NTNU

Norwegian University of
Science and Technology

Evaluation of Modern Design Methods for use in Computer Experiments

Anders Nesbakken

Master of Science in Physics and Mathematics
Submission date: February 2011
Supervisor: John Sølve Tyssedal, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

The aim of this report is to investigate the properties of the recently developed Multi-level binary replacement (MBR) design compared to other popular design methods for computer experiments, namely the Latin hypercube design and the Orthogonal array design.

Assignment given: 08. September 2010
Supervisor: John Sølve Tyssedal, MATH

Abstract

We have compared the recently developed Multi-level binary replacement (MBR) design method for use in computer experiments, to the Latin hypercube design (LHD) and the Orthogonal array (OA) design. For means of comparison, we have suggested an algorithm for drawing permutations of the MBR design, so as to obtain what we have called a MBR based Latin hypercube design. In our comparison study, the main focus have been the design scores with respect to the root mean squared error (RMSE), Max and alias sum of squares criteria. We found that the MBR design generally performed good with respect to all criteria. It scored similarly to the OA design method and better than conventional Latin hypercube sampling. The score however varied with the number of samples and the set of design generators chosen for constructing the MBR design. The MBR design performed better for designs with a relatively high number of samples compared to the number of factors.

Preface

This thesis has been carried out at the Department of Mathematical Sciences and Technology at the Norwegian University of Science and Technology (NTNU), as the final part of the Master of Technology degree within the Industrial Mathematics Program.

I would like to express my gratitude to my supervisor John Tyssedal for providing excellent guidance and help on this paper. I would also like to thank my fellow student Hege G. Thalberg for great moral support and late dicussions during the final weeks of work on this report.

Trondheim, 1. February 2011

Anders Nesbakken

Contents

Chapter 1

Introduction

Computer-based simulations are extensively used as an alternative to traditional, physical experiments. In contrast to physical experiments, an experiment based on the output of a deterministic computer code will have no random error. The main focus is to reduce the bias of an estimated metamodel. Design of experiments (DoE) methods specifically constructed for use in computer experiments, hereby referred to as modern DoE methods, therefore generally seek to be space-filling. This stands in opposition to classical DoE methods for physical experiments, which tend to place the design samples at the extremes of the experimental region, so as to reduce the effect of the random error terms.

Popular modern DoE methods include random sampling, Latin hypercube sampling (LHS) and orthogonal array (OA) sampling. The objective of this report is to conduct a comparison study of these methods and the multi-level binary replacement (MBR) design recently developed by Martens et al. [8]. The MBR design method uses a binary representation of the design factors and applies fractional factorial designs to generate designs which probe the design space at many different levels for each factor, while keeping the number of design samples needed low. For comparison of the DoE methods, we will use different criteria which reflect the designs space filling properties and ability to fit metamodels.

The report is organised as follows. In chapter 2 we will discuss some of the fundamental differences between the classical and modern experimental setting. In chapter 3 we will present some classical DoE methods. Our main focus is the fractional factorial designs, as these are used for generating the MBR designs. Chapter 4 introduces random sampling, LHS and OA sampling, along with a discussion of the advantages and disadvantages of each design method. In chapter 5 we present the MBR design method. We will also propose our own method for drawing permutations based on the MBR design, which will be used when

comparing the MBR design against other design methods. In chapter 6 we will first present some classical design criteria, and argue shortly why these might be bad measures when comparing modern DoE methods. We will then introduce alternative criteria better suited when evaluating a designs space-filling properties and ability to fit metamodels. In chapter 7 we will shortly present two different approaches for fitting metamodels, namely response surface models and kriging models. In chapter 8 we conduct a comparison study of modern design methods, where the main focus is the score of the MBR design. In the evaluation we have considered two-factor and five-factor designs.

Chapter 2

Classical vs Modern DoE

The use of computer-based simulations and analysis is becoming an increasingly popular alternative to the traditional, physical experiments. Design of experiments (DoE) methods designed for use in computer experiments differ significantly from the classical DoE-methods in their approach.

