the 4 methods of constructingUDs in C^2

Table 3.1 is used to compare the methods and select the best design by comparing their measure of uniformity. In computing the measures of uniformity 4000 evaluation points with 1000 points in four equal sized square strata are used. Thus, 4000 evaluation points and 16000 different rectangles in C^2 were used for the MSTRD criteria. The bold-faced values in Table 3.1 show the minimum measures and hence are best according to the criterion used.

For the good lattice point (glp) method and for all the measures of uniformity except STRD, the best design has generator (34;1,13). For the Cyclotomic Field (cf) method, the best UD based on the metric distances (*rmsd*, *ad* and *md*) is the design generated by prime $p = 17$, whereas the best UD based on the STRD and MSTRD measures is generated by prime $p = 13$. This suggests that different designs may be best when considering different evaluation criteria. As discussed earlier in Chapter

2, the Hammersley method produces designs with points that are more uniformly scattered than those generated by the Halton method based on discrepancy. This can be seen in Table 3.1 with all measures of uniformities (except md) smaller for the Halton method.

A comparison of the measures of uniformity among the 4 methods of constructing uniformly scattered design points suggests the glp method is the best, and this is consistent with the fact that the glp method is good for small s. This is because the glp method has multiple candidate design generators for each design size with at least one generator leading to good coverage of the experimental region. For large s, the glp method produces excellent designs, but it is computationally expensive and often infeasible to implement. Note also that in Table 3.1 the Hammersley method is the second best generating method as it has the lowest measures of uniformity after the glp method. The CF (gp) method is ranked third and the Halton method is ranked fourth as can be seen from the values in Table 3.1. This finding also agrees with what was presented in Fang and Wang (1994).

Figure 3.2 shows the magnitude of the selection criteria both for the metric distance approach and for the discrepancy (STRD and MSTRD) criteria for all the methods of constructing uniform designs and for design sizes of $n = 10, 11, ..., 50$ points. Clearly, the glp method performs best as can be seen from Figure 3.2. Hence, the glp method generates points that more uniformly cover the unit square (C^2) experimental region. As expected, a general decreasing trend is observed for all four methods. That is, as the number of design points increases, the magnitude of the evaluation criteria tends to decrease. However, a decrease or near-constant change in the magnitude of the distance-based evaluation criterion is observed for all methods of generating designs except for the glp method. The reasons for the observed patterns are:

1. For the CF(gp) method the prime number generators of the best designs are often the same for the neighboring number of design points. Hence, the expectation is a negligible change of the evaluation criteria for the metric distance approach when the generators remain the same.

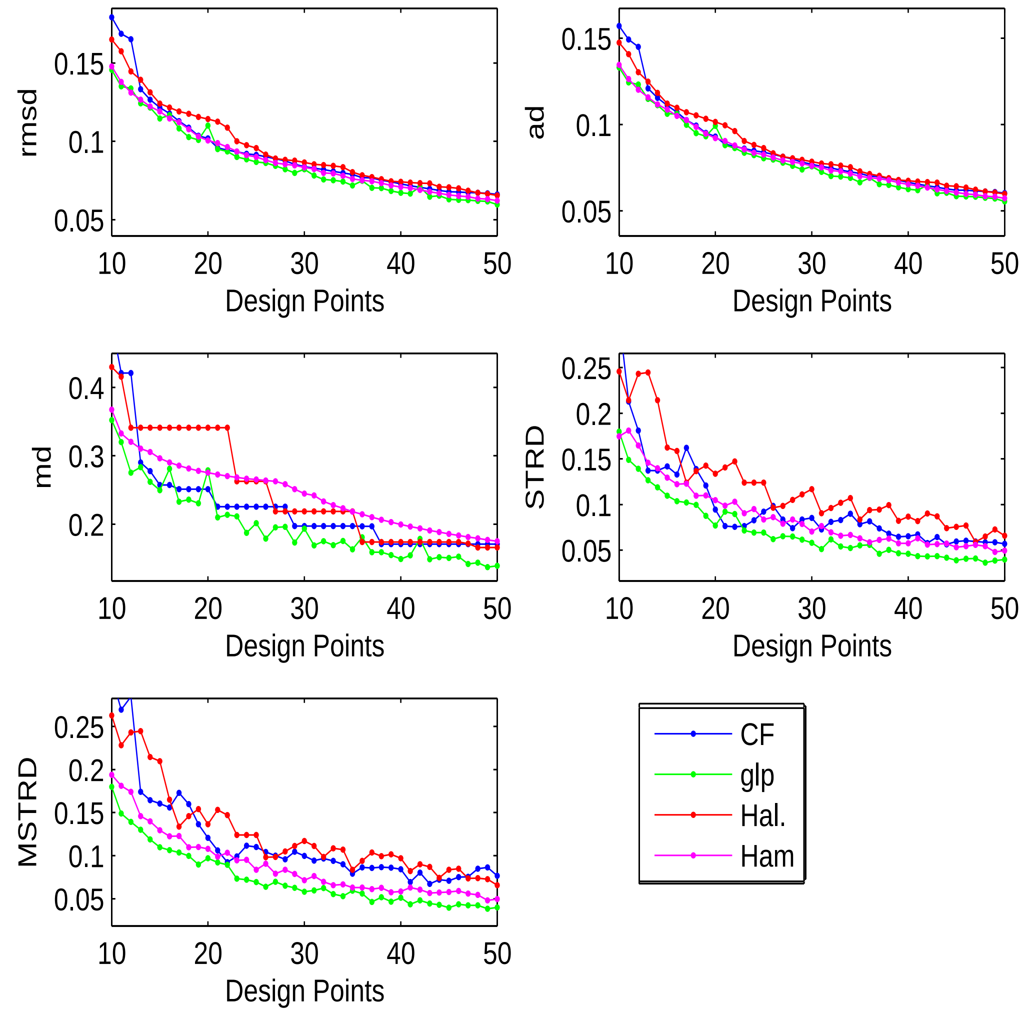
2. For the Halton method the prime number generators for $s = 2$ are always 2 and 3. Hence, all the designs in C^2 have the same prime generators which means the $n + 1$ point design is found by simply augmenting the n point design with one additional point.

3. The Hammersley method is very similar to the Halton method except the first coordinate of the design generator changes as we go from n to $n + 1$.

On the other hand, for the glp method, increasing the design size by a single point can either decrease or increase the magnitude of the evaluation criteria. When one moves from n design points to $n + 1$ design points using the glp method, it is not the addition of a new design point to the existing n-point design. That is, it is not an augmentation of the existing n-point design because the design generators for n and $n + 1$ will differ. Hence, for the glp method, it should be understood that the evaluation criteria may not always decrease when increasing the design size from n to $n+1$ design points. The evaluation criteria are highly dependent on the generator and how well the points are scattered in the entire region. This is evident in the plots for the glp method in Figure 3.2 that show cases of up and down patterns for the neighboring designs in all the evaluation criteria.

For example, for $n = 19$ and $n = 20$ the prime generators are the same in CF (gp) method, and hence produces the same or very close measures of uniformity. However, the generators are different for $n = 19$ and $n = 20$ for the glp method. In fact, there are 9 candidate generators for $n = 19$ but only 5 for $n = 20$ for the

Figure 3.2: Graphical comparison of the 4 methods of constructing UDs in C^2 for design sizes $n = 10, 11, \ldots 50$ for each Evaluation Criterion



glp method after considering only the nonequivalent generators which significantly reduces the number of distinct generators. Thus, it is not unusual to observe such distinct patterns between the CF (gp) method and the glp method. That is, we expect to observe an up and down pattern for the glp method and to observe a decrease or

Comparison of results			
Methods	Wang and Fang STRD	Dissertation STRD	MSTRD
glp	0.0642	**0.0525**	**0.0531**
CF(gp)	0.1853	**0.0900**	**0.0900**
Halton	0.1106	**0.1071**	0.1971
Hammersley	0.0864	**0.0666**	**0.0666**

Table 3.2: Comparison of STRD results from Fang and Wang (1994) and this dissertation

very little change in the distance-based measures of uniformity for the CF method in the neighboring design sizes. It is important to remember that a downward trend is not necessarily seen for neighboring design points, but rather a general downward trend across the set of all design sizes n. For the STRD and MSTRD criteria, all of the methods have plots with up and down sequences of points while the overall trend is decreasing.

For future research one can consider UD with $n^* \leq n$ that minimizes criterion and then add $n - n^*$ points. This is similar to the idea of augmented design. For example, as can be seen from Figure 3.2, the STRD measure from the glp method for $n = 31$ is smaller than for $n = 32$. Hence, to minimize the STRD measure for $n = 32$ it is better to work with $n^* = 31$ and then add 1 point to get a better design than the one obtained by the UD for $n = 32$.

The results for $n = 34$ design points using the NTM-discrepancy measure used in this study, together with the results presented in Fang and Wang (1994), are presented in Table 3.2. Table 3.2 shows that for the 4 methods, the star discrepancy (STRD) computed using 4000 stratified evaluation points (in 4 equal sized square strata) provides smaller measures of uniformity than the STRD computed in Fang and Wang (1994). The difference might partly be due to the fact that different sets of evaluation points will yield different measures of uniformity as well. What may also affect the discrepancy values is the implementation of stratified sampling yielding better discrepancy estimates in this study. That is, the proposed Modified Star Discrepancy (MSTRD) together with the stratification of evaluation points that fill the experimental region yield better results. Note that for the Halton method, the MSTRD is larger than the star discrepancy (STRD) because the STRD was underestimating the measure of uniformity due to its limited assessment of the experimental design region.

As previously stated, due to the method of constructing uniform designs, the choice of the generators and how the design points are scattered on the entire experimental region, one cannot always expect a smooth, monotonically decreasing trend for all the methods and for all the evaluation criteria. One observation from Figure 3.2 is that the STRD and MSTRD discrepancy criteria are the most volatile measures of uniformity because there are many cases of up and down patterns. Figure 3.2 also shows the Halton method as the worst method of constructing uniform designs relative to the other methods in this study. The Hammersley method is the second next best after the glp method in all the evaluation criteria except the maximum distance (md) criterion.

Overall in C^2, the glp method is found to be the best method for producing low discrepancy sets of design points. Hence, it is worth investigating the relative performance of the other 3 design generating methods in C^2 relative to the glp method. That is, the relative efficiency of the methods should be computed and plotted for each evaluation criteria and the different number of design points using the glp method as the baseline. This allows the experimenter to make an easier and proper comparison of the methods.

In this research the relative efficiency (RE) of the method for each evaluation criteria relative to the glp method is defined as

$$RE = \frac{eval_i(Method_j)}{eval_i(glp)},$$

for $i = rmsd,\ ad,\ md,\ STRD,\ MSTRD$ and $j = CF,\ Halton,$ and $Hammersley$ where $eval_i$ represents evaluation criterion i.

For example, for $i = rmsd,$ and $j = CF,$

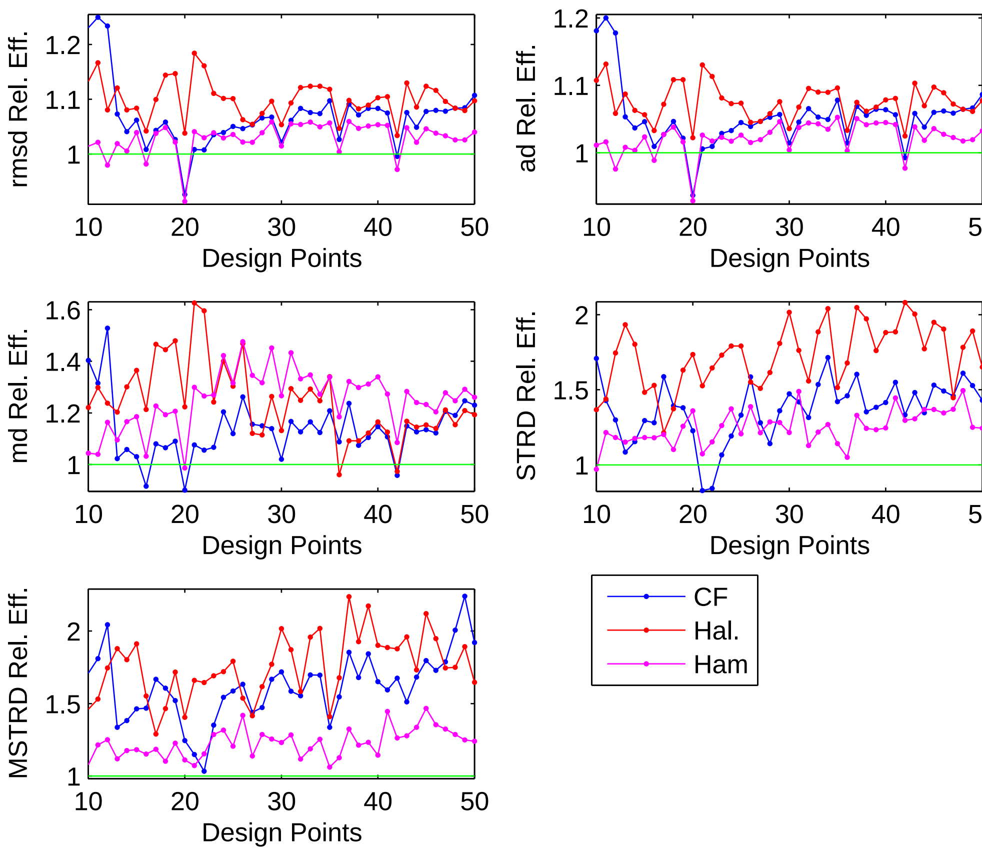
$$RE = \frac{rmsd(CF)}{rmsd(glp)}$$

Figure 3.3 shows the relative efficiency (RE) of the methods for each evaluation criterion for different numbers of design points. With the exception of the RE for the maximum distance (md) criterion where the CF(gp) method was second with the other evaluation criteria, the Hammersley method is found to be the second best after the glp method. This is consistent with what was previously observed from Figure 3.2.

A summary measure of the relative efficiency is the average of the evaluation criteria for each method relative to the average for glp method. For example, for the *rmsd* criterion, the ratio of the average *rmsd* from the CF(gp) method from each of the different design points to the average *rmsd* from the glp method, the result is 1.0684 (see Table 3.3). This measure is called the *average relative efficiency (ARE)*. For the above example, the average is calculated over the 41 design sizes ($n = 10, ..., 50$):

$$ARE(CF, glp) = \frac{\sum_{n=10}^{50} rmsd(CF(n))}{\sum_{n=10}^{50} rmsd(glp(n))}\ .$$

Figure 3.3: Relative efficiency of the methods of constructing UDs in C^2 relative to glp for each evaluation criterion for design sizes 10, 11, ... 50



It is not necessary to divide by 41 because each method had the same number (41) of designs, and thus, the ratio of the sums will equal the ratio of the means. The ARE comparison for each method in each evaluation criterion is presented in Table 3.3. The boldfaced values in Table 3.3 indicate the ARE values closest to one, and, hence, most similar to the glp method overall. For example, the ARE for the *rmsd* criterion is 1.0295 for the Hammersley method and is closest to one, and hence is the closest to the glp method. The value of $(1 - ARE)100\%$ is a percentage measure indicating how efficient a method is relative to the glp method in C^2. For example, the *rmsd* ARE of 1.0295 for Hammersley means the Hammersley method is, on average, 2.95% less efficient than the glp method for the range of design sizes in this study, while the ARE of 1.0997 for the Halton method means that the Halton method is, on average, 9.97% less efficient than the glp method. Similarly, the CF(gp) is, on average, 6.84% less efficient than the glp method by the rmsd criterion. Hence, based on the majority of the evaluation criteria presented in Table 3.3, it can be concluded that for different design sizes in C^2, the glp method is again the best method followed by the Hammersley, CF and Halton methods, respectively.

Average relative efficiency comparison					
Methods	*rmsd*	*ad*	*md*	STRD	MSTRD
CF(gp)	1.0684	1.0529	**1.1361**	1.3735	1.6148
Halton	1.0997	1.0750	1.2452	1.7243	1.7539
Hammersley	**1.0295**	**1.0216**	1.2524	**1.2503**	**1.2281**

Table 3.3: Comparison of average relative efficiency for the methods of constructing UDs in C^2 relative to glp

NEW METHODS FOR COMPUTATIONAL REDUCTION

In Chapter 2, we have seen how to generate UDs using the 4 different UD generating methods. In this chapter, an extension to computational reduction in higher dimensions will be discussed and presented for the glp method. The case where $s = 3$ will be shown both numerically and geometrically. However, because of the computational demands of calculating STRD and MSTRD, using the glp method to find the best design will also inflate the computational cost due to a large number $\binom{\phi(n)-1}{s-1}$ of possible generating vectors in s-dimensions where $\phi(n)$ is the Euler function that determines the maximum number of generators. To circumvent the computational requirements when using the glp method, two approaches are proposed and discussed in the next two sections.

Computational Reduction Method I: Equivalence

Because the MSTRD and the distance-based measures are invariant under rotation (of $90°, 180°, 270°, \ldots$), and because some generating vectors produce points that are simply a rotation of points from other generating vectors, there is no need to consider sets of generating vectors that yield the same measure of uniformity under rotation.

Two NT-nets generated by the glp method are defined to be *equivalent* if the points in one NT-net can be generated by a column permutation of the other. Therefore, because of the invariance under rotations, two equivalent NT-nets generated by the glp method will have equal values for the uniformity measures considered in this research. Computational reduction methods for the glp method for $s = 2$ and $s = 3$ will now be presented.

For $s = 2$, consider generating vectors $(1, h_i)$ and $(h_j, 1)$ where $h_j > h_i$. If $h_i \times h_j \mod (n) = 1$ then the set of NT-net points generated from $(1, h_i)$ is equivalent to the set of NT-Net points generated from $(1, h_j)$. For $s = 2$, the permutation matrix,

$$P_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

is used to permute the columns of $(h_j, 1)$ into $(1, h_j)$. That is, the set of points in the NT-net generated by $(h_j, 1) \times P_2$ is the same as those generated by $(1, h_i)$. Hence, only measures of uniformity for NT-nets generated by $(1, h_i)$ (and not $(1, h_j)$) need to be computed.

For $s = 3$, consider generating vectors $(1, h_i, h_j)$, $(h_k, 1, h_l)$ and $(h_m, h_q, 1)$ where $h_j > h_i$ and $h_k > h_i$. If the following five conditions

(i) $h_i \times h_k \mod (n) = 1$

(ii) $h_j \times h_m \mod (n) = 1$

(iii) $h_i \times h_l \times h_m \mod (n) = 1$

(iv) $h_l \times h_q \mod (n) = 1$

(v) $h_j \times h_k \times h_q \mod (n) = 1$

are satisfied, then, with respect to a uniformity measure, the set of NT-net points generated from $(1, h_i, h_j)$ is equivalent to the set of NT-net points generated by permuting the columns of $(h_k, 1, h_l)$ and $(h_m, h_q, 1)$. To see this, note the following:

1. If $h_k > h_l$, then the NT-net formed from $(1, h_i, h_j)$ is equivalent to the NT-net formed from $(h_k, 1, h_l)$. The permutation matrix,

$$P_{3A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

is used to permute the columns of $(h_k, 1, h_l)$ into $(1, h_l, h_k)$. That is, the set of points in the NT-net generated by $(h_k, 1, h_l) \times P_{3A}$ is the same as those generated by $(1, h_i, h_j)$. Hence, only measures of uniformity for NT-nets generated by $(1, h_i, h_j)$ (and not $(h_k, 1, h_l)$) need to be computed. If, however, $h_l > h_k$, then interchange the 1^{st} and 3^{rd} rows of the permutation matrix P_{3A}.

2. Similarly, if $h_m > h_q$, then the NT-net formed from $(1, h_i, h_j)$ is equivalent to the NT-net formed from $(h_m, h_q, 1)$. The permutation matrix,

$$P_{3B} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

is used to permute the columns of $(h_m, h_q, 1)$ into $(1, h_m, h_q)$. That is, the set of points in the NT-net generated by $(h_m, h_q, 1) \times P_{3B}$ is the same as those generated by $(1, h_i, h_j)$. Hence, only measures of uniformity for NT-nets generated by $(1, h_i, h_j)$ (and not $(h_m, h_q, 1)$) need to be computed. If, however, $h_q > h_m$, then interchange the 1^{st} and 2^{nd} rows of the permutation matrix P_{3B}.

Hence, using equivalence can significantly reduce the computational demands of finding the best UD using the glp method. For $s = 2$ and $s = 3$, this can lead to a maximum reduction of 1/2 and 2/3, respectively, to the number of designs evaluated.

Consider reduction using equivalence for $n = 34$ point UDs. Table 4.1 summarizes the sets of equivalent generating vectors for $s = 2$ and $s = 3$. For example, for $s = 2$, (1,3) and (23,1) are equivalent, and thus, the NT-nets generated from $(1, 3)$ and (1,23) have equal measures of uniformity. Thus, these measure only need to be computed for the NT-net generated by (1,3). Similarly, for $s = 3$, (1,3,5), (23,1,13), and (7,21,1) are equivalent, and thus, after using the appropriate permutations of (23,1,13) and (7,21,1), the NT-nets generated from (1,3,5), (1,13,23) and (1,7,21) all have equal measures of uniformity. Thus, these measures only need to be computed for the NT-net generated by (1,3,5).

After a check of equivalence, the generators that are in the first row of Tables 4.1(a) and the first column of 4.1(b) form the reduced candidate set of generators for $s = 2$ and $s = 3$, respectively, for $n = 34$. For $s = 2$, the computations for $n = 34$ are reduced from 15 to 8 generators ($\approx 47\%$ reduction), and, for $s = 3$, the computations are reduced by $\frac{2}{3}$ ($\approx 67\%$) from 105 to 35 generators.

(a) Equivalent NT-nets for $s = 2$ and $n = 34$

$(1,h_i)$	(1,3)	(1,5)	(1,9)	(1,11)	(1,13)	(1,15)	(1,27)	(1,33)
$(1,h_j)$	(1,23)	(1,7)	(1,19)	(1,31)	(1,21)	(1,25)	(1,29)	—

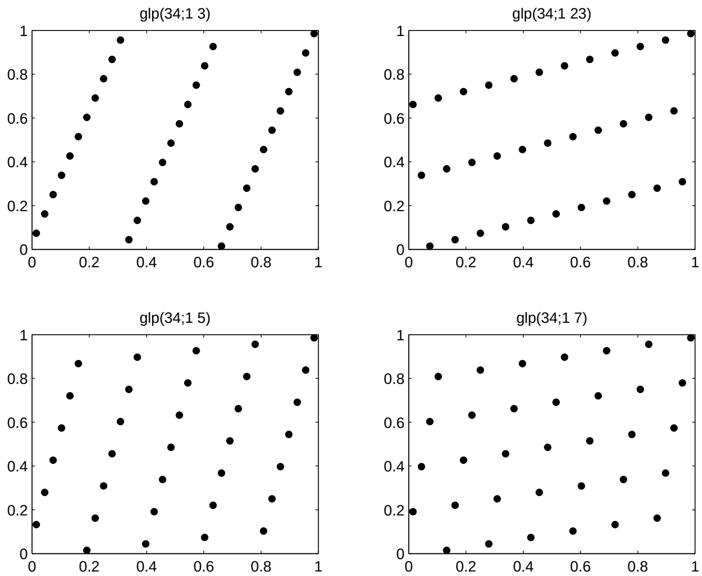
(b) Equivalent NT-nets for $s = 3$ and $n = 34$

$(1,h_i,h_j)$	$(1,h_k,h_l)$	$(1,h_m,h_q)$	$(1,h_i,h_j)$	$(1,h_k,h_l)$	$(1,h_m,h_q)$
(1,3,5)	(1,13,23)	(1,7,21)	(1,5,27)	(1,7,19)	(1,9,29)
(1,3,7)	(1,23,25)	(1,5,15)	(1,5,29)	(1,7,33)	(1,27,33)
(1,3,9)	(1,3,23)	(1,19,23)	(1,5,31)	(1,7,13)	(1,11,21)
(1,3,11)	(1,15,23)	(1,25,31)	(1,5,33)	(1,7,27)	(1,29,33)
(1,3,13)	(1,23,27)	(1,21,29)	(1,9,13)	(1,9,19)	(1,19,21)
(1,3,15)	(1,5,23)	(1,7,25)	(1,9,15)	(1,13,19)	(1,21,25)
(1,3,19)	(1,23,29)	(1,9,27)	(1,9,21)	(1,19,25)	(1,13,15)
(1,3,21)	(1,7,23)	(1,5,13)	(1,9,25)	(1,19,33)	(1,15,33)
(1,3,25)	(1,23,31)	(1,11,15)	(1,9,31)	(1,11,19)	(1,11,31)
(1,3,27)	(1,9,23)	(1,19,29)	(1,9,33)	(1,15,19)	(1,25,33)
(1,3,29)	(1,21,23)	(1,13,27)	(1,11,13)	(1,29,31)	(1,21,27)
(1,3,31)	(1,23,33)	(1,11,33)	(1,11,25)	(1,27,31)	(1,15,29)
(1,3,33)	(1,11,23)	(1,31,33)	(1,11,27)	(1,21,31)	(1,13,29)
(1,5,7)	(1,7,15)	(1,5,25)	(1,11,29)	(1,15,31)	(1,27,25)
(1,5,9)	(1,7,29)	(1,19,27)	(1,13,21)	(1,21,33)	(1,13,33)
(1,5,11)	(1,7,9)	(1,19,31)	(1,13,25)	(1,15,21)	(1,15,25)
(1,5,19)	(1,7,31)	(1,9,11)	(1,15,27)	(1,25,29)	(1,27,29)
(1,5,21)	(1,7,11)	(1,13,31)			

Table 4.1: Equivalent generators of NT-nets for $s = 2$ and $s = 3$when $n = 34$

The equivalent NT-net point generators for the first two generators for $s = 2$ and $s = 3$ from Table 4.1, are displayed graphically in Figure 4.1 and Figure 4.2, respectively. After rearranging the order of the relative primes, Figure 4.1 shows the pair of equivalent NT-nets $(1,3)$ and $(1,23)$ and the equivalent NT-nets $(1,5)$ and $(1,7)$. Figure 4.2 also shows the equivalent NT-nets $(1,3,5)$, $(1,13,23)$ and $(1,7,21)$ and the equal NT-nets $(1,3,7)$, $(1,23,25)$ and $(1,5,15)$. The uniform scatter nature of the design points in both Figures show the computational task reduction method makes sense geometrically.

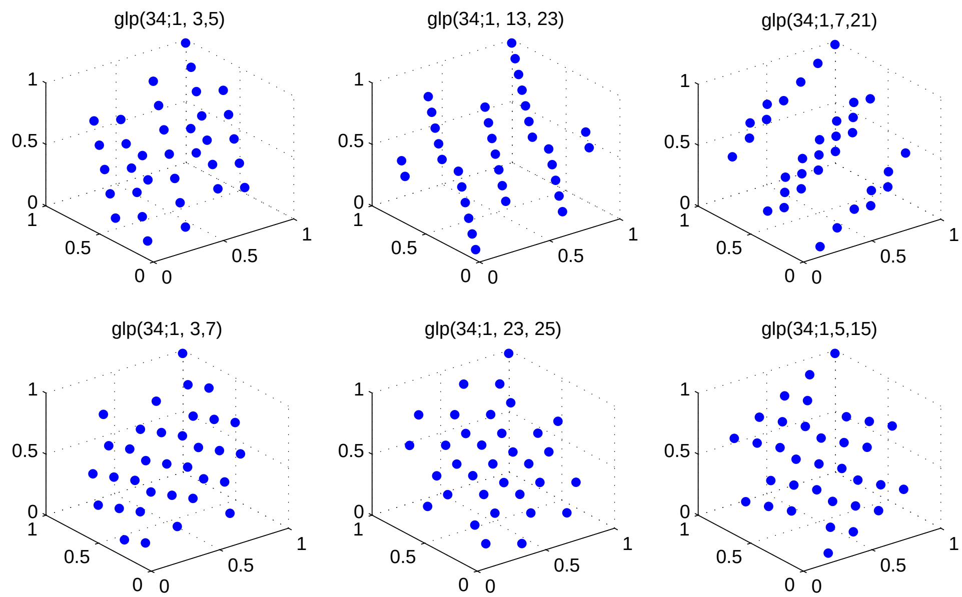
Figure 4.1: Graphical example of equivalent NT-net generators Using the reduction method in C^2 for $n = 34$ design points



That is, for $s = 2$ both the NT-nets have the same uniformity measure. Geometrically, as can be seen in Figure 4.1, both have the same pattern except that one set of points can be obtained via a rotation and reflection of the other set. The same applies for $s = 3$, as can be seen in Figure 4.2 although the invariance to rotation is not as visually clear as it is for $s = 2$.

For dimension $s = 4$, there are 23 conditions required to determine sets of equivalent generators, with the number of conditions rapidly increasing as s increases. Even with computational reduction using equivalence, the number of generating vectors to be considered will become very large as the dimension increases and especially when the number of design points is prime. For $s = 4$, a Matlab software program was written to search for the equivalent NT-nets. For higher dimensions, an alternative approach to using equivalence to reduce computations is considered and is now presented.

Figure 4.2: Graphical example of equivalent NT-net generators using the reduction method in C^3 for $n = 34$ design points



Computational Reduction Method II: Projection

The goal is to provide an alternative to using generator equivalence to reduce the computational burden of finding very good, if not the best, UDs generated by the glp method in a unit cube C^s, especially for higher dimensions. The proposed method is called the *projection method* and is defined by the following procedure:

1. Find the best glp generating vectors $(1, h_i)$ for C^2 or, when available, find the best $(1, h_i, h_j)$ for C^3. These vectors are easy to find because of the minimal computation required for low dimensions.

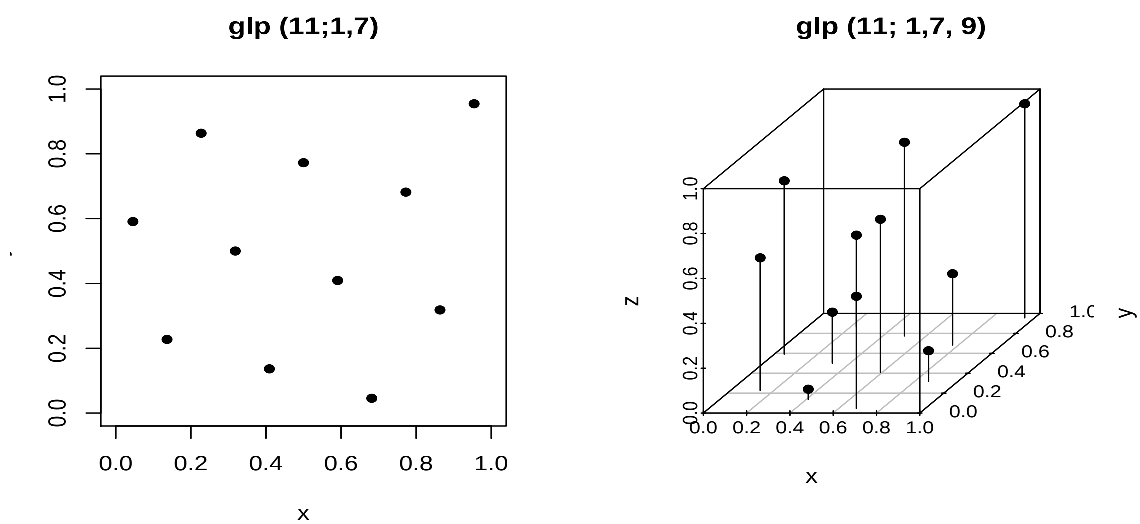
2. Project the best UDs generated in Step 1 to the next higher dimension. This is accomplished by creating a new set of generating vectors by adding an additional h value to the existing best generating vectors while retaining the sequence of

increasing values in the generating vectors. Compute the measure of uniformity for each of these vectors of length $s+1$. Retain the vectors that generated UDs having the smallest measure of uniformity.

3. Iterate the procedure in Step 2 until the desired dimension is reached.

4. Retain the UD having the smallest measure of uniformity.

Figure 4.3: Geometrical illustration of projection from C^2 intoC^3 dimensions



Note that the term projection used in this dissertation is different from the meaning that projection has in linear algebra.

For design size $n = 11$, Figure 4.3 shows the geometric projection of the best glp generator from C^2 into C^3. The best design for $s = 2$ had generator $(1, 7)$. Adding 9 to $(1, 7)$ to create generator $(1, 7, 9)$ was the optimal choice. This figure illustrates why this approach works well. That is, for any design having good space-filling properties in 3 dimensions, it must also have good space-filling properties in 2 dimensional projections.

Measures of Uniformity				
s	Generator	$rmsd$	Generator	ad
2	(1,23)	0.0425	(1,13)	0.0395
3	(1,17,23)	0.1139	(1,17,23)	0.1087
4	(1,7,37,83)	0.1993	(1,7,37,83)	0.1923
5	(1,7,37,51,83)	0.2840	(1,7,37,51,83)	0.2760
6	(1,7,37,51,83,91)	0.3698	(1,7,37,51,83,91)	0.3610
7	(1,7,37,51,69,83,91)	0.4496	(1,7,37,51,69,83,91)	0.4406
8	(1,7,11,37,51,69,83,91)	0.5263	(1,7,11,37,51,69,83,91)	0.5174
9	(1,7,11,37,51,61,69,83,91)	0.5985	(1,7,11,37,51,61,69,83,91)	0.5898
10	**(1,7,11,33,37,51,61,69,83,91)**	0.6675	**(1,7,11,33,37,51,61,69,83,91)**	0.6590

Measures of Uniformity				
s	Generator	md	Generator	MSTRD
2	(1,23)	0.0930	(1,37)	0.0224
3	(1,7,37)	0.2325	(1,23,37)	0.0352
4	(1,17,23,37)	0.4056		
5	(1,3,7,37,83)	0.5427		
6	(1,3,7,37,57,83)	0.7043		
7	(1,7,31,37,51,83,91)	0.8192		
8	(1,7,37,51,53,69,83,91)	0.9091		
9	(1,7,11,37,51,59,69,83,91)	1.0117		
10	**(1,7,11,37,41,51,61,69,83,91)**	1.1089		

Table 4.2: Best generators for $n = 100$ point glp designs using the projection method for $s = 3, ..., 10$

Given designs of size n, the projection from dimension s to $s+1$ is done by considering all the best generators found in dimension s as candidate generators to be projected to dimension $s+1$. For example, to generate $n = 100$ design points for $s = 10$, start with $s = 2$. As can be seen in Table 4.2 the best generators for $s = 2$ are (1,23) based on the $rmsd$ and md criteria, (1,13) based on the ad criterion, and (1,37) based on MSTRD. Hence, (1,23), (1,13) and (1,37) are used as candidate generators that will be projected to $s = 3$. In the rows for $s = 3$ in Table 4.2, the best generators are (1,17,23) based on the $rmsd$ and ad criteria, (1,7,37) based on the md criterion, and (1,23,37) based on MSTRD. Thus, (1,7,37), (1,17,23) and (1,23,37) are the candidate generators to be projected to $s = 4$, and those that appear in the $s = 4$ rows were found to be the best. The projection process continues from $s = 4$ until the desired dimension $s = 10$ is reached. That is, the best generators in dimension s serve as base generators for dimension $s+1$ starting for $s = 2, 3, \ldots, 9$. The bold-faced values in Table 4.2 are the best generators for $n = 100$ design points in dimension $s = 10$ using the projection method. Note that the values of the measures of uniformity in Table 4.2 increase as the dimension increases.

The equivalence approach and the projection method approach are both effective ways to reduce of the computational demands for finding very good, if not the best, designs generated using the glp method. The projection method, in particular, is simple and computationally thrifty, and is, therefore, recommended for generating good designs in higher dimensions.

UDs For $s = 3$

The same uniformity measures discussed previously are used to measure how uniformly the design points are scattered in the experimental region for $s = 3$. However, for $s = 3$, unlike the 4 rectangles created by γ, we will have $2^3 = 8$ hyperrectangles created by γ. The number of design points that fall in each of those 8 hyperrectangles will be counted and the 8 proportions relative to the cube C^3 will be measured. Then each of these proportions is compared to the corresponding proportion of the volume of the hyperrectangles to the volume of C^3. The supremum of the absolute differences of the 8 proportions of points and their corresponding ratios of volumes for sets of possible hyperrectangles created from each vertex and the infinitely many random choices for γ defined the MSTRD in Equation (3.3). That is, as in Chapter 3, instead of working with only the hyperrectangles that are formed by the origin and points γ in C^s, we propose to first divide the experimental region into 2^s hyperrectangles formed by γ and each of the 2^s vertices of C^s, and then take the maximum of the 2^s discrepancies obtained for each of the 2^s hyperrectangles. That is, find the proposed modified star discrepancy (MSTRD) given in Equation (3.3).

The five measures of uniformity (both the distance-based and discrepancy criteria) discussed in Chapter 3 are evaluated for the $s = 3$ $n = 34$ point UDs generated using the 4 methods of constructing UDs in C^3. The results are presented in Table 4.3.

For the glp method there are 105 candidate generators, but as seen in Table 4.1, only 35 generators need to be studied. Computational reduction using the equivalence method cut the required calculations by $\frac{2}{3}$. Thus, only results for the 35 non-equivalent generators are presented in Table 4.3 because, with respect to uniformity measures, the generators in the $(h_k, 1, h_l)$ and $(h_m, h_q, 1)$ columns generate equivalent NT-nets.

The best generators are those that correspond to the bold-faced values in Table 4.3. For the glp method, $(34; 1, 11, 27)$ is the best generators for the distance-based ($rmsd$ and ad) criteria and the best generators based on discrepancy criteria STRD and MSTRD are $(34; 1, 13, 25)$ and $(34; 1, 9, 21)$, respectively. This shows that different criteria can lead to different designs. For the glp method, the 34 design points generated by $(34; 1, 11, 27)$ and by $(34; 1, 9, 21)$ are displayed in Figure 4.4. For the CF(gp) method, $p = 11$ is the best generator for the distance-based criteria while $p = 23$ and $p = 17$ are the best generators for the STRD and MSTRD criteria, respectively. The 34 design points generated by $p = 11$ and $p = 17$ from the CF method are displayed in Figure 4.5. The Hammersley method with $p_1 = 2$ and $p_2 = 3$ is a better generator than the Halton method by all the measures of uniformity except the distance-based md criterion. The 34 design points from Halton and Hammersley methods are displayed in Figure 4.6.

Comparison of the 4 methods for $s = 3$ and $n = 34$ shows the glp method produces a design having the lowest measure of uniformity based on the distance-based measures and discrepancy criteria. This can be seen from the bold faced values in Table 4.3. The CF method has the lowest measure of uniformity after the glp method followed by the Hammersley and the Halton methods for the distance-based measures. However, in Chapter 3, Table 3.1 for $n = 34$ and $s = 2$, the Hammersley method was the second best after the glp method unlike what is observed in Table 4.3 based on *rmsd* and *ad*. However, according the discrepancy criteria (STRD and MSTRD) from Table 4.3, the rank order of the methods is consistent with results previously observed in Table 3.1 for $s = 2$. The space-filling nature of the designs can be clearly observed in Figures 4.4, 4.5 and 4.6.