The most important difference is the lack of random error in the output when running a computer code. In physical experiments it is assumed that some random error will always exist. Classical DoE methods seek to minimize the variance of the parameters and generally place the design points at the extremes of the experimental region. Other important aspects of classical modelling is the use of blocking and replicated points. The output of computer simulations is deterministic, so running the same code several times or on different computers will return identical answers. The concepts of blocking and replication is therefore irrelevant in the modern DoE setting.

Modern DoE techniques are mainly concerned with reducing the bias in the estimated model or metamodel. They tend to favor designs where the sample points are placed in the interior of the experimental region.

An illustrative example of the fundamental differences between the classial and the modern setting with regard to the choice of experimental design approach can be seen in figure 2.1. The two plots to the left represents the classical setting where measurement error is present. By plotting the sample points in the interior, as seen for the lower left plot, the estimated trend is of opposite effect compared to the true trend of the process. In the upper left plot the sample points are placed at the far ends of the design region, and although the true trend is underestimated, it is a far better approximation. The plots to the right represents the modern setting. The true trend is polynomial, but the researcher can only afford two experimental

samples, so the estimated trend will be linear. There is no random error in the measurements. Placing the sample points in the interior may in this situation give a better approximation than placing them at the extremes of the region.

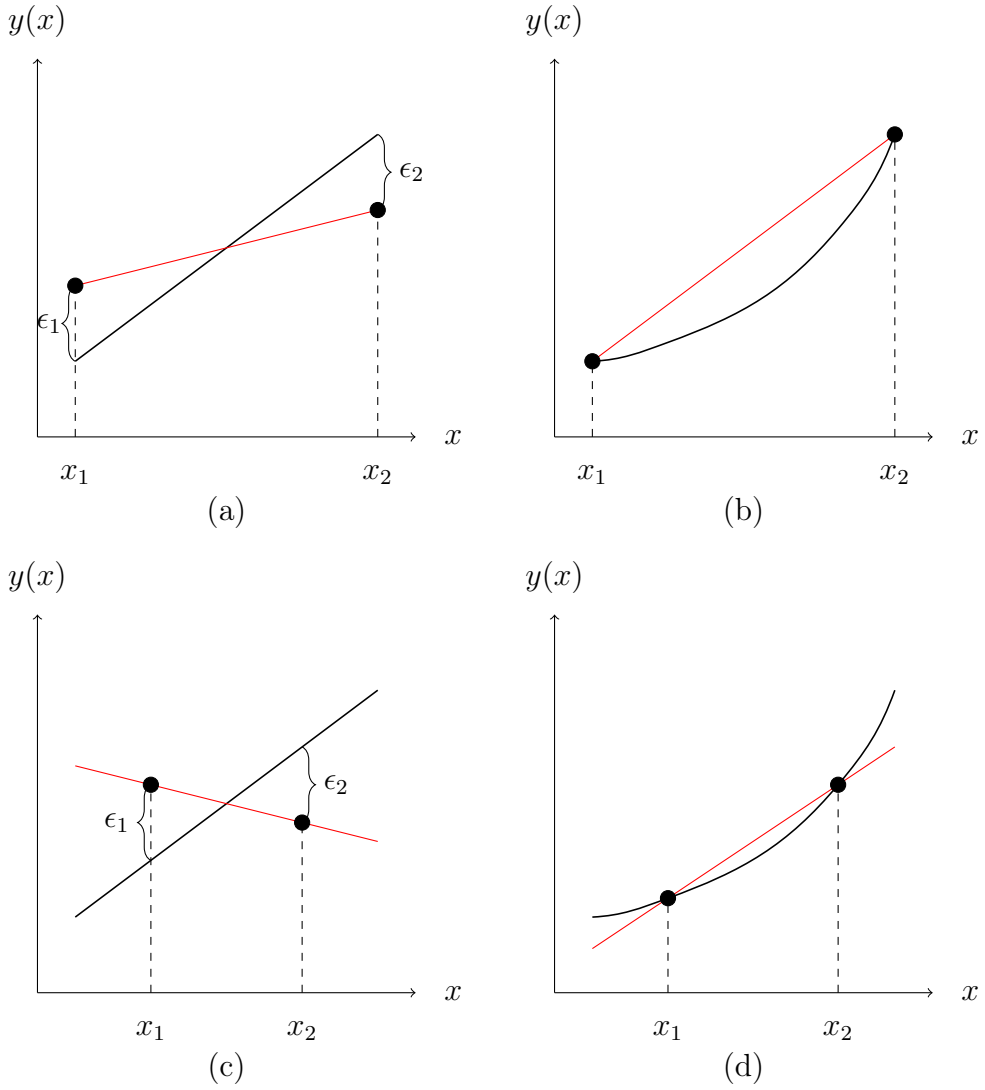

Figure 2.1: Classical versus modern modelling: The black line shows the true trend, while the red line is the estimated trend function. Classical approach: (a) and (b) places the samples at the extremes of the design space. Modern approach: (c) and (d) places the samples in the interior. Classical setting: In (a) and (c) random error occurs. Modern setting: In (b) and (d) there is no random error, only bias error.

Chapter 3

Classical Design of Experiments Methods

In the classical design of experiments setting one is considering a process where the measured outcome $y_m(\mathbf{x})$ is assumed to be a function of the true response $y_t(\mathbf{x})$ and a random error term ϵ, i. e.

$$y_m(\mathbf{x}) = y_t(\mathbf{x}) + \epsilon.$$

Here $\mathbf{x}$ represents a sample site within the experimental region.