Methods	Generator	*rmsd*	*ad*	*md*	STRD	MSTRD
glp	(34;1,3,5)	0.1958	0.1830	0.4634	0.1192	0.1192
	(34;1,3,7)	0.1818	0.1712	0.3922	0.1242	0.1271
	(34;1,3,9)	0.1729	0.1636	0.3742	0.1360	0.1360
	(34;1,3,11)	0.1730	0.1637	0.3808	0.1054	0.1283
	(34;1,3,13)	0.1706	0.1617	0.3595	0.1054	0.1158
	(34;1,3,15)	0.1834	0.1724	0.4044	0.1360	0.1360
	(34;1,3,19)	0.1837	0.1722	0.4188	0.1054	0.1307
	(34;1,3,21)	0.1719	0.1627	0.3817	0.1301	0.1390
	(34;1,3,25)	0.1714	0.1618	0.3610	0.1197	0.1307
	(34;1,3,27)	0.1795	0.1693	0.3781	0.1192	0.1334
	(34;1,3,29)	0.1977	0.1851	0.4818	0.1261	0.1261
	(34;1,3,31)	0.2956	0.2605	0.7160	0.2264	0.2759
	(34;1,3,33)	0.3006	0.2675	0.7211	0.2392	0.2537
	(34;1,5,7)	0.1797	0.1685	0.4485	0.1081	0.1081
	(34;1,5,9)	0.1801	0.1689	0.4452	0.0811	0.1063
	(34;1,5,11)	0.1845	0.1730	0.4078	0.0943	0.1429
	(34;1,5,19)	0.1819	0.1713	0.3980	0.0977	0.1268
	(34;1,5,21)	0.1707	0.1617	0.3873	0.0886	0.1231
	(34;1,5,27)	0.1809	0.1700	0.4357	0.1049	0.1161
	(34;1,5,29)	0.2880	0.2500	0.7122	0.2264	0.2629
	(34;1,5,31)	0.1960	0.1836	0.4558	0.0940	0.1241
	(34;1,5,33)	0.2867	0.2497	0.7183	0.2392	0.2590
	(34;1,9,13)	0.1776	0.1673	0.3831	0.0927	0.1032
	(34;1,9,15)	0.1769	0.1671	0.3701	0.0903	0.1054
	(34;1,9,21)	0.1785	0.1688	0.3910	0.0801	**0.0894**
	(34;1,9,25)	0.2910	0.2540	0.7098	0.2264	0.2477
	(34;1,9,31)	0.1731	0.1640	**0.3496**	0.0973	0.1143
	(34;1,9,33)	0.2860	0.2486	0.7145	0.2392	0.2459
	(34;1,11,13)	0.1942	0.1822	0.4232	0.0956	0.1013
	(34;1,11,25)	0.1788	0.1682	0.3895	0.1035	0.1146
	(34;1,11,27)	**0.1697**	**0.1608**	0.3716	0.0987	0.1185
	(34;1,11,29)	0.1806	0.1696	0.3936	0.1116	0.1217
	(34;1,13,21)	0.2883	0.2501	0.7112	0.2264	0.2464
	(34;1,13,25)	0.1779	0.1677	0.3824	**0.0732**	0.0980
	(34;1,15,27)	0.1774	0.1671	0.3961	0.0951	0.1101
CF(gp)	p=11	**0.1725**	**0.1628**	**0.3967**	0.1425	0.1990
	p=13	0.2071	0.1899	0.5125	0.1315	0.1548
	p=17	0.1816	0.1708	0.4283	0.1252	**0.1308**
	p=19	0.1961	0.1815	0.4600	0.1539	0.1810
	p=23	0.2201	0.2022	0.5222	**0.0975**	0.1829
	p=29	0.2100	0.1948	0.4696	0.1576	0.1743
	p=31	0.1841	0.1746	0.4096	0.1720	0.2007
Halton	$p_1 = 2$, $p_2 = 3$, and $p_3 = 5$	0.1799	0.1693	**0.4119**	0.1186	0.1358
Hammersley	$p_1 = 2$ and $p_2 = 3$	**0.1761**	**0.1666**	0.4175	**0.0806**	**0.1027**

Table 4.3: Measures of uniformity from the 4 methods of constructingUDs in C^3

Figure 4.4: Best 34-point glp UDs in C^3 for the *rmsd* and *ad* criteria (left) and MSTRD criterion (right)

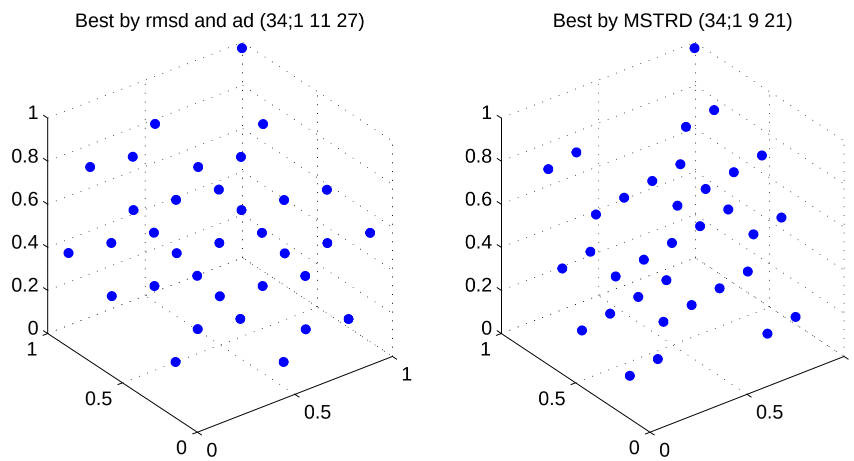
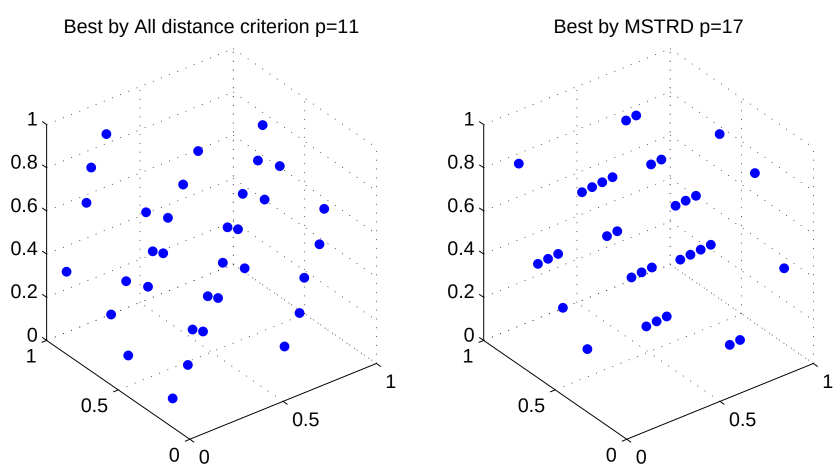
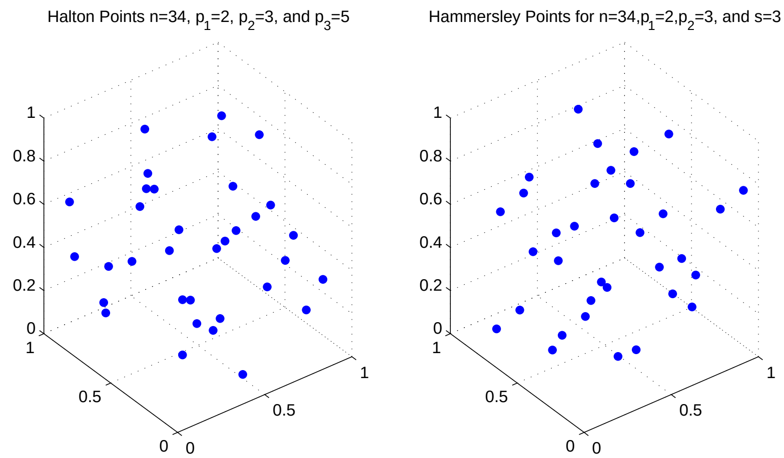


Figure 4.5: Best 34-point CF(gd) UDs in C^3 for the distance-based (left) and MSTRD Criteria (right)



In Figure 4.7, the values of the *rmsd* and MSTRD criteria for $s = 3$ and the *rmsd* criterion for $s = 4$ obtained from the equivalence and projection methods are plotted for the best generators for design sizes $n = 10, 11 \ldots, 50$. Figure 4.7 indicates that for the *rmsd* criterion, both methods are equally efficient for generating good UDs in 3

Figure 4.6: Best 34-point UDs in C^3 from Halton (left) and Hammersley (right) in C^3



and 4 dimensions. However, for the MSTRD criterion, the equivalence method tends to produce UDs having smaller measures of uniformity than UDs resulting from the projection method. In general, however, the differences in the measures of uniformity between the two methods is very small.

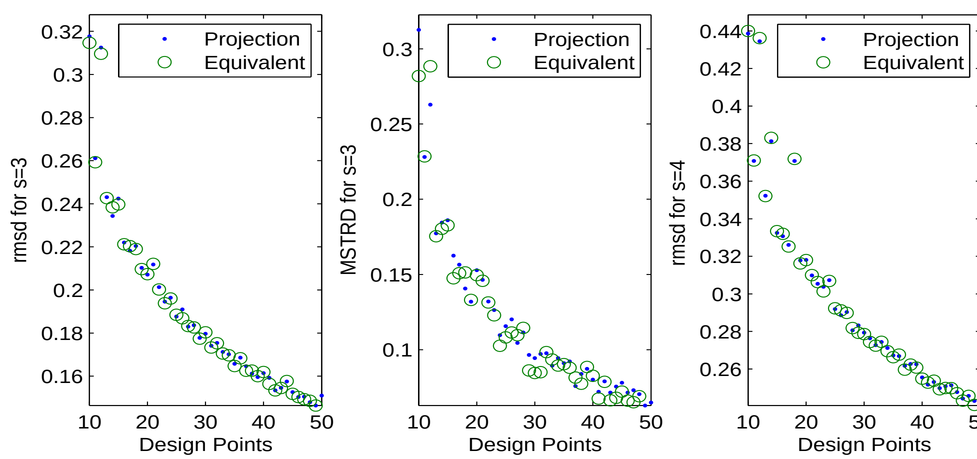


Figure 4.7: The projection and equivalent methods for design sizes $10, 11, \ldots, 50$ based on *rmsd* for dimension 3 and 4 and based on MSTRD for dimension 3 only.

DISCREPANCY MEASURES IN B^2 AND B^3 DESIGN REGIONS

Experimental design regions vary in form and shape with the hypercube and hypersphere (or ball) being the most common. In the field of response surface methodology, extensive research exists for designs in spherical design regions. For example, model-based designs like the popular spherical central composite designs (Box and Wilson, 1951), the Box-Behnken designs (Box and Behnken, 1960), and the hybrid designs (Roquemore, 1976), are designs for which spherical design regions are assumed.

An s-dimensional spherical design region of radius ρ (denoted $B^s(\rho)$) is defined to be the s-dimensional ball of radius ρ containing both the spherical surface and interior design points (Myers et al. (2009)). That is,

$$B^s(\rho) = \left\{ (x_1, x_2, \ldots, x_s) : \sum_{i=1}^{s} x_i^2 \leq \rho^2 \right\}.$$

The ball $B^s(1)$ of radius 1 will be denoted simply as B^s.

Transformation Of UDs From C^2 into B^2

Fang and Wang (1994) and Yuan and Fang (2010) discuss how UDs can be obtained in a two-dimensional ball B^2 which corresponds to the unit disk:

$$B^2 = \left\{ (x, y) : x^2 + y^2 \leq 1 \right\}.$$

Let the set $\mathcal{Q}_n = \{\mathcal{Q}_n(i) = (r_i, \theta_i), i = 1, 2, \ldots, n\}$ be points from a glp in C^2. Then, the set of points $\mathcal{P}_n = \{\mathcal{P}_n(i) = (r_i \cos(2\pi\theta_i), r_i \sin(2\pi\theta_i)), i = 1, 2, \ldots, n\}$ contains the n points in $\mathcal{Q}_n$ transformed from C^2 into B^2. However, despite the points of $\mathcal{Q}_n$ being uniformly scattered in C^2, this transformation does not preserve the uniformity of the scatter of the design points in B^2. To achieve uniformity in B^2 a different transformation is needed. Consider the probability density functions (pdf's) for r and θ given by

$$f_r(x) = \begin{cases} 2x & \text{for } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad f_\theta(x) = \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

with cumulative density function (CDF's) over the support given by

$$F_r(x) = x^2, \text{ for } 0 \le x \le 1 \quad \text{and} \quad F_\theta(x) = x, \text{ for } 0 \le x \le 1.$$

Applying inverse CDFs $F_r^{-1}(r_i)$ and $F_\theta^{-1}(\theta_i)$ for $i = 1, 2, \ldots, n$ leads to the transformations

$$x_i = \sqrt{r_i}\cos(2\pi\theta_i) \quad \text{and} \quad y_i = \sqrt{r_i}\sin(2\pi\theta_i), \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{5.1}$$

which are now uniformly scattered in B^2. Therefore, when (r_i, θ_i) are a set of uniformly scattered points in C^2, the set of transformed points (x_i, y_i) in (5.1) are uniformly scattered in B^2 (Fang and Wang (1994) and Yuan and Fang (2010)). Figure 5.1 is an illustration of the transformation from C^2 to B^2. That is, the shaded part of the square constructed by (r, θ) in C^2 is transformed into the sector shown in the unit disk B^2 in Figure 5.1.

Figure 5.1: Geometrical illustration of the transformation from C^2 to B^2

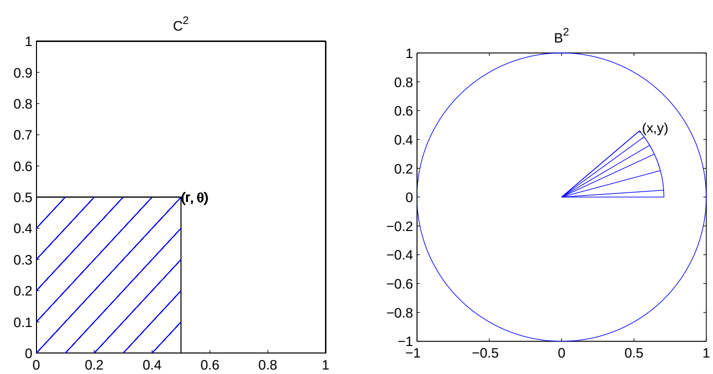
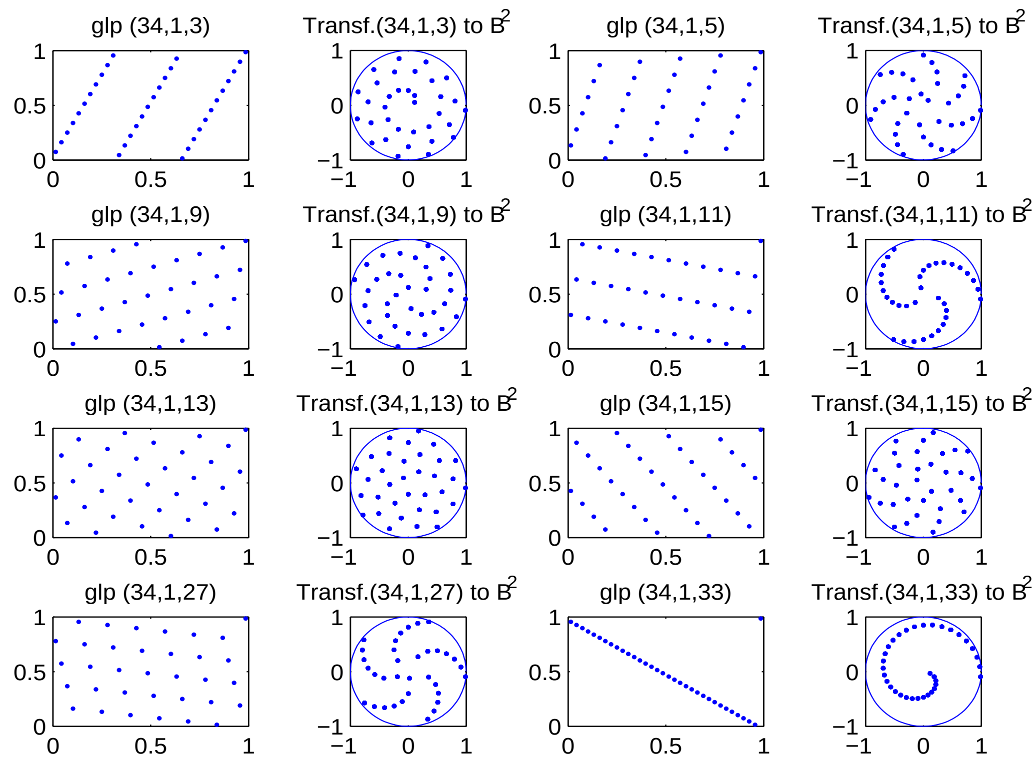


Figure 5.2 shows how the $n = 34$ design points for eight glp designs are scattered before and after the transformation from C^2 to B^2. However, note that the 8 glp designs in Figure 5.2 correspond to the non-equivalent generators in C^2.

Figure 5.2: Transformation of the non-equivalent 34 glp design pointsfrom C^2 to B^2



Non-equivalent glp generators are a set of generating vectors whose design points cannot be found by a column permutation of other generators from the set. In this dissertation the measures of uniformity developed are invariant under 90^o rotations in C^2 and, hence, there is no reason to have two or more equivalent generating vectors as candidate generators. Details of the proposed methods that identify the equivalent and non-equivalent generators were presented in Chapter 4.

For example, Table 5.1 gives the equivalent generators for $s = 2$ and $s = 3$ in C^s for $n = 11$ design points. Those that are in the first row of Table 5.1(a) and first column of Table 5.1(b) are referred to as the non-equivalent candidate generators in C^2 and C^3, respectively.

(a) Equivalent glp generators for $s = 2$ and $n = 11$

(1,h_i)	(1,2)	(1,3)	(1,5)	(1,7)	(1,10)
(1,h_j)	(1,6)	(1,4)	(1,9)	(1,8)	—

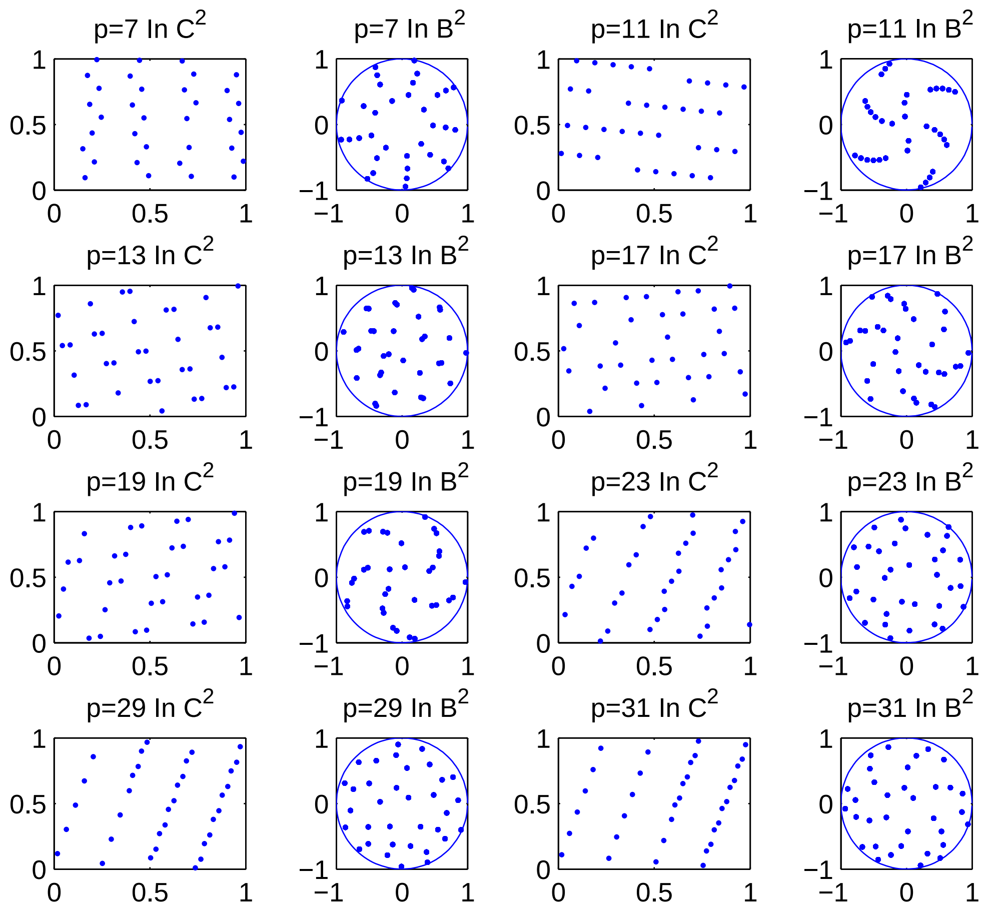
(b) Equivalent glp generators for $s = 3$ and $n = 11$

$(1, h_i, h_j)$	$(1, h_k, h_l)$	$(1, h_m, h_q)$
(1,2,3)	(1,6,7)	(1,4,8)
(1,2,4)	(1,2,6)	(1,3,6)
(1,2,5)	(1,6,8)	(1,7,9)
(1,2,7)	(1,6,9)	(1,5,8)
(1,2,8)	(1,4,6)	(1,3,7)
(1,2,9)	(1,6,10)	(1,5,10)
(1,2,10)	(1,5,6)	(1,9,10)
(1,3,4)	(1,4,5)	(1,3,9)
(1,3,5)	(1,4,9)	(1,5,9)
(1,3,8)	(1,4,10)	(1,7,10)
(1,3,10)	(1,4,7)	(1,8,10)
(1,5,7)	(1.8,9)	(1,7,8)

Table 5.1: Equivalent glp generators for $s = 2$ and $s = 3$ when $n = 11$

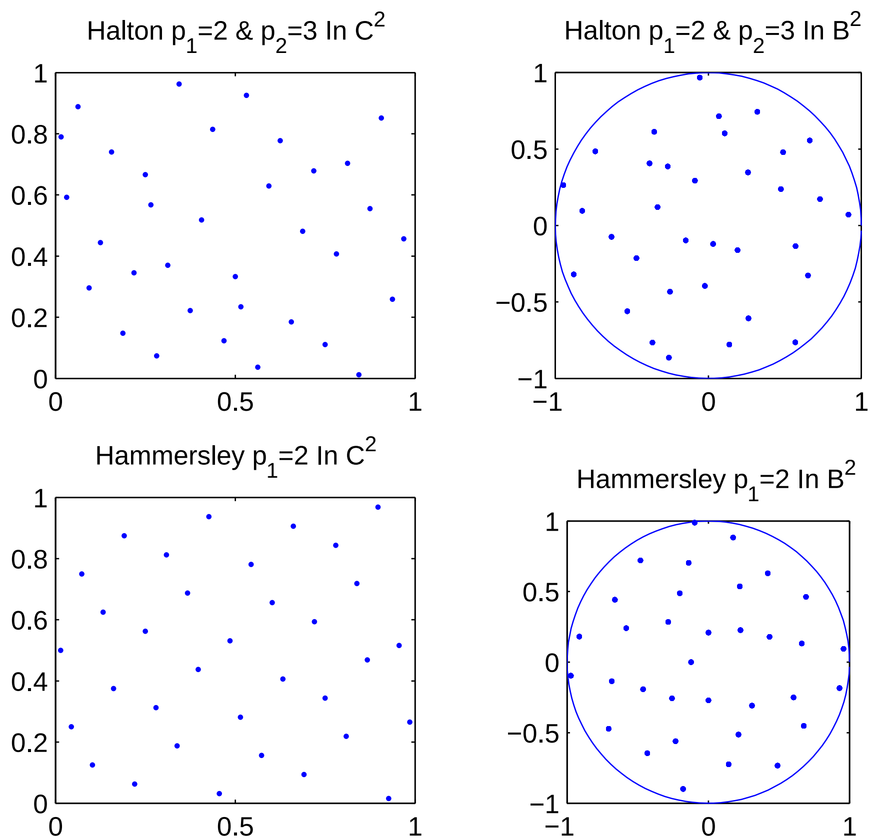
The design points generated by these non-equivalent generators are then transformed into the s-dimensional ball B^s $(s = 2, 3)$. In this dissertation the candidates or UD generating vectors in C^s and B^s are the same in the first stage. In other words, the first step is to transform the design points from the non-equivalent generators in C^s to B^s and identify the best UD generators using the measures of uniformities and then the search for the best UDs generators is restricted to the ones identified in the first step and their equivalent generators. For example, in C^2 for $n = 11$ all the design points from non-equivalent generators in C^2 are transformed to B^2 and the best UDs generators are reported as step 1. That is, for example, if the best glp has generator (1,2) for the *rmsd* criterion after transforming into B^2, in step 2, it will then be compared to the glp with its corresponding equivalent generator (1,6), and then the best design generator that has smallest *rmsd* criteria is reported as the final best generator in B^2.

Figure 5.3: Transformation of 34 CF(gp) design points from C^2 to B^2



Figures 5.2, 5.3 and 5.4 shows how the $n = 34$ design points are scattered after the transformation from C^2 to B^2 using glp method, CF method and (Halton and Hammersley) methods respectively.

Figure 5.4: Transformation of 34 Halton and Hammersley design points from C^2 to B^2



A New Discrepancy Measure Of Uniformity For B^2

In B^2, as in C^2, the degree of how well the points are uniformly scattered or the degree of uniformity has to be assessed numerically. In this subsection we present methods of measuring the uniformity of design points in B^2.

Suppose $\mathcal{Q}_n = \{\mathcal{Q}_n(i) = (r_i, \theta_i), i = 1, 2, \ldots, n\}$ is a glp set of points in C^2. The so-called star discrepancy measure is given by

$$D(\mathcal{Q}_n) = \sup_{(r,\theta)\in C^2} \left| \frac{N(r,\theta)}{n} - r\theta \right|, \tag{5.2}$$

where $N(r, \theta)$ is the number of points in $\mathcal{Q}_n$ that satisfy $r_i \leq r$ and $\theta_i \leq \theta$ (Fang and Wang (1994) and Yuan and Fang (2010)). Geometrically, $N(r, \theta)$ is the number of points that fall in the shaded rectangle of Figure 5.1 for C^2, and $r\theta$ is its area.

Analogously, let $\mathcal{P}_n$ be the set of points of $\mathcal{Q}_n$ transformed into B^2 using (5.1). Let (r^*, θ^*) be the polar coordinates for (x, y) in Figure 5.1. The corresponding star discrepancy in B^2 is measured using

$$D(\mathcal{P}_n) = \sup_{(r^*,\theta^*)\in B^2} \left| \frac{N(r^*, \theta^*)}{n} - \frac{r^{*2}\theta^*}{2\pi} \right|, \tag{5.3}$$

where $N(r^*, \theta^*)$ is the number of design points in B^2 that fall in the shaded sector in Figure 5.1 constructed by $r_i \leq r^*$ and $\theta_i \leq \theta^*$. Thus, $D(\mathcal{P})$ in (5.3) is a new measure of discrepancy in B^2 which is analogous to the discrepancy measured in C^2. It is calculated from the proportion of points $\frac{N(r^*,\theta^*)}{n}$ in the shaded sector constructed by r^* and θ^* and the x-axis to the ratio of the areas $\frac{Area(r^*,\theta^*)}{Area(B^2)} = \frac{r^{*2}\theta^*}{2\pi}$. In summary, to compute an estimated discrepancy for a design in B^2:

1. Transform the n points (r_i, θ_i) of a UD from C^2 into B^2 using the transformation $x_i = \sqrt{r_i}\cos(2\pi\theta_i)$ and $y_i = \sqrt{r_i}\sin(2\pi\theta_i)$ for $i = 1, 2, \ldots, n$.

2. Using the inverse tangent (arctan) function and the Pythagorean Theorem, convert each (x_i, y_i) in Step 1 into an angle θ^* and radius r^*.

3. Estimate the discrepancy $D(\mathcal{P}_n) = \sup_{(r^*,\theta^*)\in B^2} \left| \frac{N(r^*,\theta^*)}{n} - \frac{r^{*2}\theta^*}{2\pi} \right|$, where $N(r^*, \theta^*)$ is the number of points that fall in the sector constructed by r^* and θ^*. The supremum is estimated by taking the maximum of the absolute values over 4000 random evaluation points in B^2. The 4000 evaluation points are a stratified random sample of points in C^2 which are then transformed into B^2 using the above transformations.

As alternatives to discrepancy measures, the *rmsd*, *ad*, and *md* distance criteria do not require converting the random evaluation points (x_i, y_i) into angles and radii.

These three criteria are computed directly from the 4000 random evaluation points in B^2.

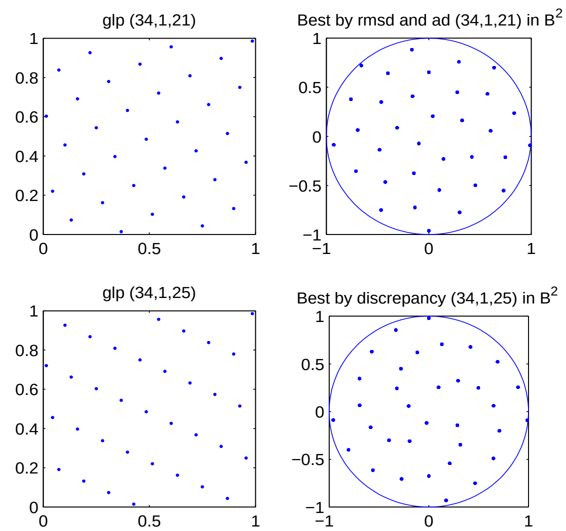
The 8 pairs of plots in Figure 5.2 show that the scatter pattern of the $n = 34$ glp points in C^2 is not preserved after the transformation into B^2 with most of the points scattered in patterns emanating from the center of the circle. Table 5.2 shows the values of the new discrepancy, *rmsd*, *ad*, and *md* measures of uniformity in B^2 for $n = 34$ design points from the non-equivalent glp generators after being transformed from C^2. A total of 4000 evaluation points in B^2 were used. The best measures of uniformity from the non-equivalent generators are shown by the bold-faced figures in Table 5.2. That is, after transformation from C^2 to B^2, the glp generator (1,13) is found to be the best for the three distance-based criteria whereas the glp generator (1,15) is the best for the new discrepancy measure. Now these best generators have to be compared with their corresponding equivalent generators. That is, (1,21) and (1,25) are the corresponding equivalent generators for (1,13) and (1,15), respectively, and their measures of uniformity are presented in Table 5.3. Thus, the bold-faced values in Table 5.3 shows (1,21) is the best based on *rmsd* and *ad*, based on *md* (1,13) is the best and the generator (1,25) is the best based on the new discrepancy measure. The best UDs generated by (1,21) and (1,25) are shown in Figure 5.5.

Measures of Uniformity				
Generator	*rmsd*	*ad*	*md*	$D(\mathcal{P}_n)$
(1,3)	0.1372	0.1260	0.2864	0.1032
(1,5)	0.1511	0.1369	0.3530	0.0875
(1,9)	0.1343	0.1244	0.2944	0.0727
(1,11)	0.2016	0.1759	0.4722	0.0854
(1,13)	**0.1314**	**0.1220**	**0.2766**	0.0536
(1,15)	0.1403	0.1284	0.3088	**0.0534**
(1,27)	0.1676	0.1483	0.4002	0.0678
(1,33)	0.2542	0.2166	0.6273	0.2460

Table 5.2: Measures of uniformity in B^2 for the glp method

Table 5.4 shows the values of four measures of uniformity for $n = 34$ design points after being transformed from C^2 to B^2 using the four methods of constructing uni-

Figure 5.5: Best UDs from glp method for $n = 34$ design points in B^2



Measures of Uniformity				
Generator	*rmsd*	*ad*	*md*	$D(\mathcal{P}_n)$
(1,13)	0.1314	0.1220	**0.2766**	0.0536
(1,15)	0.1403	0.1284	0.3088	0.0534
(1,21)	**0.1306**	**0.1212**	0.2909	0.0542
(1,25)	0.1354	0.1248	0.3058	**0.0498**

Table 5.3: Measures of uniformity in B^2 for the final best glp generators for $n = 34$ design points

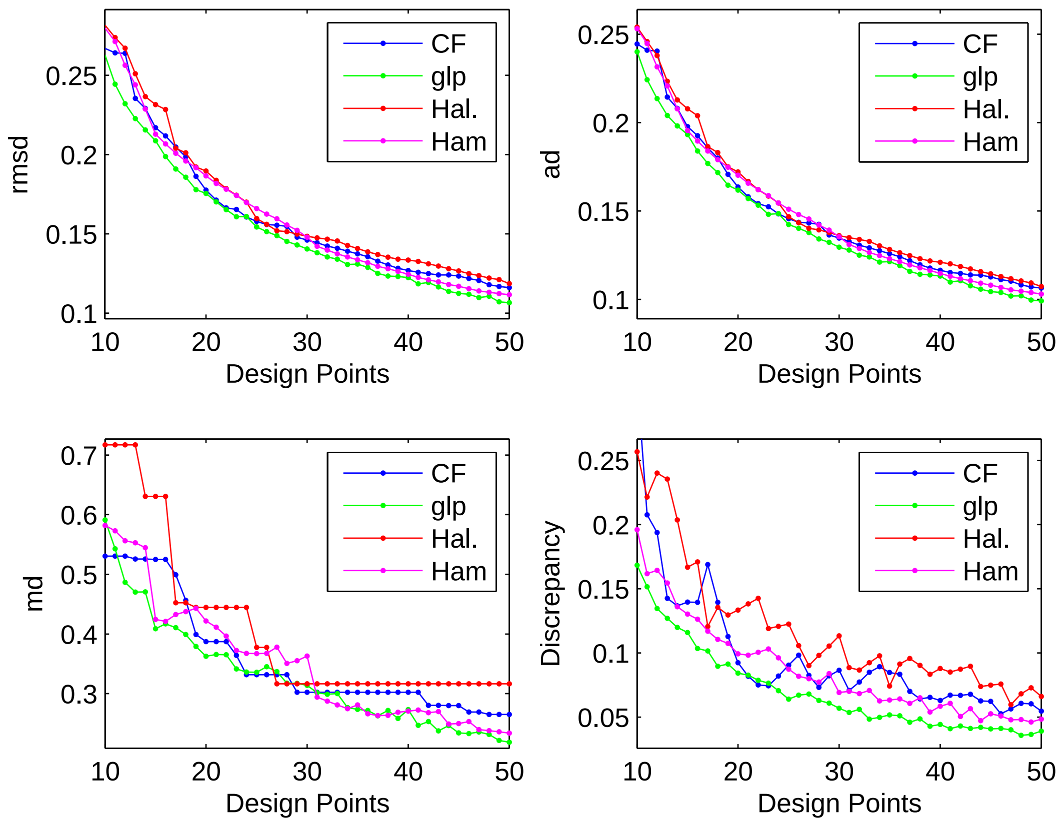
formly scattered design points. For the CF method and the (Halton and Hammersley) methods and the glp method the designs are plotted in Figures 5.3, 5.4 and 5.5, respectively. In the glp method after transformation from C^2 to B^2, the design generators selected as the best are different from those selected in C^2 meaning what is best in C^2 is not necessarily best in B^2. This is the main reason why the best non-equivalent generators have to be compared to their corresponding equivalent generators. For the CF (gp) method, the prime generator $p = 17$ was the best generator in C^2, but now after being transformed into B^2, $p = 29$ is found to be the best generator using the distance based criteria. However, the discrepancy criteria ($D(\mathcal{P}_n)$) indicated $p = 13$

Measures of Uniformity					
Methods	Generator	*rmsd*	*ad*	*md*	$D(\mathcal{P}_n)$
glp	(1,13)	0.1314	0.1220	**0.2766**	0.0536
	(1,15)	0.1403	0.1284	0.3088	0.0534
	(1,21)	**0.1306**	**0.1212**	0.2909	0.0542
	(1,25)	0.1354	0.1248	0.3058	**0.0498**
CF(gp)	p=7	0.1667	0.1495	0.3985	0.2079
	p=11	0.2078	0.1834	0.4958	0.1089
	p=13	0.1597	0.1469	0.3851	**0.0892**
	p=17	0.1635	0.1475	0.3975	0.1206
	p=19	0.1798	0.1608	0.4184	0.0928
	p=23	0.1434	0.1309	0.3170	0.1161
	p=29	**0.1391**	**0.1276**	**0.3025**	0.1408
	p=31	0.1459	0.1326	0.3242	0.1605
Halton	$p_1 = 2$, $p_2 = 3$	0.1427	0.1303	0.3166	0.0977
Hammersley	$p_1 = 2$	**0.1354**	**0.1248**	**0.2750**	**0.0626**

Table 5.4: Measures of uniformity in B^2 for the 4 methods ofconstructing UDs

as the best generator which is the same generator as in C^2. Looking at the magnitude of the evaluation criteria from Table 5.4, it seems the performance rank of the 4 methods is also preserved after transformation. Is it always true that the rank will be preserved for any number of design points after transformation from C^2 to B^2? To answer this question, we will consider how the methods perform for each evaluation criterion for $n = 10, ..., 50$. Figure 5.6 shows a graphical comparison of the methods in B^2 for different design sizes. Like the comparisons in C^2 in Figure 3.2, from Figure 5.6 one can clearly see that the glp method performs best for all 4 evaluation criteria followed by Hammersley method, CF method and Halton method, respectively. In order to investigate this further as in C^2, in B^2 we also need to consider plotting the relative efficiency (RE) of each method for each evaluation criterion. Since the glp method is expected to be the best, the way RE is defined is the same as in C^2 except that now we have only 4 evaluation criteria: *rmsd*, *ad*, *md*, and Discrepancy ($D(\mathcal{P}_n)$). There are no STRD and MSTRD associated with B^2. Figure 5.7 shows the relative efficiency of the methods in B^2 for each evaluation criterion.

Figure 5.6: Comparison of the 4 methods of constructing UDs in B^2 for $n = 10, 11, \ldots, 50$ design points in each evaluation criterion



The average relative efficiency (ARE) is also computed as in C^2 for each method and presented in Table 5.5. Table 5.5 clearly shows the Hammersley method is the second best after the glp method using all the criteria, followed by the CF(gp) method and the Halton method, respectively. Thus, this indicates that the performance rank of the methods in C^2 observed before transformation is preserved in B^2. However, note that in C^2 the ARE for the *md* criterion in Table 3.3 shows the CF method is the second best after the glp method followed by the Halton method and the Hammersley method, respectively. As previously seen, the UDs constructed using the glp method are the best. However, the number of candidate generating sets increases as the number of design points and dimensions increase, and thus it is computationally expensive.

Average Relative Efficiency Comparison				
Methods	*rmsd*	*ad*	*md*	$D(\mathcal{P}_n)$
CF(gp)	1.0548	1.0457	1.0943	1.4113
Hammersley	**1.0503**	**1.0412**	**1.0640**	**1.2520**
Halton	1.0903	1.0701	1.2342	1.7301

Table 5.5: Comparison of average relative efficiency in B^2 relative to glp

For future research discrepancy in B^2 may be computed by transforming the hyperrectangels in the modified star discrepancy in C^2 into sectors in B^2.

Transformation Of UDs From C^3 into B^3

In this section, methods of constructing UDs in B^3 via transformation of UDs in C^3 will be discussed. In texts, such as Fang and Wang (1994), a "sphere" corresponds only to the spherical surface and not the interior points. However, the spherical design region appearing in the response surface design literature is the three-dimensional unit ball B^3:

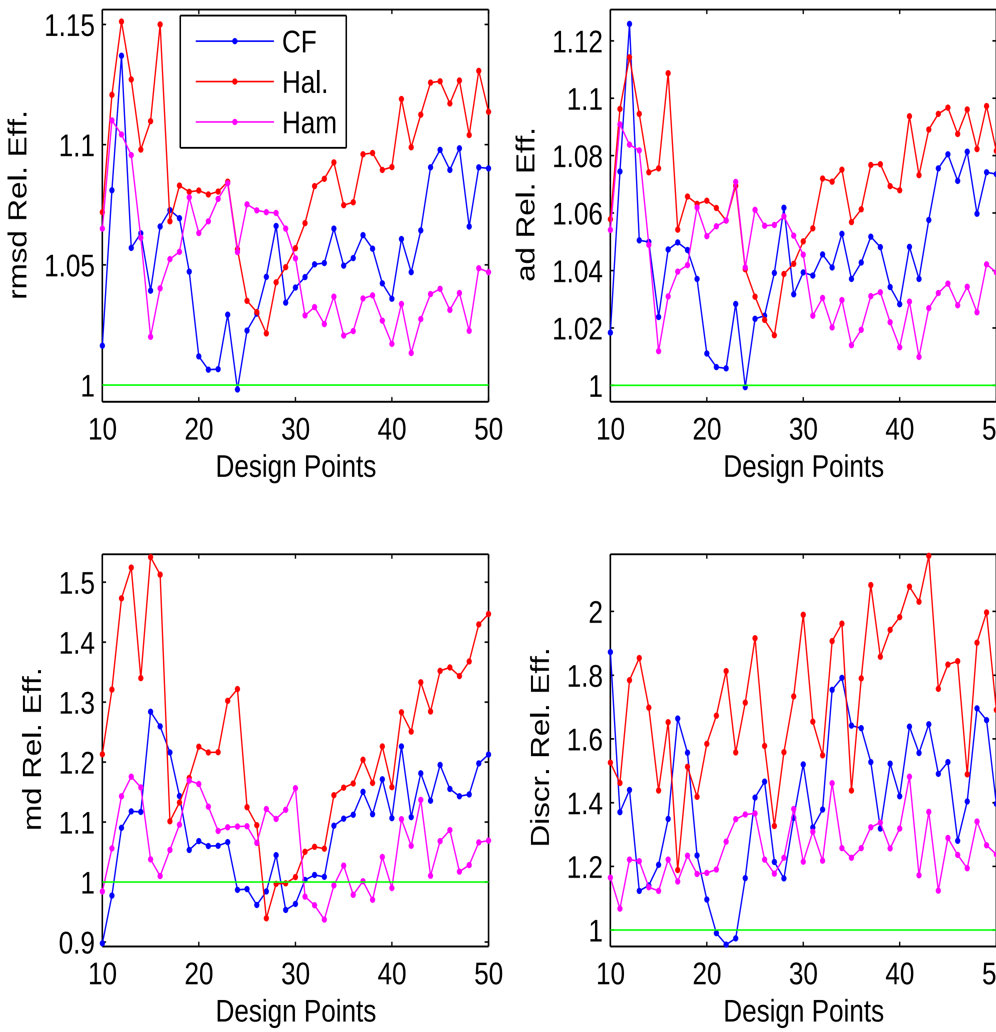
$$B^3 = \left\{ (x_1, x_2, x_3) : x_1^2 + x_2^2 + x_3^2 \leq 1 \right\}.$$

Let the set

$$\mathcal{Q}_n = \{\mathcal{Q}_n(j) = (c_{j1}, c_{j2}, c_{j3}), j = 1, 2, \ldots, n\}$$

be a set of UD points in C^3. Then, the spherical coordinate transformation given by

Figure 5.7: Relative efficiency of the methods of constructing UDs in B^2 relative to glp for each evaluation criterion for $n = 10, 11, \ \ldots \ , 50$ design points



$$x_{j1} = c_{j1}^{\frac{1}{3}}(1 - 2c_{j2}), \qquad x_{j2} = 2c_{j1}^{\frac{1}{3}}\left(\sqrt{c_{j2}(1 - c_{j2})}\right)\cos(2\pi c_{j3}) \qquad \text{and}$$

$$x_{j3} = 2c_{j1}^{\frac{1}{3}}\left(\sqrt{c_{j2}(1 - c_{j2})}\right)\sin(2\pi c_{j3}) \quad \text{for } j = 1, \ldots, n \tag{5.4}$$

transforms the UD points in C^3 into the ball B^3 (Fang and Wang, 1994).

Recall that a point (x_1,x_2) in B^2 can be represented in polar coordinates (r^*, θ^*) with a radius r^* and angle θ^*. Similarly, in B^3, each point $P = (x_1, x_2, x_3)$ in rectangular coordinates can be represented uniquely by the spherical coordinates (ρ, θ, ϕ) where

$$\rho = \sqrt{x_1^2 + x_2^2 + x_3^2}, \quad \theta = \arctan\left(\frac{x_2}{x_1}\right), \quad \text{and} \quad \phi = \arccos\left(\frac{x_3}{\rho}\right)$$

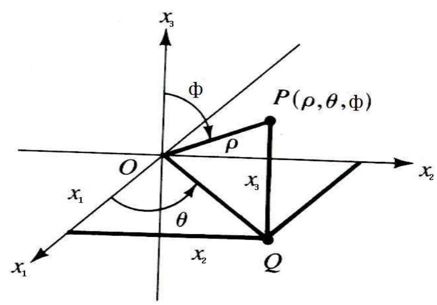
where ρ is the distance from origin O to P, θ is the angle formed by the projection of point P to the point Q in the x_1x_2-plane and ϕ is the angle between the x_3-axis and the line segment from origin O to P (see Figure 5.8, Riddle (1977)). Thus, $\rho \geq 0$, $0 \leq \theta \leq 2\pi$, and $0 \leq \phi \leq \pi$ (Riddle (1977); Varberg and Rigdon (2000); Harris and Stocker (1998).

In this dissertation the $\arctan 2$ function in Matlab (2010) is used to find θ instead of the arctan function. The reason is that the arctan function does not identify the quadrant of the inverse tangent whereas the $\arctan 2$ uses the sign of x_1 and x_2 to determine the quadrant and thus gives the four-quadrant inverse tangent of x_1 and x_2 (Aguilera and Aguila (2004) and Matlab (2010)). For example, $\arctan(\frac{1}{1}) = \frac{\pi}{4}$ and $\arctan(\frac{-1}{-1}) = \frac{\pi}{4}$, because arctan does not take signs into account. However, $\arctan 2(\frac{1}{1}) = \frac{\pi}{4}$, while $\arctan 2(\frac{-1}{-1}) = \frac{-3\pi}{4}$ gives the signed angle because it identifies the correct quadrant of the arctan of the angle.

A New Discrepancy Measure Of Uniformity For B^3

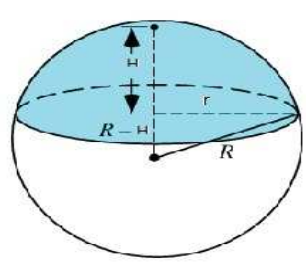
No statistical literature was found that addresses how discrepancy for uniformly scattered sets of design points can be measured for the set of transformed points in B^3. In this section, a new discrepancy measure is proposed and is applied to sets of points in B^3. Although it is simple to find the area of the random sector in B^2, finding the volume of a three-dimensional sector in B^3 is much more complicated. It depends

Figure 5.8: Geometrical illustration of the spherical coordinates



on multiple angles and a non-constant radius. Thus, even though one can visualize a 3D sector in the interior of B^3, there is no general closed-form formula to compute its volume because the height is not easily defined. Thus, a new alternative measure of discrepancy for B^3 is proposed. Unlike discrepancy in B^2, the new discrepancy for B^3 is defined in terms of the proportion of points in a varying volume formed by a random ball cap, or more simply, a random cap. Figure 5.9 is the graphical display of a cap and it is a modified version of the image from Weisstein (2011). In Figure 5.9, H is the height of the cap and r is the radius at the base of the cap, and R is the radius of B^3 which is scaled to have radius $R = 1$ in this research.

Figure 5.9: Geometric illustration of a cap in B^3



The height H uniquely determines the cap. It is determined by the direction of the axis in which slicing is done. In Figure 5.9 slicing is done along the x_3 axis. If x_3

is greater than 0 then $H = 1 - x_3$ and forms an upper cap. The volume of an upper cap when $R = 1$ is:

$$V_{cap}(H) = \frac{\pi}{3} H^2(3 - H). \tag{5.5}$$

However, if $x_3 < 0$, then the height H is obtained by $H = 1 - |x_3|$, and this corresponds to a lower cap. Before defining a new measure of discrepancy for designs in B^3, we must first define discrepancy in terms of the proportion of design points in an upper or lower cap.

Let $\mathcal{P}_n$ be the set of design points in B^3 formed by applying the spherical coordinate transformation in (5.4) to the points of a UD in C^3. Next, define $\mathcal{P}_n(\theta_R, \phi_R)$ be the set of points formed by rotating the points in $\mathcal{P}_n$ by rotation angles (θ_R, ϕ_R) which are, respectively, sets to be multiples of $\pi/3$ and $\pi/6$ radians. Specifically, $\theta_R \in \Theta = \{\frac{\pi}{3}, \frac{2\pi}{3}, \ldots, 2\pi\}$ and $\phi_R \in \Phi = \{\frac{\pi}{6}, \frac{\pi}{3}, \ldots, \pi\}$. Therefore, 36 different rotation combinations of θ_R and ϕ_R are considered. Finally, let $\gamma_i = (e_{i1}, e_{i2}, e_{i3})$ for $i = 1, \ldots, N$ form the set of N random evaluation points in B^3. The random points are stratified random points in C^3 and then transformed to B^3.

At first 144 different rotation combinations were considered but it was computationally expensive and hence we decided to use the 36 different rotation combinations. In fact when the 144 different angle rotations was used for $n = 34$ the glp generators (1,11,27) (1,11,27), (1,25,33), and (1,3,15) are best based on $rmsd$, ad, md and discrepancy criterions, respectively with measures of uniformity 0.3125, 0.2887, 0.7371, and 0.1868 respectively. The running time was 527min and hence this was the main reason why the 36 angle rotation is considered.

For a given (θ_R, ϕ_R), the discrepancies associated with an upper and lower cap now will be defined. Suppose B^3 is to be sliced perpendicular to the x_j-axis for $j =$

1, 2, or 3. If $e_{ij} \geq 0$ for evaluation point γ_i, we then form the upper cap in B^3 sliced at $x_j = e_{ij}$. For this upper cap compute:

$$D_{ij}(\mathcal{P}_n(\theta_R, \phi_R)) = \left| \frac{N_{ij}(\theta_R, \phi_R)}{n} - \frac{V_{cap}(H_{ij})}{4\pi/3} \right|, \tag{5.6}$$

where $N_{ij}(\theta_R, \phi_R)$ is the number of design points that fall in the upper cap (i.e., the number of design points (x_{i1}, x_{i2}, x_{i3}) in $\mathcal{P}_n(\theta_R, \phi_R)$ that satisfy $e_{ij} \leq x_{ij}$). Then the volume of the upper cap is computed by replacing H with $H_{ij} = 1 - e_{ij}$ in (5.5):

$$V_{cap}(H_{ij}) = \frac{\pi}{3} \left[H_{ij}^2 (3 - H_{ij}) \right]. \tag{5.7}$$

However, if $e_{ij} < 0$ for evaluation point γ_i, then we form the lower cap in B^3 sliced at $x_j = e_{ij}$. For this lower cap, interest is in $N_{ij}(\theta_R, \phi_R)$ = the number of design points that fall above the lower cap along the x_j axis (i.e., the number of design points (x_{i1}, x_{i2}, x_{i3}) in $\mathcal{P}_n(\theta_R, \phi_R)$ that satisfy $e_{ij} \leq x_{ij}$). The volume for the lower cap case is:

$$V_{cap}(H_{ij}) = \frac{\pi}{3} \left[4 - H_{ij}^2 (3 - H_{ij}) \right], \text{ for } i = 1, \ldots, N \tag{5.8}$$

where $H_{ij} = 1 - |e_{ij}|$. Note that the volume of the lower cap is subtracted from the volume of B^3 in (5.8) to find $V_{cap}(H_{ij})$ which is needed to calculate $D_{ij}(\mathcal{P}_n(\theta_R, \phi_R))$ in (5.6).

For each evaluation point $\gamma_i = (e_{i1}, e_{i2}, e_{i3})$, we slice along the three axes yielding three discrepancies, and then the maximum is taken:

$$D_i(\mathcal{P}_n(\theta_R, \phi_R)) = \max(\, D_{i1}(\mathcal{P}_n(\theta_R, \phi_R))\,,\, D_{i2}(\mathcal{P}_n(\theta_R, \phi_R))\,,\, D_{i3}(\mathcal{P}_n(\theta_R, \phi_R))\,).$$

This process is repeated for each of the $N = 10000$ γ_i evaluation points, and a maximum is taken again. This defines the discrepancy measure for a given rotation (θ_R, ϕ_R):

$$D(\mathcal{P}_n(\theta_R, \phi_R)) = \max_{i=1,2,\ldots,N} D_i(\mathcal{P}_n(\theta_R, \phi_R))) \tag{5.9}$$

Note that evaluation point γ_i determines which cap to consider. However, discrepancies defined in upper and lower caps or their complementary regions yield the same result. Thus, we only need to compute a total of 3 different discrepancies for each evaluation point.

By definition, the discrepancy measure associated with design $\mathcal{P}_n$ is the maximum of $D(\mathcal{P}_n(\theta_R, \phi_R))$ taken over the set of rotations. That is, $D(\mathcal{P}_n)$, the new measure of discrepancy of designs in B^3, is defined as

$$D(\mathcal{P}_n) \;=\; \max_{\theta_R \in \Theta, \phi_R \in \Phi} D(\mathcal{P}_n(\theta_R, \phi_R)) \tag{5.10}$$

Finding the best UD in B^3 for the glp method using $D(\mathcal{P}_n)$ requires generating all n-point non-equivalent glp designs in C^3. Then, for each of these glp-based designs and for CF(gp), Halton, and Hammersley designs, transform the points to form a design $\mathcal{P}_n$ in B^3, and the design $\mathcal{P}_n^*$ which minimizes $D(\mathcal{P}_n)$ is the best UD in B^3. The same procedure applies when finding the best UD in B^3 using any of the three distance-based criteria ($rmsd$, ad, or md).

Table 5.6 contains the measures of uniformity for the non-equivalent glp generators transformed from C^3 to B^3 for $n = 34$ design points calculated over a set of 10000 evaluation points in B^3. Also, the 36 different angle rotations $\theta_R \in \Theta$ and $\phi_R \in \Phi$ were used to compute the discrepancies $D(\mathcal{P}_n)$ in Table 5.6. The best measures of uniformity from the non-equivalent glp generators are shown by the bold-faced values. That is, (1,13,25), (1,11,27), (1,3,25), and (1,3,11) are the best generators based on the $rmsd$, md, ad and the discrepancy criteria, respectively. However, these best generators have to be compared with their corresponding equivalent generators. Comparison of the measures of uniformity is done among the equivalent glp generators shown in Table 5.7. This is because (1,13,25) is equivalent to (1,15,21) and (1,15,25), (1,11,27) is equivalent to (1,21,31) and (1,13,29), and (1,3,11) is equivalent to (1,15,23) and (1,25,31). The measures of uniformities are presented in Table 5.7 and the bold-faced values show the glp generators (1,13,25), (1,11,27), (1,3,25) and (1,15,21) as the best based on *rmsd*, *ad*, *md* and the new discrepancy criteria, respectively. The equivalence computational reduction method reduces the running time by approximately 1/3.

Generator	$rmsd$	ad	md	$D(\mathcal{P}_n)$	Generator	$rmsd$	ad	md	$D(\mathcal{P}_n)$
(1,3,5)	0.3144	0.2928	0.7437	0.1978	(1,5,27)	0.3156	0.2926	0.7430	0.2036
(1,3,7)	0.3273	0.3022	0.8377	0.2305	(1,5,29)	0.4834	0.4277	1.1312	0.4104
(1,3,9)	0.3177	0.2955	0.7346	0.2086	(1,5,31)	0.3265	0.3018	0.7577	0.1929
(1,3,11)	0.3118	0.2906	0.7626	**0.1745**	(1,5,33)	0.3478	0.3200	0.8272	0.2189
(1,3,13)	0.3263	0.2995	0.8069	0.2029	(1,9,13)	0.3348	0.3087	0.8261	0.1925
(1,3,15)	0.3211	0.2968	0.7495	0.1868	(1,9,15)	0.3157	0.2922	0.7807	0.1912
(1,3,19)	0.3216	0.2967	0.7696	0.1946	(1,9,21)	0.3373	0.3089	0.8558	0.2102
(1,3,21)	0.3217	0.2970	0.8267	0.2419	(1,9,25)	0.4829	0.4246	1.1589	0.4180
(1,3,25)	0.3180	0.2959	**0.7149**	0.2207	(1,9,31)	0.3642	0.3301	0.9645	0.2296
(1,3,27)	0.3283	0.3040	0.7988	0.2328	(1,9,33)	0.3227	0.3003	0.7405	0.2024
(1,3,29)	0.3150	0.2931	0.7349	0.1944	(1,11,13)	0.3221	0.2975	0.7709	0.1803
(1,3,31)	0.4795	0.4231	1.1154	0.4236	(1,11,25)	0.3232	0.2974	0.8042	0.2191
(1,3,33)	0.3938	0.3618	0.9935	0.2299	(1,11,27)	0.3125	**0.2887**	0.7681	0.1935
(1,5,7)	0.3110	0.2892	0.7573	0.1867	(1,11,29)	0.3208	0.2964	0.7441	0.1842
(1,5,9)	0.3185	0.2946	0.7480	0.1834	(1,13,21)	0.4739	0.4172	1.1542	0.4244
(1,5,11)	0.3293	0.3048	0.8360	0.1975	(1,13,25)	**0.3102**	0.2889	0.7515	0.1800
(1,5,19)	0.3230	0.3012	0.7309	0.2324	(1,15,27)	0.3193	0.2960	0.7607	0.2064
(1,5,21)	0.3639	0.3274	0.9403	0.2300					

Table 5.6: Measures of uniformity in B^3 for glp-transformed 34point designs

Table 5.8 shows the 4 measures of uniformity for $n = 34$ design points on B^3 from the 4 methods of constructing uniformly scattered design points. The bold-faced values indicate the best UDs generators. The integers following "/ " in Table 5.8 indicate which pair of angles of rotations $\theta_R \in \Theta$ and $\phi_R \in \Phi$ (θ and ϕ) are the ones that maximize $D(\mathcal{P}_n(\theta_R, \phi_R))$, the reported value before "/ " . That means, the values appearing in Table 5.8 before "/ " are the maximum of $D(\mathcal{P}_n(\theta_R, \phi_R))$ after rotating the design points an angle $\theta_R = (q_1 + 1) \times \frac{\pi}{3}$ and an angle $\phi_R = q_2 \times \frac{\pi}{6}$, where q_1 is the integer obtained after dividing the integer after "/ " by 6, and q_2 is the integer remainder. Note that if there is no remainder after dividing the number

Measures of Uniformity				
Generator	$rmsd$	ad	md	$D(\mathcal{P}_n)$
(1,13,25)	**0.3102**	0.2889	0.7515	0.1800
(1,15,21)	0.3117	0.2889	0.7472	**0.1734**
(1,15,25)	0.3207	0.2966	0.7731	0.1955
(1,11,27)	0.3125	**0.2887**	0.7681	0.1935
(1,21,31)	0.3144	0.2914	0.7856	0.2096
(1,13,29)	0.3238	0.2984	0.7785	0.2300
(1,3,25)	0.3180	0.2959	**0.7149**	0.2207
(1,23,31)	0.3261	0.3006	0.8382	0.2059
(1,11,25)	0.3643	0.3303	0.9664	0.2049
(1,3,11)	0.3118	0.2906	0.7626	0.1745
(1,15,23)	0.3219	0.2985	0.8033	0.2119
(1,25,31)	0.3637	0.3299	0.9406	0.2056

Table 5.7: Measures of uniformity in B^3 for the final best glpgenerators for $n = 34$

after "/ " by 6 then the pair of angle of rotations used are $\theta_R = q_1 \times \frac{\pi}{3}$ and an angle $\phi_R = \pi$. Also if the number after "/ " is less than or equal to 6 use $q_1 = 0$ and q_2 equals that number. For example, the first row of Table 5.8 for the glp method for the $rmsd$ criterion shows 0.3102 /6, which means the $rmsd$ for generator (1,13,25) is 0.3102 and the maximum $rmsd$ occurs when the design points are rotated at an angle $(\frac{\pi}{3})$ for θ_R and $\phi_R = \pi$. This is because the number after "/ " is less than or equal to 6 and hence use $q_1 = 0$, and $q_2 = 6$, in the formula for θ_R and ϕ_R above. Similarly, the md criterion for (1,13,25) is 0.7515 /2 indicates the maximum md reported is obtained when the design points are rotated at an angle $(\frac{\pi}{3})$ for θ_R and $2(\frac{\pi}{6})$ for ϕ_R, and corresponds to $q_1 = 0$ and $q_2 = 2$. For the CF method based on the distance-based criteria, $p = 23$ is the best generator in B^3 but after rotating the design points at an angle $\theta_R = 4(\frac{\pi}{3})$ and $\phi_R = 1(\frac{\pi}{6})$ (for $rmsd$ and ad), while based on the discrepancy criterion ($D(\mathcal{P}_n)$), $p = 13$ is the best generator but after rotating the design points at an angle $\theta_R = 4(\frac{\pi}{3})$ and $\phi_R = 3(\frac{\pi}{6})$. Similarly, the bold faced values in Table 5.8 show the design points generated in B^3 using the Hammersley method are better relative to the Halton method based on ($rmsd$, ad, and $D(\mathcal{P}_n)$) after rotating the design points. However, the pair of rotation angles are different for the distance-based ($rmsd$ and

ad) and discrepancy ($D(\mathcal{P}_n)$) criteria as shown in the last row of Table 5.8. The best measures of uniformity for all the methods are shown by the bold faced values in Table 5.8.

The rank order of the methods in C^3 in Table 4.3 is not preserved in B^3 at least for the $n = 34$ point designs. Based on ($rmsd$, ad, and $D(\mathcal{P}_n)$) the rank order of the methods as shown in Table 4.3 from best to worst is

$$\text{(Best)} \quad \text{glp} \Longrightarrow \text{Hammersley} \Longrightarrow \text{Halton} \Longrightarrow \text{CF(gp)} \quad \text{(Worst)}$$

and based on md, the rank order of the methods from best to worst is

$$\text{(Best)} \quad \text{glp} \Longrightarrow \text{Halton} \Longrightarrow \text{CF(gp)} \Longrightarrow \text{Hammersley} \quad \text{(Worst)}.$$

Note that even though the glp method is best by all measures of uniformity, it is computationally expensive. The run-time for the Matlab codes for the glp method is about 138 minutes, about 20 minutes for the CF(gp) method and about 6 minutes for the combined Halton and the Hammersley methods. If the computational task reduction method was not introduced, then the estimated running time would have tripled to approximately 7.5 hours. These times are hardware dependent and it may vary depending on the speed of the computer. In this research, a laptop that has a processor of 2.13GHz was used.

The $n = 34$ design points for the best designs generated by the glp, the CF, and the Halton and Hammersley methods are displayed graphically in Figures 5.10, 5.11 and 5.12 respectively. The built in functions plot3, scatter3 and surf of Matlab (2010) are used to plot Figures 5.10, 5.11 and 5.12.

Methods	Generator	$rmsd$	ad	md	$D(\mathcal{P}_n)$
glp	(1,13,25)	**0.3102 /6**	0.2889 /6	0.7515 /2	0.1800 /6
	(1,15,21)	0.3117 /18	0.2889 /34	0.7472 /3	**0.1734 /35**
	(1,15,25)	0.3207 /31	0.2966 /15	0.7731 /34	0.1955 /33
	(1,11,27)	0.3125 /24	**0.2887 /24**	0.7681 /19	0.1935 /6
	(1,21,31)	0.3144 /11	0.2914 /11	0.7856 /14	0.2096 /18
	(1,13,29)	0.3238 /5	0.2984 /5	0.7785 /34	0.2300 /33
	(1,3,25)	0.3180 /5	0.2959 /5	**0.7149 /5**	0.2207 /29
	(1,23,31)	0.3261 /5	0.3006 /5	0.8382 /34	0.2059 /16
	(1,11,25)	0.3643 /22	0.3303 /33	0.9664 /4	0.2049 /5
	(1,3,11)	0.3118 /23	0.2906 /23	0.7626 /2	0.1745 /36
	(1,15,23)	0.3219 /33	0.2985 /30	0.8033 /32	0.2119 /23
	(1,25,31)	0.3637 /33	0.3299 /17	0.9406 /22	0.2056 /23
CF(gp)	p=11	0.3807 /20	0.3518 /21	0.9372 / 33	0.2475 /5
	p=13	0.3760 /20	0.3483 /20	0.8463 /17	**0.2332 /21**
	p=17	0.3575 /33	0.3282 /33	0.8838 /16	0.2372 /18
	p=19	0.3670 /12	0.3400 /11	0.8035 /2	0.2375 /13
	p=23	**0.3492 /19**	**0.3227 /19**	**0.7896 /24**	0.2706 /31
	p=29	0.3526 /8	0.3257 /7	0.8192 /11	0.2659 /7
	p=31	0.3565 /5	0.3296 /5	0.8125 /17	0.2980 /31
Halton	$p = 2, 3$ and 5	0.3241 /29	0.2991 /29	**0.7619 / 10**	0.2280 /21
Hammersley	$p = 2,$ and 3	**0.3205 /23**	**0.2954 /23**	0.8340 /4	**0.2273 /24**

Table 5.8: Measures of uniformity in B^3 for the 4 methods of constructing UDs for $n = 34$

In summary, the algorithm for computing an estimated discrepancy and distance-based measures of uniformity on a unit ball B^3 is given below:

1. Find an NT-net in the three-dimensional unit cube C^3.
2. Transform the uniformly scattered points on C^3 into B^3 to find (x_{j1}, x_{j2}, x_{j3}) for $j = 1, ..., n$. That is, find $\mathcal{P}_n$.
3. Convert the n points in step 2 into spherical coordinate triplets $(\rho_j, \theta_j, \phi_j)$.
4. Rotate the design points for a set of all possible ordered pairs of angles (θ_R, ϕ_R) for each generator.

Figure 5.10: $n = 34$ design points in B^3 from glp method

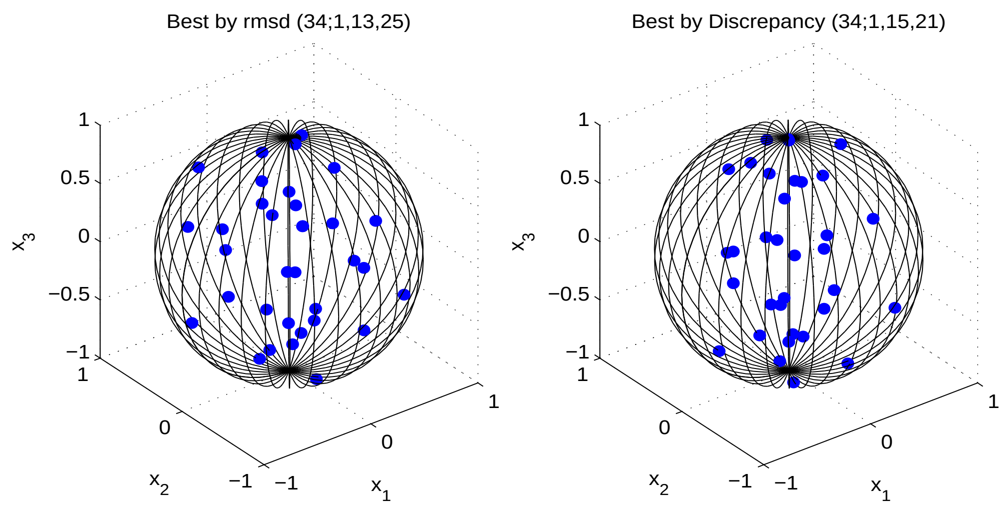
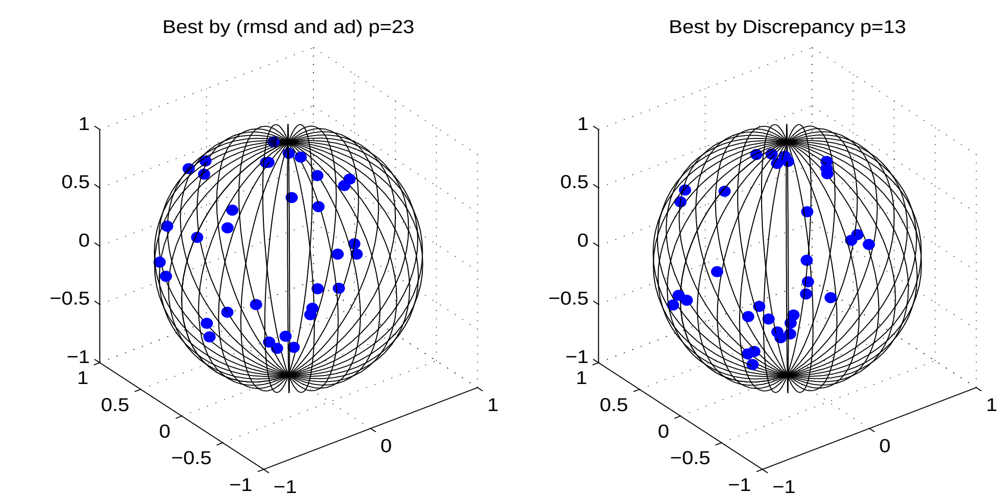
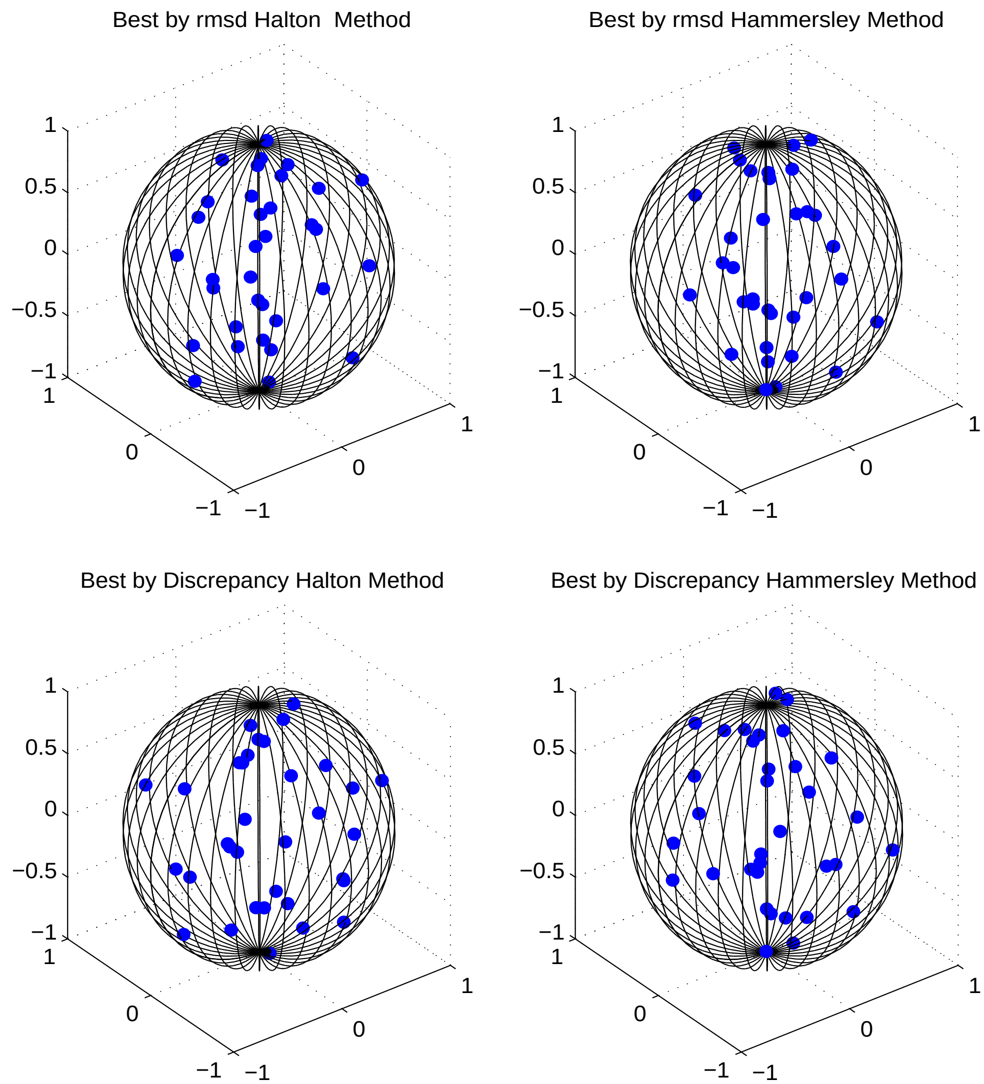


Figure 5.11: $n = 34$ design points in B^3 from CF method



5. Compute $D_{ij}(\mathcal{P}_n(\theta_R, \phi_R))$ in equation 5.6 for $i = 1, ..., N$, evaluation points and by slicing the experimental region along the three primary axes or x_j axis for $j = 1, 2, 3$.

Figure 5.12: $n = 34$ design points in B^3 from Halton andHammersley methods



6. Find $D_i(\mathcal{P}_n(\theta_R, \phi_R))$ for each of the $N = 10000$ evaluation points and then find $D(\mathcal{P}_n(\theta_R, \phi_R))$ in equation 5.9.

7. Compute the 3 distance-based ($rmsd$,ad and md) criteria for the design points from the generator for each rotation. The maxima for each of the rotation-

generated distance criterion values from the 3 primary axes are saved, and the maximum of the 36 maxima is reported for each generator.

8. Repeat steps 1-7 for all candidate generators for each method.

9. Finally, the generator that has the smallest measure of uniformity is considered to be the best for each method.

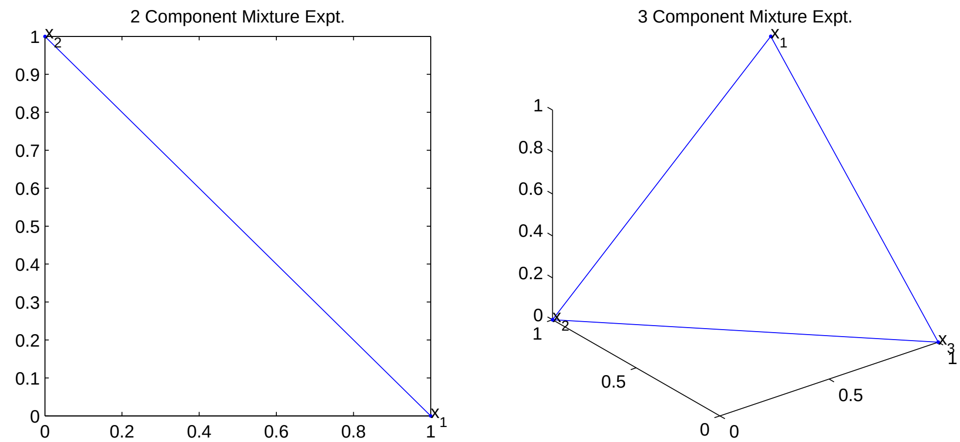
For future research discrepancy may be computed for regions other than cap such as spherical wedge.

MIXTURE DESIGNS

In product formulation and process engineering, end products are formed by mixing two or more components (ingredients). For example, in the building construction industry, concrete is made by mixing sand, water and cement. A second example is the formulation of a fruit juice blend, such as by mixing pineapple and orange juices. The overall aim of a mixture experiment is to search for a blended product or mixture that is cost effective, has superior quality, and is more desirable than a single ingredient product. For the fruit blend example, instead of having pure 100% orange juice or pure 100% pineapple juice, the idea is to find a mixture at a certain proportion of each to create a blended product, and see if this blend could increase demand, quality of juice, and boost profit from the manufacturer's point of view. Similarly, the strength (quality) of the concrete depends on the proportions of sand, water and cement present. The quality and property of the blended juice depends on the proportion of each of the fruits present. These examples highlight why a mixture experiment is a special type of response surface experiment. Specifically, it is the proportion of the input variables (ingredients or components) that affect the response rather than the amounts of the ingredients in the mixture. Excellent extensive reviews and discussion about mixture experiments can be found in Cornell (2002) and Smith (2005).

Geometrically, the experimental region for the 2 and 3 component mixture experiments are presented in Figure 6.1. The experimental region for the 2 component mixture experiment is the diagonal line segment ($x_1 + x_2 = 1$) between (1,0) and (0,1). The fruit juice example is a 2 component mixture experiment with x_1 denoting the proportion of orange juice and x_2 denoting the proportion of pineapple juice. The pure orange blend and pure pineapple blend occur at the indicated end points $(1, 0)$ and $(0, 1)$ of the line segment respectively, whereas the binary blended mixtures of the two components occur on the remaining part of the line segment $x_1 + x_2 = 1$. For

Figure 6.1: Two and three component mixture experiments geometrically



a 3 component mixture experiment, the experimental region is a simplex (equilateral triangle) as shown in Figure 6.1. For a 3 component mixture experiment of fruit juice, the pure blends occur at the 3 vertices of the simplex $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. A binary blend has one of the 3 components occur at a point on one of the three sides of the equilateral triangle, and the ternary blends that has a positive proportion for each of the 3 components occur in the interior of the triangle. For a 4 component mixture experiment the experimental region is a 3-dimensional simplex (tetrahedron). A k component mixture experiment has a maximum of k pure blends, $\binom{k}{2}$ choices of 2 components in a binary mixture blend, $\binom{k}{3}$ choices of 3 components in a ternary mixture blend and so on. However, due to scarcity of resources it is often not possible to consider every possible mixture in an experiment and test for the best end product quality in the entire experimental region. Thus, how should the proportions of each ingredient for each experimental mixture be selected from the infinite possible mixtures in the entire experimental region? The simplex lattice designs introduced by Scheffe (1958) and the simplex centroid and axial designs have been used to address this question (Cornell, 2002; Myers et al., 2009; Smith, 2005; Fang and Wang, 1994). In an axial design there are only s design points and these are obtained by taking

s equidistant design points from the line segment connecting each vertex and the centroid of the simplex (Fang and Wang, 1994). The distance d of the s design points from the centroid is selected such that $0 < d < \frac{s-1}{s}$. However, these designs have problems because the experimental design points obtained are not evenly scattered on the experimental region. Optimal computer generated designs have also been used but these designs put a large proportion of design points on the boundary of the experimental region. Each of these designs are also model dependent (Chan, 2000). Thus, a method that provides more uniform coverage of the experimental region is required and one approach is to use number theoretic methods (NTMs), specifically Uniform Designs of Experiments with Mixtures (UDEM) to find the proportion of ingredients for a set of mixtures that are uniformly scattered in the mixture design region. One approach would be to use a Scheffe type design and to contract the experimental points by moving them to the inner part of the experimental region while preserving the pattern of the points (Fang and Wang, 1994). The complete review of the Scheffe type of design is found in (Fang and Wang, 1994). However, in this study, the conditional distribution Monte-Carlo approach developed by Fang and Yang (2000) will be used.

A second method would be to use mesh generating as was done by Linse (2008). However, the mesh approach also has a limitation as it places a large number of design points on the boundary of the experimental mixture region. Even though this problem could be solved using a Scheffe type of design and contracting the experimental points while preserving the pattern of the design points, the mesh method currently does not exist for higher dimensional cases (Personal communication with Borkowski 2010).

Next, consider the most common situation when there are additional constraints placed on the component proportions. For example, consider a 3 component fruit blend mixture experiment. Suppose that the proportion of orange juice has to be between 10% and 70%, the proportion of pineapple juice between 0% and 80% and the proportion of watermelon juice between 10% and 60%. Suppose also that the proportion of the total blend from orange and pineapple mixed has to be between 10% and 80%. If x_1, x_2 and x_3 respectively denote orange, pineapple and watermelon juice proportions, mathematically these constraints are summarized below:

$$0.1 \leq x_1 \leq 0.7, \quad 0 \leq x_2 \leq 0.8, \quad 0.1 \leq x_3 \leq 0.6, \quad 0.1 \leq x_1 + x_2 \leq 0.8,$$

in addition to the basic constraint $x_1 + x_2 + x_3 = 1$

The constraints for each ingredient are determined from experience or from the natural property of the ingredients by the investigator or experts on the field. The size of the experimental region is reduced from the simplex due to the restrictions imposed. That is, the more restrictions (constraints) imposed to the components or combination of components, the smaller the size of the experimental region and the more irregular in shape it may become.

Suppose the mixture experiment has q different ingredients and let x_i denote the i^{th} proportion for the i^{th} ingredient. Then, a mixture experiment has the following constraints that need to be met:

1. Non-negative component proportion constraint. That is $x_i \geq 0$ for $i = 1, ..., q$.

2. $\sum_{i=1}^{q} x_i = 1$. That is the mixture component proportions (percentages) must sum to 1 (100%) according to this constraint. As a result, the level of each ingredient cannot be selected independently. For example, if we know the proportion of orange and pineapple in the 3 component blend then we also know the proportion of watermelon. It is this condition that makes the levels of $x_i's$ dependent and also makes a mixture experiment different than a traditional response surface experiment (Borkowski and Piepel (2009), Myers et al. (2009) and Cornell (2002)).

   Constraints 1 and 2 have to always be met for every mixture design. A mixture design that also has lower and upper limit constraints for each component is called a single component constraint (SCC) design. If, in addition to the restriction imposed for a SCC design, there is also at least one constraint on a linear combination of component proportions then the mixture design is called a multiple component constraint (MCC) design. Thus, depending on whether the mixture design is a SCC or MCC design, the following restrictions are added to the constraints imposed in 1 and 2:

3. For a SCC mixture experiment with q components, the lower and upper limits for each ingredient proportion must be between 0 and 1. That is

$$0 \leq a_i \leq x_i \leq b_i \leq 1$$

   for $i = 1, ..., q$ where a_i and b_i are lower and upper limits of the i^{th} component respectively.

4. For a MCC mixture experiment, there are constraints formed by linear combinations of q components that satisfy

$$L_m \leq \sum_{i=1}^{q} D_{im} x_i \leq U_m$$

   where L_m, D_{im} and U_m are all constants for the m^{th} MCC.

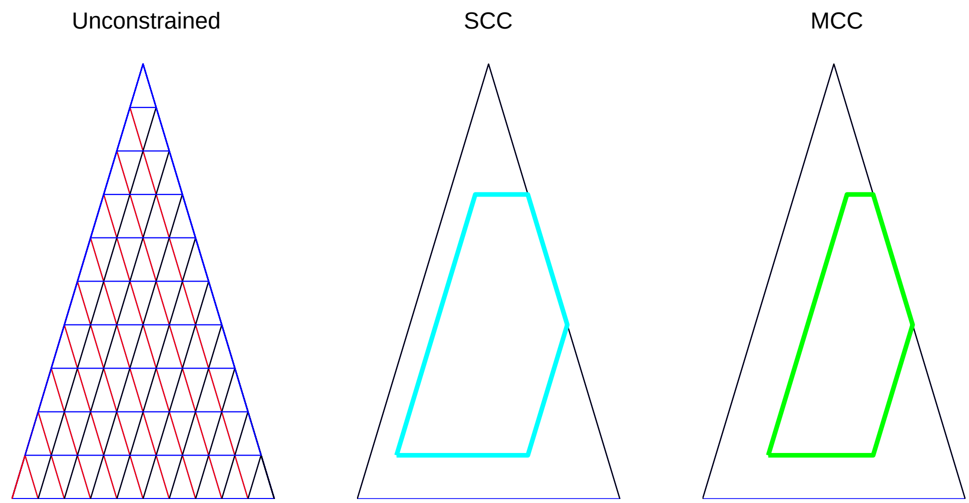
The MCC imposed on the fruit juice example above is

$$0.1 \leq x_1 + x_2 \leq 0.8.$$

If the MCC is dropped then the mixture design becomes a SCC design. Thus, note that the SCC is a special case of the MCC design.

Figure 6.2 shows how the experimental region for a 3 component mixture design changes with the different type of constraints. That is, the mixture experimental region gets smaller and smaller with the increase in the number and type of constraints.