A typical objective of the experimental study is to fit a response surface model in order to analyse the effect of the of the experimental factors on the response. This model can for example be a regression model on the form

$$\mathbf{y}_m = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{y}_m$ is a vector of responses, $\mathbf{X}$ is the design matrix and $\boldsymbol{\beta}$ a vector of regression coefficients which needs to be estimated. The vector $\boldsymbol{\epsilon}$ is assumed to consist of independent, identically distributed random variables, which are often considered normal with mean zero and variance σ^2.

The key challenge in such an experimental study lies in choosing a DoE which minimizes the effect of the error term on the measured response, i. e. reducing the variance error. To achieve this a typical classical experimental design method will tend to place the samples near the boundaries of the design space.

In this chapter we will present the two level factorial design and the central composite design (CCD), which are among the most popular of the classical DoE methods. The multilevel binary replacement (MBR) design method later to be studied, uses fractional factorials in order to generate designs.

3.1 The 2^k Factorial Design

The 2^k factorial designs are a special class of designs where each factor k is studied at exactly two levels, usually referred to as high and low. These designs are often used to fit first-order response surface models. An advantage of the factorial designs is the low computational cost and the ease in estimating the regression parameters. We will not go into the details of estimation in 2^k experiments in this report, but refer to Montgomery et al.[10] for further reading.

In the full factorial 2^k design all combinations of high/low between all experimental factors are being studied. The experiment would require a total of $n = 2^k$ runs.

3.1.1 Fractional Factorial Designs

If the number of factors is large, or it is expected that some of the factors will not have a significant influence on the response, running a fractional factorial experiment might suffice. Fractional factorial designs are often used in screening experiments.

We use the notation 2^{k-p} for a two-level fractional design, where p describes the reduction factor for the experiment, i. e. the $\frac{1}{2^p}$-fraction of the full factorial design. A key feature lies in choosing the design generator so that as little information as possible is lost. The choice of design generator yields an alias structure, which defines how the effects are confounded with each other. That effects or interaction of effects are confounded with each other means that they can not be estimated independently.