Figure 6.2: Geometrical illustration of the experimental regions for Unconstrained, SCC and MCC mixture designs



Construction Of SCC Mixture Designs

Fang and Yang (2000), Prescott (2008) and Pham (2006) have an excellent review of the method of constructing mixture designs using NTMs. In constructing mixture designs the results from Fang and Yang (2000) will be used. Their method is based on the application of conditional distributions for the mixture components and will be presented later in this section. In C^2, the good lattice point (glp) method was the best in yielding low discrepancy design points. In constructing mixture design points, the glp method will be used to find the design points that are equally spaced in C^2 and then using the inverse transformation from Fang and Yang (2000) to map the design points into the full simplex. The full simplex, denoted S^3, refers to the mixture design space that satisfies only constraints 1 and 2 stated above. The full simplex is also called the non-restricted simplex. Fang and Yang (2000), Pham (2006) and Prescott (2008) used the inverse transformation approach to find the design points in the full simplex. The statistical literature has been focused on design points for the non-restricted simplex region that are obtained from the inverse transformation used by Fang and Yang (2000), Pham (2006) and Prescott (2008). However, in this dissertation, an additional step is implemented. That is, the G-mapping function that is used for restricted simplex region will be applied and discussed later in this section. Borkowski and Piepel (2009) have used the G function which tends to yield evenly scattered points in subspaces of the full simplex that preserve the general pattern of the design points observed in C^{3-1}. The method and steps will now be described and illustrated with a 3 components mixture experiment:

1. For designs in the full simplex (S^3), generate n design points in C^2 using the glp method. Suppose the design points in C^2 are denoted by $\mathbf{c}_k = (c_{k1}, c_{k2})$ for $k = 1, ..., n$. Then, these design points are transformed using the following inverse transformation into an NT-net on the full simplex S^3 :

$$x_{k1} = 1 - (c_{k1})^{\frac{1}{2}}, \quad x_{k2} = (c_{k1})^{\frac{1}{2}}(1 - c_{k2}) \quad \text{and} \quad x_{k3} = (c_{k1})^{\frac{1}{2}} c_{k2}$$

   for $k = 1, ..., n$. Then, a matrix multiplier is required to project the 3 dimensional points ($\mathbf{x}_k = (x_{k1}, x_{k2}, x_{k3})$) into a 2 dimensional equilateral triangle that preserves relative locations of points. The projected design points in 2 dimensions are used to calculate the discrepancy measure that will be discussed later. The multiplier for the 3 components mixture experiment is the transpose of the matrix
   $A = \begin{pmatrix} \frac{1}{2} & 0 & 1 \\ \sqrt{\frac{3}{4}} & 0 & 0 \end{pmatrix}$. The projection of S^3 into the 2 dimensions will be denoted by S_p^2. That is, $\mathbf{y}_k = \mathbf{x}_k A'$ yields the uniformly scattered design points on S_p^2. Note that the columns of the matrix A are the vertices for the simplex (equilateral triangle) in Figure 6.1 projected into 2 dimension. The projected design points ($\mathbf{y}_k$) for the non-restricted (full) simplex region (S^3) are used to compute the new proposed discrepancy measure. In this dissertation (as was

mentioned above) the G-mapping function that will be discussed in step 4 can be used by taking 0 and 1 as the lower and upper limits respectively for each component.

2. Suppose that there are lower and upper limit restrictions on the mixture components (restricted simplex region). That is, there exist a_i and b_i such that

$$0 \leq a_i \leq x_i \leq b_i \leq 1$$

for $i = 1, 2, 3$. The simplex for a 3 components mixture experiment with restrictions is denoted by $S^3_{a,b}$, where $a = (a_1, a_2, a_3)$ and $b = (b_1, b_2, b_3)$. The experimental region for $S^3_{a,b}$ defined by the restrictions is non-empty if and only if $\sum_{i=1}^{3} a_i \leq 1 \leq \sum_{i=1}^{3} b_i$. Thus, if this condition is not met then the set $S^3_{a,b}$ is empty (Pham, 2006).

The following steps are used to generate design points in $S^3_{a,b}$:

(i) Given all the constraints imposed, some of the restrictions may be redundant. Hence, to get rid of the redundancies in the constraints, replace $a_i \leq x_i \leq b_i$ with the constraint $a_{i0} \leq x_i \leq b_{i0}$, where $a_{i0} = \max(a_i, b_i + 1 - b_0)$ and $b_{i0} = \min(b_i, a_i + 1 - a_0)$ for $i = 1, 2, 3$ and a_0 and b_0 are the sum of the lower and upper limits of the mixture component proportions. That is, $a_0 = a_1 + a_2 + a_3$ and $b_0 = b_1 + b_2 + b_3$. The restricted simplex region with a_{i0} and b_{i0} as the lower and upper limits respectively is denoted by $S^3_{\mathbf{A},\mathbf{B}}$, where

$$\mathbf{A} = (a_{10}, a_{20}, a_{30}) \quad \text{and} \quad \mathbf{B} = (b_{10}, b_{20}, b_{30}) \tag{6.1}$$

(ii) Now the design points $\mathbf{x}_k = (x_{k1}, x_{k2}, x_{k3})$ $k = 1, ..., n$ in the non-restricted (full simplex) S^3 are transformed into $\mathbf{u}_k$ in C^2 using the inverse transformation. That is, if the design point $\mathbf{x}_k = (x_{k1}, x_{k2}, x_{k3})$ is in S^3, then the design point in C^2 is $\mathbf{u}_k = (u_{k2}, u_{k3})$ with the coordinates of u_{k2} and u_{k3} given by:

$$u_{k2} = \left[\frac{x_{k2}}{(1 - x_{k3})}\right] \quad \text{and} \quad u_{k3} = 1 - [1 - x_{k3}]^2.$$

The vector $\mathbf{u}_k$ is described as a random point in Fang and Yang (2000) and as a result is a bit confusing. However, Prescott (2008) clarified how u_{k2} and u_{k3} are obtained as above. Even though both $\mathbf{u}_k$ and $\mathbf{c}_k$ are called design points in C^2, it is worth mentioning that $\mathbf{u}_k \neq \mathbf{c}_k$. Pham (2006) used $\mathbf{c}_k$ in the G-function for a restricted simplex region. This step is a bit tricky and is stated in Fang and Yang (2000) though they did not make it clear how $\mathbf{u}$ is obtained. As stated above from Prescott (2008), $\mathbf{u}_k$ has to be used in the G-function.

(iii) The G function defined by Fang and Yang (2000) and also used by Borkowski and Piepel (2009) is given by

$$y_i = G(u_i, d_i, \phi_i, \Delta_i, q-1) = \Delta_i \left(1 - [u_i(1-\phi_i)^{q-1} + (1-u_i)(1-d_i)^{q-1}]^{\frac{1}{q-1}}\right) \tag{6.2}$$

for $i = q, ..., 2$, where Δ_i, ϕ_i, and d_i, for $i = 2, 3$ are defined below.

Because the component proportions must sum to 1 (100%), by definition, $y_1 = 1 - \sum_{i=2}^{q} y_i$. The G function can be used to map the design points from C^2 to a non-restricted simplex region (S^3) or to a restricted irregular-shaped polyhedral region ($S^3_{\mathbf{A},\mathbf{B}}$). For S^3, the lower and upper limits are 0 and 1, respectively, for each component and, the steps are the same. The following illustrates how the G function works for a 3 component mixture. Now the design point defined by $\mathbf{u} = (u_2, u_3)$ is used and

$$\Delta_3 = 1, \quad d_3 = \max\left(\frac{a_{30}}{\Delta_3}, 1 - \frac{\sum_{i=1}^{2} b_{i0}}{\Delta_3}\right), \quad \text{and} \quad \phi_3 = \min\left(\frac{b_{30}}{\Delta_3}, 1 - \frac{\sum_{i=1}^{2} a_{i0}}{\Delta_3}\right)$$

and then substitute Δ_3, d_3, ϕ_3 and u_3 in the G function to find

$$y_3 = G(u_3, d_3, \phi_3, \Delta_3, 2)$$

in ($S^3_{\mathbf{A},\mathbf{B}}$). Similarly for S^3, $\Delta_3 = 1$, $d_3 = \max(0, 1)$ and $\phi_3 = 1$ are used in G function to find y_3.

Next, we need to find y_2 with the condition that y_3 is given. That is, find $y_2|y_3$ by first calculating

$$\Delta_2 = \Delta_3 - y_3, \quad d_2 = \max\left(\frac{a_{20}}{\Delta_2}, 1 - \frac{\sum_{i=1}^{1} b_{i0}}{\Delta_2}\right),$$

and

$$\phi_2 = \min\left(\frac{b_{20}}{\Delta_2}, 1 - \frac{\sum_{i=1}^{1} a_{i0}}{\Delta_2}\right)$$

and then

$$y_2 = G(u_2, d_2, \phi_2, \Delta_2, 1).$$

For S^3, use $\Delta_2 = \Delta_3 - y_3$, $d_2 = \max\left(0, 1 - \frac{1}{\Delta_2}\right)$ and $\phi_2 = \min\left(\frac{1}{\Delta_2}, 1\right)$ to find $y_2 = G(u_2, d_2, \phi_2, \Delta_2, 1)$.

Finally, given y_3 and y_2 then find y_1. This is done using the fact that the sum of the components need to sum to 100%. That is

$$y_1 = 1 - \sum_{i=2}^{3} y_i$$

The idea is simply based on using conditional distributions. That is, first obtain y_3 from the G function. Then given y_3, find y_2. Finally given y_2 and y_3, find y_1. Detailed derivation and discussion of the G function and the transformation is found in Fang and Yang (2000).

(iv) The design point in the restricted simplex region is then given by $\mathbf{y} = (y_1, y_2, y_3)$. For $k = 1, ..., n$, the design points are given by $\mathbf{y}_k = (y_{k1}, y_{k2}, y_{k3})$ then $\mathbf{Y}_k = \mathbf{y}_k * A'$, is a uniformly scattered projected design point on the restricted region denoted by S^2_{pc} projected into 2 dimensions. Note that A' is the matrix multiplier stated in step 1.

A numerical example and illustration of the method for $n = 21$ design points obtained from the glp method for the SCC mixture fruit juice example discussed earlier is in Figure 6.3. Figure 6.3 shows the different design candidates in C^{3-1}, non-restricted simplex region (S^3) and restricted simplex region $S^3_{\mathbf{A},\mathbf{B}}$. The figure shows that both S^3 and $S^3_{\mathbf{A},\mathbf{B}}$ preserves the structure of the design patterns from (C^{3-1}) but the systematic pattern of the uniformity spacing is not strictly preserved. Note also that the straight-line scatter of design points in C^{3-1} becomes curvilinear in the full simplex (S^3) and in $S^3_{\mathbf{A},\mathbf{B}}$. The degree of uniformity of the design points for each designs varies from experimental region to experimental region and thus there is a need to find the numerical measure of uniformity in each experimental region. Hence, a criterion has to be developed to measure uniformity in each experimental region. This is because what is best in C^{3-1} may not necessarily be best in S^3 or $S^3_{\mathbf{A},\mathbf{B}}$ or that the best design in S^3 may not be best in $S^3_{\mathbf{A},\mathbf{B}}$.

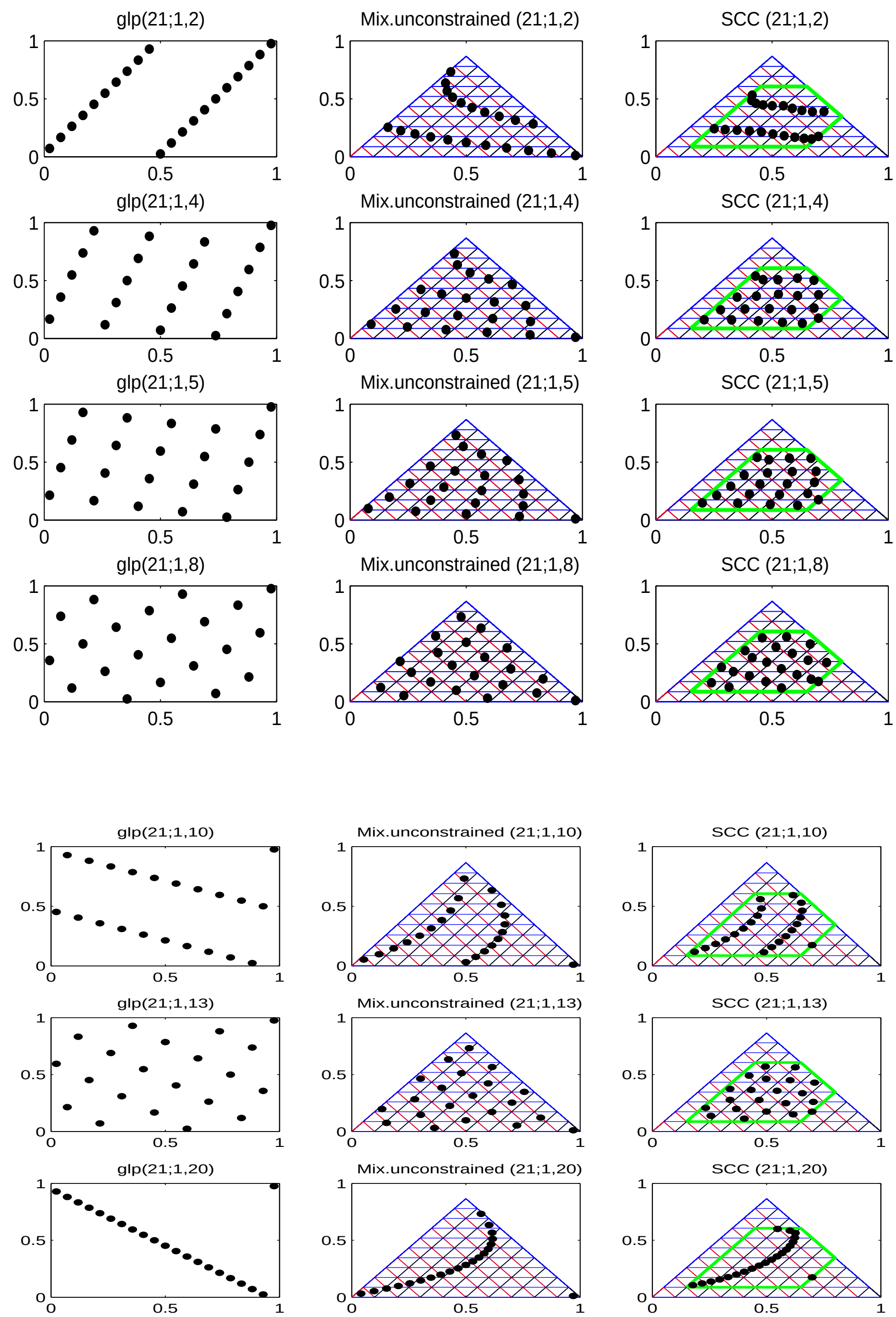
New Measures Of Uniformity For Mixture Experiments

In this section, the distance-based criteria proposed by Borkowski and Piepel (2009) will be used and two new discrepancy measures of uniformity will be proposed.

Distance-Based Criteria

As was discussed in the previous section for C^s, the three distance-based evaluation criteria *rmsd*, *ad* and *md* will be used. However, for mixture designs the minimum distance between a point $\mathbf{x}$ and matrix $\mathbf{X}$ in C^s will be replaced by the following

Figure 6.3: SCC uniform mixture design from glp method for$n = 21$ design points



scaled minimum distance proposed by Borkowski and Piepel (2009). That is , for any point $\mathbf{x}$ in S^3 or $S^3_{a,b}$ and matrix $\mathbf{X}$, the distance between a point $\mathbf{x}$ and matrix $\mathbf{X}$ is defined as

$$d(\mathbf{x}, \mathbf{X}) = \min_j d\left(\mathbf{x}(\mathbf{B}-\mathbf{A}), \mathbf{x}_j(\mathbf{B}-\mathbf{A})\right),$$

for $j = 1, 2, ..., N$ where $\mathbf{A}$ and $\mathbf{B}$ are vectors of the lower and upper component constraints as defined in equation (6.1), $\mathbf{x}(\mathbf{B}-\mathbf{A})$ and $\mathbf{x}_j(\mathbf{B}-\mathbf{A})$ denote vectors of scaled mixture proportions. Specifically, $\mathbf{x}(\mathbf{B}-\mathbf{A}) = \left(\frac{\mathbf{x}_1}{b_1-a_1}, \frac{\mathbf{x}_2}{b_2-a_1}, \ldots, \frac{\mathbf{x}_q}{b_q-a_q}\right)$. The same scaling applies to $\mathbf{x}_j$ to produce $\mathbf{x}_j(\mathbf{B}-\mathbf{A})$. The reason behind this scaling is to ensure that each component is treated as equally important. That is, each component contributes equally towards the computed distance measure. For example, a component with a very small range in its limits may contribute very little towards the computation of the Euclidean distance. Thus, in this study the three distance-based evaluation criteria mentioned in Chapter 3 with this scaled distance criterion will be used to assess uniformity of scatter for the design points for the restricted simplex region $S^3_{\mathbf{A},\mathbf{B}}$. Note that the three distance-based evaluation criteria are larger in $S^3_{\mathbf{A},\mathbf{B}}$ than in S^3 as shown in Table 6.1 and this is due to the scaling factor used.

Discrepancy Criteria

In this study, two new discrepancy criteria for measuring uniformity in a 3 component mixture experiment are proposed: directional and systematic discrepancy (DSD) and non-directional random discrepancy (NDRD). When the mixture design does not have any SCC or MCC constraints, then the mixture design region is the full simplex.

Directional and Systematic Discrepancy (DSD): The first approach is shown geometrically in Figure 6.4 (a) and more clearly presented in Figure 6.5. That is, for each vertex, form an equilateral triangle with side length (2γ) where $(0 \leq \gamma \leq 0.5)$ that contain a vertex from which the direction of the discrepancy is computed. In

Figure 6.4: Geometrical illustration for Non-restricted mixture discrepancy

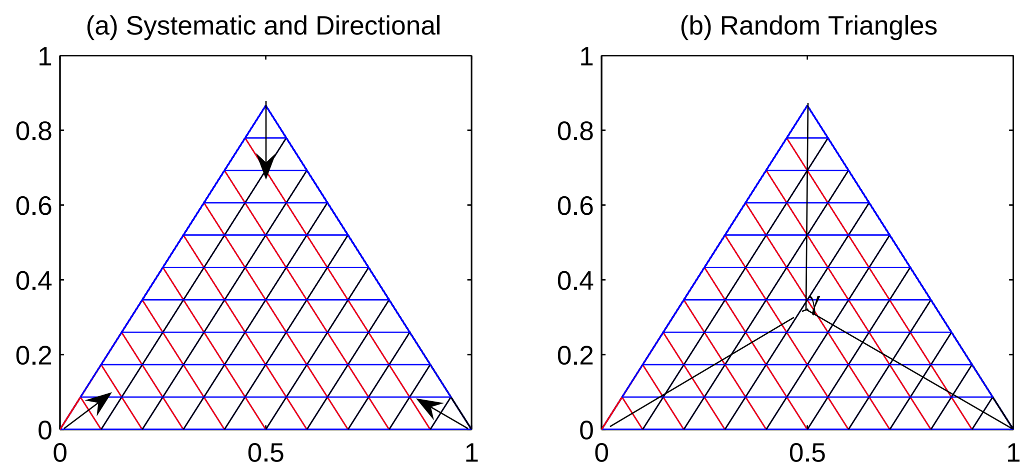


Figure 6.5: Geometrical illustration of the DSD for Non-restricted mixture discrepancy

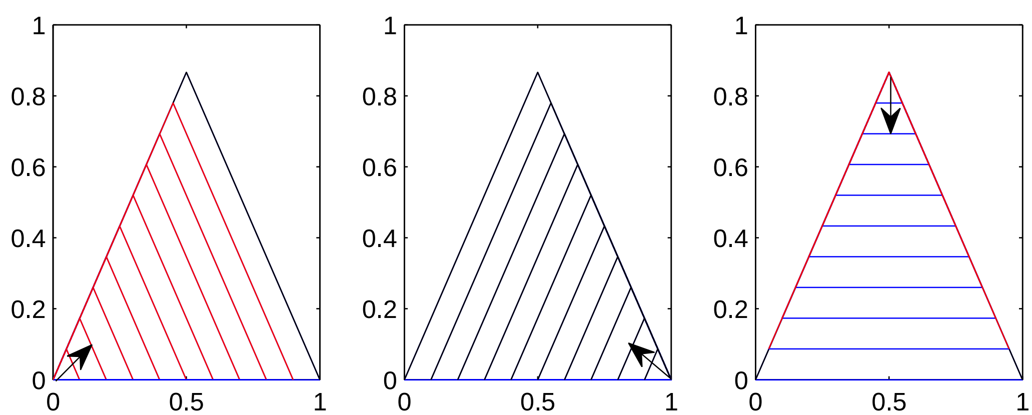


Figure 6.5, in a 3 dimensional setting, move from (0,1,0) to (0.5,0,0.5) along the vector $(\gamma, 1-2\gamma, \gamma)$ which is the midpoint of the equilateral triangle opposite the vertex, then move from (1,0,0) to (0,0.5,0.5) along the vector $(1-2\gamma, \gamma, \gamma)$, and finally move from (0,0,1) to (0.5,0.5,0) along the vector $(\gamma, \gamma, 1-2\gamma)$. In other words it means

to systematically progress from one vertex so that each of the created triangles are equilateral triangles. First compute a discrepancy in the j^{th} direction:

$$D(n, \mathcal{P}, V_j) = \sup_{0 \leq \gamma \leq 0.5} \left| \frac{N(\gamma, \mathcal{P})}{n} - (2\gamma)^2 \right|, \tag{6.3}$$

where V_j is the jth starting vertex. Then the new DSD criterion is

$$DSD = \max_j D(n, \mathcal{P}, V_j).$$

Non-Directional Random Discrepancy (NDRD) in the Full Simplex: The second new approach is shown geometrically in Figure 6.4(b). This is analogous to the modified star discrepancy (MSTRD) in the C^s case. That is, the experimental region is first divided into three triangles formed by the random evaluation point (γ) as shown in Figure 6.4 (b). Then a star discrepancy is computed for each triangle and then the supremum (maximum) discrepancy across the 3 triangles is used as the measure of uniformity. Mathematically, the discrepancy

$$D(n, \mathcal{P}, \gamma) = \max_{j=1,2,3} \left| \frac{N(\gamma, \mathcal{P})}{n} - \frac{4\text{Area}(T_j)}{\sqrt{3}} \right|, \tag{6.4}$$

is computed for each triangle where T_j is the jth triangle and Area(T_j) is the area of triangle j. Then the new NDRD discrepancy criterion is

$$NDRD = \sup D(n, \mathcal{P}, \gamma).$$

The second (NDRD) discrepancy is also used to compute the discrepancy for designs in a restricted simplex region ($S^3_{\mathbf{A},\mathbf{B}}$) which is an irregularly-shaped polygon. Thus, the random evaluation point (γ) will divide the polygon into t triangles where t is the number of sides of the polygon. For the fruit juice example shown in Figure 6.3, the restricted simplex region is an irregularly-shaped pentagon and, hence 5 triangles are formed by any point (γ) in the interior of $S^3_{\mathbf{A},\mathbf{B}}$. The discrepancy is measured for each triangle, and then the maximum over the t triangles is used as the NDRD. It is not computationally easy to compute the NDRD for a restricted simplex region, especially for higher dimensions. An alternative approach in higher dimensions is to use the distance-based evaluation criteria used by Borkowski and Piepel (2009).

The measures of uniformity for $n = 21$ mixture design points using the glp method for different experimental regions (C^{3-1}, S^3, $S^3_{\mathbf{A},\mathbf{B}}$) are presented in Table 6.1. Note that 4000 evaluation points have been used to estimate the measures of uniformity for both the distance-based and the new DSD and NDRD criteria. For C^{3-1} the 6^{th} and 7^{th} columns are the star and modified star discrepancy (STRD and MSTRD respectively), and for S^3 these columns contain the proposed DSD and NDRD measures. For $S^3_{\mathbf{A},\mathbf{B}}$ the DSD criterion is not applicable so we have only non-directional random discrepancy (NDRD). The bold faced figures in Table 6.1 are the smallest evaluation criteria values and, hence, the best. Thus, for the distance-based *rmsd* and *ad* criteria, the NTMDs points obtained by the generator (21;1,13) are the best in all experimental regions as shown in Table 6.1, whereas for the *md* criterion, different generators were selected in different experimental regions. Comparison of the NDRD discrepancy criteria in $S^3_{\mathbf{A},\mathbf{B}}$ and S_3 shows the criterion picks the NTMDs obtained by the generator (21;1,13) and (21;1,8) as the best, respectively. This is a justification as to why we need to compute discrepancies in each experimental region. Overall, NT-net (21;1,13) is the best NTMDs generator according to the distance-based criteria (*rmsd* and *ad*) in all experimental regions. Based on STRD and DSD, (21;1,13) is the best generator in C^{3-1} and S^3 and for the NDRD criterion, (21;1,13) is the best design generator in $S^3_{\mathbf{A},\mathbf{B}}$.

As was seen from Figure 6.3 the structure of the design patterns from (C^{3-1}) are preserved in both S^3 and $S^3_{\mathbf{A},\mathbf{B}}$ although the systematic uniformity of spacing is not strictly preserved. Hence, what is best in C^{3-1} may not necessarily be best in S^3 or $S^3_{\mathbf{A},\mathbf{B}}$. Therefore, as was done in B^2, the non-equivalent best generators from

Measures of Uniformity						
Exp. Region	Generator	*rmsd*	*ad*	*md*	STRD	MSTRD
C^{3-1}	(21;1,2)	0.1641	0.1391	0.4533	0.1512	0.1567
	(21;1,4)	0.0996	0.0912	0.2618	0.1090	0.1090
	(21;1,5)	0.0975	0.0897	0.2219	0.0922	0.0922
	(21;1,8)	0.0972	0.0899	0.2411	0.0925	0.0925
	(21;1,10)	0.1471	0.1278	0.4026	0.1286	0.1409
	(21;1,13)	**0.0950**	**0.0881**	**0.2097**	**0.0922**	**0.0922**
	(21;1,20)	0.2451	0.2056	0.6539	0.2453	0.2453
Exp. Region	Generator	*rmsd*	*ad*	*md*	DSD	NDRD
S^3	(21;1,2)	0.1426	0.1218	0.4229	0.4971	0.2939
	(21;1,4)	0.0950	0.0871	**0.2111**	0.5067	0.2037
	(21;1,5)	0.0948	0.0868	0.2228	0.4971	0.2167
	(21;1,8)	0.0916	0.0845	0.2492	0.5067	**0.1895**
	(21;1,10)	0.1302	0.1130	0.3168	0.5543	0.3044
	(21; 1,13)	**0.0904**	**0.0836**	0.2346	**0.4590**	0.2093
	(21;1,20)	0.2056	0.1737	0.4330	0.5543	0.4981
Exp. Region	Generator	*rmsd*	*ad*	*md*	DSD	NDRD
$S^3_{\mathbf{A},\mathbf{B}}$	(21;1,2)	0.1698	0.1478	0.3932		0.2281
	(21;1,4)	0.1085	0.0998	0.2622		0.1540
	(21;1,5)	0.1066	0.0981	0.2626		0.1840
	(21;1,8)	0.1071	0.0989	**0.2364**		0.1515
	(21;1,10)	0.1457	0.1275	0.3891		0.2105
	(21;1,13)	**0.1057**	**0.0973**	0.2705		**0.1438**
	(21;1,20)	0.2245	0.1923	0.5103		0.3926

Table 6.1: Measures Of uniformity using glp method in different experimental regions

Table 6.1 for both S^3 and $S^3_{\mathbf{A},\mathbf{B}}$ are evaluated and compared with their corresponding equivalent generators. In this example, what happens in S^3 with the non-equivalent generators also happens to be the best, and hence nothing changes. In $S^3_{\mathbf{A},\mathbf{B}}$, the generators selected by the three distance-based criteria did not change either. However, the generator (1,4) in S^3 is equivalent to (1,16) and using the NDRD criterion the generator (1,16) is found to be the best with a NDRD value 0.1372 which is lower value than the NDRD value for (1,13) obtained in Table 6.1. Thus, note that the tables of uniform mixture designs in S^3 for 3, 4 and 5 component mixtures presented in Appendix C were obtained this way.

Construction Of MCC Mixture Designs

The G-mapping function defined in Equation (6.2) only maps SCC mixture design points from C^{q-1} to a non-restricted simplex region (S^q) and to a restricted irregularly-shaped polyhedra region ($S^q_{\mathbf{A},\mathbf{B}}$). Thus, finding a function that maps mixture design points into the space formed by MCCs is an open problem. As a result generating number theoretic mixture designs (NTMDs) with MCCs is not easy. Borkowski and Piepel (2009) proposed two approaches for highly constrained mixture experiments with MCCs. These are the one-pass exchange algorithm and the power modulo a prime (PMP) method. For example, if one wants to construct a MCC mixture experiment of say size n in S^3, the one-pass exchange algorithm can be summarized in the following four steps.

1. For integers n^* such that $n \leq n^* \leq n^*_{max}$, use the relative primes of each n^* to find the generators defined by $(n^*; h^*)$, where $h^* = (1, h_2,\ h_3,\ \ldots,\ h_{q-2})$ and $h_i's$ are positive integers such that the $gcd(n^*, h_i) = 1$, for $i = 2, 3, \ldots, q-2$. The choice of the integer n^*_{max} as an upper limit depends on the volume of the region defined by SCCs and MCCs. That is, the size of n^*_{max} is directly related to n and the number of MCCs.

2. Use the generators in step 1 to generate n^* points in C^{q-1} and then map the design points using the G-function to generate NTMDs in SCCs.

3. Retain only those designs from step 2 that have exactly n points that satisfy all the MCCs. The remaining $n^* - n$ points are discarded.

4. Repeat the process over the set of n^* values up to n^*_{max}.

Note that even if one tries values of $n^* > n^*_{max}$ at some integer N^*_{max}, no new exact n point designs will be generated for every $n^* > n^*_{max}$. That is, there will always be a finite number of generators. In Figure 6.3, the SCC NTMD plots were presented. To illustrate the one-pass exchange algorithm method reconsider the 3 component MCC mixture experiment and the goal is to generate $n = 21$ design points. Suppose the constraint $0.1 \leq x_1 + x_2 \leq 0.8$ is added turning the SCC mixture experiment into a MCC mixture experiment. The first step is to choose n^* such that $n \leq n^* \leq n^*_{max}$. Integers $n^* = 25, ..., 84$ are used and the generators for each n^* using the glp method are found. For example for $n^* = 32$, the generators are

$$(32;1,3) \quad (32;1,5) \quad (32;1,7) \quad (32;1,9) \quad (32;1,11) \quad (32;1,13) \quad (32;1,15)$$
$$(32;1,17) \quad (32;1,19) \quad (32;1,21) \quad (32;1,23) \quad (32;1,25) \quad (32;1,27) \quad (32;1,29)$$
$$(32;1,31).$$

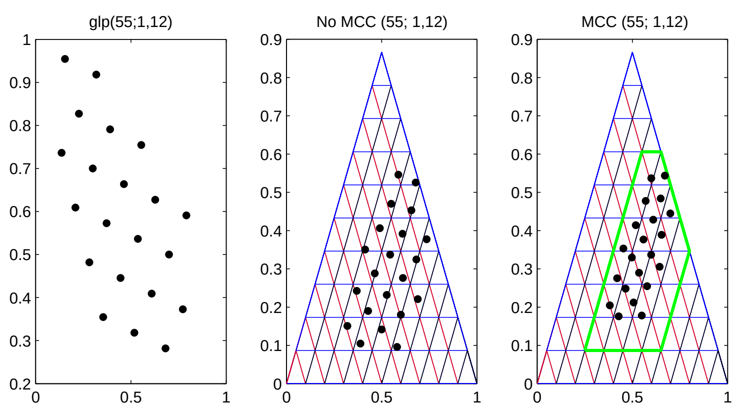
However, because some of the generators are equal when the uniformity is measured in C^2, the task reduction method proposed in this study, reduces the original set to the following 9 generators for $n^* = 32$:

$$(32;1,3) \quad (32;1,5) \quad (32;1,7) \quad (32;1,9) \quad (32;1,15)$$
$$(32;1,17) \quad (32;1,19) \quad (32;1,21) \quad \text{and} \quad (32;1,31).$$

The second step is to use these generators for $n^* = 32$ to generate design points in C^{3-1} using the glp method and then use the G-function to transform the points to generate NTMDs in S^3, the design region in the simplex defined by SCCs. The third step is to retain only those designs from step 2 that have exactly 21 design points that meet all the MCCs. The only generator that satisfies the SCC and MCC mixture constraints is (32;1,31) and the other 8 designs are discarded. That is, the other generators produce designs that either have fewer or more than $n = 21$ design points that satisfy all constraints and, hence are discarded.

The MCC design generators of NTMDs, together with their corresponding measures of uniformity, from the one-pass exchange algorithm method are presented in Table 6.2. Table 6.2 shows the 39 candidate design generators that meet all the SCC and MCC constraints and yielded exactly $n = 21$ design points. The bold faced values are those that have minimum measures of uniformity and thus are best. For distance-based criteria (*rmsd* and *ad*), generator (55;1,12) is the best. However, based on the non-directional random discrepancy (NDRD) criterion, generator (49;1,38) is best. This shows different designs are generated using different criteria. Figures 6.6 and 6.7 show the best $n = 21$ designs that meet the MCC mixture experiment constraint based on the constrained distance-based (*rmsd* and *ad*) criteria and the NDRD criterion, respectively. The plots of the mixture designs without MCC in figures 6.6 and 6.7 shows that the design points provide good coverage in the center of the simplex. Visual inspection of the two figures show the design selected based on the NDRD criterion seems to have slightly more uniform coverage than the design selected using the *rmsd* and *ad* criteria given the position of the top point.

Figure 6.6: Best MCC mixture UD based on *rmsd* and *ad* criteria from One-Pass Exchange Algorithm

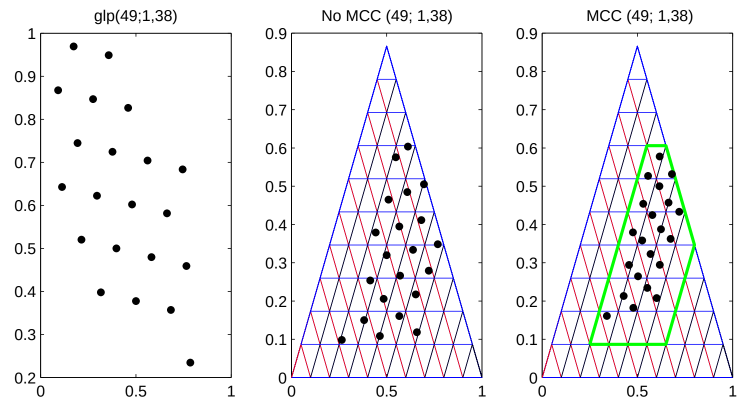


In this dissertation, a new procedure similar to the one-pass exchange algorithm called the one-step exchange algorithm is proposed. In the one-step exchange algorithm instead of retaining only those that have exactly n design points, we also retain those that have $n+1$, $n+2$,..., $1.05n$ design points. And then for each of the retained number of design points (n_p), where $n+1 \leq n_p \leq 1.05n$, randomly sample exactly n design points. Note that $1.05n$ is rounded up or rounded down to the closest integer.

Generator	$rmsd$	ad	md	NDRD
(32;1, 31)	0.1597	0.1338	0.3925	0.5561
(33;1,32)	0.1594	0.1336	0.3917	0.5775
(48; 1,19)	0.0769	0.0696	0.1968	0.2741
(49; 1,18)	0.0714	0.0660	0.1519	0.2699
(49; 1,23)	0.0947	0.0828	0.2374	0.3945
(49; 1,38)	0.0690	0.0639	0.1529	**0.2419**
(51;1,2)	0.1562	0.1330	0.4037	0.3771
(51; 1,19)	0.0683	0.0630	0.1579	0.2813
(51; 1,20)	0.0707	0.0648	0.1699	0.3052
(51;1,38)	0.0828	0.0737	0.1946	0.3761
(52;1,5)	0.0756	0.0687	0.1515	0.3150
(52;1,11)	0.0702	0.0644	0.1579	0.2671
(52;1,23)	0.0742	0.0676	0.1809	0.3275
(52;1,31)	0.0756	0.0686	0.1748	0.2863
(52;1,37)	0.0671	0.0622	0.1698	0.3147
(53;1,3)	0.1134	0.0983	0.2822	0.3318
(53;1,4)	0.0929	0.0824	0.2126	0.3301
(53;1,7)	0.0705	0.0649	0.1510	0.2920
(53;1,11)	0.0714	0.0654	0.1528	0.3126
(53;1,12)	0.0706	0.0646	0.1739	0.2847
(53;1,14)	0.0729	0.0666	0.1516	0.3393
(53;1,17)	0.0997	0.0868	0.2637	0.3099
(53;1,24)	0.0792	0.0712	0.2014	0.3247
(54;1,5)	0.0778	0.0702	0.2049	0.3118
(54;1,7)	0.0675	0.0623	**0.1455**	0.2926
(54;1,23)	0.0684	0.0630	0.1573	0.3965
(55;1,3)	0.1122	0.0976	0.2801	0.3250
(55;1,12)	**0.0663**	**0.0614**	0.1502	0.3217
(55;1,16)	0.0689	0.0633	0.1961	0.3416
(55;1,19)	0.0994	0.0862	0.3242	0.3385
(55;1,24)	0.0727	0.0656	0.2045	0.3272
(55;1,38)	0.0731	0.0669	0.1499	0.3529
(57;1,4)	0.0966	0.0848	0.2499	0.3229
(57;1,10)	0.0692	0.0632	0.2093	0.3761
(57;1,16)	0.0701	0.0641	0.1773	0.3652
(57;1,28)	0.2230	0.1798	0.4952	0.6472
(58;1,11)	0.0735	0.0665	0.1738	0.2884
(59;1,4)	0.0933	0.0825	0.2103	0.3497
(59;1,33)	0.0669	0.0618	0.1476	0.3765

Table 6.2: MCC mixture UDs of size $n = 21$ from the One-Pass Exchange Algorithm

Figure 6.7: Best MCC mixture UD based on NDRD criteria from One-Pass Exchange Algorithm



The randomly selected design points are replaced (interchanged) with each of the non-selected design points one at a time and that is why the method is called one-step exchange algorithm. The idea of considering an additional 5% for the number of design points is to minimize the impact of deviating from a quasi Monte-Carlo method to generate UDs.

This approach will be investigated to determine if the 5% drift from the quasi Monte-Carlo can lead to superior designs using the exchange algorithm. That is, it may allow some regions that are not represented by other designs to be covered. This approach assesses the effect of design augmentation on the uniformity of the design points in the restricted experimental region. The steps of the one-step exchange algorithm are the following:

1. For integers n^* such that $n \leq n^* \leq n^*_{max}$, use relative primes relative to each n^* to find the generators defined by $(n^*; h^*)$, where $h^* = (1,\ h_2,\ h_3,\ \ldots,\ h_{q-2})$ and $h_i's$ are positive integers such that the $gcd(n^*, h_i) = 1$, for $i = 2, 3, \ldots, q-2$.

2. Use the generators in step 1 to generate points in C^{q-1} and then map the design points using the G-function to generate NTMDs in SCCs.

3. Retain those designs from step 2 that satisfies all the MCCs and have exactly $n+1$ $n+2$,..., $1.05n$ design points and the rest are discarded.

4. Randomly select n design points from the $n+1$ points. Do the same for $n+2$,...,$1.05n$.

5. Store the randomly selected n design points as candidate design points from each of the retained number of design points in step 3 and also find the remaining candidate designs from the randomly selected designs by exchanging each of the design points by the non-selected design points one at a time.

6. Measure the uniformity of each of the candidate designs and select the design with low discrepancy of design points.

7. Repeat the process over the set of n^* values up to n^*_{max}.

8. Finally the design with the minimum measure of uniformity is selected as the best design from all the candidate designs considered.

To illustrate the one-step exchange algorithm with an example, the idea is to generate NTMDs $n = 21$ design points from MCC mixture experiment. The first and second steps are the same as in the one-pass exchange algorithm but here we retained only those that give exactly 22 design points because approximately the $5\%n$ is just 1. The third step is to randomly select $n = 21$ design points from the 22. The fourth step is to store the randomly selected $n = 21$ design points as one candidate design to be considered and later the exchange algorithm is applied to find the remaining candidate designs by replacing each of the randomly selected $n = 21$ design points with the non-selected design point one at a time. In other words, for this particular example since $5\%n$ is just 1, this is equivalent to saying generate 21 design points from the 22 possible design points by deleting one point at a time. That means for this example, the generator that generates exactly $n = 22$ design points will have $n = 21$ different designs that are obtained by exchanging each of the 21 selected design points with the non-selected design point plus we have the $22nd$ design which is the one that was randomly selected. Thus, for $n^* = 25, ..., 84$ we have found only 40 generators that meet all the MCCs as shown in column one of Table 6.3. The minimum measures of uniformity are reported for each of the 40 generators and are presented in detail in Table 6.3. Note that in Table 6.3, the integers following "/" indicates the point to be removed from the 22 possible design points. For example, the first row shows the generator (34;1,33) and the minimum measures of uniformity are reported for each criteria and this occurs when no point is deleted from the randomly selected 21 design points. The 3rd row generator from Table 6.3 is (49;1,39) and the measures of uniformity reported are minimum when the 17th point of the original randomly selected design point is deleted and replaced by the non-selected design point according *rmsd* and *ad* criteria but *md* is minimum when the 3rd point is replaced by the non-selected point and the NDRD is minimum when the 5th design point is replaced with the non-selected design point. As shown in Table 6.3 we have 40 different generators and within each generator we have 22 different design candidates (21 are obtained by exchanging each design point one at a time and the $22nd$ is the one obtained randomly) and thus it means we have in total $40 \times 22 = 880$ design candidates from which the best design is to chose from. For all the indicated integers greater than 1 following "/" in Table 6.3 it means the measure of uniformity is minimum when the indicated number shown minus one is replaced.

The different generators together with their minimum measure of uniformity are presented in Table 6.3. The bold faced values in Table 6.3 shows the measures of uniformity that are minimum and hence best. Using the distance-based criteria, the NTMDs points obtained from the generator (53;1,15) are the best and using the NDRD criteria the generator (52; 1,7) is the best. Moreover, the best NTMDs for $n = 21$ design points from the one-step exchange algorithm based on the distance criteria and the discrepancy (NDRD) criteria are presented in Figure 6.8 and 6.9 respectively. The generator (53;1,15) is chosen as the best because even though generator (58;1,33) is the best according to *rmsd* and *ad* (and are very close), (53; 1,15) is better for the *md* and NDRD as can be seen in Table 6.3. Note that both for the one-pass exchange algorithm and the proposed one-step exchange algorithm, 16000 evaluation points have been used in C^{3-1} of which only around 6000 met both the SCC and MCC constraints. Hence, to find the measures of uniformity for both the distance-based and the proposed discrepancy criteria, the 6000 evaluation points that meet both the SCC and MCC are used. Also note that computation is expensive for the one-step exchange algorithm. For this example, the running time was around 6444 seconds whereas the one-pass exchange algorithm running time is around 237 seconds.