An important property regarding the alias structure is the resolution of the fractional design. Montgomery et al.[10] gives the following definition of resolution III-V designs:

Definition 1 (Design Resolution)**.** •

- **Resolution III Designs**. *A design is called a resolution III design if no main effects are confounded with any other main effects, but main effects are confounded with two-factor interactions and two-factor interactions may be confounded with each other.*

- **Resolution IV Designs**. *A design is called a resolution IV design if no main effect are confounded with any other main effect or two-factor interaction, but two-factor interactions are confounded with each other.*
- **Resolution V Designs**. *A design is called a resolution V design if no main effect or two-factor interaction are confounded with any other main effect or two-factor interaction, but two-factor interactions are confounded with three-factor interactions.*

Generally one would always seek a design with the highest resolution possible. A resolution II design would imply that main effects are confounded with each other, and are not useful for experiments. Designs of higher resolutions than V are also possible to construct, but not so commonly used since their run size normally will be large.

The consepts of design generators and alias matrices are easiest understood by an example. Consider an experiment with $k = 5$ factors, denoted by letters A to E. If one is interested in constructing a half fraction of the full 2^5 design, one would choose as design generator $I = ABCDE$. As we can see from the alias structure in table 3.1, no first order or interaction effects are confounded with each other. This would give us a resolution V design. We denote the design by 2^{5-1}_V.

$A = BCDE$	$AB = CDE$	$BD = ACE$
$B = ACDE$	$AC = BDE$	$BE = ACD$
$C = ABDE$	$AD = BCE$	$CD = ABE$
$D = ABCE$	$AE = BCD$	$CE = ABD$
$E = ABCD$	$BC = ADE$	$DE = ABC$

Table 3.1: Alias Structure for the 2^{5-1}_V design, with $I = ABCDE$.

Consider now an experiment with $k = 6$ factors, denoted by A to F. Assume we want to construct only a quarter fraction of the full design. We can now at best achieve a resolution IV design, by for example choosing the design generators $I = ABCE$ and $I = BCDF$. This yields the alias structure given in table 3.2. We see that no main effect are confounded with any other main effect or two-factor interaction, but two-factor interactions are confounded with each other.

$A = BCE = DEF = ABCDF$	$AB = CE = ACDF = BDEF$
$B = ACE = CDF = ABDEF$	$AC = BE = ABDF = CDEF$
$C = ABE = BDF = ACDEF$	$AD = EF = ABCF = BCDE$
$D = AEF = BCF = ABCDE$	$AE = BC = DF = ABCDEF$
$E = ABC = ADF = BCDEF$	$AF = DE = ABCD = BCEF$
$F = ADE = BCD = ABCEF$	$BD = CF = ABEF = ACDE$
	$BF = CD = ABDE = ACEF$
$ABD = ACF = BEF = CDE$	
$ABF = ACD = BDE = CEF$	

Table 3.2: Alias Structure for the 2^{6-2}_V design, with $I = ABCE$ and $I = BCDF$.

3.2 Central Composite Design

The central composite design (CCD) can be seen as a full factorial design augmented with axial points and center points. The axial points allows estimation of squared effects, so the design is especially useful in fitting second order response surface models. It requires a total of $n = 2^k + 2k + n_c$ samples, where n_c represents the number of center points.

Figure 3.1 shows the central composite design for three factors, along with a full factorial and fractional factorial design.

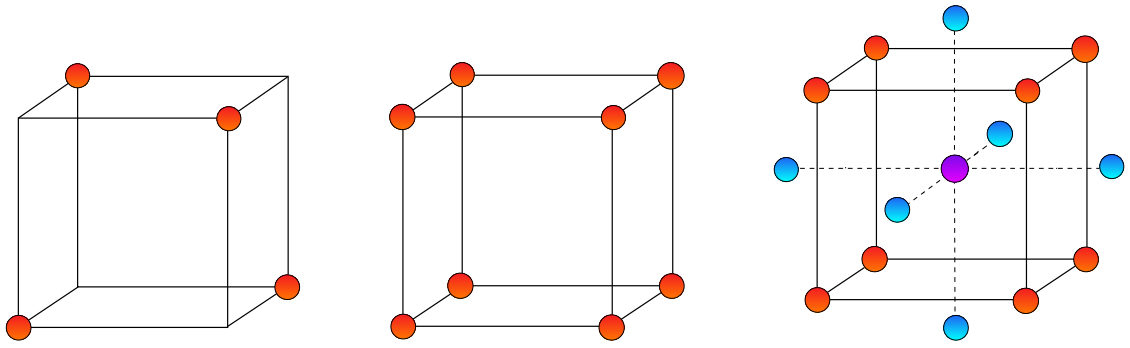

Figure 3.1: Left: 2^{3-1} fractional design. Middle: 2^3 full factorial design. Right: Central composite design with 3 factors.

Chapter 4

Modern Design of Experiments Methods

There are numerous problems to which modern DoE methods can be applied. The goal of the first computer experiments was to approximate numerical integration. A similar problem might consist of estimating some statistic, for example the mean or second order moments of a complicated function. In these cases exact values of the statistics might be impossible to obtain and traditional numerical integrations methods might be very expensive.

Another goal of a computer experiment can be to find a suitable metamodel. The true response trend is then assumed to be unknown. Also, if the underlying function requires a lot of computation time, it might be desirable to fit a metamodel so that the value at an arbitrary point within the design space can be easily computed.

We distinguish between two main types of modern DoE methods. Optimal designs seek a design which is optimal with respect to some criterion. The main focus in this report will be the so-called space-filling designs, as these have a wider variety of applications. In many situations it can be hard or impossible to choose one criterion which will be most important for a design, as this often depends on the goal of the computer simulation or the form of the underlying computer code.

In this chapter we will first introduce the Monte-Carlo (MC) sampling alghorithms, originally proposed as an alternative to numerical integration. We will then present the Latin hypercube design (LHD), which was the first design proposed specifically for applications in computer experiments. Several extensions of Latin hypercube sampling (LHS) exist, and among these maximin-LHS and orthogonal array (OA) based LHS will be further discussed.

4.1 Monte Carlo Sampling

Monte Carlo (MC) sampling in computer experiments was originally developed as an alternative to numerical integration of complex functions [13].

4.1.1 Pseudo-Monte Carlo Sampling

Pseudo-Monte Carlo sampling might also be referred to as pseudo-random sampling, stochastic sampling or just random sampling. The MC sampling algorithm draws, for each sample point, a random number within the interval $[x_{j,L}, x_{j,U}]$ of each factor j. A design of n sample points will simply be a collection of n points drawn randomly from the experimental region $[x_{j,L}, x_{j,U}]^k$, which will usually be scaled to the hypercube $[0, 1]^k$.

The clearest advantage of MC-sampling is that it can be easily implemented, one only needs a reliable pseudo-random number generator, which is readily available in most computer languages. The algorithm may also be extended to non rectangular design spaces and non-uniform distributions. A drawback is that the user has no control as to where the sample sites are placed within the experimental region, which can leave large areas unexplored. If the number of samples is large the MC-sampling algorithm can be expected to cover the design space well. But if the user can only afford a small number of samples, which is often the case in computer experiments, one might end up with unsatisfactory designs.

4.1.2 Stratified Monte Carlo Sampling

Stratified Monte Carlo sampling was developed as an alternative to the simpler pseudo-MC sampling to ensure better coverage of the design space. The algorithm divides each factor k into p_k subintervals, or bins, of equal probability. The sample sites are then drawn randomly within each bin. This provides a better overall coverage by ensuring that no region of the experimental region is left totally unexplored.

The user is free to decide the numbers of bins within each factor, so that the sample size can be chosen according to need or budget. However, stratified MC sampling requires at least a number of 2^k samples (two bins for each factor), which

might not be affordable if the numbers of factors is large.

In figure 4.1 we can see a pseudo-MC sample at the left, and a stratified MC-sample at the right. The stratified sample clearly has a better coverage than the random sample.