Comparison of the two methods for this particular example shows the results for the two methods are close to each other with the one-pass exchange algorithm being better with respect to the measures of uniformity and running time. This may be due to the fact that we allowed it to drift from the quasi Monte-Carlo approach with 5% of the design points. The proposed method expands the search for better designs but at the expense of being computationally expensive. The important message from this example is that finding the exact number of design points requires restricting the search within the frame of the quasi Monte-Carlo method in the one-pass exchange algorithm proposed by Borkowski and Piepel (2009) whereas the proposed method allows 5% drift from the quasi Monte-Carlo method and checks for the best augmented design.

Figure 6.8: Best MCC mixture UD Based on distance criteria from the One-Step Exchange Algorithm

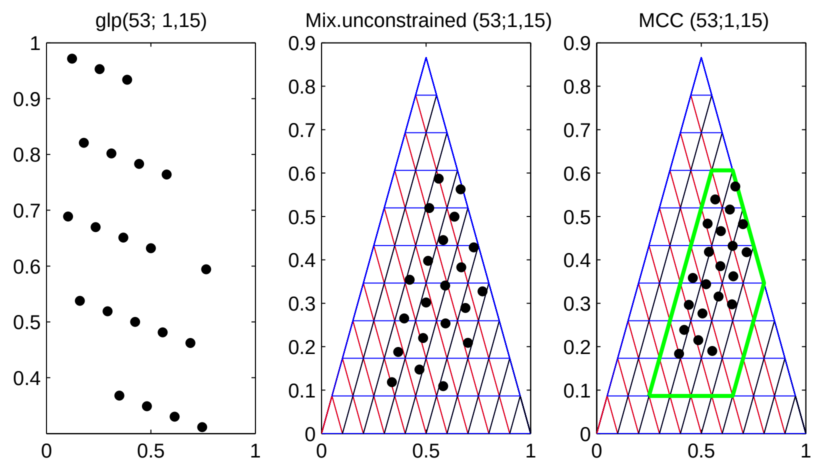
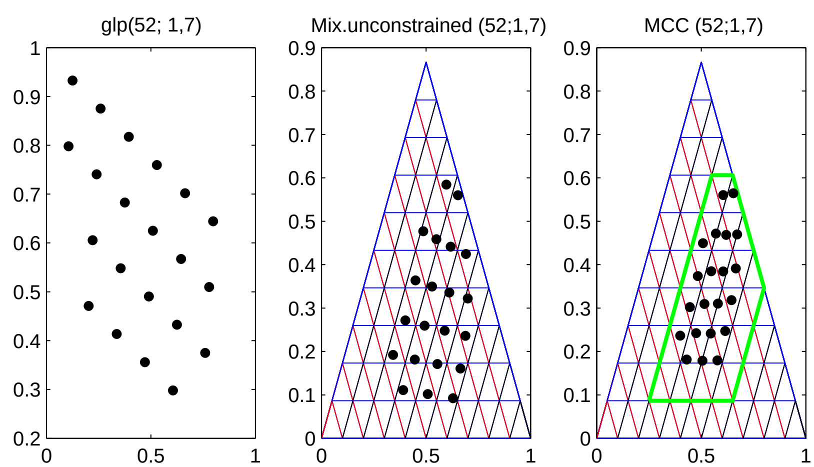


Figure 6.9: Best MCC mixture UD based on NDRD criteria from the One-Step Exchange Algorithm



Generator	$rmsd$	ad	md	NDRD
(34;1,33)	0.1591 / 1	0.1336/ 1	0.3912/ 1	0.5451/ 1
(49;1,2)	0.1558 / 1	0.1327 / 1	0.3994 / 1	0.2865 / 2
(49;1,39)	0.0759 / 18	0.0693 / 18	0.1831 / 4	0.2335 / 6
(50;1,7)	0.0692 / 5	0.0639 / 13	0.1413 / 5	0.2300 / 21
(52;1,3)	0.1139 / 22	0.0987 / 12	0.2839 / 1	0.2924 / 3
(52;1,7)	0.0695 / 18	0.0635 / 18	0.1852 / 1	**0.2098** / 3
(53;1,6)	0.0700 / 5	0.0644 / 14	0.1578 / 1	0.2277 / 2
(53;1,15)	**0.0676** / 8	**0.0623** / 5	**0.1411** / 12	0.3431 / 1
(53;1,21)	0.0762 / 1	0.0690 / 1	0.1873 / 1	0.2338 / 4
(53;1,33)	0.0679 / 9	0.0625 / 9	0.1440 / 8	0.2868 / 2
(53;1,35)	0.1131 / 6	0.0962 / 6	0.3209 / 1	0.4404 / 2
(53;1,37)	0.0697 / 14	0.0644 / 14	0.1527 / 2	0.2849 / 2
(54;1,17)	0.0886 / 16	0.0786 / 16	0.2130 / 1	0.3102 / 3
(54;1,29)	0.0803 / 12	0.0725 / 12	0.2049 / 1	0.2728 / 2
(54;1,43)	0.0768 / 18	0.0691 / 18	0.1866 / 1	0.2528 / 6
(55;1,13)	0.0789 / 20	0.0713 / 20	0.2011 / 1	0.2807 / 1
(55;1,18)	0.1144 / 6	0.0971 / 6	0.3197 / 1	0.4516 / 2
(56;1,3)	0.1146 / 22	0.0992 / 22	0.2798 / 1	0.3157 / 3
(56;1,9)	0.0696 / 7	0.0638 / 7	0.1534 / 2	0.3257 / 2
(56;1,17)	0.0736 / 13	0.0668 / 13	0.1821 / 2	0.2955 / 2
(56;1,31)	0.0710 / 20	0.0650 / 20	0.1569 / 2	0.3344 / 2
(56;1,37)	0.1156 / 6	0.0978 / 6	0.3207 / 1	0.4734 / 1
(57;1,17)	0.0703 / 13	0.0641 / 13	0.2051 / 1	0.2583 / 3
(57;1,31)	0.0732 / 2	0.0666 / 2	0.1730 / 1	0.2939 / 2
(57;1,32)	0.0692 / 6	0.0636 / 6	0.1466 / 3	0.3757/ 3
(58;1,5)	0.0765 / 18	0.0692 / 18	0.1519 / 1	0.3052 / 2
(58;1,21)	0.0703 / 18	0.0644 / 18	0.1583 / 2	0.3393 / 2
(58;1,33)	**0.0675** / 3	**0.0623** / 7	0.1739 / 2	0.3933 / 2
(58;1,45)	0.0680 / 11	0.0628 / 18	0.1512 / 4	0.2861 / 2
(59;1,5)	0.0764 / 8	0.0690 / 8	0.1764 / 1	0.3729 / 2
(59;1,11)	0.0692 / 2	0.0632 / 2	0.1578 / 2	0.3733 / 2
(59;1,25)	0.0696 / 13	0.0638 / 13	0.1616 / 2	0.3846 / 2
(59;1,29)	0.2229 / 20	0.1798 / 1	0.4916 / 1	0.6409 / 2
(61;1,4)	0.0960 / 5	0.0843 / 12	0.2354 / 1	0.3615 / 2
(61;1,9)	0.0677 / 22	0.0622 / 22	0.1973 / 2	0.2939 / 2
(61;1,21)	0.0997 / 1	0.0862 / 7	0.3255 / 1	0.3035 / 2
(61;1,30)	0.2247 / 1	0.1812 / 22	0.5084 / 1	0.6885 / 1
(62;1,15)	0.0898 / 20	0.0781 / 20	0.2158 / 2	0.4329 / 2
(63;1,10)	0.0689 / 15	0.0633 / 15	0.1888 / 1	0.3938 / 2
(63;1,31)	0.2246 / 7	0.1811 / 22	0.5044 / 1	0.7297 / 1

Table 6.3: MCC mixture UDs of Size $n = 21$ from the One-Step Exchange Algorithm

EXAMPLES OF UD APPLICATIONS AND DATA ANALYSIS

In the previous chapters of this dissertation, different methods of generating uniform designs (UDs) using NTMs were discussed and presented in detail. The good lattice point (glp) method was found to be the best despite being computationally expensive. This chapter will contain a presentation and discussion of the experimental data layout and data analysis for UDs generated by the glp method. This chapter will contain two examples to show the robustness of UDs in data analysis using a simulation study.

Data Layout

Tables of uniform designs generated using the glp method provide the best generators assuming all input factors have the same number of levels. That is, the UD tables do not provide the best design generators when the input factors have different numbers of levels. In this section, we will describe the experimental data layout for UDs generated using the glp method for two cases: when all the input factors have the same number of levels and when they have different numbers of levels.

Consider the case where an experimenter is planning an experiment with 9 runs having two input factors x_1 and x_2. The UD tables obtained from the glp method indicate the NT-net with (9;1,4) as the best design generator. As was discussed previously, one of the advantages of UDs is that they allow a maximum number of n levels to be used for each factor. As a result both x_1 and x_2 can have each 9 equally spaced levels in the range between -1 and 1. The data layout for the $U_9(9^2)$ design is shown in Table 7.1. Note that the order of the experimental runs should be randomized.

Next consider the case where a researcher wants to conduct an experiment with 18 runs having 3 input factors x_1, x_2 and x_3 with 6, 3 and 2 levels, respectively. The

Uniform Design				
No.	$U_9(9^2)$		x_1	x_2
1	1	4	-1.00	-0.25
2	2	8	-0.75	0.75
3	3	3	-0.50	-0.5
4	4	4	-0.25	0.5
5	5	2	0.00	-0.75
6	6	6	0.25	0.25
7	7	1	0.50	-1.00
8	8	5	0.75	0.00
9	9	9	1.00	1.00

Table 7.1: UD data layout for $U_9(9^2)$ design

UD tables give only the NTM generator for $U_{18}(18^3)$ but not for $U_{18}(6\times3\times2)$. What should the data layout of this experiment look like? To answer this question one needs to start with the design obtained from the uniform design table. That is, start with the best generator for $U_{18}(18^3)$ which is (18;1,5,7), and then use the pseudo-level technique described in Fang et al. (2006) to redefine the original levels in each column. The goal is to convert the three columns of this UD having each 18 levels into a mixed design with 6, 3, and 2 factor levels in three columns, respectively. Because the goal for column 1 is to have 6 levels instead of 18 levels, we redefine the original levels as follows: $(1,2,3) \Longrightarrow 1$, $(4,5,6) \Longrightarrow 2$, $(7,8,9) \Longrightarrow 3$, $(10,11,12) \Longrightarrow 4$, $(13,14,15) \Longrightarrow 5$, $(16,17,18) \Longrightarrow 6$. Similarly for column 2, redefine the original levels $(1,2,3,4,5,6) \Longrightarrow 1$, $(7,8,9,10,11,12) \Longrightarrow 2$, and $(13,14,15,16,17,18) \Longrightarrow 3$. Similarly for column 3, redefine the original levels $(1,2,3,4,5,6,7,8,9) \Longrightarrow 1$ and $(10,11,12,13,14,15,16,17,18) \Longrightarrow 2$. The data layout for $U_{18}(6\times3\times2)$ for the mixed levels $6\times3\times2$ is given in Table 7.2 in the columns x_1, x_2 and x_3, respectively.

In the previous example $U_{18}(18^3)$ was used as a starting design for $U_{18}(6\times3\times2)$. Now what if instead the interest is in obtaining a 17 point $U_{17}(6\times3\times2)$ design. Because 17 is not divisible by 6, 3 and 2, the recommendation is to use $U_{18}(18^3)$ in Table 7.2 as a starting design and then delete the last row. This approach is recommended because the levels of each variable in the last row of any uniform design

Uniform Design						
No.	$U_{18}(6 \times 3 \times 2)$			x_1	x_2	x_3
1	1	5	7	1	1	1
2	2	10	14	1	2	2
3	3	15	3	1	3	1
4	4	2	10	2	1	2
5	5	7	17	2	2	2
6	6	12	6	2	2	1
7	7	17	13	3	3	2
8	8	4	2	3	1	1
9	9	9	9	3	2	1
10	10	14	16	4	3	2
11	11	1	5	4	1	1
12	12	6	12	4	1	2
13	13	11	1	5	2	1
14	14	16	8	5	3	1
15	15	3	15	5	1	2
16	16	8	4	6	2	1
17	17	13	11	6	3	2
18	18	18	18	6	3	2

Table 7.2: UD data layout for $U_{18}(6 \times 3 \times 2)$ design

table are identically n, and, in this case, $n = 18$. This point, therefore, is always in the same relative position in C^s. Thus, applying the proposed approach, the data layout for $U_{17}(6 \times 3 \times 2)$ will be the same as Table 7.2 but with the last row omitted.

Now, consider another example of an unusual situation that can occur with UDs. Suppose an experimenter plans to use the glp method to find a uniform design with $n = 6$ and 2 input factors. Based on the glp method, the only candidate NT-net design is generated by (6;1,5), and as shown in Figure 7.1, it is a very poor design. Thus, if the number of candidate designs that can yield uniformly scattered design points is very limited, then the idea is to start with a $U_{n+1}([n+1]^s)$ design and delete the last row to generate the n point design. For the example with $n = 6$ and 2 input factors, a better option than the UD generated by (6;1,5) is to start with $U_7(7^2)$ and delete the last row of the design to form the 6 point uniform design denoted $U_{7-1}(6^2)$ which is shown graphically in Figure 7.1 (b) and is better than the $U_6(6^2)$ design in

Figure 7.1: 6 design points from NTM's glp(6;1,5) and glp(7;1,3) with last row deleted

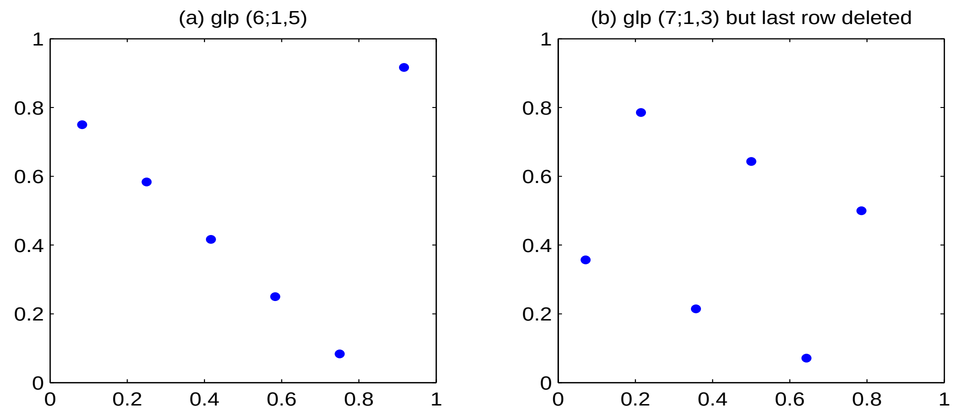


Figure 7.1 (a). Note that the range and the number of levels for each input factor and the number of experimental runs are determined by the experimenter either from experience or based on the availability of resources.

Data Analysis

In this section, a data analysis for UDs using a simulated data set will be presented. Because a significant amount of resources is often wasted due to a poor experimental design, the main focus of this dissertation is on design generation and not on the analysis of experimental data. However, because UDs are not orthogonal, the data analysis is, in general, not straightforward given the linear model coefficients cannot be independently estimated. Hence, this is an area that requires future research. In this dissertation, a stepwise regression analysis using the R statistical package will be considered as a preliminary analysis of the data collected from UDs. Note that the R-statistical package of R Development Core Team (2011) requires a package called MASS of Venables and Ripley (2002) to run a stepwise regression to find best model according to AIC criteria. We will present two examples for simulated data sets.

Example 1: Consider the deterministic model

$$y = \exp(0.5x_1 - 1.5x_2) + 5.$$

From a CCD example contained in Borkowski (2010). The goal is to find an approximating model from the uniform design having 2 input factors each having 9 levels for the purpose of predicting future responses. Table 7.3 contains the uniform design $U_9(9^2)$, which is the same as Table 7.1 but modified with a column of response (y) values corresponding to the deterministic model. As was discussed in Chapter 1,

Uniform Design					
No.	$U_9(9^2)$		x_1	x_2	y
1	1	4	-1.00	-0.25	5.8825
2	2	8	-0.75	0.75	5.2231
3	3	3	-0.50	-0.5	6.6487
4	4	4	-0.25	0.5	5.4169
5	5	2	0.00	-0.75	8.0802
6	6	6	0.25	0.25	5.7788
7	7	1	0.50	-1.00	10.7546
8	8	5	0.75	0.00	6.4550
9	9	9	1.00	1.00	5.3679

Table 7.3: Data layout for $U_9(9^2)$ UD and response values (y)
from the deterministic model

CCDs are good for fitting a second order (quadratic) polynomial regression model. However, UDs, unlike CCDs, can be used to fit higher order models such as cubic polynomial regression models. Thus, for the data in Table 7.3, both backwards and forwards stepwise regression using stepAIC was performed to find the best model according to AIC criteria from a second-order polynomial model plus $x_1x_2^2$ and $x_2x_1^2$ terms using the lm function in the statistical package R. The final recommended model from the stepwise regression is given by

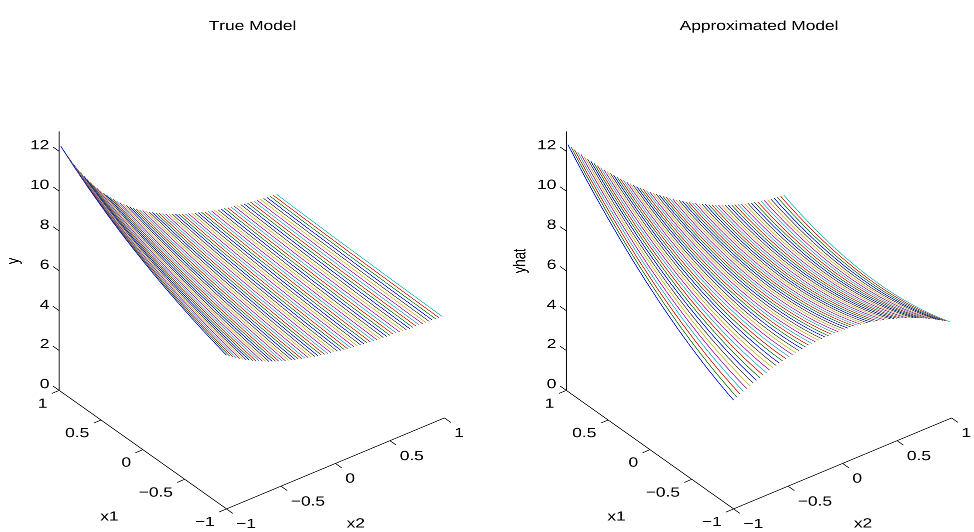
$$y = 6.1941 + 0.3401x_1 - 1.9455x_2 - 1.6218x_1x_2 + 0.8412x_2^2 + 1.5580x_1x_2^2.$$

The residual standard error = 0.04372 and R^2 = 0.9998, and the p-value for the model F-test = 1.147×10^{-5}. The summary for the final recommended model is presented in Table 7.4.

Terms	Estimates	Std. Error	t value	$Pr(>\|t\|)$
Intercept	6.19407	0.02870	215.831	2.19×10^{-7}
x_1	0.34007	0.04025	8.449	0.003480
x_2	-1.94551	0.02493	-78.031	4.64×10^{-6}
x_2^2	0.84124	0.07570	11.113	0.001561
x_1x_2	-1.62178	0.06311	-25.699	0.000129
$x_1x_2^2$	1.55800	0.12096	12.881	0.001010

Table 7.4: Summary of the approximated model for the UD with $n = 9$ design points

Figure 7.2: True and approximated model fit from UD data



The true and approximated models are displayed graphically in Figure 7.2, and demonstrates the robustness of UDs to fit alternative models that cannot be fit using a classical response surface design (e.g. a CCD).

Example 2: This example will provide an application of UDs in accelerated stress testing (AST). AST is a method used by engineers to study the life time (reliability) of products at faster or "accelerated" times than those used under normal usage conditions. Thus, accelerated testing is used to study the life of products when it takes an impractical amount of time to observe failures under normal operating conditions. Engineers use AST to collect life data (i.e., time to failure data) under accelerated stress conditions for factors that cause the product to fail more quickly. That is, the levels of the factors that cause the product to fail under normal conditions are increased so that the product fails more quickly (Pham, 2006; Dodson and Schwab, 2006). For example, if a device operates normally at a temperature of 110^{o}F then, for the product to fail more quickly, higher temperatures are applied (for example, at 130^{o}F, 150^{o}F etc). Dodson and Schwab (2006) have an excellent review of the different AST models. A model in the book by Pham (2006) will be used to show the application of a UD in accelerated testing. Assume a product operates under settings for three factors: voltage V (Volts), temperature T (Kelvin) and relative humidity H(%), and suppose the underlying model for the median time to failure t is given by

$$t = 240V^{-1.04} \exp\left(\frac{1062}{T} - 0.021H\right).$$

The logarithmic transformation of this model is given by

$$\ln t = \ln 240 - 1.04 \ln V + \frac{1062}{T} - 0.021H.$$

The motivation of the problem is how to design an experiment so that you get a model that best approximates the unknown underlying model. In this example an experiment with $n = 21$ runs and 7 levels on each of the factors V, T, and H will be used. Pham (2006) used $n = 8$ experimental runs with 4 equally spaced levels on each of the input factors $\ln V$, $\frac{1}{T}$, and H. The UD tables obtained from the glp method indicate that (21;1,4,10) is the best generator, and thus was used to design the experiment. The data layout together with the observed median time to failure with random error (ε) from a standard normal distribution added is shown in Table 7.5.

No.	$U_{21}(7^3)$			V	T	H	True (t)	$t+\varepsilon$	$\hat{t}$	$\lvert t-\hat{t}\rvert$
1	1	4	10	2.0000	356	92.5	330.4356	331.8060	331.0777	0.6420
2	2	8	20	2.0000	359	100.0	275.3335	276.3156	276.1265	0.7930
3	3	12	9	2.0000	363	90.0	328.7790	329.1092	329.8991	1.1202
4	4	16	19	2.3300	370	100.0	215.1222	215.6398	215.2801	0.1579
5	5	20	8	2.3300	373	90.0	259.3351	259.3635	259.2715	0.0635
6	6	3	18	2.3300	353	97.5	260.3260	259.1340	259.9699	0.3560
7	7	7	7	2.7144	359	90.0	247.2358	247.7669	246.6710	0.5648
8	8	11	7	2.7144	363	97.5	204.4340	202.7049	204.0885	0.3455
9	9	15	6	2.7144	366	87.5	246.2296	245.8973	245.7929	0.4367
10	10	19	16	3.1623	373	97.5	161.2526	159.9727	160.5779	0.6747
11	11	2	5	3.1623	353	87.5	233.7579	233.3295	232.5021	1.2558
12	12	6	15	3.1623	356	95.0	194.6948	194.0003	193.9122	0.7827
13	13	10	4	3.6840	363	87.5	183.5707	184.2904	183.7252	0.1545
14	14	14	14	3.6840	366	95.0	153.1043	153.6162	153.2310	0.1268
15	15	18	3	3.6840	370	85.0	183.0483	182.5066	183.0712	0.0228
16	16	1	13	4.2919	353	95.0	145.3492	145.0044	145.8836	0.5345
17	17	5	2	4.2919	356	85.0	174.8251	175.4374	175.6942	0.8691
18	18	9	12	4.2919	359	92.5	145.6720	147.1866	146.5331	0.8611
19	19	13	1	5.0000	363	85.0	140.8145	141.3336	141.1976	0.3831
20	20	17	11	5.0000	370	92.5	113.8172	112.8804	114.0497	0.2325
21	21	21	21	5.0000	373	100.0	95.0125	96.0535	95.1201	0.1076

Table 7.5: Data layout from the glp(21;1,4,10) based on *rmsd* and *ad* for the AST example

First the second-order polynomial model plus a cubic term was fit using $(t + \varepsilon)$ as the median time to failure. However, the performance of the model is better when $\log(t + \varepsilon)$ is used as the log(median time to failure). Both backwards and forwards stepwise regression using stepAIC in the statistical package R is used to select the best model from a second-order polynomial plus a cubic term with factors, relative humidity (H), temperature (T) and voltage (V). Second those non-significant model terms from the model selected according to the AIC criterion are dropped sequentially starting with the one that has the highest p-value. Finally, after back transforming, the recommended (approximated) model for the true model of the AST example is given by

$$\hat{t} = \exp\left(12.1621 - 0.0210\text{H} - 0.0080\text{T} - 1.0967\text{V} + 0.1821\text{V}^2 - 0.0127\text{V}^3\right).$$

The summary for the final recommended best model is presented in Table 7.6 and the residual standard error = 0.0052 and $R^2 = 0.9998$, and the p-value $\approx 2.2 \times 10^{-16}$ for the model F-test. This model can not be fit using CCDs because CCDs can only fit a second-order polynomial with no cubic term added as discussed in Chapter 1. UDs unlike the CCDs provide flexibility in the number of runs, number of levels and potential models to consider. Thus, this result clearly shows the advantage of UDs to approximate a nonlinear AST model.

terms	Estimates	Std. Error	t value	$Pr(>\|t\|)$
Intercept	12.1620592	0.0780252	155.873	$< 2 \times 10^{-16}$
H	-0.0209956	0.0002411	-87.100	$< 2 \times 10^{-16}$
T	-0.0080085	0.0001847	-43.360	$< 2 \times 10^{-16}$
V	-1.0966720	0.0628535	-17.448	$< 2 \times 10^{-11}$
V^2	0.1820758	0.0188073	9.681	7.64×10^{-8}
V^3	-0.0127105	0.0017967	-7.074	3.78×10^{-6}

Table 7.6: Summary of the approximated model for the glp(21;1,4,10) UD for the AST example

An alternative approach would be to fit a nonlinear model without having to transform the response t.

Next, reconsider the true model in equation (1.1) given by

$$Y(\mathbf{x}) = h(\mathbf{x}) + \varepsilon. \tag{7.1}$$

Then, the true model is given in equation (7.1), which can also be written as

$$Y(\mathbf{x}) = \beta' g(\mathbf{x}) + f(\mathbf{x}) + \varepsilon \tag{7.2}$$

where $\beta' g(\mathbf{x})$ is an approximated linear model and $f(\mathbf{x})$ is the systematic error (i.e, the bias) as explained in Chapter 1. Thus, $f(\mathbf{x}) = h(\mathbf{x}) - \beta' g(\mathbf{x})$ which is the difference between the true and approximating models. It is also worth remembering that there is also a random error in the model, and, as it is explained in Borkowski (2010), the systematic error cannot be avoided but is assumed to be negligible. Thus, bias is often ignored putting emphasis on the random errors. It was also explained in Chapter 1 that UDs minimize the maximum bias under certain conditions (Wiens, 1991).

No.	t	$\hat{t}$	$\|t - \hat{t}\|$
1	330.4356	331.0777	0.6420
2	275.3335	276.1265	0.7930
3	328.7790	329.8991	1.1202
4	215.1222	215.2801	0.1579
5	259.3351	259.2715	0.0635
6	260.3260	259.9699	0.3560
7	247.2358	246.6710	0.5648
8	204.4340	204.0885	0.3455
9	246.2296	245.7929	0.4367
10	161.2526	160.5779	0.6747
11	233.7579	232.5021	1.2558
12	194.6948	193.9122	0.7827
13	183.5707	183.7252	0.1545
14	153.1043	153.2310	0.1268
15	183.0483	183.0712	0.0228
16	145.3492	145.8836	0.5345
17	174.8251	175.6942	0.8691
18	145.6720	146.5331	0.8611
19	140.8145	141.1976	0.3831
20	113.8172	114.0497	0.2325
21	95.0125	95.1201	0.1076

Table 7.7: The true (t), approximated ($\hat{t}$) and true-approximated deviations $|t - \hat{t}|$ for the AST example

.

The goal is to have an approximation model that minimizes the systematic error or bias. The absolute difference for the median time (hours) to failure between the true and approximated model is presented in Table 7.7. Table 7.7 clearly shows the biases are small and, hence, the approximated model is very close to the true model. This example shows one of the benefits and applications of UDs in real world experimentation.

CONCLUSIONS AND FUTURE RESEARCH

When the number of experimental runs allowed to conduct a certain type of experiment is small due to the expensive nature of the experiment, the sizes of existing traditional experimental designs such as orthogonal arrays and fractional factorial designs may exceed the desired number of runs. In such cases, a robust alternative is to use model independent UDs.

The overall aim of this dissertation was to prepare a handy catalog of UD generator tables for practitioners to use because the existing available UD generator tables are limited in scope with respect to design sizes with most containing impractically large numbers of runs. The existing tables also do not consider multiple methods for design generators. Thus, the UDs tables provided in this study address the practical experimental issues related to varying numbers of factors and reasonable design sizes and shows the glp method as the superior method. Unlike the existing UD tables, in this dissertation UD tables for experimental regions other than the hypercubes are also provided. Specifically, UD generator tables were presented for designs in spherical regions as well for mixture experiments in a simplex.

Several methods of constructing UDs are available. In this research, four methods of constructing UDs have been used. These were the glp method, the CF (gp) method, the Halton method and the Hammersley method which were discussed in Chapter 2. It was consistently found that the glp method is the best method. That is, the glp method produces design points with superior measures of uniformity in all the experimental regions studied. However, the glp method is computationally expensive and the degree of complexity increases with the increase in the dimension of the experimental region and number of design points.

To circumvent the computationally expensive nature of the glp method, two methods were proposed in Chapter 4. These were the equivalence method and the projection method. The latter is a much simpler method and, hence, it is recommended for higher dimensions.

Measures of assessing uniformity in hypercubes were discussed in Chapter 3. These include the star discrepancy (STRD) and the proposed new discrepancy measure of uniformity called modified star discrepancy (MSTRD). Also considered were the three distance-based measures of uniformity (*rmsd*, *ad*, and *md*) used by Borkowski and Piepel (2009). The STRD and MSTRD methods are defined only for low dimensional hypercubes because they are computationally expensive whereas the distance-based measures of uniformity are easy to implement. Hence, the distance-based measures of uniformity are the recommended alternative to the discrepancy measures for high dimensional hypercubes. However, note that both the STRD and MSTRD are defined for any dimension.

UDs for low dimensional spherical regions are discussed in Chapter 5. Chapter 5 also introduces new discrepancy measures of uniformity for two-dimensional and three-dimensional spherical regions. The distance-based criteria used in hypercubes are also used for spherical experimental regions as an alternative to discrepancy measures, especially for higher dimensional spherical regions. The proposed new discrepancy measures are computationally expensive even after the two methods of computational task reduction are applied to the glp method. For the other non-glp methods the proposed discrepancy measures are also computationally expensive although less so because of the smaller numbers of available candidate generators. In the spherical experimental region, as was the case in the hypercubes, the glp method also yields low discrepancy design points.

In Chapter 6, UDs for mixture experiments were discussed and new measures of uniformity were proposed in both the unconstrained and constrained simplex regions for three component mixture experiments. A new discrepancy measure was proposed for single component constraint (SCC) mixture experiments but like the previously proposed discrepancy measures, these discrepancy measures are computationally expensive. As alternatives, the distance-based measures of uniformity proposed by Borkowski and Piepel (2009) were applied. However, unlike the hypercube and spherical experimental regions, for mixture experimental regions distance is scaled to ensure each component is treated as equally important. For a multiple component constraint (MCC) mixture experiment, the transformation of design points from a unit cube to a simplex is not as straightforward as the case with only SCCs. The one pass algorithm proposed by Borkowski and Piepel (2009) is used and compared with the one-step algorithm proposed in this study.

Finally the details of UD data layout and a preliminary data analysis are provided in Chapter 7. The data analysis involved a stepwise regression in R, but as discussed above, development of more appropriate methods of data analysis is a future research problem for UDs. Two simulated data examples are provided which demonstrated the robustness of UDs and supports what was proved mathematically by Wiens (1991).

UDs are examples of highly fractionated factorial designs but unlike the fractional factorial designs, UDs are not orthogonal and thus possess a certain level of collinearity. The collinearity will minimally affect prediction, but it can certainly affect parameter estimation. Therefore, UDs may require a more sophisticated data analysis. Finding appropriate methods of data analysis for UDs is in itself a good area of future research. That is, methods of model construction and variable selection procedures have to be developed to account for the collinearity. In this dissertation, however, the focus was on design generation and not on developing methods of data analysis.

For mixture designs the G-mapping function works for SCC mixtures in transforming the design points in the s-dimensional unit cube to a $s+1$ component mixture design in a simplex. However, for MCCs, there is no analogous function that can transform points from a cube of dimension s to a MCC region in the $s+1$ simplex. Hence, this is another open problem for future research.

BIBLIOGRAPHY

Bibliography

Aguilera, A., Aguila, R. P., 2004. General n-Dimensional Rotations. *Communication Papers Proceedings*, Plzen, Czech Republic, Copyright Union Agency-Science Press.

Borkowski, J. J., 2003. A Comparison of Prediction Variance Criteria for Response Surface Designs. *Journal of Quality Technology* 35 (1), 70–77.

Borkowski, J. J., 2010. *Lecture Notes on Response Surface Methodology*. Montana State University, Bozeman.

Borkowski, J. J., Piepel, G. F., 2009. Space-Filling Designs for Highly-Constrained Mixture Experiments. *Journal of Quality Technology* 41 (1), 35–47.

Box, G. E. P., Behnken, D. W., 1960. Some New Three Level Designs for the Study of Quantitative Variables. *Technometrics* 2, 455–475.

Box, G. E. P., Wilson, K., 1951. On the Experimental Attainment of Optimum Conditions (with discussion). *Journal of the Royal Statistical Society Series B* 13 (1), 1–45.

Casella, G., Berger, R. L., 2002. *Statistical Inference*, 2nd Edition. Duxbury, Advanced Series. Pacific Grove, CA.

Chan, L. Y., 2000. Optimal Designs For Experiments With Mixtures: A Survey. *Communincations in Statistics: Theory and Methods* 29, 2281–2312.

Cornell, J., 2002. *Experiments with Mixtures, Designs, Models, and the Analysis of Mixture Data*. 3rd ed. Wiley Series in Probability and Statistics. New York.

Dodson, L. B., Schwab, H. L., 2006. *Accelerated Testing: A Practitioners Guide to Accelerated and Reliability Testing*. SAE International, Warrendale, PA.

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., Li, R., Sudjianto, A., 2006. *Design And Modeling for Computer Experiments*. Chapman and Hall/CRC, Boca Raton.

Fang, K. T., Lin, D. K. J., 2003. Uniform Experimental Designs and Their Application in Industry. Vol. 22. *HandBook of Statistics*, Elsevier Science.

Fang, K. T., Lin, D. K. J., Winker, P., Zhang, Y., 2000. Uniform Design: Theory and Application. *Technometrics* 42, 237–248.

Fang, K. T., Shiu, W., Pan, J., 1999. Uniform Design Based on Latin Squares. *Statistica Sinica* (3), 905–912.

Fang, K. T., Wang, Y., 1994. *Number-Theoretic Methods in Statistics*. Chapman and Hall, London.

Fang, K. T., Wang, Y., Bentler, P. M., 1994. Some Applications of Number-Theoretic Methods in Statistics. *Statistical Science* 9 (3), 416–428.

Fang, K. T., Yang, Z. H., 2000. On Uniform Design of Experiments with Restricted Mixtures and Generation of Uniform Distribution on Some Domains. *Statistics and Probability Letters* (46), 113–120.

Halton, J. H., 1960. On Efficiency of Certain Quasi-random Sequence of Points in Evaluating Multidimensional Integrals. *Numerische Mathematik* (2), 84–90.

Hammersley, J. M., 1964. Monte-Carlo Methods for Solving Multivariable Problems. *Annals of the New York Academy of Science* 86, 844–874.

Harris, J. W., Stocker, H., 1998. *Handbook of Mathematics and Computational Science* . Springer-Verlag, NewYork, Inc.

Hickernell, F. J., 1998a. A Generalized Discrepancy and Quadrature Error Bound. *Mathematics of Computation* 67 (221), 299–332.

Hickernell, F. J., 1998b. *Lattice Rules: How Well Do They Measure Up? In: Random and Quasi- Random Point Sets*, eds. by P. Hellekalek, and G. Larcher, Lecture Notes in Statistics 138, 109–166.

Hua, L. K., Wang, Y., 1981. Application of Number Theory to Numerical Analysis . Berlin: Springer-Verlag, Berlin, Science Press, Beijing.

Johnson, M. E., Moore, L. M., Ylvisaker, D., 1990. Minimax and Maximin Distance Designs. *Journal of Statistical Planning and Inference* 26 (2), 131–148.

Koehler, J. R., Owen, A. B., 1996. Computer Experiments. In S. Ghosh and C. R. Rao (Eds.), *Handbook of Statistics*. Vol. 13. Elsevier Science.

Krommer, A. R., UeberHuber, C. W., 1998. *Computational Integration*. SIAM (Society of Industrial and Applied Mathematics). Philadelphia.

Li, R., Lin, D. K. J., Chen, Y., 2004. Uniform Design: Design, Analysis and Application. *International Journal of Material and Product Technology* 20, 101–114.

Liang, Y., Fang, K. T., Xu, Q., 2001. Uniform Designs and Its Application in Chemistry and Chemical Engineering. *Chemometrics and Intelligent Laboratory Systems* 58, 44–57.

Linse, G., 2008. *Generating Space-Filling Designs Using Meshes*. Department of Mathematical Sciences, Masters degree writing project.

Matlab, 2010. *The Language of Technical Computing R2010a*. MathWorks, Natick, MA.

Myers, R. H., Montgomery, D. C., Anderson-Cook, C., 2009. *Response Surface Methodology, Process and Product Optimization Using Designed Experiments* . Third ed. John Wiley and Sons, Inc., Hoboken, New Jersey.

Niederreiter, H., 1992. Random Number Generation and Quasi-Monte Carlo Methods . *CBMS-NSF Regional Conference Series in Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Pham, H., 2006. *Springer Handbook of Engineering Statistics*. Springer-Verlag, London.

Prescott, P., 2008. Design of Experiments: Nearly Uniform Designs for Mixture Designs. *Communications in Statistics-Theory and Methods* (37), 2095–2115.

R Development Core Team, 2011. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
URL `http://www.R-project.org/`

Riddle, D. F., 1977. *Analytic Geometry with Vectors* . Wadsworth Publishing, Belmont, California.

Roquemore, K. G., 1976. Hybrid Designs for Quadratic Response Surfaces. *Technometrics* (18), 419–423.

Sacks, J., S. S. B., Welch, W. J., 1989. Designs for Computer Experiments. *Technometrics* 31 (1), 41–47.

Santner, T. J., W. B. J., Notz, W. I., 2003. *The Design and Analysis of Computer Experiments* . Springer-Verlag, Inc. NewYork.

Smith, W. F., 2005. *Experimental Design For Formulation*. The American Statistical Association and Society for Industrial and Applied Mathematics .

Tony, T. W., 1972. Computational Investigations of Low-Discrepancy Point Sets. *Applications of Number Theory to Numerical Analysis*. Academic Press, New York, 319–343.

Varberg, D., P. W. J., Rigdon, S. E., 2000. *CALCULUS*. Prentice Hall, Upper Saddle River, New Jersey.

Venables, W. N., Ripley, B. D., 2002. Modern Applied Statistics with S, 4th Edition. Springer, New York, iSBN 0-387-95457-0.
URL `http://www.stats.ox.ac.uk/pub/MASS4`

Wang, Y., Fang, K. T., 1981. A Note on Uniform Distribution and Experimental Design. *KeXue TongBao* 6, 485–489.

Wang, Y., Fang, K. T., 1990. Number Theroretic Methods in Applied Statistics. *Chinese Annals of Mathematics* 11B:1, 51–65.

Weisstein, E. W., 2011. "Spherical Cap." From MathWorld-A Wolfram Web Resource, http://mathworld.wolfram.com/SphericalCap.html .

Wiens, D. P., 1991. Designs for Approximately Linear Regression: Two Optimality Properties of Uniform Designs. *Statistics and Probability Letters* (12), 217–221.

Yuan, W., Fang, K. T., 2010. On Number-Theoretic Methods in Statistics Simulation. *Science China Mathematics* 53 (1), 179–186.

Zhang, L., Liang, Y., Jiang, J., Yu, R., Fang, K. T., 1998. Uniform Designs Applied to Nonlinear Multivariate Calibration by Artificial Neural Network. *Analytica Chimica Acta* 370 (1), 65–77.

APPENDICES

The Matlab codes that generated the UD Tables and all the results in this dissertation are available from the author.

APPENDIX A

TABLES IN C^S FOR $S = 2, 3, 4$ DESIGN VARIABLES

glp method in C^2										
	rmsd		*ad*		*md*		STRD		MSTRD	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3)	0.1456	(1,3)	0.1331	(1,3)	0.3522	(1,3)	0.1798	(1,3)	0.1798
11	(1,7)	0.1350	(1,7)	0.1244	(1,7)	0.3200	(1,5)	0.1490	(1,7)	0.1490
12	(1,5)	0.1338	(1,5)	0.1232	(1,5)	0.2754	(1,5)	0.1393	(1,5)	0.1393
13	(1,5)	0.1242	(1,5)	0.1149	(1,4)	0.2834	(1,4)	0.1265	(1,5)	0.1302
14	(1,9)	0.1215	(1,9)	0.1113	(1,9)	0.2620	(1,9)	0.1189	(1,9)	0.1189
15	(1,11)	0.1145	(1,11)	0.1062	(1,11)	0.2497	(1,11)	0.1096	(1,11)	0.1096
16	(1,5)	0.1167	(1,5)	0.1062	(1,5)	0.2810	(1,7)	0.1037	(1,7)	0.1063
17	(1,5)	0.1082	(1,5)	0.0999	(1,5)	0.2326	(1,11)	0.1022	(1,11)	0.1037
18	(1,5)	0.1027	(1,5)	0.0950	(1,5)	0.2359	(1,7)	0.0996	(1,7)	0.0996
19	(1,14)	0.1008	(1,14)	0.0932	(1,8)	0.2304	(1,14)	0.0875	(1,14)	0.0897
20	(1,13)	0.1100	(1,13)	0.0993	(1,13)	0.2787	(1,9)	0.0770	(1,9)	0.0971
21	(1,13)	0.0950	(1,13)	0.0881	(1,13)	0.2097	(1,5)	0.0922	(1,5)	0.0922
22	(1,5)	0.0936	(1,5)	0.0864	(1,5)	0.2136	(1,13)	0.0895	(1,13)	0.0895
23	(1,9)	0.0901	(1,9)	0.0837	(1,17)	0.2114	(1,9)	0.0717	(1,9)	0.0734
24	(1,19)	0.0886	(1,19)	0.0822	(1,7)	0.1874	(1,11)	0.0693	(1,17)	0.0722
25	(1,7)	0.0869	(1,7)	0.0804	(1,7)	0.2014	(1,11)	0.0693	(1,11)	0.0693
26	(1,11)	0.0862	(1,11)	0.0797	(1,11)	0.1787	(1,11)	0.0622	(1,11)	0.0639
27	(1,16)	0.0844	(1,5)	0.0778	(1,8)	0.1951	(1,16)	0.0652	(1,16)	0.0695
28	(1,11)	0.0821	(1,5)	0.0760	(1,11)	0.1961	(1,11)	0.0650	(1,11)	0.0650
29	(1,18)	0.0799	(1,18)	0.0740	(1,18)	0.1730	(1,18)	0.0615	(1,9)	0.0628
30	(1,7)	0.0821	(1,7)	0.0758	(1,7)	0.1932	(1,19)	0.0580	(1,19)	0.0580
31	(1,12)	0.0781	(1,5)	0.0725	(1,7)	0.1689	(1,14)	0.0513	(1,14)	0.0595
32	(1,7)	0.0757	(1,7)	0.0702	(1,7)	0.1750	(1,19)	0.0617	(1,9)	0.0623
33	(1,14)	0.0751	(1,14)	0.0699	(1,14)	0.1691	(1,14)	0.0540	(1,14)	0.0554
34	(1,13)	0.0743	(1,13)	0.0691	(1,13)	0.1753	(1,15)	0.0525	(1,13)	0.0531
35	(1,29)	0.0719	(1,29)	0.0665	(1,8)	0.1631	(1,13)	0.0552	(1,13)	0.0593
36	(1,7)	0.0749	(1,7)	0.0691	(1,7)	0.1808	(1,11)	0.0559	(1,11)	0.0559
37	(1,8)	0.0703	(1,6)	0.0654	(1,23)	0.1590	(1,23)	0.0461	(1,23)	0.0463
38	(1,7)	0.0701	(1,7)	0.0650	(1,7)	0.1590	(1,27)	0.0504	(1,27)	0.0516
39	(1,7)	0.0684	(1,7)	0.0636	(1,17)	0.1547	(1,25)	0.0467	(1,25)	0.0467
40	(1,17)	0.0672	(1,17)	0.0625	(1,9)	0.1490	(1,17)	0.0461	(1,31)	0.0510
41	(1,9)	0.0667	(1,34)	0.0620	(1,9)	0.1543	(1,17)	0.0435	(1,17)	0.0435
42	(1,25)	0.0710	(1,25)	0.0650	(1,25)	0.1784	(1,19)	0.0433	(1,19)	0.0480
43	(1,25)	0.0647	(1,25)	0.0601	(1,25)	0.1487	(1,19)	0.0434	(1,19)	0.0444
44	(1,7)	0.0654	(1,27)	0.0604	(1,13)	0.1517	(1,27)	0.0418	(1,27)	0.0428
45	(1,26)	0.0629	(1,26)	0.0585	(1,19)	0.1506	(1,19)	0.0388	(1,19)	0.0395
46	(1,13)	0.0627	(1,13)	0.0583	(1,27)	0.1523	(1,13)	0.0405	(1,27)	0.0435
47	(1,18)	0.0625	(1,7)	0.0581	(1,18)	0.1417	(1,14)	0.0408	(1,14)	0.0422
48	(1,41)	0.0619	(1,41)	0.0575	(1,41)	0.1435	(1,31)	0.0364	(1,41)	0.0423
49	(1,9)	0.0616	(1,38)	0.0571	(1,9)	0.1370	(1,15)	0.0385	(1,22)	0.0385
50	(1,11)	0.0597	(1,11)	0.0555	(1,9)	0.1389	(1,23)	0.0398	(1,21)	0.0399

Table A.1: Best generators in C^2 from the glp method

glp method in C^3										
	rmsd		*ad*		*md*		STRD		MSTRD	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3,9)	0.3147	(1,3,9)	0.2874	(1,3,9)	0.7697	(1,3,7)	0.2555	(1,3,7)	0.2819
11	(1,3,5)	0.2593	(1,3,5)	0.2445	(1,2,4)	0.5206	(1,3,4)	0.2285	(1,3,4)	0.2285
12	(1,5,11)	0.3096	(1,5,11)	0.2810	(1,5,7)	0.7544	(1,5,11)	0.2228	(1,5,7)	0.2884
13	(1,4,6)	0.2426	(1,4,6)	0.2297	(1,4,11)	0.5184	(1,3,5)	0.1590	(1,4,5)	0.1754
14	(1,3,5)	0.2384	(1,3,5)	0.2245	(1,3,5)	0.5925	(1,3,5)	0.1804	(1,3,5)	0.1804
15	(1,7,13)	0.2397	(1,7,13)	0.2249	(1,2,4)	0.5103	(1,7,13)	0.1500	(1,7,13)	0.1827
16	(1,3,7)	0.2212	(1,3,7)	0.2103	(1,3,7)	0.4328	(1,5,9)	0.1408	(1,5,9)	0.1474
17	(1,4,10)	0.2203	(1,3,5)	0.2083	(1,4,10)	0.4358	(1,8,12)	0.1249	(1,4,5)	0.1508
18	(1,5,7)	0.2190	(1,5,7)	0.2069	(1,5,7)	0.4426	(1,5,7)	0.1404	(1,5,7)	0.1513
19	(1,3,12)	0.2097	(1,3,12)	0.1991	(1,3,7)	0.4175	(1,6,8)	0.1190	(1,6,8)	0.1331
20	(1,3,13)	0.2073	(1,3,13)	0.1965	(1,3,13)	0.4192	(1,3,11)	0.1374	(1,3,17)	0.1496
21	(1,4,10)	0.2121	(1,4,10)	0.1989	(1,2,16)	0.4756	(1,4,13)	0.1186	(1,4,5)	0.1458
22	(1,5,7)	0.2002	(1,5,7)	0.1897	(1,5,7)	0.4050	(1,5,7)	0.1263	(1,5,13)	0.1314
23	(1,9,20)	0.1938	(1,9,20)	0.1847	(1,9,20)	0.3856	(1,5,19)	0.1101	(1,7,18)	0.1229
24	(1,11,19)	0.1960	(1,11,19)	0.1855	(1,17,19)	0.3955	(1,11,19)	0.0895	(1,11,17)	0.1026
25	(1,8,14)	0.1885	(1,3,16)	0.1791	(1,6,9)	0.3602	(1,8,14)	0.0943	(1,6,11)	0.1081
26	(1,3,17)	0.1869	(1,3,17)	0.1770	(1,3,9)	0.3950	(1,5,7)	0.0925	(1,5,11)	0.1114
27	(1,4,17)	0.1833	(1,4,17)	0.1741	(1,4,10)	0.3966	(1,8,22)	0.0928	(1,5,19)	0.1096
28	(1,5,13)	0.1826	(1,5,13)	0.1728	(1,9,15)	0.3797	(1,5,13)	0.0921	(1,5,13)	0.1145
29	(1,16,20)	0.1774	(1,16,20)	0.1688	(1,4,18)	0.3569	(1,9,17)	0.0856	(1,9,17)	0.0862
30	(1,7,13)	0.1803	(1,7,13)	0.1710	(1,7,11)	0.3655	(1,7,13)	0.0821	(1,7,13)	0.0845
31	(1,10,17)	0.1733	(1,6,9)	0.1646	(1,10,14)	0.3331	(1,5,19)	0.0827	(1,6,22)	0.0850
32	(1,3,13)	0.1753	(1,3,13)	0.1659	(1,9,29)	0.3506	(1,7,9)	0.0877	(1,5,7)	0.0984
33	(1,4,23)	0.1705	(1,4,23)	0.1621	(1,4,23)	0.3426	(1,7,23)	0.0779	(1,8,14)	0.0934
34	(1,11,27)	0.1697	(1,11,27)	0.1608	(1,9,31)	0.3496	(1,13,25)	0.0732	(1,9,21)	0.0894
35	(1,11,19)	0.1647	(1,11,19)	0.1571	(1,11,19)	0.3107	(1,8,11)	0.0770	(1,8,22)	0.0904
36	(1,7,17)	0.1683	(1,7,17)	0.1593	(1,5,19)	0.3666	(1,5,13)	0.0781	(1,5,7)	0.0880
37	(1,7,17)	0.1625	(1,4,27)	0.1547	(1,4,13)	0.3158	(1,17,29)	0.0618	(1,7,20)	0.0814
38	(1,7,17)	0.1626	(1,7,17)	0.1542	(1,7,27)	0.3258	(1,9,25)	0.0676	(1,9,21)	0.0773
39	(1,4,14)	0.1600	(1,4,14)	0.1523	(1,4,14)	0.3222	(1,11,23)	0.0715	(1,11,23)	0.0885
40	(1,13,17)	0.1619	(1,13,17)	0.1534	(1,13,17)	0.3349	(1,7,17)	0.0664	(1,17,33)	0.0826
41	(1,8,19)	0.1564	(1,8,19)	0.1489	(1,10,28)	0.3115	(1,9,26)	0.0631	(1,9,26)	0.0675
42	(1,25,31)	0.1534	(1,25,31)	0.1462	(1,11,25)	0.3017	(1,25,31)	0.0627	(1,11,25)	0.0786
43	(1,12,35)	0.1545	(1,12,35)	0.1470	(1,8,12)	0.3065	(1,17,19)	0.0604	(1,10,18)	0.0663
44	(1,7,27)	0.1577	(1,7,27)	0.1491	(1,7,27)	0.3300	(1,25,27)	0.0680	(1,25,27)	0.0680
45	(1,4,17)	0.1518	(1,4,17)	0.1447	(1,11,29)	0.2932	(1,14,37)	0.0569	(1,28,37)	0.0721
46	(1,5,33)	0.1503	(1,5,33)	0.1432	(1,7,11)	0.3068	(1,7,17)	0.0630	(1,13,29)	0.0661
47	(1,4,18)	0.1493	(1,4,18)	0.1423	(1,7,22)	0.2962	(1,10,14)	0.0535	(1,15,18)	0.0651
48	(1,13,19)	0.1486	(1,13,19)	0.1415	(1,13,19)	0.2908	(1,13,19)	0.0566	(1,13,19)	0.0690
49	(1,6,15)	0.1464	(1,6,15)	0.1394	(1,8,13)	0.2775	(1,15,29)	0.0518	(1,15,29)	0.0602
50	(1,7,23)	0.1463	(1,7,23)	0.1392	(1,7,27)	0.3033	(1,7,11)	0.0528	(1,11,21)	0.0619

Table A.2: Best generators in C^3 from the glp method

glp method in C^4						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3,7,9)	0.4400	(1,3,7,9)	0.4142	(1,3,7,9)	0.8936
11	(1,2,5,7)	0.3710	(1,2,5,7)	0.3561	(1,2,5,7)	0.6965
12	(1,5,7,11)	0.4362	(1,5,7,11)	0.4094	(1,5,7,11)	0.8895
13	(1,4,5,11)	0.3521	(1,4,5,11)	0.3383	(1,4,5,11)	0.6553
14	(1,3,5,9)	0.3831	(1,3,5,9)	0.3616	(1,3,5,9)	0.7920
15	(1,2,4,7)	0.3334	(1,2,4,7)	0.3211	(1,2,4,8)	0.5946
16	(1,3,5,9)	0.3320	(1,3,5,9)	0.3193	(1,3,5,7)	0.6640
17	(1,2,8,13)	0.3252	(1,2,8,13)	0.3127	(1,2,8,13)	0.5964
18	(1,5,7,13)	0.3719	(1,5,7,13)	0.3491	(1,7,13,17)	0.7807
19	(1,3,5,9)	0.3162	(1,3,5,9)	0.3040	(1,3,5,9)	0.5817
20	(1,3,7,11)	0.3180	(1,3,7,11)	0.3046	(1,3,7,11)	0.6518
21	(1,2,5,10)	0.3100	(1,2,5,10)	0.2975	(1,2,5,13)	0.5804
22	(1,3,7,13)	0.3063	(1,3,7,13)	0.2942	(1,5,7,13)	0.6339
23	(1,3,7,11)	0.3013	(1,2,10,18)	0.2892	(1,2,11,17)	0.5619
24	(1,5,7,13)	0.3069	(1,5,7,13)	0.2940	(1,5,13,17)	0.6251
25	(1,4,9,19)	0.2923	(1,4,9,19)	0.2814	(1,4,11,19)	0.5615
26	(1,3,5,15)	0.2912	(1,3,5,15)	0.2799	(1,5,7,15)	0.6124
27	(1,5,17,19)	0.2899	(1,4,17,19)	0.2781	(1,2,7,16)	0.5467
28	(1,5,23,25)	0.2818	(1,5,23,25)	0.2718	(1,3,5,9)	0.5795
29	(1,4,6,13)	0.2793	(1,4,6,13)	0.2690	(1,4,7,16)	0.5225
30	(1,7,11,13)	0.2786	(1,7,11,13)	0.2685	(1,7,11,13)	0.5148
31	(1,6,10,13)	0.2744	(1,6,10,13)	0.2641	(1,5,19,22)	0.5141
32	(1,7,15,19)	0.2725	(1,3,15,25)	0.2622	(1,5,7,9)	0.5716
33	(1,5,13,31)	0.2745	(1,5,13,31)	0.2637	(1,4,23,31)	0.5305
34	(1,9,13,15)	0.2695	(1,9,13,15)	0.2592	(1,9,13,15)	0.5226
35	(1,4,19,26)	0.2663	(1,4,19,26)	0.2561	(1,4,9,24)	0.5150
36	(1,5,11.13)	0.2677	(1,5,11.13)	0.2574	(1,5,13,29)	0.4861
37	(1,3,16,30)	0.2597	(1,3,16,30)	0.2503	(1,3,7,16)	0.4705
38	(1,9,21,35)	0.2624	(1,3,7,25)	0.2525	(1,5,7,29)	0.4916
39	(1,14,22,29)	0.2608	(1,14,22,29)	0.2500	(1, 2,7,28)	0.5150
40	(1,3,7,19)	0.2549	(1,3,7,19)	0.2458	(1,3,13,33)	0.4891
41	(1,5,13,16)	0.2528	(1,8,19,36)	0.2440	(1,5,22,33)	0.4623
42	(1,5,19,31)	0.2538	(1,5,19,31)	0.2444	(1,5,11,23)	0.4869
43	(1,7,19,40)	0.2492	(1,7,19,40)	0.2400	(1,6,13,16)	0.2517
44	(1,3,25,37)	0.2499	(1,3,25,37)	0.2405	(1,3,13,25)	0.4627
45	(1,7,17, 32)	0.2498	(1,4,7,29)	0.2399	(1,8,26,31)	0.4717
46	(1,11,17,43)	0.2471	(1,11,17,43)	0.2378	(1,3,7,19)	0.4543
47	(1,10,15,44)	0.2432	(1,10,15,44)	0.2343	(1,4,10,25)	0.4455
48	(1,5,11,23)	0.2457	(1, 5,11,23)	0.2364	(1,5,11,35)	0.4571
49	(1,12,29,46)	0.2409	(1,12,29,46)	0.2321	(1,3,10,31)	0.4509
50	(1, 3,19,43)	0.2396	(1,3,19,43)	0.2308	(1,3,13,21)	0.4487

Table A.3: Best generators in C^4 from the glp method

CF method in C^2										
	rmsd		*ad*		*md*		STRD		MSTRD	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	P=7	0.1792	P=7	0.1572	P=7	0.4945	P=7	0.3072	P=7	0.3072
11	P=11	0.1687	P=7	0.1493	P=11	0.4210	P=11	0.2128	P=7	0.2695
12	P=11	0.1651	P=7	0.1451	P=11	0.4210	P=11	0.1810	P=11	0.2846
13	P=13	0.1333	P=13	0.1210	P=7	0.2900	P=13	0.1372	P=13	0.1740
14	P=13	0.1265	P=13	0.1154	P=13	0.2774	P=13	0.1372	P=13	0.1643
15	P=13	0.1216	P=13	0.1111	P=13	0.2574	P=13	0.1419	P=13	0.1603
16	P=13	0.1177	P=13	0.1072	P=13	0.2574	P=13	0.1328	P=13	0.1560
17	P=17	0.1129	P=17	0.1026	P=13	0.2512	P=13	0.1622	P=17	0.1728
18	P=13	0.1087	P=13	0.0994	P=13	0.2512	P=13	0.1388	P=13	0.1599
19	P=19	0.1035	P=19	0.0952	P=13	0.2512	P=19	0.1208	P=19	0.1364
20	P=13	0.1019	P=17	0.0930	P=13	0.2512	P=19	0.0945	P=13	0.1208
21	P=13	0.0958	P=13	0.0886	P=17	0.2256	P=17	0.0764	P=17	0.1060
22	P=17	0.0943	P=17	0.0872	P=17	0.2256	P=13	0.0755	P=13	0.0925
23	P=17	0.0933	P=17	0.0861	P=17	0.2256	P=17	0.0764	P=17	0.0992
24	P=17	0.0921	P=17	0.0849	P=17	0.2256	P=17	0.0826	P=13	0.1114
25	P=17	0.0913	P=17	0.0840	P=17	0.2256	P=17	0.0922	P=23	0.1099
26	P=17	0.0902	P=17	0.0828	P=17	0.2256	P=17	0.0986	P=23	0.1043
27	P=17	0.0889	P=17	0.0814	P=17	0.2256	P=17	0.0835	P=23	0.1000
28	P=17	0.0875	P=17	0.0800	P=17	0.2256	P=23	0.0742	P=19	0.0957
29	P=17	0.0853	P=17	0.0782	P=17	0.1971	P=19	0.0836	P=13	0.1047
30	P=17	0.0839	P=17	0.0769	P=17	0.1971	P=19	0.0854	P=13	0.0996
31	P=17	0.0829	P=17	0.0758	P=17	0.1971	P=13	0.0728	P=13	0.0943
32	P=17	0.0820	P=17	0.0748	P=17	0.1971	P=19	0.0811	P=13	0.0968
33	P=17	0.0808	P=17	0.0736	P=17	0.1971	P=13	0.0829	P=13	0.0940
34	P=17	0.0798	P=17	0.0725	P=17	0.1971	P=13	0.0900	P=13	0.0900
35	P=17	0.0789	P=17	0.0717	P=17	0.1971	P=13	0.0784	P=13	0.0792
36	P=17	0.0769	P=17	0.0701	P=17	0.1967	P=13	0.0816	P=13	0.0864
37	P=17	0.0767	P=17	0.0699	P=17	0.1967	P=37	0.0739	P=13	0.0858
38	P=17	0.0751	P=17	0.0686	P=17	0.1709	P=19	0.0682	P=13	0.0866
39	P=17	0.0741	P=17	0.0677	P=17	0.1709	P=23	0.0646	P=23	0.0860
40	P=17	0.0728	P=17	0.0665	P=17	0.1709	P=19	0.0652	P=23	0.0842
41	P=17	0.0717	P=17	0.0655	P=17	0.1709	P=23	0.0674	P=23	0.0693
42	P=17	0.0707	P=17	0.0645	P=17	0.1709	P=17	0.0577	P=13	0.0804
43	P=17	0.0696	P=17	0.0636	P=17	0.1709	P=17	0.0643	P=13	0.0671
44	P=17	0.0686	P=17	0.0627	P=17	0.1709	P=17	0.0563	P=13	0.0720
45	P=17	0.0678	P=17	0.0620	P=17	0.1709	P=41	0.0594	P=17	0.0709
46	P=17	0.0677	P=17	0.0619	P=17	0.1709	P=17	0.0604	P=13	0.0752
47	P=17	0.0674	P=17	0.0615	P=17	0.1709	P=17	0.0595	P=43	0.0754
48	P=17	0.0671	P=17	0.0612	P=17	0.1709	P=43	0.0586	P=13	0.0848
49	P=17	0.0668	P=17	0.0609	P=17	0.1709	P=19	0.0588	P=31	0.0862
50	P=17	0.0661	P=17	0.0603	P=17	0.1709	P=19	0.0569	P=13	0.0766

Table A.4: Best generators in C^2 from the CF method

CF method in C^3										
	$rmsd$		ad		md		STRD		MSTRD	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	-	-	-	-	-	-	-	-	-	-
11	P=11	0.2753	P=11	0.2583	P=11	0.5881	P=11	0.2308	P=11	0.3525
12	P=11	0.2680	P=11	0.2516	P=11	0.5460	P=11	0.2308	P=11	0.3525
13	P=11	0.2610	P=11	0.2444	P=11	0.5460	P=11	0.2352	P=13	0.2870
14	P=11	0.2570	P=11	0.2405	P=11	0.5460	P=11	0.2407	P=13	0.2628
15	P=11	0.2479	P=11	0.2321	P=11	0.5387	P=11	0.2308	P=13	0.2566
16	P=11	0.2404	P=11	0.2252	P=11	0.5387	P=11	0.2232	P=11	0.2927
17	P=11	0.2317	P=11	0.2177	P=11	0.4876	P=17	0.1803	P=17	0.1978
18	P=11	0.2265	P=11	0.2127	P=11	0.4876	P=17	0.1489	P=17	0.1906
19	P=11	0.2236	P=11	0.2098	P=11	0.4876	P=17	0.1440	P=17	0.1731
20	P=11	0.2203	P=11	0.2063	P=11	0.4876	P=17	0.1479	P=17	0.1697
21	P=11	0.2174	P=11	0.2035	P=11	0.4876	P=17	0.1551	P=19	0.1823
22	P=11	0.2122	P=11	0.1994	P=11	0.4829	P=11	0.1543	P=19	0.2020
23	P=11	0.2094	P=11	0.1965	P=11	0.4829	P=23	0.1356	P=17	0.1809
24	P=11	0.2053	P=11	0.1926	P=11	0.4829	P=23	0.1405	P=17	0.1729
25	P=11	0.2027	P=11	0.1902	P=11	0.4829	P=23	0.1261	P=23	0.1660
26	P=11	0.1999	P=11	0.1871	P=11	0.4829	P=23	0.1308	P=17	0.1584
27	P=11	0.2971	P=11	0.1842	P=11	0.4829	P=23	0.1165	P=17	0.1545
28	P=11	0.1923	P=11	0.1806	P=11	0.4225	P=23	0.1033	P=13	0.1557
29	P=11	0.1884	P=11	0.1771	P=11	0.4225	P=23	0.1045	P=13	0.1619
30	P=11	0.1856	P=11	0.1743	P=11	0.4225	P=23	0.1036	P=19	0.1437
31	P=11	0.1828	P=11	0.1716	P=11	0.4096	P=23	0.1176	P=13	0.1394
32	P=11	0.1787	P=11	0.1683	P=31	0.4096	P=23	0.1191	P=19	0.1479
33	P=11	0.1771	P=11	0.1666	P=31	0.4096	P=23	0.1024	P=17	0.1511
34	P=11	0.1725	P=11	0.1628	P=11	0.3967	P=23	0.0975	P=17	0.1308
35	P=11	0.1711	P=11	0.1616	P=31	0.3967	P=23	0.0870	P=17	0.1365
36	P=11	0.1703	P=11	0.1608	P=31	0.3815	P=23	0.0922	P=17	0.1371
37	P=11	0.1691	P=11	0.1595	P=31	0.3815	P=23	0.1032	P=17	0.1264
38	P=11	0.1672	P=11	0.1578	P=17	0.3404	P=23	0.0940	P=17	0.1168
39	P=11	0.1662	P=11	0.1567	P=17	0.3404	P=11	0.1073	P=23	0.1166
40	P=11	0.1650	P=11	0.1557	P=17	0.3404	P=11	0.0923	P=23	0.1130
41	P=11	0.1639	P=11	0.1545	P=17	0.3404	P=41	0.0742	P=41	0.1027
42	P=11	0.1630	P=11	0.1537	P=17	0.3404	P=41	0.0824	P=17	0.1016
43	P=11	0.1622	P=11	0.1529	P=17	0.3404	P=41	0.0704	P=41	0.1010
44	P=11	0.1614	P=11	0.1520	P=17	0.3404	P=41	0.0625	P=17	0.1061
45	P=17	0.1601	P=11	0.1511	P=17	0.3404	P=41	0.0676	P=17	0.0952
46	P=11	0.1588	P=11	0.1497	P=17	0.3404	P=17	0.0643	P=17	0.0899
47	P=11	0.1570	P=11	0.1481	P=17	0.3404	P=43	0.0764	P=17	0.0890
48	P=11	0.1558	P=11	0.1469	P=17	0.3404	P=43	0.0697	P=17	0.0966
49	P=11	0.1551	P=11	0.1462	P=17	0.3404	P=43	0.0685	P=31	0.1002
50	P=11	0.1539	P=11	0.1451	P=17	0.3404	P=43	0.0737	P=31	0.1013

Table A.5: Best generators in C^3 from the CF method

CF method in C^4						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
10	-	-	-	-	-	-
11	P=11	0.4187	P=11	0.3950	P=11	0.8954
12	P=11	0.4143	P=11	0.3903	P=11	0.8954
13	P=13	0.3817	P=13	0.3616	P=13	0.8457
14	P=13	0.3764	P=13	0.3562	P=13	0.8457
15	P=13	0.3710	P=13	0.3512	P=13	0.8457
16	P=13	0.3677	P=13	0.3475	P=11	0.8066
17	P=17	0.3346	P=17	0.3199	P=17	0.6731
18	P=17	0.3280	P=17	0.3136	P=17	0.6731
19	P=17	0.3231	P=17	0.3092	P=17	0.6516
20	P=17	0.3171	P=17	0.3036	P=17	0.6516
21	P=17	0.3135	P=17	0.3002	P=17	0.6516
22	P=17	0.3119	P=17	0.2986	P=17	0.6516
23	P=17	0.3094	P=17	0.2962	P=17	0.6516
24	P=17	0.3051	P=17	0.2918	P=17	0.6516
25	P=17	0.3021	P=17	0.2889	P=17	0.6516
26	P=17	0.2976	P=17	0.2847	P=17	0.6516
27	P=17	0.2947	P=17	0.2823	P=17	0.6516
28	P=17	0.2921	P=17	0.2797	P=13	0.6400
29	P=17	0.2889	P=17	0.2764	P=29	0.6126
30	P=17	0.2843	P=17	0.2725	P=29	0.6126
31	P=17	0.2816	P=17	0.2698	P=31	0.6024
32	P=17	0.2781	P=17	0.2666	P=13	0.5890
33	P=17	0.2754	P=17	0.2639	P=13	0.5890
34	P=17	0.2720	P=17	0.2607	P=13	0.5890
35	P=17	0.2686	P=17	0.2574	P=13	0.5890
36	P=17	0.2639	P=17	0.2536	P=17	0.5434
37	P=17	0.2627	P=17	0.2526	P=17	0.5434
38	P=17	0.2610	P=17	0.2510	P=17	0.5434
39	P=17	0.2597	P=17	0.2497	P=17	0.5434
40	P=17	0.2580	P=17	0.2480	P=17	0.5434
41	P=41	0.2555	P=41	0.2453	P=41	0.5389
42	P=41	0.2538	P=41	0.2438	P=41	0.5389
43	P=41	0.2532	P=41	0.2433	P=41	0.5389
44	P=17	0.2522	P=17	0.2423	P=41	0.5389
45	P=17	0.2502	P=17	0.2404	P=41	0.5389
46	P=17	0.2500	P=17	0.2402	P=41	0.5389
47	P=31	0.2469	P=31	0.2375	P=31	0.5220
48	P=31	0.2454	P=31	0.2361	P=31	0.4889
49	P=31	0.2447	P=31	0.2354	P=31	0.4889
50	P=31	0.2436	P=31	0.2344	P=31	0.4889

Table A.6: Best generators in C^4 from the CF method

Halton and Hammersley methods in C^2										
	rmsd		*ad*		*md*		STRD		MSTRD	
n	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.
10	0.1649	0.1477	0.1474	0.1346	0.4299	0.3676	0.2457	0.1746	0.2626	0.1939
11	0.1575	0.1379	0.1408	0.1264	0.4156	0.3327	0.2143	0.1811	0.2281	0.1811
12	0.1446	0.1311	0.1304	0.1202	0.3409	0.3204	0.2431	0.1646	0.2431	0.1742
13	0.1392	0.1266	0.1249	0.1158	0.3409	0.3104	0.2445	0.1457	0.2445	0.1457
14	0.1313	0.1222	0.1183	0.1117	0.3409	0.3055	0.2143	0.1398	0.2143	0.1398
15	0.1241	0.1190	0.1122	0.1087	0.3409	0.2961	0.1625	0.1295	0.2095	0.1295
16	0.1216	0.1146	0.1097	0.1050	0.3409	0.2901	0.1586	0.1224	0.1649	0.1224
17	0.1190	0.1123	0.1071	0.1026	0.3409	0.2854	0.1241	0.1228	0.1337	0.1228
18	0.1175	0.1077	0.1053	0.0986	0.3409	0.2814	0.1367	0.1098	0.1459	0.1098
19	0.1156	0.1030	0.1033	0.0947	0.3409	0.2780	0.1427	0.1100	0.1540	0.1100
20	0.1142	0.1005	0.1015	0.0922	0.3409	0.2751	0.1336	0.1047	0.1364	0.1080
21	0.1125	0.0989	0.0996	0.0904	0.3409	0.2725	0.1407	0.0989	0.1530	0.0989
22	0.1087	0.0964	0.0962	0.0879	0.3409	0.2703	0.1472	0.1032	0.1472	0.1032
23	0.1001	0.0936	0.0905	0.0856	0.2625	0.2684	0.1241	0.0905	0.1241	0.0944
24	0.0976	0.0912	0.0882	0.0836	0.2625	0.2666	0.1241	0.0951	0.1241	0.0951
25	0.0957	0.0900	0.0863	0.0825	0.2625	0.2651	0.1241	0.0836	0.1241	0.0836
26	0.0916	0.0881	0.0833	0.0809	0.2625	0.2638	0.0965	0.0863	0.0982	0.0906
27	0.0890	0.0862	0.0814	0.0793	0.2186	0.2625	0.0984	0.0791	0.0984	0.0791
28	0.0882	0.0853	0.0804	0.0783	0.2186	0.2582	0.1050	0.0836	0.1050	0.0836
29	0.0876	0.0846	0.0796	0.0774	0.2186	0.2512	0.1112	0.0788	0.1112	0.0788
30	0.0865	0.0833	0.0785	0.0761	0.2186	0.2447	0.1169	0.0705	0.1169	0.0714
31	0.0854	0.0824	0.0774	0.0752	0.2186	0.2421	0.0904	0.0764	0.1113	0.0764
32	0.0849	0.0798	0.0769	0.0733	0.2186	0.2331	0.0961	0.0696	0.0986	0.0696
33	0.0844	0.0795	0.0762	0.0729	0.2186	0.2278	0.1018	0.0658	0.1084	0.0658
34	0.0835	0.0780	0.0753	0.0715	0.2186	0.2230	0.1071	0.0666	0.1071	0.0666
35	0.0804	0.0760	0.0729	0.0700	0.2186	0.2184	0.0836	0.0630	0.0836	0.0630
36	0.0784	0.0752	0.0714	0.0693	0.1737	0.2142	0.0938	0.0587	0.0938	0.0630
37	0.0772	0.0745	0.0703	0.0687	0.1737	0.2102	0.0944	0.0613	0.1035	0.0613
38	0.0759	0.0734	0.0690	0.0677	0.1737	0.2065	0.0994	0.0626	0.0994	0.0626
39	0.0745	0.0719	0.0679	0.0664	0.1737	0.2029	0.0822	0.0576	0.1014	0.0576
40	0.0741	0.0708	0.0674	0.0653	0.1737	0.1996	0.0867	0.0574	0.0969	0.0584
41	0.0737	0.0702	0.0670	0.0646	0.1737	0.1965	0.0820	0.0629	0.0820	0.0629
42	0.0734	0.0690	0.0666	0.0635	0.1737	0.1936	0.0901	0.0561	0.0901	0.0606
43	0.0731	0.0678	0.0663	0.0624	0.1737	0.1908	0.0870	0.0567	0.0870	0.0567
44	0.0710	0.0668	0.0646	0.0615	0.1737	0.1882	0.0741	0.0572	0.0741	0.0572
45	0.0707	0.0658	0.0642	0.0606	0.1737	0.1857	0.0756	0.0531	0.0837	0.0579
46	0.0700	0.0651	0.0635	0.0599	0.1737	0.1834	0.0771	0.0545	0.0847	0.0589
47	0.0685	0.0646	0.0623	0.0594	0.1717	0.1811	0.0590	0.0559	0.0736	0.0559
48	0.0671	0.0635	0.0612	0.0585	0.1657	0.1790	0.0649	0.0544	0.0740	0.0544
49	0.0665	0.0632	0.0606	0.0582	0.1657	0.1770	0.0728	0.0481	0.0728	0.0481
50	0.0655	0.0621	0.0598	0.0573	0.1657	0.1751	0.0657	0.0495	0.0657	0.0495

Table A.7: Measures of uniformity in C^2 from the Halton and Hammersley
Measures of uniformity in C^2 from the Halton and Hammersley methods

Halton and Hammersley methods in C^3										
	rmsd		*ad*		*md*		STRD		MSTRD	
n	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.
10	0.3008	0.2750	0.2790	0.2587	0.7128	0.5713	0.3194	0.2232	0.3573	0.2448
11	0.2952	0.2621	0.2729	0.2474	0.7128	0.5440	0.3194	0.2152	0.3703	0.2277
12	0.2879	0.2566	0.2649	0.2418	0.7128	0.5397	0.3194	0.2228	0.3573	0.2448
13	0.2661	0.2514	0.2472	0.2365	0.6210	0.5055	0.3194	0.2292	0.3194	0.2292
14	0.2515	0.2428	0.2350	0.2288	0.5981	0.4918	0.2479	0.2037	0.2479	0.2165
15	0.2421	0.2364	0.2268	0.2230	0.5650	0.4807	0.2173	0.1897	0.2677	0.2168
16	0.2389	0.2303	0.2238	0.2176	0.5650	0.4715	0.2337	0.1857	0.2346	0.1857
17	0.2314	0.2284	0.2170	0.2154	0.5650	0.4638	0.1969	0.1720	0.2129	0.1720
18	0.2273	0.2230	0.2130	0.2106	0.5650	0.4574	0.1936	0.1676	0.2149	0.1676
19	0.2216	0.2198	0.2080	0.2074	0.5594	0.4542	0.1614	0.1599	0.1710	0.1617
20	0.2150	0.2156	0.2023	0.2032	0.5594	0.4689	0.1658	0.1600	0.1815	0.1600
21	0.2118	0.2117	0.1989	0.1993	0.5594	0.4808	0.1801	0.1424	0.1911	0.1517
22	0.2086	0.2074	0.1956	0.1952	0.5594	0.4885	0.1885	0.1388	0.1885	0.1388
23	0.1978	0.2011	0.1875	0.1902	0.4119	0.4373	0.1614	0.1344	0.1614	0.1344
24	0.1962	0.1962	0.1858	0.1859	0.4119	0.4348	0.1241	0.1199	0.1420	0.1296
25	0.1955	0.1939	0.1851	0.1837	0.4119	0.4326	0.1358	0.1174	0.1421	0.1235
26	0.1929	0.1919	0.1826	0.1819	0.4119	0.4307	0.1333	0.1192	0.1585	0.1192
27	0.1915	0.1897	0.1812	0.1799	0.4119	0.4290	0.1337	0.1099	0.1658	0.1099
28	0.1900	0.1882	0.1794	0.1783	0.4119	0.4275	0.1221	0.1040	0.1530	0.1058
29	0.1880	0.1869	0.1773	0.1767	0.4119	0.4262	0.1143	0.0914	0.1333	0.0990
30	0.1863	0.1840	0.1758	0.1739	0.4119	0.4251	0.1197	0.0976	0.1439	0.0997
31	0.1852	0.1823	0.1745	0.1722	0.4119	0.4240	0.1157	0.1010	0.1600	0.1010
32	0.1843	0.1792	0.1736	0.1696	0.4119	0.4231	0.1232	0.1034	0.1496	0.1034
33	0.1823	0.1782	0.1714	0.1685	0.4119	0.4223	0.1150	0.0885	0.1346	0.1004
34	0.1799	0.1761	0.1693	0.1666	0.4119	0.4175	0.1186	0.0806	0.1358	0.1027
35	0.1781	0.1737	0.1677	0.1644	0.4119	0.4201	0.1001	0.0800	0.1434	0.0930
36	0.1758	0.1713	0.1658	0.1623	0.4119	0.4074	0.1104	0.0861	0.1505	0.0895
37	0.1746	0.1699	0.1646	0.1610	0.4119	0.4029	0.1202	0.0841	0.1302	0.0947
38	0.1734	0.1685	0.1632	0.1596	0.4119	0.3987	0.1294	0.0765	0.1294	0.0892
39	0.1699	0.1664	0.1603	0.1578	0.3630	0.3949	0.1126	0.0753	0.1184	0.0911
40	0.1690	0.1652	0.1594	0.1566	0.3630	0.3913	0.1215	0.0778	0.1255	0.0911
41	0.1671	0.1644	0.1575	0.1557	0.3630	0.3880	0.1157	0.0780	0.1206	0.0978
42	0.1664	0.1631	0.1568	0.1542	0.3630	0.3849	0.1239	0.0790	0.1303	0.0834
43	0.1655	0.1610	0.1558	0.1522	0.3630	0.3820	0.1172	0.0782	0.1172	0.0819
44	0.1628	0.1589	0.1535	0.1503	0.3252	0.3793	0.1008	0.0783	0.1033	0.0804
45	0.1616	0.1570	0.1523	0.1485	0.3252	0.3768	0.1074	0.0704	0.1074	0.0866
46	0.1606	0.1558	0.1513	0.1473	0.3252	0.3745	0.1137	0.0727	0.1137	0.0747
47	0.1594	0.1547	0.1501	0.1463	0.3252	0.3723	0.0984	0.0713	0.0984	0.0741
48	0.1582	0.1532	0.1490	0.1449	0.3252	0.3702	0.1030	0.0709	0.1030	0.0709
49	0.1570	0.1526	0.1478	0.1443	0.3252	0.3683	0.0899	0.0626	0.0910	0.0766
50	0.1564	0.1521	0.1472	0.1437	0.3252	0.3665	0.0826	0.0693	0.0826	0.0730

Table A.8: Measures of uniformity in C^3 from the Halton and Hammersley
Measures of uniformity in C^3 from the Halton and Hammersley methods