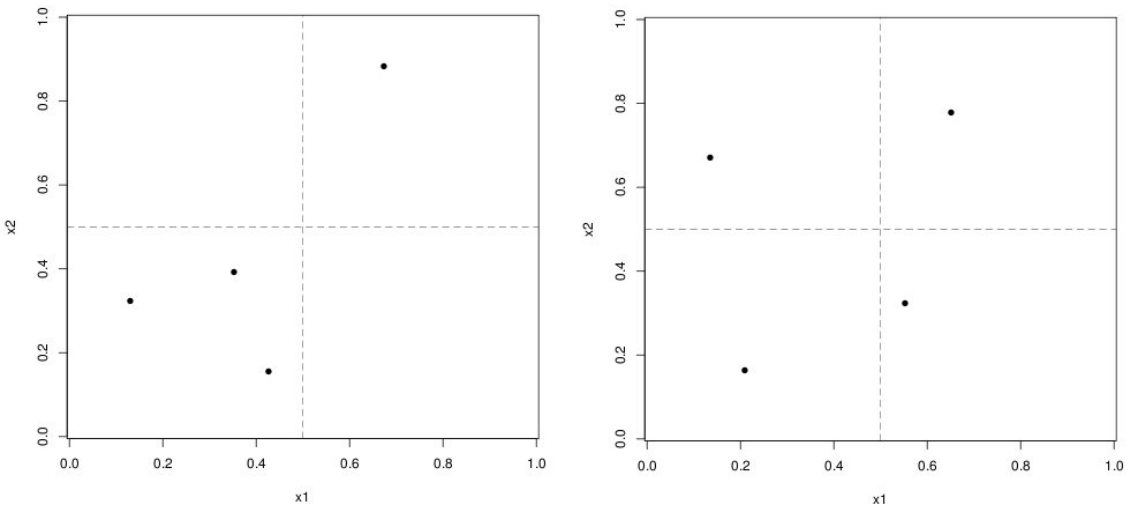

Figure 4.1: Plots of the MC sampling algorithms. Left: Pseudo-MC sample. Right: Stratified-MC sample

4.2 Latin Hypercube Sampling

Latin hypercube sampling (LHS) was originally developed by McKay et al.[9], as an alternative to the random sampling algorithms for computer experiments. They showed that under certain assumptions, LHS gave more accurate estimations of the mean, the variance and the distribution function based on output from a computer code, than the simple Monte-Carlo sampling and the stratified Monte-Carlo sampling algorithms. The basic idea was to construct a design that ensured a more uniform distribution of the samples throughout the design space. It was the first method proposed specifically for applications in computer experiments.

The idea behind the LHS algorithm is based on the definition of the Latin square:

Definition 2 (Latin Square)**.** *A $n \times n$ square grid containing sample points taking n different values is a Latin square if and only if every value appears exactly one*

time for each row and each column.

A	E	C	B	D
D	B	A	C	E
C	A	E	D	B
E	D	B	A	C
B	C	D	E	A

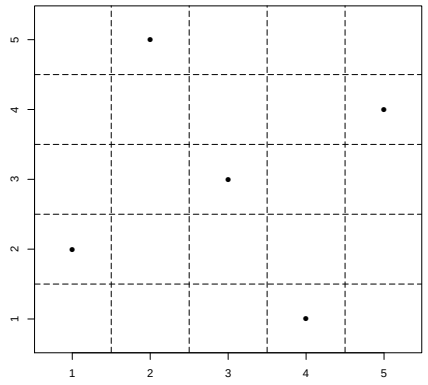

Figure 4.2: Left: Example of a latin square. Right: A latin square sample based on the character E from the latin square to the left.

An example of a 5×5 latin square with character coding can be seen to the left in figure 4.2. The plot at the right shows a latin square sample obtained by taking the sample points from the E-values in the latin square. We could obtain another sample by choosing any of the characters A, B, C or D, or by constructing a different latin square. The definition of the Latin square can be generalized to apply for an arbitrary number of dimensions, and is then called a Latin hypercube. A sample taken from such cube is called a latin hypercube sample.

For constructing a Latin hypercube sample of size n in k dimensions, we partition the range of each variable k into n equally spaced subintervals. The design space will then be divided into a total of n^k bins with equal probability. We then do a random permutation of the values $1, 2, ..., n$ for each variable k, and combine the columns to achieve a $n \times k$ matrix representing a Latin hypercube sample. When applying the LHS algorithm to computer experiments, the design space is usually scaled to the hypercube $[0, 1]^k$ and the samples are placed randomly within each bin.

There are several advantages to LHS that makes it a popular design among many scientists. The LHS algorithm yields design points that are evenly spread out when looking at each dimension separately. Furthermore there are no restrictions

with regard to the number of samples, so the sample size can be chosen exactly according to the researchers need or budget. Many other methods, for example the central composite design and the stratified Monte-Carlo sampling, requires at least 2^k runs, while other designs are only applicable for certain sample sizes n given the number of variables k. LHS scales as $O(k)$, so for experiments where the simulations are too expensive or time consuming to allow a high number of samples, LHS will be a good option.

There are also some drawbacks with LHS that the user should be aware of. When projecting the sample points into one dimension, the samples will have a balanced distribution. When considering multiple dimensions simultaneously however, the randomness in choosing the permutations can lead to designs with good coverage of the design space, but also to designs that leave large areas of the design space unexplored. In figure 4.3 we can see plots of four notably different two-dimensional Latin hypercubes with $n = 8$ sample points. In the upper left plot the samples cover the design space well. For the upper right plot, only a half fraction of the design space is explored. The bottom plots show high spatial correlation, the samples are nearly co-linear. This can lead to ill-conditioned systems, for example when fitting a response surface model by linear regression.

4.2.1 The LHS algorithm

For generating a Latin hypercube sample with n sample points in k dimensions, denoted LHS(n, k), the samples can be found as:

$$x_{ij} = \frac{\pi_{ij} - U_{ij}}{n}, \quad \text{for } 1 \leq i \leq n,\ 1 \leq j \leq k,$$

where each $\pi_{\cdot j}$ is a random permutation of the integers $1, 2, \ldots, n$, and each U_{ij} is a random value drawn from the uniform distribution U$[0, 1]$. The subscript j denotes the dimension index while i represents the number of the experimental sample point.

Example 1 (LHS(n=6,k=2))**.** *We draw two vectors $\boldsymbol{\pi}_{\cdot 1}$ and $\boldsymbol{\pi}_{\cdot 2}$ being random permutations of the sequence [1,2,3,4,5,6] and 12 random numbers from $U[0, 1]$:*

$$\boldsymbol{\pi}_{\cdot 1} = \begin{bmatrix} 5 \\ 2 \\ 4 \\ 6 \\ 3 \\ 1 \end{bmatrix}, \quad \boldsymbol{\pi}_{\cdot 2} = \begin{bmatrix} 2 \\ 1 \\ 6 \\ 4 \\ 5 \\ 3 \end{bmatrix}, \mathbf{U} = \begin{bmatrix} 0.576 & 0.175 \\ 0.240 & 0.374 \\ 0.620 & 0.147 \\ 0.294 & 0.837 \\ 0.877 & 0.616 \\ 0.865 & 0.018 \end{bmatrix}$$

This gives us the LHS design:

$$\frac{1}{n}(\boldsymbol{\pi} - \mathbf{U}) = \frac{1}{n}\left(\begin{bmatrix} 5 & 2 \\ 2 & 1 \\ 4 & 4 \\ 6 & 4 \\ 3 & 5 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} 0.576 & 0.175 \\ 0.240 & 0.374 \\ 0.620 & 0.147 \\ 0.294 & 0.837 \\ 0.877 & 0.616 \\ 0.865 & 0.018 \end{bmatrix}\right) = \begin{bmatrix} 0.737 & 0.284 \\ 0.304 & 0.027 \\ 0.626 & 0.853 \\ 0.937 & 0.563 \\ 0.396 & 0.689 \\ 0.142 & 0.496 \end{bmatrix}$$

A plot of the design can be seen in figure 4.4

Instead of drawing random values for each U_{ij}, one might simply set the value of each U_{ij} to the fixed value 0.5. The sample points will then be placed in the center of each bin, in what is called a midpoint Latin hypercube design.