Halton and Hammersley methods in C^4						
	rmsd		*ad*		*md*	
n	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.
10	0.4163	0.3898	0.3928	0.3720	1.0223	0.8518
11	0.3970	0.3831	0.3763	0.3651	0.8471	0.8446
12	0.3920	0.3768	0.3705	0.3585	0.8471	0.8165
13	0.3681	0.3652	0.3504	0.3480	0.7744	0.7978
14	0.3587	0.3569	0.3421	0.3402	0.7013	0.7770
15	0.3544	0.3499	0.3377	0.3340	0.7013	0.7600
16	0.3524	0.3432	0.3357	0.3283	0.7013	0.7458
17	0.3450	0.3391	0.3290	0.3244	0.6932	0.7339
18	0.3419	0.3343	0.3257	0.3198	0.6932	0.7282
19	0.3370	0.3307	0.3211	0.3165	0.6932	0.7277
20	0.3284	0.3271	0.3135	0.3130	0.6578	0.7160
21	0.3268	0.3210	0.3117	0.3074	0.6578	0.7014
22	0.3229	0.3167	0.3075	0.3031	0.6578	0.6927
23	0.3156	0.3119	0.3009	0.2985	0.6578	0.6841
24	0.3142	0.3089	0.2996	0.2956	0.6578	0.6780
25	0.3134	0.3043	0.2987	0.2914	0.6578	0.6844
26	0.3116	0.3025	0.2968	0.2894	0.6578	0.6917
27	0.3070	0.3012	0.2929	0.2880	0.6578	0.6996
28	0.3033	0.2989	0.2895	0.2857	0.6578	0.6739
29	0.3002	0.2967	0.2864	0.2834	0.6578	0.6808
30	0.2984	0.2929	0.2846	0.2801	0.6578	0.6882
31	0.2965	0.2900	0.2827	0.2773	0.6578	0.6961
32	0.2934	0.2867	0.2800	0.2746	0.2800	0.6023
33	0.2910	0.2836	0.2774	0.2717	0.6556	0.5958
34	0.2857	0.2800	0.2731	0.2685	0.6107	0.5898
35	0.2811	0.2771	0.2691	0.2659	0.6006	0.5842
36	0.2783	0.2743	0.2666	0.2635	0.6006	0.5792
37	0.2766	0.2721	0.2647	0.2614	0.6006	0.5745
38	0.2731	0.2698	0.2614	0.2592	0.6006	0.5701
39	0.2713	0.2663	0.2596	0.2561	0.6006	0.5629
40	0.2699	0.2637	0.2582	0.2538	0.6006	0.5496
41	0.2673	0.2611	0.2558	0.2514	0.6006	0.5447
42	0.2667	0.2596	0.2550	0.2498	0.6006	0.5446
43	0.2654	0.2577	0.2537	0.2479	0.6006	0.5446
44	0.2634	0.2557	0.2519	0.2460	0.6006	0.5445
45	0.2612	0.2534	0.2497	0.2439	0.6006	0.5376
46	0.2595	0.2519	0.2479	0.2424	0.6006	0.5301
47	0.2567	0.2506	0.2456	0.2411	0.5737	0.5229
48	0.2554	0.2486	0.2445	0.2393	0.5737	0.5161
49	0.2544	0.2472	0.2436	0.2380	0.5737	0.5096
50	0.2512	0.2453	0.2410	0.2363	0.5396	0.5034

Table A.9: Measures of uniformity in C^4 from the Halton and Hammersley
Measures of uniformity in C^4 from the Halton and Hammersley methods

APPENDIX B

TABLES IN B^S FOR $S = 2, 3, 4$ DESIGN VARIABLES

glp method in B^2								
	$rmsd$		ad		md		$D(\mathcal{P}_n)$	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3)	0.2627	(1,3)	0.2401	(1,7)	0.5910	(1,7)	0.1683
11	(1,2)	0.2444	(1,3)	0.2243	(1,3)	0.5428	(1,5)	0.1515
12	(1,5)	0.2321	(1,5)	0.2136	(1,5)	0.4866	(1,5)	0.1346
13	(1,5)	0.2227	(1,5)	0.2041	(1,2)	0.4703	(1,8)	0.1270
14	(1,11)	0.2155	(1,11)	0.1981	(1,11)	0.4706	(1,9)	0.1199
15	(1,2)	0.2087	(1,2)	0.1932	(1,2)	0.4090	(1,11)	0.1159
16	(1,7)	0.1987	(1,7)	0.1839	(1,7)	0.4169	(1,7)	0.1034
17	(1,5)	0.1909	(1,5)	0.1769	(1,2)	0.4107	(1,11)	0.1015
18	(1,13)	0.1857	(1,13)	0.1718	(1,13)	0.3993	(1,13)	0.0896
19	(1,7)	0.1779	(1,7)	0.1646	(1,7)	0.3789	(1,8)	0.0913
20	(1,9)	0.1755	(1,9)	0.1618	(1,9)	0.3627	(1,9)	0.0842
21	(1,13)	0.1702	(1,13)	0.1571	(1,2)	0.3656	(1,13)	0.0826
22	(1,17)	0.1653	(1,15)	0.1533	(1,17)	0.3654	(1,17)	0.0786
23	(1,18)	0.1607	(1,18)	0.1482	(1,2)	0.3414	(1,18)	0.0764
24	(1,13)	0.1609	(1,13)	0.1485	(1,13)	0.3363	(1,11)	0.0705
25	(1,9)	0.1543	(1,9)	0.1424	(1,2)	0.3358	(1,16)	0.0639
26	(1,19)	0.1514	(1,19)	0.1402	(1,19)	0.3450	(1,11)	0.0670
27	(1,5)	0.1488	(1,5)	0.1378	(1,2)	0.3371	(1,8)	0.0679
28	(1,5)	0.1452	(1,5)	0.1342	(1,15)	0.3176	(1,23)	0.0629
29	(1,8)	0.1429	(1,8)	0.1323	(1,12)	0.3173	(1,17)	0.0608
30	(1,19)	0.1404	(1,19)	0.1296	(1,19)	0.3141	(1,19)	0.0569
31	(1,13)	0.1381	(1,13)	0.1280	(1,12)	0.3015	(1,22)	0.0535
32	(1,5)	0.1354	(1,5)	0.1250	(1,3)	0.2991	(1,19)	0.0560
33	(1,26)	0.1340	(1,26)	0.1241	(1,23)	0.3000	(1,19)	0.0484
34	(1,21)	0.1306	(1,21)	0.1212	(1,13)	0.2766	(1,25)	0.0498
35	(1,8)	0.1309	(1,8)	0.1214	(1,16)	0.2736	(1,11)	0.0516
36	(1,11)	0.1288	(1,11)	0.1191	(1,11)	0.2720	(1,23)	0.0510
37	(1,29)	0.1250	(1,29)	0.1159	(1,17)	0.2630	(1,29)	0.0459
38	(1,5)	0.1233	(1,5)	0.1142	(1,5)	0.2718	(1,31)	0.0486
39	(1,25)	0.1230	(1,25)	0.1139	(1,14)	0.2583	(1,25)	0.0429
40	(1,7)	0.1224	(1,7)	0.1133	(1,7)	0.2734	(1,29)	0.0443
41	(1,11)	0.1185	(1,11)	0.1099	(1,11)	0.2468	(1,30)	0.0409
42	(1,5)	0.1193	(1,5)	0.1106	(1,5)	0.2531	(1,19)	0.0430
43	(1,5)	0.1165	(1,5)	0.1077	(1,8)	0.2375	(1,19)	0.0412
44	(1,13)	0.1137	(1,13)	0.1058	(1,13)	0.2465	(1,25)	0.0420
45	(1,8)	0.1124	(1,8)	0.1044	(1,8)	0.2342	(1,19)	0.0408
46	(1,27)	0.1118	(1,27)	0.1039	(1,27)	0.2332	(1,29)	0.0411
47	(1,13)	0.1097	(1,34)	0.1020	(1,13)	0.2357	(1,18)	0.0401
48	(1,5)	0.1106	(1,5)	0.1021	(1,17)	0.2315	(1,31)	0.0358
49	(1,31)	0.1071	(1,31)	0.0997	(1,18)	0.2215	(1,22)	0.0364
50	(1,21)	0.1065	(1,21)	0.0992	(1,21)	0.2188	(1,37)	0.0391

Table B.1: Best generators in B^2 from the glp method

glp method in B^3								
	rmsd		*ad*		*md*		$D(\mathcal{P}_n)$	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,7,9)	0.4713	(1,7,9)	0.4428	(1,3,9)	1.0599	(1,7,9)	0.3159
11	(1,5,7)	0.4384	(1,5,7)	0.4131	(1,8,9)	0.9515	(1,2,5)	0.2305
12	(1,7,11)	0.4340	(1,7,11)	0.4063	(1,7,11)	0.9594	(1,7,11)	0.2694
13	(1,4,6)	0.4100	(1,4,6)	0.3865	(1,4,6)	0.8815	(1,4,6)	0.2188
14	(1,3,9)	0.4195	(1,3,9)	0.3907	(1,3,9)	0.9358	(1,5,11)	0.2495
15	(1,11,14)	0.4066	(1,11,14)	0.3788	(1,2,11)	0.8967	(1,4,14)	0.2235
16	(1,3,9)	0.3994	(1,3,9)	0.3719	(1,3,9)	0.8803	(1,3,11)	0.2448
17	(1,4,11)	0.3854	(1,4,11)	0.3608	(1,3,8)	0.8315	(1,2,8)	0.2117
18	(1,11,13)	0.3828	(1,11,13)	0.3572	(1,13,17)	0.8326	(1,5,11)	0.2205
19	(1,6,9)	0.3689	(1,6,9)	0.3447	(1,6,9)	0.8260	(1,2,9)	0.1991
20	(1,3,11)	0.3662	(1,3,11)	0.3421	(1,3,11)	0.8213	(1,11,19)	0.2201
21	(1,4,10)	0.3535	(1,4,10)	0.3332	(1,2,5)	0.8215	(1,4,10)	0.2044
22	(1,7,13)	0.3601	(1,7,13)	0.3351	(1,3,13)	0.7684	(1,9,17)	0.2156
23	(1,3,7)	0.3495	(1,3,7)	0.3270	(1,11,13)	0.7963	(1,3,7)	0.1928
24	(1,5,11)	0.3476	(1,5,11)	0.3234	(1,5,11)	0.7917	(1,5,11)	0.1843
25	(1,8,12)	0.3413	(1,8,12)	0.3183	(1,8,12)	0.7709	(1,6,16)	0.1865
26	(1,3,19)	0.3368	(1,3,7)	0.3140	(1,3,19)	0.7635	(1,9,15)	0.1911
27	(1,4,13)	0.3313	(1,4,13)	0.3097	(1,5,13)	0.7175	(1,8,25)	0.1863
28	(1,9,11)	0.3295	(1,9,11)	0.3073	(1,5,13)	0.7371	(1,5,19)	0.1951
29	(1,4,14)	0.3207	(1,4,14)	0.3014	(1,3,4)	0.7387	(1,8,17)	0.1817
30	(1,7,17)	0.3231	(1,17,19)	0.3007	(1,19,29)	0.7634	(1,7,17)	0.1788
31	(1,6,15)	0.3151	(1,6,15)	0.2964	(1,6,15)	0.7218	(1,3,20)	0.1747
32	(1,9,15)	0.3120	(1,9,15)	0.2915	(1,5,15)	0.7243	(1,9,17)	0.1778
33	(1,4,26)	0.3138	(1,4,26)	0.2925	(1,14,23)	0.7587	(1,2,19)	0.1812
34	(1,13,25)	0.3102	(1,11,27)	0.2887	(1,3,25)	0.7149	(1,15,21)	0.1734
35	(1,3,11)	0.3068	(1,6,13)	0.2856	(1,3,26)	0.7028	(1,11,19)	0.1770
36	(1,5,19)	0.3002	(1,11,19)	0.2806	(1,5,17)	0.7001	(1,5,23)	0.1778
37	(1,4,7)	0.3005	(1,4,27)	0.2787	(1,2,6)	0.6853	(1,11,35)	0.1689
38	(1,5,29)	0.3001	(1,27,31)	0.2788	(1,3,21)	0.7068	(1,5,29)	0.1696
39	(1,11,37)	0.2961	(1,5,8)	0.2759	(1,7,19)	0.6799	(1,4,25)	0.1702
40	(1,13,21)	0.2905	(1,13,21)	0.2706	(1,3,7)	0.6937	(1,9,19)	0.1687
41	(1,6,8)	0.2870	(1,6,8)	0.2676	(1,6,20)	0.6549	(1,5,28)	0.1643
42	(1,19,25)	0.2875	(1,19,25)	0.2680	(1,29,41)	0.6712	(1,11,29)	0.1660
43	(1,6,21)	0.2868	(1,4,10)	0.2656	(1,2,36)	0.6681	(1,8,12)	0.1581
44	(1,13,23)	0.2808	(1,13,23)	0.2626	(1,7,23)	0.6446	(1,31,35)	0.1687
45	(1,4,32)	0.2818	(1,4,32)	0.2622	(1,17,22)	0.6619	(1,2,32)	0.1631
46	(1,7,9)	0.2787	(1,25,27)	0.2595	(1,17,45)	0.6621	(1,25,27)	0.1589
47	(1,11,45)	0.2777	(1,10,25)	0.2576	(1,2,34)	0.6391	(1,17,26)	0.1624
48	(1,5,23)	0.2762	(1,5,23)	0.2571	(1,7,23)	0.6470	(1,5,37)	0.1686
49	(1,6,32)	0.2715	(1,6,32)	0.2521	(1,2,19)	0.6457	(1,3,32)	0.1584
50	(1,9,11)	0.2735	(1,9,11)	0.2537	(1,19,49)	0.6346	(1,7,23)	0.1566

Table B.2: Best generators in B^3 from the glp method

glp method in B^4						
	$rmsd$		ad		md	
n	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3,7,9)	0.5416	(1,3,7,9)	0.5185	(1,3,7,9)	1.0882
11	(1,2,7,8)	0.5101	(1,2,5,7)	0.4902	(1,3,4,5)	0.9062
12	(1,5,7,11)	0.5154	(1,5,7,11)	0.4931	(1,5,7,11)	1.0870
13	(1,2,4,7)	0.4840	(1,2,4,7)	0.4654	(1,2,3,7)	0.8610
14	(1,3,5,13)	0.4792	(1,3,5,13)	0.4612	(1,3,5,13)	0.8688
15	(1,2,4,7)	0.4678	(1,2,4,7)	0.4501	(1,2,7,14)	0.8806
16	(1,3,7,15)	0.4587	(1,3,9,11)	0.4413	(1,5,9,13)	0.8315
17	(1,3,4,11)	0.4459	(1,3,4,11)	0.4293	(1,2,10,14)	0.8159
18	(1,7,13,17)	0.4504	(1,7,13,17)	0.4326	(1,7,13,17)	0.8449
19	(1,2,7,11)	0.4329	(1,2,7,11)	0.4165	(1,2,12,16)	0.7937
20	(1,3,9,13)	0.4361	(1,3,9,13)	0.4181	(1,3,7,11)	0.7910
21	(1,2,4,5)	0.4256	(1,2,4,5)	0.4087	(1,2,5,10)	0.7944
22	(1,3,7,9)	0.4194	(1,3,7,9)	0.4027	(1,3,7,21)	0.7799
23	(1,2,11,15)	0.4093	(1,2,11,15)	0.3940	(1,7,11,13)	0.7679
24	(1,5,13,17)	0.4067	(1,5,13,17)	0.3917	(1,7,13,23)	0.7633
25	(1,2,12,17)	0.4022	(1,3,12,18)	0.3874	(1,4,6,14)	0.7461
26	(1,3,17,21)	0.3984	(1,3,17,21)	0.3828	(1,3,5,7)	0.7655
27	(1,4,5,7)	0.3938	(1,4,5,7)	0.3789	(1,8,20,25)	0.7309
28	(1,3,13,17)	0.3926	(1,3,13,17)	0.3774	(1,9,11,25)	0.7455
29	(1,3,19,25)	0.3848	(1,3,5,11)	0.3702	(1,8,12,18)	0.7175
30	(1,17,19,23)	0.3814	(1,17,19,23)	0.3673	(1,17,19,23)	0.7229
31	(1,2,20,23)	0.3768	(1,2,20,23)	0.3626	(1,12,13,14)	0.7020
32	(1,3,17,21)	0.3765	(1,3,17,21)	0.3620	(1,3,5,13)	0.7235
33	(1,2,13,25)	0.3699	(1,2,13,25)	0.3565	(1,8,13,31)	0.6993
34	(1,3,5,11)	0.3684	(1,3,5,11)	0.3546	(1,13,25,33)	0.6961
35	(1,2,19,23)	0.3649	(1,2,19,23)	0.3511	(1,4,6,19)	0.6956
36	(1,5,19,23)	0.3640	(1,5,19,23)	0.3502	(1,5,19,23)	0.7189
37	(1,4,6,15)	0.3602	(1,4,6,22)	0.3468	(1,17,29,30)	0.6712
38	(1,3,21,23)	0.3589	(1,3,21,23)	0.3451	(1,5,7,29)	0.6733
39	(1,2,28,31)	0.3534	(1,2,28,31)	0.3399	(1,4,10,17)	0.6799
40	(1,7,9,17)	0.3533	(1,7,9,17)	0.3400	(1,3,11,21)	0.7028
41	(1,4,10,13)	0.3492	(1,4,10,13)	0.3364	(1,6,10,32)	0.6581
42	(1,13,25,31)	0.3488	(1,13,25,31)	0.3360	(1,19,25,37)	0.6525
43	(1,4,6,17)	0.3453	(1,4,6,17)	0.3324	(1,19,20,30)	0.6501
44	(1,3,7,31)	0.3450	(1,3,7,31)	0.3311	(1,21,29,31)	0.6584
45	(1,2,11,19)	0.3409	(1,2,11,19)	0.3280	(1,8,28,38)	0.6624
46	(1,3,17,35)	0.3387	(1,3,17,35)	0.3260	(1,21,27,41)	0.6519
47	(1,6,28,39)	0.3359	(1,6,28,39)	0.3235	(1,6,28,43)	0.6359
48	(1,11,13,19)	0.3361	(1,11,13,19)	0.3230	(1,11,17,47)	0.6564
49	(1,12,15,29)	0.3335	(1,2,13,27)	0.3211	(1,12,15,20)	0.6185
50	(1,7,13,41)	0.3327	(1,7,13,41)	0.3204	(1,3,19,37)	0.6503

Table B.3: Best generators in B^4 from the glp method

CF method in B^2								
	$rmsd$		ad		md		$D(\mathcal{P}_n)$	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	P=7	0.2670	P=7	0.2445	P=7	0.5305	P=7	0.3152
11	P=7	0.2642	P=7	0.2410	P=7	0.5305	P=11	0.2076
12	P=7	0.2639	P=7	0.2405	P=7	0.5305	P=11	0.1938
13	P=13	0.2354	P=13	0.2144	P=13	0.5256	P=13	0.1426
14	P=13	0.2291	P=13	0.2080	P=13	0.5256	P=13	0.1367
15	P=13	0.2169	P=13	0.1978	P=13	0.5251	P=13	0.1396
16	P=13	0.2118	P=13	0.1926	P=13	0.5251	P=13	0.1395
17	P=13	0.2048	P=13	0.1857	P=17	0.4993	P=13	0.1689
18	P=13	0.1986	P=13	0.1799	P=7	0.4565	P=13	0.1395
19	P=13	0.1863	P=13	0.1707	P=13	0.3991	P=19	0.1127
20	P=13	0.1776	P=13	0.1636	P=13	0.3874	P=19	0.0923
21	P=13	0.1713	P=13	0.1581	P=13	0.3874	P=17	0.0818
22	P=13	0.1664	P=13	0.1542	P=13	0.3874	P=13	0.0750
23	P=23	0.1654	P=23	0.1524	P=23	0.3640	P=17	0.0744
24	P=23	0.1606	P=23	0.1484	P=23	0.3318	P=17	0.0820
25	P=23	0.1578	P=23	0.1457	P=23	0.3318	P=17	0.0905
26	P=23	0.1559	P=23	0.1436	P=23	0.3318	P=17	0.0982
27	P=23	0.1555	P=23	0.1432	P=23	0.3318	P=23	0.0824
28	P=23	0.1548	P=23	0.1425	P=23	0.3318	P=23	0.0731
29	P=29	0.1478	P=29	0.1365	P=29	0.3025	P=23	0.0822
30	P=29	0.1461	P=29	0.1347	P=29	0.3025	P=13	0.0865
31	P=29	0.1443	P=29	0.1329	P=29	0.3025	P=13	0.0707
32	P=29	0.1422	P=29	0.1307	P=29	0.3025	P=19	0.0772
33	P=29	0.1408	P=29	0.1292	P=29	0.3025	P=13	0.0849
34	P=29	0.1391	P=29	0.1276	P=29	0.3025	P=13	0.0892
35	P=29	0.1374	P=29	0.1259	P=29	0.3025	P=13	0.0847
36	P=29	0.1356	P=29	0.1242	P=29	0.3025	P=19	0.0833
37	P=29	0.1328	P=29	0.1219	P=29	0.3025	P=19	0.0701
38	P=29	0.1303	P=29	0.1197	P=29	0.3025	P=19	0.0641
39	P=29	0.1282	P=29	0.1178	P=29	0.3025	P=23	0.0653
40	P=29	0.1268	P=29	0.1165	P=29	0.3025	P=19	0.0629
41	P=29	0.1257	P=29	0.1152	P=29	0.3025	P=23	0.0670
42	P=41	0.1249	P=41	0.1147	P=41	0.2805	P=23	0.0669
43	P=41	0.1240	P=41	0.1139	P=41	0.2805	P=19	0.0678
44	P=41	0.1240	P=41	0.1138	P=23	0.2799	P=13	0.0626
45	P=29	0.1234	P=29	0.1128	P=23	0.2799	P=41	0.0623
46	P=43	0.1218	P=43	0.1113	P=43	0.2694	P=17	0.0526
47	P=29	0.1205	P=29	0.1103	P=43	0.2694	P=17	0.0563
48	P=29	0.1179	P=29	0.1082	P=29	0.2653	P=19	0.0607
49	P=29	0.1168	P=29	0.1071	P=29	0.2653	P=19	0.0604
50	P=29	0.1161	P=29	0.1065	P=29	0.2653	P=19	0.0546

Table B.4: Best generators in B^2 from the CF method

CF Method in B^3								
	$rmsd$		ad		md		$D(\mathcal{P}_n)$	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	-	-	-	-	-	-	-	-
11	P=11	0.5141	P=11	0.4793	P=11	1.0220	P=11	0.3680
12	P=11	0.5054	P=11	0.4701	P=11	1.0220	P=11	0.3743
13	P=13	0.4571	P=13	0.4265	P=13	0.9543	P=13	0.3217
14	P=13	0.4557	P=13	0.4249	P=13	0.9543	P=13	0.3102
15	P=13	0.4528	P=13	0.4213	P=13	0.9543	P=13	0.3057
16	P=13	0.4315	P=13	0.4026	P=13	0.9386	P=13	0.2516
17	P=13	0.4220	P=13	0.3940	P=13	0.9386	P=13	0.2510
18	P=17	0.4061	P=17	0.3780	P=17	0.9147	P=13	0.2443
19	P=17	0.3879	P=17	0.3620	P=19	0.8536	P=13	0.2489
20	P=17	0.3798	P=17	0.3560	P=19	0.8536	P=13	0.2542
21	P=17	0.3762	P=17	0.3499	P=19	0.8536	P=13	0.2378
22	P=17	0.3739	P=17	0.3478	P=19	0.8536	P=13	0.2211
23	P=17	0.3725	P=17	0.3460	P=13	0.8463	P=13	0.2316
24	P=17	0.3714	P=17	0.3444	P=19	0.8282	P=13	0.2307
25	P=17	0.3702	P=17	0.3438	P=19	0.8282	P=13	0.2391
26	P=17	0.3690	P=17	0.3426	P=19	0.8282	P=19	0.2291
27	P=17	0.3672	P=17	0.3407	P=19	0.8282	P=19	0.2177
28	P=17	0.3648	P=17	0.3381	P=19	0.8282	P=17	0.2356
29	P=17	0.3645	P=17	0.3368	P=29	0.8129	P=17	0.2418
30	P=17	0.3642	P=17	0.3356	P=19	0.8035	P=17	0.2476
31	P=31	0.3630	P=17	0.3352	P=19	0.8035	P=17	0.2412
32	P=23	0.3544	P=23	0.3283	P=19	0.8035	P=17	0.2510
33	P=23	0.3499	P=23	0.3238	P=23	0.7896	P=19	0.2473
34	P=23	0.3492	P=23	0.3227	P=23	0.7896	P=13	0.2332
35	P=23	0.3460	P=23	0.3192	P=23	0.7896	P=19	0.2282
36	P=29	0.3411	P=29	0.3161	P=29	0.7190	P=19	0.2195
37	P=37	0.3325	P=23	0.3081	P=37	0.6955	P=13	0.2374
38	P=23	0.3322	P=23	0.3077	P=23	0.6955	P=17	0.2302
39	P=23	0.3310	P=23	0.3064	P=37	0.6955	P=17	0.2206
40	P=23	0.3307	P=23	0.3061	P=37	0.6955	P=17	0.2132
41	P=41	0.3263	P=41	0.3023	P=41	0.6875	P=41	0.1958
42	P=41	0.3240	P=41	0.3003	P=41	0.6875	P=41	0.1899
43	P=41	0.3225	P=41	0.2989	P=41	0.6875	P=17	0.1944
44	P=41	0.3224	P=41	0.2987	P=41	0.6875	P=17	0.1885
45	P=41	0.3222	P=41	0.2986	P=41	0.6875	P=43	0.1937
46	P=43	0.3214	P=43	0.2973	P=41	0.6875	P=43	0.1879
47	P=43	0.3198	P=23	0.2952	P=41	0.6875	P=11	0.1995
48	P=23	0.3195	P=23	0.2946	P=41	0.6875	P=31	0.1933
49	P=23	0.3162	P=23	0.2915	P=41	0.6875	P=17	0.1994
50	P=23	0.3161	P=23	0.2915	P=41	0.6875	P=17	0.2023

Table B.5: Best generators in B^3 from the CF method

CF method in B^4						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
10	-	-	-	-	-	-
11	P=11	0.5715	P=11	0.5413	P=11	1.0645
12	P=11	0.5634	P=11	0.5330	P=11	1.0645
13	P=13	0.5073	P=13	0.4833	P=13	1.0097
14	P=13	0.4958	P=13	0.4731	P=13	1.0097
15	P=13	0.4917	P=13	0.4690	P=13	1.0097
16	P=13	0.4836	P=13	0.4612	P=13	1.0097
17	P=13	0.4683	P=13	0.4479	P=13	0.9412
18	P=13	0.4603	P=13	0.4407	P=13	0.9412
19	P=17	0.4493	P=17	0.4307	P=17	0.9387
20	P=17	0.4420	P=17	0.4233	P=17	0.9387
21	P=17	0.4332	P=17	0.4159	P=17	0.8914
22	P=17	0.4291	P=17	0.4114	P=17	0.8914
23	P=17	0.4272	P=17	0.4094	P=23	0.8630
24	P=17	0.4230	P=17	0.4053	P=23	0.8394
25	P=17	0.4197	P=17	0.4019	P=23	0.8394
26	P=17	0.4124	P=17	0.3952	P=23	0.8394
27	P=17	0.4078	P=17	0.3904	P=23	0.8394
28	P=17	0.4030	P=17	0.3856	P=23	0.8394
29	P=17	0.4002	P=17	0.3825	P=23	0.8394
30	P=17	0.3995	P=17	0.3818	P=23	0.8291
31	P=17	0.3991	P=17	0.3812	P=23	0.8291
32	P=13	0.3947	P=13	0.3771	P=23	0.8291
33	P=13	0.3927	P=13	0.3750	P=23	0.8291
34	P=13	0.3890	P=13	0.3712	P=23	0.8291
35	P=13	0.3876	P=13	0.3697	P=23	0.7935
36	P=17	0.3842	P=17	0.3672	P=31	0.7706
37	P=11	0.3822	P=11	0.3644	P=31	0.7227
38	P=13	0.3780	P=13	0.3612	P=31	0.7227
39	P=31	0.3714	P=31	0.3559	P=31	0.7227
40	P=31	0.3669	P=31	0.3516	P=31	0.7227
41	P=31	0.3635	P=31	0.3484	P=31	0.7227
42	P=31	0.3589	P=31	0.3445	P=31	0.6761
43	P=31	0.3570	P=31	0.3428	P=31	0.6761
44	P=31	0.3531	P=31	0.3390	P=31	0.6761
45	P=31	0.3489	P=31	0.3350	P=31	0.6761
46	P=31	0.3458	P=31	0.3321	P=31	0.6761
47	P=31	0.3433	P=31	0.3295	P=31	0.6761
48	P=31	0.3414	P=31	0.3278	P=31	0.6761
49	P=31	0.3404	P=31	0.3269	P=31	0.6761
50	P=31	0.3395	P=31	0.3259	P=31	0.6761

Table B.6: Best generators in B^4 from the CF method

Halton and Hammersley methods in B^2								
	rmsd		*ad*		*md*		$D(\mathcal{P}_n)$	
n	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.
10	0.2816	0.2798	0.2540	0.2531	0.7168	0.5817	0.2567	0.1960
11	0.2739	0.2713	0.2459	0.2447	0.7168	0.5731	0.2214	0.1617
12	0.2672	0.2563	0.2380	0.2315	0.7168	0.5562	0.2401	0.1644
13	0.2510	0.2440	0.2234	0.2208	0.7168	0.5528	0.2354	0.1545
14	0.2366	0.2287	0.2128	0.2078	0.6305	0.5448	0.2036	0.1360
15	0.2316	0.2129	0.2078	0.1955	0.6305	0.4244	0.1667	0.1302
16	0.2285	0.2067	0.2039	0.1896	0.6305	0.4210	0.1709	0.1263
17	0.2039	0.2009	0.1865	0.1839	0.4522	0.4326	0.1206	0.1170
18	0.2011	0.1960	0.1831	0.1790	0.4522	0.4374	0.1355	0.1105
19	0.1922	0.1918	0.1750	0.1748	0.4445	0.4429	0.1295	0.1074
20	0.1897	0.1866	0.1722	0.1702	0.4445	0.4220	0.1334	0.0993
21	0.1837	0.1818	0.1668	0.1658	0.4445	0.4115	0.1382	0.0983
22	0.1786	0.1781	0.1621	0.1621	0.4445	0.3965	0.1425	0.1004
23	0.1743	0.1742	0.1585	0.1587	0.4445	0.3725	0.1190	0.1030
24	0.1700	0.1698	0.1545	0.1546	0.4445	0.3674	0.1208	0.0961
25	0.1597	0.1659	0.1468	0.1511	0.3777	0.3670	0.1224	0.0873
26	0.1560	0.1624	0.1434	0.1480	0.3777	0.3674	0.1057	0.0818
27	0.1520	0.1595	0.1402	0.1455	0.3166	0.3781	0.0901	0.0799
28	0.1514	0.1556	0.1394	0.1421	0.3166	0.3509	0.0980	0.0772
29	0.1499	0.1522	0.1379	0.1392	0.3166	0.3554	0.1054	0.0839
30	0.1484	0.1478	0.1361	0.1355	0.3166	0.3632	0.1132	0.0691
31	0.1474	0.1421	0.1350	0.1311	0.3166	0.2941	0.0885	0.0700
32	0.1466	0.1398	0.1340	0.1288	0.3166	0.2875	0.0867	0.0682
33	0.1455	0.1374	0.1329	0.1266	0.3166	0.2811	0.0923	0.0707
34	0.1427	0.1354	0.1303	0.1248	0.3166	0.2750	0.0977	0.0626
35	0.1407	0.1336	0.1283	0.1231	0.3166	0.2811	0.0742	0.0633
36	0.1386	0.1317	0.1264	0.1214	0.3166	0.2662	0.0913	0.0641
37	0.1370	0.1295	0.1248	0.1195	0.3166	0.2633	0.0956	0.0607
38	0.1352	0.1279	0.1230	0.1179	0.3166	0.2637	0.0903	0.0650
39	0.1340	0.1263	0.1218	0.1164	0.3166	0.2690	0.0833	0.0539
40	0.1335	0.1245	0.1210	0.1148	0.3166	0.2706	0.0878	0.0584
41	0.1326	0.1225	0.1202	0.1131	0.3166	0.2726	0.0850	0.0606
42	0.1311	0.1209	0.1187	0.1117	0.3166	0.2683	0.0873	0.0504
43	0.1296	0.1197	0.1173	0.1106	0.3166	0.2700	0.0896	0.0565
44	0.1280	0.1180	0.1158	0.1092	0.3166	0.2490	0.0738	0.0472
45	0.1266	0.1169	0.1145	0.1081	0.3166	0.2501	0.0748	0.0526
46	0.1249	0.1153	0.1130	0.1068	0.3166	0.2533	0.0758	0.0508
47	0.1236	0.1139	0.1118	0.1055	0.3166	0.2398	0.0597	0.0479
48	0.1221	0.1131	0.1105	0.1047	0.3166	0.2380	0.0681	0.0480
49	0.1211	0.1123	0.1094	0.1039	0.3166	0.2361	0.0727	0.0461
50	0.1186	0.1115	0.1073	0.1031	0.3166	0.2339	0.0661	0.0484

Table B.7: Measures of uniformity in B^2 from the Halton and Hammersley
Measures of uniformity in B^3 from the Halton and Hammersley methods

Halton and Hammersley methods in B^3								
	$rmsd$		ad		md		$D(\mathcal{P}_n)$	
n	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.
10	0.5043	0.4869	0.4666	0.4558	1.1194	0.9989	0.3627	0.3241
11	0.4802	0.4726	0.4459	0.4410	1.0413	0.9967	0.3104	0.3229
12	0.4660	0.4602	0.4306	0.4285	1.0413	0.9948	0.3104	0.3277
13	0.4582	0.4495	0.4225	0.4170	1.0413	0.9929	0.3104	0.3142
14	0.4360	0.4280	0.4025	0.3985	1.0413	0.9484	0.2390	0.2751
15	0.4356	0.4093	0.4019	0.3836	1.0413	0.9136	0.2437	0.2611
16	0.4325	0.4045	0.3985	0.3781	1.0413	0.9128	0.2516	0.2518
17	0.4259	0.3954	0.3914	0.3689	1.0413	0.9123	0.2516	0.2564
18	0.4148	0.3903	0.3820	0.3635	0.9585	0.9118	0.2549	0.2828
19	0.4131	0.3876	0.3798	0.3610	0.9585	0.9114	0.2578	0.2646
20	0.3873	0.3833	0.3590	0.3556	0.8892	0.9105	0.2604	0.2683
21	0.3760	0.3739	0.3490	0.3473	0.8721	0.8725	0.2628	0.2241
22	0.3743	0.3655	0.3468	0.3400	0.8721	0.8705	0.2650	0.2250
23	0.3645	0.3617	0.3387	0.3349	0.8052	0.8697	0.2315	0.2178
24	0.3638	0.3599	0.3378	0.3327	0.8052	0.8702	0.2409	0.2227
25	0.3619	0.3447	0.3355	0.3217	0.8052	0.8630	0.2209	0.2084
26	0.3528	0.3414	0.3267	0.3194	0.8052	0.8585	0.2240	0.2174
27	0.3500	0.3376	0.3237	0.3144	0.8052	0.8543	0.2269	0.2028
28	0.3456	0.3352	0.3192	0.3117	0.7914	0.8505	0.2295	0.2098
29	0.3366	0.3323	0.3119	0.3096	0.7914	0.8469	0.2179	0.2164
30	0.3354	0.3297	0.3103	0.3071	0.7914	0.8436	0.2202	0.2225
31	0.3330	0.3273	0.3045	0.3078	0.8406	0.7914	0.2284	0.2223
32	0.3322	0.3257	0.3069	0.3025	0.7914	0.8378	0.2243	0.2168
33	0.3270	0.3223	0.2990	0.3023	0.8358	0.7619	0.2221	0.2262
34	0.3241	0.3205	0.2991	0.2954	0.7619	0.8340	0.2280	0.2273
35	0.3224	0.3198	0.2972	0.2946	0.7619	0.8316	0.2297	0.2321
36	0.3203	0.3176	0.2947	0.2922	0.7619	0.8294	0.2313	0.2366
37	0.3181	0.3128	0.2925	0.2891	0.7619	0.8256	0.2328	0.2140
38	0.3143	0.3103	0.2889	0.2863	0.7577	0.8218	0.2342	0.1994
39	0.3130	0.3084	0.2876	0.2842	0.7577	0.8208	0.2099	0.1915
40	0.3113	0.3076	0.2858	0.2832	0.7577	0.8198	0.2118	0.1922
41	0.3104	0.3069	0.2846	0.2822	0.7577	0.8189	0.1893	0.1912
42	0.3099	0.3041	0.2838	0.2795	0.7577	0.8181	0.1916	0.1952
43	0.3078	0.3018	0.2819	0.2773	0.7512	0.8173	0.1938	0.2002
44	0.3052	0.2997	0.2792	0.2750	0.7512	0.8166	0.1959	0.2049
45	0.3040	0.2993	0.2777	0.2745	0.7512	0.8159	0.1980	0.2094
46	0.2984	0.2985	0.2729	0.2735	0.7476	0.8153	0.1999	0.2074
47	0.2977	0.2966	0.2721	0.2714	0.7476	0.8148	0.2017	0.2103
48	0.2949	0.2949	0.2694	0.2696	0.7366	0.8142	0.2035	0.2131
49	0.2945	0.2941	0.2688	0.2686	0.7366	0.8137	0.2052	0.2158
50	0.2938	0.2914	0.2679	0.2659	0.7366	0.8133	0.2068	0.1984

Table B.8: Measures of uniformity in B^3 from the Halton and Hammersley
Measures of uniformity in B^3 from the Halton and Hammersley methods

Halton and Hammersley methods in B^4						
	rmsd		*ad*		*md*	
n	Hal.	Hamm.	Hal.	Hamm.	Hal.	Hamm.
10	0.5462	0.5619	0.5210	0.5360	1.0760	1.1364
11	0.5356	0.5482	0.5102	0.5217	1.0760	1.1242
12	0.5180	0.5359	0.4946	0.5098	1.0269	1.1136
13	0.4934	0.5121	0.4737	0.4886	0.9184	1.0141
14	0.4796	0.4954	0.4607	0.4736	0.9017	0.9833
15	0.4783	0.4878	0.4593	0.4657	0.9017	0.9776
16	0.4746	0.4796	0.4551	0.4583	0.9017	0.9725
17	0.4666	0.4616	0.4470	0.4427	0.9017	0.8717
18	0.4624	0.4567	0.4429	0.4378	0.9017	0.8698
19	0.4546	0.4431	0.4358	0.4256	0.9017	0.8582
20	0.4459	0.4361	0.4269	0.4192	0.9017	0.8525
21	0.4410	0.4292	0.4226	0.4123	0.9017	0.8512
22	0.4369	0.4215	0.4182	0.4052	0.9017	0.8500
23	0.4297	0.4152	0.4117	0.3991	0.9017	0.8489
24	0.4236	0.4096	0.4060	0.3942	0.9017	0.7817
25	0.4216	0.4063	0.4038	0.3908	0.9017	0.7787
26	0.4159	0.4001	0.3985	0.3851	0.9017	0.7760
27	0.4110	0.3964	0.3946	0.3813	0.8281	0.7734
28	0.4066	0.3930	0.3905	0.3781	0.7930	0.7708
29	0.4037	0.3892	0.3874	0.3743	0.7930	0.7677
30	0.3993	0.3855	0.3836	0.3706	0.7639	0.7647
31	0.3975	0.3829	0.3818	0.3680	0.7639	0.7619
32	0.3939	0.3807	0.3782	0.3658	0.7639	0.7600
33	0.3916	0.3784	0.3760	0.3634	0.7639	0.7602
34	0.3878	0.3762	0.3720	0.3612	0.7237	0.7605
35	0.3840	0.3732	0.3684	0.3582	0.7237	0.7588
36	0.3807	0.3701	0.3654	0.3555	0.7237	0.7567
37	0.3777	0.3675	0.3624	0.3528	0.7237	0.7548
38	0.3748	0.3645	0.3594	0.3500	0.7237	0.7530
39	0.3738	0.3623	0.3585	0.3477	0.7086	0.7512
40	0.3726	0.3603	0.3571	0.3459	0.7086	0.7496
41	0.3701	0.3580	0.3544	0.3437	0.7086	0.7481
42	0.3692	0.3568	0.3534	0.3424	0.7086	0.7466
43	0.3684	0.3555	0.3525	0.3409	0.7086	0.7452
44	0.3652	0.3531	0.3493	0.3387	0.7086	0.7439
45	0.3623	0.3512	0.3469	0.3368	0.7086	0.7426
46	0.3580	0.3482	0.3431	0.3340	0.7086	0.7414
47	0.3555	0.3460	0.3408	0.3318	0.7086	0.7403
48	0.3525	0.3437	0.3381	0.3298	0.6746	0.6860
49	0.3499	0.3425	0.3356	0.3286	0.6651	0.6844
50	0.3480	0.3412	0.3336	0.3273	0.6651	0.6829

Table B.9: Measures of uniformity in B^4 from the Halton and Hammersley
Measures of uniformity in B^4 from the Halton and Hammersley methods

APPENDIX C

TABLES IN S^Q FOR $Q = 3, 4, 5$ COMPONENT MIXTURE DESIGN

3 component mixture design from glp method										
	rmsd		*ad*		*md*		DSD		NDRD	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3)	0.1371	(1,3)	0.1257	(1,3)	0.3023	(1,9)	0.4600	(1,3)	0.2579
11	(1,7)	0.1284	(1,7)	0.1179	(1,7)	0.2743	(1,2)	0.4582	(1,8)	0.2491
12	(1,5)	0.1244	(1,5)	0.1140	(1,5)	0.2657	(1,7)	0.4733	(1,7)	0.2282
13	(1,8)	0.1166	(1,8)	0.1072	(1,4)	0.2806	(1,2)	0.4862	(1,8)	0.2133
14	(1,9)	0.1158	(1,9)	0.1054	(1,9)	0.2679	(1,3)	0.4829	(1,1)	0.2458
15	(1,11)	0.1107	(1,11)	0.1019	(1,11)	0.2489	(1,2)	0.5067	(1,11)	0.2483
16	(1,13)	0.1106	(1,13)	0.1009	(1,7)	0.2393	(1,5)	0.4650	(1,9)	0.2282
17	(1,7)	0.1018	(1,10)	0.0934	(1,4)	0.2219	(1,7)	0.4282	(1,5)	0.1955
18	(1,7)	0.0989	(1,7)	0.0914	(1,7)	0.2497	(1,13)	0.4733	(1,7)	0.1962
19	(1,8)	0.0970	(1,11)	0.0891	(1,12)	0.2077	(1,7)	0.4716	(1,12)	0.1786
20	(1,13)	0.1037	(1,13)	0.0928	(1,9)	0.2295	(1,17)	0.4400	(1,11)	0.2282
21	(1,13)	0.0904	(1,13)	0.0836	(1,4)	0.2111	(1,13)	0.4590	(1,8)	0.1895
22	(1,13)	0.0883	(1,13)	0.0816	(1,13)	0.2008	(1,19)	0.4582	(1,17)	0.1933
23	(1,9)	0.0864	(1,9)	0.0797	(1,9)	0.1970	(1,3)	0.4661	(1,7)	0.1878
25	(1,7)	0.0835	(1,7)	0.0768	(1,11)	0.1880	(1,3)	0.4400	(1,11)	0.1926
25	(1,18)	0.0830	(1,18)	0.0764	(1,11)	0.1880	(1,3)	0.4400	(1,16)	0.1772
26	(1,11)	0.0810	(1,11)	0.0746	(1,19)	0.1723	(1,21)	0.4554	(1,19)	0.1863
27	(1,17)	0.0804	(1,17)	0.0739	(1,8)	0.1827	(1,10)	0.4548	(1,19)	0.1725
28	(1,11)	0.0790	(1,11)	0.0729	(1,11)	0.1926	(1,3)	0.4614	(1,11)	0.1935
29	(1,11)	0.0761	(1,11)	0.0702	(1,9)	0.1889	(1,8)	0.4676	(1,9)	0.1744
30	(1,13)	0.0760	(1,13)	0.0700	(1,19)	0.1664	(1,7)	0.4733	(1,11)	0.1664
31	(1,22)	0.0737	(1,22)	0.0681	(1,12)	0.1664	(1,2)	0.4529	(1,24)	0.1767
32	(1,9)	0.0725	(1,9)	0.0670	(1,9)	0.1833	(1,5)	0.4650	(1,25)	0.1646
33	(1,14)	0.0718	(1,14)	0.0665	(1,26)	0.1560	(1,4)	0.4582	(1,10)	0.1641
34	(1,13)	0.0701	(1,13)	0.0650	(1,25)	0.1664	(1,5)	0.4635	(1,21)	0.1701
35	(1,22)	0.0695	(1,13)	0.0644	(1,8)	0.1576	(1,22)	0.4400	(1,22)	0.1641
36	(1,11)	0.0726	(1,11)	0.0661	(1,11)	0.1613	(1,11)	0.4733	(1,13)	0.1592
37	(1,23)	0.0674	(1,29)	0.0624	(1,7)	0.1521	(1,5)	0.4616	(1,23)	0.1590
38	(1,27)	0.0665	(1,27)	0.0616	(1,27)	0.1523	(1,3)	0.4716	(1,31)	0.1715
39	(1,11)	0.0660	(1,11)	0.0612	(1,11)	0.1557	(1,16)	0.4554	(1,25)	0.1623
40	(1,29)	0.0643	(1,29)	0.0594	(1,33)	0.1505	(1,7)	0.4650	(1,11)	0.1588
41	(1,12)	0.0638	(1,24)	0.0588	(1,30)	0.1488	(1,11)	0.4693	(1,26)	0.1601
42	(1,19)	0.0681	(1,19)	0.0620	(1,19)	0.1554	(1,19)	0.4590	(1,29)	0.1555
43	(1,25)	0.0615	(1,25)	0.0568	(1,8)	0.1437	(1,4)	0.4540	(1,30)	0.1520
44	(1,13)	0.0616	(1,13)	0.0569	(1,17)	0.1406	(1,7)	0.4582	(1,13)	0.1572
45	(1,28)	0.0606	(1,28)	0.0561	(1,28)	0.1443	(1,4)	0.4622	(1,28)	0.1552
46	(1,29)	0.0603	(1,33)	0.0558	(1,27)	0.1446	(1,7)	0.4704	(1,27)	0.1575
47	(1,18)	0.0589	(1,18)	0.0547	(1,37)	0.1361	(1,4)	0.4570	(1,34)	0.1565
48	(1,11)	0.0599	(1,13)	0.0550	(1,11)	0.1388	(1,41)	0.4525	(1,13)	0.1546
49	(1,11)	0.0584	(1,38)	0.0541	(1,15)	0.1342	(1,19)	0.4727	(1,15)	0.1533
50	(1,11)	0.0573	(1,11)	0.0528	(1,11)	0.1444	(1,41)	0.4400	(1,11)	0.1553

Table C.1: Best generators, 3 component mixture design fromthe glp method

4 Component mixture design from glp method						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3,9)	0.2121	(1,3,9)	0.1953	(1,3,9)	0.4937
11	(1,5,9)	0.1842	(1,5,9)	0.1731	(1,2,8)	0.3962
12	(1,7,11)	0.2068	(1,7,11)	0.1891	(1,7,11)	0.4872
13	(1,5,11)	0.1725	(1,5,11)	0.1625	(1,5,11)	0.3660
14	(1,3,5)	0.1693	(1,3,5)	0.1598	(1,3,5)	0.4164
15	(1,7,13)	0.1711	(1,7,13)	0.1601	(1,2,11)	0.3760
16	(1,3,7)	0.1587	(1,5,7)	0.1501	(1,5,9)	0.3078
17	(1,3,5)	0.1578	(1,11,13)	0.1488	(1,4,10)	0.3213
18	(1,5,7)	0.1541	(1,5,7)	0.1455	(1,5,11)	0.3357
19	(1,6,15)	0.1518	(1,6,15)	0.1434	(1,6,8)	0.3164
20	(1,11,13)	0.1486	(1,11,13)	0.1404	(1,11,13)	0.3428
21	(1,13,19)	0.1486	(1,13,19)	0.1393	(1,13,16)	0.3310
22	(1,5,7)	0.1427	(1,5,7)	0.1348	(1,3,5)	0.2970
23	(1,13,17)	0.1396	(1,13,17)	0.1320	(1,5,16)	0.2925
24	(1,11,19)	0.1384	(1,11,19)	0.1308	(1,17,19)	0.2970
25	(1,4,6)	0.1368	(1,14,19)	0.1295	(1,4,14)	0.2766
26	(1,11,17)	0.1355	(1,11,17)	0.1276	(1,11,17)	0.2851
27	(1,10,23)	0.1317	(1,10,23)	0.1245	(1,5,19)	0.2815
28	(1,5,13)	0.1303	(1,5,13)	0.1230	(1,3,13)	0.2786
29	(1,16,24)	0.1279	(1,16,24)	0.1209	(1,4,13)	0.2567
30	(1,7,13)	0.1283	(1,7,13)	0.1214	(1,7,11)	0.2632
31	(1,5,14)	0.1255	(1,5,14)	0.1185	(1,5,7)	0.2508
32	(1,5,17)	0.1256	(1,5,17)	0.1185	(1,9,17)	0.2654
33	(1,5,8)	0.1236	(1,5,8)	0.1170	(1,4,7)	0.2542
34	(1,15,27)	0.1239	(1,15,27)	0.1167	(1,3,27)	0.2690
35	(1,19,24)	0.1189	(1,19,24)	0.1129	(1,19,24)	0.2326
36	(1,11,17)	0.1184	(1,11,17)	0.1119	(1,5,17)	0.2384
37	(1,10,20)	0.1162	(1,10,20)	0.1103	(1,10,20)	0.2330
38	(1,7,17)	0.1168	(1,7,17)	0.1107	(1,27,31)	0.2472
39	(1,7,22)	0.1152	(1,7,22)	0.1089	(1,7,22)	0.2492
40	(1,29,33)	0.1170	(1,29,33)	0.1099	(1,17,33)	0.2458
41	(1,11,16)	0.1127	(1,11,16)	0.1067	(1,9,15)	0.2301
42	(1,5,13)	0.1118	(1,5,13)	0.1057	(1,5,23)	0.2315
43	(1,7,27)	0.1117	(1,7,27)	0.1055	(1,8,13)	0.2464
44	(1,13,23)	0.1127	(1,13,23)	0.1065	(1,13,23)	0.2475
45	(1,8,13)	0.1094	(1,8,13)	0.1037	(1,4,29)	0.2323
46	(1,9,13)	0.1079	(1,9,13)	0.1019	(1,7,11)	0.2279
47	(1,10,32)	0.1071	(1,10,32)	0.1015	(1,14,26)	0.2111
48	(1,13,19)	0.1059	(1,13,19)	0.1004	(1,11,41)	0.2215
49	(1,27,41)	0.1056	(1,6,15)	0.1000	(1,15,27)	0.2085
50	(1,7,23)	0.1068	(1,7,23)	0.1008	(1,7,41)	0.2323

Table C.2: Best generators, 4 component mixture design fromthe glp method

5 Component mixture design from glp method						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3,7,9)	0.2477	(1,3,7,9)	0.2326	(1,3,7,9)	0.5945
11	(1,2,5,8)	0.2156	(1,2,5,8)	0.2058	(1,2,3,10)	0.4791
12	(1,5,7,11)	0.2430	(1,5,7,11)	0.2269	(1,5,7,11)	0.5664
13	(1,2,6,9)	0.2044	(1,2,6,9)	0.1955	(1,2,6,9)	0.4691
14	(1,3,9,13)	0.2166	(1,3,9,11)	0.2046	(1,3,5,13)	0.5361
15	(1,2,4,7)	0.1969	(1,2,4,8)	0.1879	(1,2,7,14)	0.4698
16	(1,3,7,11)	0.1934	(1,3,7,11)	0.1850	(1,3,7,11)	0.4059
17	(1,2,4,9)	0.1886	(1,2,4,9)	0.1804	(1,2,3,7)	0.4357
18	(1,5,11,17)	0.2054	(1,5,11,17)	0.1931	(1,7,13,17)	0.4545
19	(1,6,11,14)	0.1888	(1,6,11,14)	0.1798	(1,2,3,14)	0.4673
20	(1,3,9,13)	0.1832	(1,3,9,13)	0.1749	(1,11,13,17)	0.4246
21	(1,2,4,13)	0.1809	(1,2,5,10)	0.1730	(1,2,5,10)	0.4251
22	(1,3,7,17)	0.1791	(1,3,5,7)	0.1713	(1,3,7,17)	0.3679
23	(1,2,3,16)	0.1806	(1,2,3,16)	0.1721	(1,2,3,16)	0.4199
24	(1,7,13,19)	0.1731	(1,7,13,19)	0.1653	(1,11,17,19)	0.4397
25	(1,3,6,14)	0.1742	(1,3,6,14)	0.1658	(1,2,4,7)	0.4207
26	(1,5,17,19)	0.1682	(1,5,17,19)	0.1608	(1,3,5,11)	0.4564
27	(1,5,10,25)	0.1705	(1,5,10,25)	0.1627	(1,2,5,11)	0.3950
28	(1,3,5,15)	0.1634	(1,3,5,15)	0.1566	(1,3,5,13)	0.3478
29	(1,2,4,15)	0.1707	(1,2,4,15)	0.1623	(1,2,4,15)	0.4095
30	(1,7,13,19)	0.1615	(1,7,13,19)	0.1547	(1,7,13,19)	0.4137
31	(1,2,3,19)	0.1706	(1,2,3,19)	0.1618	(1,2,4,7)	0.4251
32	(1,3,9,15)	0.1600	(1,3,9,15)	0.1531	(1,3,5,11)	0.4304
33	(1,2,5,13)	0.1631	(1,2,5,13)	0.1554	(1,2,5,13)	0.3799
34	(1,3,13,23)	0.1573	(1,3,13,23)	0.1503	(1,3,5,9)	0.3482
35	(1,2,4,11)	0.1636	(1,2,4,11)	0.1555	(1,2,3,31)	0.4027
36	(1,11,13,17)	0.1530	(1,11,13,17)	0.1465	(1,11,23,31)	0.3369
37	(1,2,3,10)	0.1654	(1,2,3,10)	0.1565	(1,2,4,7)	0.3924
38	(1,3,5,21)	0.1538	(1,3,5,21)	0.1468	(1,3,5,11)	0.4107
39	(1,2,5,11)	0.1581	(1,2,5,11)	0.1501	(1,2,5,11)	0.3755
40	(1,3,7,21)	0.1492	(1,3,7,21)	0.1425	(1,3,7,11)	0.3279
41	(1,8,19,30)	0.1623	(1,8,19,30)	0.1525	(1,2,4,7)	0.3854
42	(1,5,11,23)	0.1464	(1,5,11,19)	0.1403	(1,11,13,37)	0.3620
43	(1,2,3,12)	0.1618	(1,4,5,29)	0.1521	(1,2,4,7)	0.3960
44	(1,5,23,11)	0.1469	(1,5,23,11)	0.1401	(1,3,5,27)	0.3969
45	(1,8,13,26)	0.1498	(1,8,13,26)	0.1424	(1,2,7,11)	0.3584
46	(1,17,27,31)	0.1456	(1,17,27,31)	0.1396	(1,3,5,35)	0.2987
47	(1,2,3,34)	0.1593	(1,2,3,34)	0.1501	(1,2,4,7)	0.3889
48	(1,5,11,19)	0.1405	(1,5,11,19)	0.1348	(1,5,11,19)	0.3193
49	(1,2,4,9)	0.1578	(1,2,4,9)	0.1495	(1,2,4,9)	0.3739
50	(1,29,33,43)	0.1424	(1,29,33,43)	0.1357	(1,9,13,27)	0.3582

Table C.3: Best generators, 5 component mixture design fromthe glp method

APPENDIX D

TABLES GENERATED USING PROJECTION METHOD

glp method in C^2										
	rmsd		*ad*		*md*		STRD		MSTRD	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3)	0.1456	(1,3)	0.1331	(1,7)	0.3504	(1,7)	0.1746	(1,7)	0.1746
11	(1,7)	0.1350	(1,7)	0.1244	(1,8)	0.3182	(1,5)	0.1490	(1,7)	0.1490
12	(1,5)	0.1338	(1,5)	0.1232	(1,5)	0.2754	(1,5)	0.1393	(1,5)	0.1393
13	(1,8)	0.1238	(1,8)	0.1141	(1,10)	0.2720	(1,11)	0.1249	(1,5)	0.1302
14	(1,9)	0.1215	(1,9)	0.1113	(1,11)	0.2547	(1,9)	0.1189	(1,9)	0.1189
15	(1,11)	0.1145	(1,11)	0.1062	(1,11)	0.2497	(1,11)	0.1096	(1,11)	0.1096
16	(1,5)	0.1167	(1,5)	0.1062	(1,13)	0.2678	(1,7)	0.1037	(1,7)	0.1063
17	(1,5)	0.1082	(1,5)	0.0999	(1,5)	0.2326	(1,11)	0.1022	(1,12)	0.1022
18	(1,5)	0.1027	(1,5)	0.0950	(1,5)	0.2359	(1,13)	0.0993	(1,13)	0.0993
19	(1,14)	0.1008	(1,14)	0.0932	(1,8)	0.2304	(1,14)	0.0875	(1,14)	0.0897
20	(1,13)	0.1100	(1,13)	0.0993	(1,17)	0.2735	(1,9)	0.0770	(1,9)	0.0971
21	(1,13)	0.0950	(1,13)	0.0881	(1,13)	0.2097	(1,5)	0.0922	(1,5)	0.0922
22	(1,5)	0.0936	(1,5)	0.0864	(1,9)	0.2117	(1,17)	0.0805	(1,17)	0.0805
23	(1,9)	0.0901	(1,9)	0.0837	(1,19)	0.2112	(1,9)	0.0717	(1,9)	0.0734
24	(1,19)	0.0886	(1,19)	0.0822	(1,7)	0.1874	(1,11)	0.0693	(1,17)	0.0722
25	(1,7)	0.0869	(1,7)	0.0804	(1,7)	0.2014	(1,16)	0.0637	(1,16)	0.0673
26	(1,11)	0.0862	(1,11)	0.0797	(1,11)	0.1787	(1,11)	0.0622	(1,11)	0.0639
27	(1,16)	0.0844	(1,5)	0.0778	(1,17)	0.1907	(1,16)	0.0652	(1,22)	0.0669
28	(1,11)	0.0821	(1,5)	0.0760	(1,23)	0.1943	(1,11)	0.0650	(1,11)	0.0650
29	(1,18)	0.0799	(1,18)	0.0740	(1,18)	0.1730	(1,13)	0.0596	(1,13)	0.0606
30	(1,7)	0.0821	(1,7)	0.0758	(1,13)	0.1919	(1,19)	0.0580	(1,19)	0.0580
31	(1,13)	0.0775	(1,13)	0.0720	(1,7)	0.1689	(1,14)	0.0513	(1,14)	0.0595
32	(1,7)	0.0757	(1,7)	0.0702	(1,27)	0.1732	(1,25)	0.0599	(1,25)	0.0599
33	(1,14)	0.0751	(1,14)	0.0699	(1,14)	0.1691	(1,14)	0.0540	(1,14)	0.0554
34	(1,21)	0.0735	(1,21)	0.0684	(1,29)	0.1725	(1,25)	0.0486	(1,13)	0.0531
35	(1,29)	0.0719	(1,29)	0.0665	(1,8)	0.1631	(1,16)	0.0521	(1,27)	0.0543
36	(1,7)	0.0749	(1,7)	0.0691	(1,31)	0.1764	(1,23)	0.0496	(1,23)	0.0518
37	(1,8)	0.0703	(1,6)	0.0654	(1,23)	0.1590	(1,23)	0.0461	(1,23)	0.0463
38	(1,31)	0.0697	(1,31)	0.0645	(1,7)	0.1590	(1,31)	0.0473	(1,27)	0.0516
39	(1,7)	0.0684	(1,7)	0.0636	(1,17)	0.1547	(1,25)	0.0467	(1,25)	0.0467
40	(1,17)	0.0672	(1,17)	0.0625	(1,9)	0.1490	(1,33)	0.0452	(1,33)	0.0488
41	(1,9)	0.0667	(1,34)	0.0620	(1,9)	0.1543	(1,17)	0.0435	(1,17)	0.0435
42	(1,25)	0.0710	(1,25)	0.0650	(1,25)	0.1784	(1,19)	0.0433	(1,19)	0.0480
43	(1,31)	0.0647	(1,25)	0.0601	(1,31)	0.1469	(1,34)	0.0408	(1,25)	0.0444
44	(1,37)	0.0651	(1,37)	0.0601	(1,17)	0.1503	(1,27)	0.0418	(1,27)	0.0428
45	(1,26)	0.0629	(1,26)	0.0585	(1,19)	0.1506	(1,19)	0.0388	(1,19)	0.0395
46	(1,13)	0.0627	(1,33)	0.0583	(1,29)	0.1481	(1,29)	0.0398	(1,29)	0.0408
47	(1,27)	0.0619	(1,27)	0.0573	(1,34)	0.1352	(1,37)	0.0363	(1,37)	0.0407
48	(1,41)	0.0619	(1,41)	0.0575	(1,41)	0.1435	(1,31)	0.0364	(1,31)	0.0423
49	(1,11)	0.0615	(1,40)	0.0571	(1,11)	0.1338	(1,29)	0.0368	(1,29)	0.0368
50	(1,29)	0.0597	(1,11)	0.0555	(1,9)	0.1389	(1,23)	0.0398	(1,21)	0.0399

Table D.1: Best generators in C^2 from the glp method without the reduction methods

glp method projected from C^2 into C^3										
	rmsd		*ad*		*md*		STRD		MSTRD	
n	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3,7)	0.3177	(1,3,9)	0.2902	(1,7,9)	0.7523	(1,3,9)	0.2340	(1,3,7)	0.3126
11	(1,7,9)	0.2611	(1,7,9)	0.2467	(1,2,7)	0.5331	(1,2,7)	0.2281	(1,2,7)	0.2281
12	(1,5,11)	0.3125	(1,5,11)	0.2830	(1,5,11)	0.7488	(1,5,11)	0.2255	(1,5,7)	0.2628
13	(1,5,11)	0.2431	(1,5,11)	0.2299	(1,2,8)	0.5164	(1,6,8)	0.1734	(1,4,5)	0.1773
14	(1,9,11)	0.2343	(1,9,11)	0.2228	(1,9,11)	0.5820	(1,3,9)	0.1844	(1,9,11)	0.1844
15	(1,8,11)	0.2424	(1,8,11)	0.2278	(1,7,11)	0.5180	(1,2,11)	0.1860	(1,2,11)	0.1860
16	(1,3,7)	0.2220	(1,3,7)	0.2110	(1,5,7)	0.4371	(1,5,13)	0.1499	(1,3,5)	0.1625
17	(1,12,14)	0.2182	(1,12,14)	0.2073	(1,12,14)	0.4585	(1,5,7)	0.1359	(1,5,7)	0.1565
18	(1,11,13)	0.2204	(1,11,13)	0.2085	(1,5,7)	0.4530	(1,7,13)	0.1321	(1,7,13)	0.1406
19	(1,11,14)	0.2103	(1,11,14)	0.1993	(1,11,14)	0.4567	(1,6,14)	0.1280	(1,8,14)	0.1319
20	(1,3,13)	0.2071	(1,3,13)	0.1961	(1,3,9)	0.4125	(1,9,17)	0.1307	(1,9,13)	0.1528
21	(1,13,19)	0.2118	(1,13,16)	0.1987	(1,2,5)	0.4791	(1,10,13)	0.1249	(1,4,13)	0.1464
22	(1,5,15)	0.2013	(1,5,15)	0.1906	(1,5,15)	0.4064	(1,7,17)	0.1094	(1,3,17)	0.1320
23	(1,9,20)	0.1946	(1,9,20)	0.1851	(1,9,17)	0.4172	(1,5,9)	0.1070	(1,9,10)	0.1263
24	(1,13,17)	0.1964	(1,13,17)	0.1853	(1,5,17)	0.4065	(1,11,19)	0.0968	(1,11,17)	0.1096
25	(1,16,22)	0.1877	(1,16,22)	0.1780	(1,3,16)	0.3799	(1,6,7)	0.1013	(1,16,21)	0.1156
26	(1,11,23)	0.1910	(1,11,23)	0.1804	(1,11,17)	0.3973	(1,5,11)	0.1059	(1,5,11)	0.1201
27	(1,19,22)	0.1830	(1,19,22)	0.1736	(1,7,16)	0.3890	(1,8,22)	0.0899	(1,8,22)	0.1045
28	(1,11,25)	0.1836	(1,11,25)	0.1736	(1,3,11)	0.4010	(1,11,23)	0.0999	(1,11,15)	0.1114
29	(1,5,13)	0.1778	(1,5,13)	0.1689	(1,4,18)	0.3570	(1,13,24)	0.0784	(1,13,18)	0.0965
30	(1,7,11)	0.1797	(1,7,11)	0.1701	(1,13,19)	0.3820	(1,19,23)	0.0817	(1,7,13)	0.0944
31	(1,5,14)	0.1741	(1,11,14)	0.1655	(1,14,20)	0.3458	(1,13,27)	0.0839	(1,7,13)	0.0971
32	(1,21,25)	0.1754	(1,21,25)	0.1662	(1,3,25)	0.3837	(1,7,9)	0.0793	(1,25,27)	0.0976
33	(1,14,25)	0.1712	(1,14,25)	0.1628	(1,14,25)	0.3513	(1,10,14)	0.0841	(1,10,14)	0.0893
34	(1,13,31)	0.1701	(1,13,31)	0.1612	(1,13,31)	0.3656	(1,15,21)	0.0733	(1,15,21)	0.0944
35	(1,26,29)	0.1657	(1,26,29)	0.1574	(1,12,27)	0.3580	(1,8,29)	0.0757	(1,13,27)	0.0910
36	(1,23,31)	0.1686	(1,23,31)	0.1597	(1,7,17)	0.3744	(1,17,23)	0.0757	(1,5,7)	0.0922
37	(1,8,11)	0.1646	(1,8,11)	0.1563	(1,6,23)	0.3482	(1,21,23)	0.0688	(1,11,23)	0.0758
38	(1,27,31)	0.1612	(1,27,31)	0.1534	(1,27,31)	0.3330	(1,17,27)	0.0699	(1,17,27)	0.0839
39	(1,4,25)	0.1596	(1,4,25)	0.1519	(1,4,25)	0.3240	(1,7,22)	0.0722	(1,25,35)	0.0872
40	(1,29,33)	0.1614	(1,29,33)	0.1528	(1,11,17)	0.3779	(1,9,17)	0.0662	(1,9,33)	0.0801
41	(1,13,17)	0.1592	(1,13,17)	0.1511	(1,13,17)	0.3156	(1,17,19)	0.0663	(1,9,29)	0.0720
42	(1,11,25)	0.1534	(1,11,25)	0.1464	(1,11,25)	0.3170	(1,25,31)	0.0608	(1,25,31)	0.0789
43	(1,31,35)	0.1546	(1,31,35)	0.1469	(1,8,31)	0.3156	(1,16,31)	0.0595	(1,13,25)	0.0715
44	(1,17,37)	0.1575	(1,17,37)	0.1493	(1,17,27)	0.3452	(1,31,37)	0.0618	(1,25,27)	0.0755
45	(1,16,26)	0.1526	(1,26,26)	0.1449	(1,19,29)	0.3036	(1,19,28)	0.0611	(1,26,31)	0.0780
46	(1,13,41)	0.1503	(1,13,41)	0.1431	(1,13,41)	0.3049	(1,9,13)	0.0576	(1,11,29)	0.0715
47	(1,32,37)	0.1505	(1,32,37)	0.1428	(1,17,27)	0.3135	(1,22,37)	0.0611	(1,18,37)	0.0730
48	(1,5,31)	0.1479	(1,5,31)	0.1407	(1,5,31)	0.3049	(1,31,43)	0.0583	(1,31,37)	0.0704
49	(1,13,29)	0.1464	(1,13,29)	0.1398	(1,6,29)	0.3040	(1,15,29)	0.0586	(1,15,29)	0.0629
50	(1,29,33)	0.1510	(1,29,33)	0.1424	(1,21,37)	0.3364	(1,11,21)	0.0523	(1,29,31)	0.0648

Table D.2: Best generators from the glp method projected fromC^2 into C^3

glp method projected from C^3 into C^4						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
10	(1,3,7,9)	0.4386	(1,3,7,9)	0.4134	(1,3,7,9)	0.9256
11	(1,7,8,9)	0.3708	(1,7,8,9)	0.3561	(1,7,8,9)	0.6775
12	(1,5,7,11)	0.4345	(1,5,7,11)	0.4081	(1,5,7,11)	0.9238
13	(1,4,5,11)	0.3523	(1,4,5,11)	0.3388	(1,4,5,11)	0.6474
14	(1,9,11,13)	0.3813	(1,9,11,13)	0.3596	(1,5,9,11)	0.8038
15	(1,8,11,13)	0.3325	(1,8,11,13)	0.3201	(1,8,11,13)	0.5995
16	(1,3,7,11)	0.3307	(1,3,7,11)	0.3180	(1,3,5,9)	0.6607
17	(1,10,12,14)	0.3261	(1,10,12,14)	0.3135	(1,4,12,14)	0.6399
18	(1,7,13,17)	0.3708	(1,7,13,17)	0.3481	(1,5,7,13)	0.7887
19	(1,8,14,17)	0.3178	(1,8,14,17)	0.3052	(1,11,14,16)	0.6282
20	(1,3,9,13)	0.3181	(1,3,9,13)	0.3047	(1,3,9,13)	0.6595
21	(1,13,17,19)	0.3098	(1,13,17,19)	0.2973	(1,4,13,19)	0.5879
22	(1,3,13,17)	0.3054	(1,3,13,17)	0.2932	(1,3,13,17)	0.6486
23	(1,9,18,20)	0.3038	(1,5,9,20)	0.2917	(1,9,18,20)	0.6049
24	(1,5,11,17)	0.3073	(1,5,11,17)	0.2945	(1,13,17,19)	0.6164
25	(1,16,19,21)	0.2919	(1,16,19,21)	0.2812	(1,16,19,21)	0.5634
26	(1,5,11,23)	0.2886	(1,5,11,23)	0.2777	(1,9,11,23)	0.6025
27	(1,8,10,22)	0.2903	(1,8,17,22)	0.2788	(1,8,17,22)	0.5473
28	(1,11,23,25)	0.2806	(1,11,23,25)	0.2704	(1,11,15,23)	0.5834
29	(1,4,13,18)	0.2832	(1,13,18,20)	0.2722	(1,5,11,13)	0.5747
30	(1,7,11,13)	0.2794	(1,7,11,13)	0.2695	(1,7,13,19)	0.5418
31	(1,7,13,28)	0.2763	(1,7,13,28)	0.2659	(1,3,7,13)	0.5537
32	(1,21,23,25)	0.2727	(1,21,23,25)	0.2626	(1,23,25,27)	0.5605
33	(1,10,14,16)	0.2744	(1,10,14,16)	0.2635	(1,10,14,31)	0.5420
34	(1,7,13,31)	0.2712	(1,7,13,31)	0.2606	(1,15,21,25)	0.5257
35	(1,11,26,29)	0.2672	(1,11,26,29)	0.2571	(1,11,26,29)	0.5366
36	(1,5,7,17)	0.2669	(1,5,7,17)	0.2568	(1,23,25,31)	0.5016
37	(1,5,11,23)	0.2617	(1,5,11,23)	0.2519	(1,6,11,23)	0.5289
38	(1,17,27,35)	0.2627	(1,17,27,35)	0.2523	(1,17,23,27)	0.5071
39	(1,23,25,35)	0.2625	(1,23,25,35)	0.2517	(1,4,7,25)	0.5230
40	(1,27,29,33)	0.2555	(1,27,29,33)	0.2462	(1,27,29,33)	0.4828
41	(1,13,17,19)	0.2517	(1,13,17,19)	0.2429	(1,13,17,19)	0.4709
42	(1,23,25,31)	0.2531	(1,23,25,31)	0.2439	(1,23,25,31)	0.4962
43	(1,29,31,35)	0.2498	(1,29,31,35)	0.2407	(1,3,13,25)	0.4862
44	(1,15,17,37)	0.2508	(1,15,17,37)	0.2415	(1,15,17,37)	0.4800
45	(1,26,31,37)	0.2513	(1,26,31,37)	0.2417	(1,16,26,34)	0.5074
46	(1,11,25,29)	0.2477	(1,11,25,29)	0.2385	(1,11,13,41)	0.4705
47	(1,32,37,44)	0.2443	(1,32,37,44)	0.2355	(1,18,37,44)	0.4784
48	(1,23,31,37)	0.2457	(1,23,31,37)	0.2363	(1,23,31,37)	0.4804
49	(1,13,29,44)	0.2430	(1,15,26,29)	0.2335	(1,15,29,41)	0.4653
50	(1,7,29,33)	0.2406	(1,7,29,33)	0.2317	(1,29,33,43)	0.4652

Table D.3: Best generators from the glp method projected fromC^3 into C^4

glp method projected from C^4 into C^5						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
11	(1,6,7,8,9)	0.4732	(1,6,7,8,9)	0.4584	(1,5,7,8,9)	0.8972
13	(1,3,4,5,11)	0.4541	(1,3,4,5,11)	0.4404	(1,2,4,5,11)	0.8961
14	(1,3,5,9,11)	0.4900	(1,3,5,9,11)	0.4688	(1,3,5,9,11)	0.9828
15	(1,2,8,11,13)	0.4538	(1,2,8,11,13)	0.4365	(1,7,8,11,13)	0.8746
16	(1,3,5,9,15)	0.4555	(1,3,7,9,15)	0.4377	(1,3,5,9,15)	0.8755
17	(1,10,11,12,14)	0.4283	(1,10,11,12,14)	0.4146	(1,4,9,12,14)	0.8085
18	(1,5,7,11,13)	0.4808	(1,5,7,11,13)	0.4587	(1,5,7,13,17)	0.9830
19	(1,9,11,14,16)	0.4169	(1,9,11,14,16)	0.4041	(1,8,9,14,17)	0.7943
20	(1,3,9,13,17)	0.4441	(1,3,9,11,13)	0.4254	(1,3,9,13,17)	0.8431
21	(1,4,5,13,19)	0.4070	(1,4,5,13,19)	0.3945	(1,4,5,13,19)	0.8227
22	(1,3,13,15,17)	0.4070	(1,3,13,15,17)	0.3937	(1,3,13,15,17)	0.7580
23	(1,7,9,18,20)	0.4004	(1,7,9,18,20)	0.3876	(1,6,9,18,20)	0.7619
24	(1,7,13,17,19)	0.4342	(1,7,13,17,19)	0.4156	(1,5,11,17,19)	0.8499
25	(1,11,16,19,21)	0.3909	(1,14,16,19,21)	0.3790	(1,16,19,21,22)	0.7445
26	(1,9,11,19,23)	0.3940	(1,7,9,11,23)	0.3809	(1,5,9,11,23)	0.7607
27	(1,8,17,20,22)	0.3858	(1,8,17,20,22)	0.3740	(1,8,17,20,22)	0.7629
28	(1,11,15,19,23)	0.3807	(1,11,15,19,23)	0.3691	(1,11,15,19,23)	0.7353
29	(1,4,13,18,21)	0.3804	(1,4,13,18,21)	0.3683	(1,4,13,17,18)	0.7007
30	(1,7,11,13,23)	0.4143	(1,7,11,13,23)	0.3955	(1,7,13,19,23)	0.8169
31	(1,5,7,13,28)	0.3728	(1,5,7,13,28)	0.3615	(1,3,7,11, 13)	0.7140
32	(1,13, 21,23,25)	0.3704	(1,13,21,23,25)	0.3591	(1,15,23,25,27)	0.7109
33	(1,5,10,14,16)	0.3647	(1,5,10,14,16)	0.3539	(1,5,10,14,16)	0.6701
34	(1,5,7,13,31)	0.3662	(1,5,7,13,31)	0.3549	(1,15,21,25,27)	0.7104
35	(1,11,22,26,29)	0.3606	(1,11,22,26,29)	0.3497	(1,12,22,26,29)	0.6806
36	(1,5,7,17,25)	0.3679	(1,5,7,13,17)	0.3564	(1,19,23,25,31)	0.7155
37	(1,6,9,11,23)	0.3582	(1,6,11,23,35)	0.3474	(1,6,11,19,23)	0.6828
38	(1,17,27,33,35)	0.3564	(1,17,27,33,35)	0.3455	(1,15,17,27,35)	0.6981
39	(1,4,7,16,25)	0.3530	(1,4,7,16,25)	0.3423	(1,4,7,25,37)	0.6835
40	(1,27,29,33,37)	0.3532	(1,27,29,33,37)	0.3424	(1,3,27,29,33)	0.6963
41	(1,13,15,17,19)	0.3496	(1,13,15,17,19)	0.3392	(1,13,17,19,25)	0.6646
42	(1,23,25,29,31)	0.3511	(1,23,25,29,31)	0.3402	(1,13,23,25,31)	0.7005
43	(1,24,29,31,35)	0.3443	(1,24,29,31,35)	0.3340	(1,28,29,31,35)	0.6462
44	(1,15,17,37,41)	0.3449	(1,15,17,37,41)	0.3342	(1,5,15,17,37)	0.6907
45	(1,26,29,31,37)	0.3418	(1,26,29,31,37)	0.3315	(1,16,22,26,34)	0.6584
46	(1,11,25,27,29)	0.3421	(1,11,25,27,29)	0.3318	(1,3,11,13,41)	0.6756
47	(1,20,32,37,44)	0.3391	(1,20,32,37,44)	0.3287	(1,29,32,37,44)	0.6416
48	(1,23,29,31,37)	0.3409	(1,23,29,31,37)	0.3307	(1,13,23,31,37)	0.6721
49	(1,13,29,44,46)	0.3345	(1,3,13,29,44)	0.3247	(1,15,18,29,44)	0.6329
50	(1,7,9,29,33)	0.3378	(1,7,9,29,33)	0.3272	(1,7,29,33,43)	0.6816

Table D.4: Best generators from the glp method projected fromC^4 into C^5

glp method projected from C^5 into C^6						
	rmsd		*ad*		*md*	
n	Gen.	Value	Gen.	Value	Gen.	Value
11	(1,3,6,7,8,9)	0.5752	(1,3,6,7,8,9)	0.5594	(1,6,7,8,9,10)	1.0335
13	(1,3,4,5,7,11)	0.5485	(1,3,4,5,7,11)	0.5347	(1,2,3,4,5,11)	1.0049
14	(1,3,5,9,11,13)	0.5812	(1,3,5,9,11,13)	0.5622	(1,3,5,9,11,13)	1.1251
15	(1,2,7,8,11,13)	0.5519	(1,2,7,8,11,13)	0.5346	(1,2,4,8,11,13)	1.0351
16	(1,3,5,9,13,15)	0.5536	(1,3,5,9,13,15)	0.5362	(1,3,5,9,11,15)	1.0500
17	(1,4,9,12,14,15)	0.5207	(1,4,9,12,14,15)	0.5077	(1,4,10,11,12,14)	0.9642
18	(1,5,7,11,13,17)	0.5723	(1,5,7,11,13,17)	0.5527	(1,5 ,7,11,13,17)	1.1185
19	(1,9,11,14,15,16)	0.5086	(1,9,11,14,15,16)	0.4960	(1,3,9,11,14,16)	0.9528
20	(1,3,9,13,17,19)	0.5425	(1,3,9,13,17,19)	0.5243	(1,3,9,13,17,19)	1.0565
21	(1,4,5,10,13,19)	0.4966	(1,4,5,10,13,19)	0.4843	(1,4,5,11,13,19)	0.9206
22	(1,3,13,15,17,21)	0.5152	(1,3,13,15,17,21)	0.4996	(1,3,13,15,17,21)	0.9900
23	(1,7,9,10,18,20)	0.4908	(1,7,9,10,18,20)	0.4784	(1,2,6,9,18,20)	0.9014
24	(1,7,13,17,19,23)	0.5335	(1,7,13,17,19,23)	0.5155	(1,5,11,13,17,19)	1.0329
25	(1,11,16,18,19,21)	0.4834	(1,8,16,19,21,22)	0.4714	(1,12,16,19,21,22)	0.8884
26	(1,5,7,9,11,23)	0.4861	(1,5,7,9,11,23)	0.4733	(1,9,11,19,21,23)	0.8491
27	(1,8,14,17,20,22)	0.4761	(1,8,14,17,20,22)	0.4641	(1,8,14,17,20,22)	0.8642
28	(1,11,15,19,23,25)	0.4698	(1,11,15,19,23,25)	0.4586	(1,11,15,19,23,25)	0.8165
29	(1,4,13,18,21,27)	0.4687	(1,4,13,18,21,27)	0.4568	(1,4,13,17,18,21)	0.8590
30	(1,7,11,13,23,29)	0.5179	(1,7,11,13,23,29)	0.4991	(1,7,13,19,23,29)	0.9902
31	(1,5,7,9,13,28)	0.4635	(1,5,7,9,13,28)	0.4518	(1,5,7,13,17,28)	0.8584
32	(1,13,21,23,25,29)	0.4629	(1,13,21,23,25,29)	0.4513	(1,3,13,21,23,25)	0.8486
33	(1,5,10,14,16,25)	0.4530	(1,5,10,14,16,25)	0.4420	(1,5,10,14,16,25)	0.8405
34	(1,5,7,11,13,31)	0.4585	(1,5,7,11,13,31)	0.4467	(1,5,7,13,25,31)	0.8540
35	(1,3,11,22,26,29)	0.4526	(1,3,11,22,26,29)	0.4411	(1,2,11,22,26,29)	0.8167
36	(1,5,7,13,17,25)	0.4564	(1,5,7,13,17,25)	0.4452	(1,19,23,25,29,31)	0.8195
37	(1,6,9,11,19,23)	0.4456	(1,6,9,11,19,23)	0.4348	(1,6,11,19,23,34)	0.8112
38	(1,7,17,27,33,35)	0.4483	(1,7,17,27,33,35)	0.4370	(1,9,17,27,33,35)	0.8099
39	(1,4,7,20,25,37)	0.4386	(1,4,7,20,25,37)	0.4282	(1,4,7,20,25,37)	0.8259
40	(1,3,9,27,29,33)	0.4437	(1,3,9,27,29,33)	0.4326	(1,21,27,29,33,37)	0.8005
41	(1,10,13,15,17,19)	0.4388	(1,10,13,15,17,19)	0.4280	(1,6,13,15,17,19)	0.8060
42	(1,13,23,25,31,37)	0.4402	(1,13,23,25,31,37)	0.4291	(1,13,23,25,31,37)	0.8344
43	(1,5,28,29,31,35)	0.4354	(1,3,24,29,31,35)	0.4246	(1,15,24,29,31,35)	0.7895
44	(1,15,17,35,37,41)	0.4342	(1,15,17,35,37,41)	0.4233	(1,15,15,17,21,37)	0.8003
45	(1,16,22,26,34,43)	0.4274	(1,16,22,26,34,43)	0.4173	(1,11,26,29,31,37)	0.7968
46	(1,3,7,11,13,41)	0.4295	(1,3,7,11,13,41)	0.4192	(1,3,9,11,13,41)	0.7704
47	(1,20,31,32,37,44)	0.4286	(1,20,31,32,37,44)	0.4181	(1,25,29,32,37,44)	0.7701
48	(1,13,23,31,37,43)	0.4305	(1,13,23,31,37,43)	0.4199	(1,5,23,29,31,37)	0.7637
49	(1,9,13,29,44,46)	0.4213	(1,9,13,29,44,46)	0.4112	(1,8,13,29,44,46)	0.7703
50	(1,7,9,29,33,39)	0.4276	(1,7,9,29,33,39)	0.4166	(1,7,9,23,29,33)	0.8015

Table D.5: Best generators from the glp method projected fromC^5 into C^6