The LHS algorithm can easily be adopted to variables having non-uniform distributions. If sampling variables from for example the normal-, loguniform- or Weibull-distribution, instead of assigning bins of the same size, one simply assigns bins having the same probability of being drawn out.

4.2.2 Modifications of the LHS algorithm

Several modifications of the LHS algorithm exist that seek to optimize the designs with respect to certain criteria, or avoid designs with high spatial correlation or inadequate coverage of the design space. As the standard LHS algorithm only ensures uniformity for one-dimensional projections, it is desirable to find extensions of the algorithm that will be balanced also in higher dimensions. For each variable in the LHS algorithm there exists $n!$ different permutations. This gives a total of $(n!)^k$ possible LHS designs for given values of n and k, so creating and comparing all designs will be computationally very expensive.

The maximin Latin hypercube design is a space-filling design that seeks to find a Latin hypercube that is optimal with respect to the maximin criterion. A maximin distance design, originally proposed by Johnson et al.[7], is defined as:

Definition 3 (Maximin distance design)**.** *Let $\mathcal{S} \subset \mathbb{R}^k$ be the design space, and $d(x_1, x_2)$ a distance measure on $\mathcal{S}$, for example the Euclidian distance. Let $S_n = \{x_1, \ldots, x_n\}$ denote a possible design. The design S_n^0 is called a maximin distance design if*

$$\max_{\mathcal{S}_n} \min_{x,x' \in \mathcal{S}_n} d(x, x') = \min_{x,x' \in \mathcal{S}_n^0} d(x, x').$$

A drawback to the maximin criteria is that if the number of sample points is smaller than 2^k, the optimal design with respect to the criterion will be one that places all the sample points at the extremes of the design space, i.e. a factorial design. If the number of samples is large, however, a maximin distance design will try to maximize the minimum distance between all design points, thereby seeking a design where the design points are as evenly spread out as possible.

Several other modifications to the LHS algorithm exist, which optimizes LHS with respect to some criteria. Latin hypercube samples can also be constructed based on orthogonal arrays, and this will be furthered discussed in the next section.

4.3 Orthogonal Array Sampling

Tang [13] introduced the idea of using orthogonal arrays (OA's) to construct Latin hypercube samples. He proved that the bias variance, when used for numerical integration, was substantially lower when using LHS based on OA's than for random sampling or the standard LHS algorithm. An important feature of OA based LHD's is that with the strength r associated with the OA, any r-dimensional projection of the design will yield a balanced design. This is an improvement from LHS which is only ensured to be balanced in one dimension.

Tang [13] defines an orthogonal array as follows:

Definition 4 (Orthogonal Array)**.** *An $n \times m$ matrix $\mathbf{A}$, with entries from a set of $p \geq 2$ symbols, is called an OA of strength r, size n, with k constraints (factors) and p levels if each $n \times r$ submatrix of $\mathbf{A}$ contains all possible $1 \times r$ row vectors with the same frequency λ.*

We refer to λ as the index of the array, and the relationship $n = \lambda p^r$ applies for all OA's. We denote such an OA by $\text{OA}(n, k, p, r; \lambda)$.

As mentioned, we associate an OA with the strength r, which reflects the number of dimensions for which the OA is balanced. LHS has strength $r = 1$, and can therfore be seen as an $\text{OA}(n, k, p, 1)$.

From a given OA, a OA based LHS can easily be constructed in a similiar fashion to the LHS algorithm. For each column, instead of drawing a random permutation of all numbers $1, 2, .., n$, we draw for each entry $s = 1, 2, ..., p$ a permutation of the numbers $(s-1)\lambda p^{r-1} + 1, (s-1)\lambda p^{r-1} + 2, ..., s\lambda p^{r-1}$. We then draw the samples randomly from each of the design bins. Tang refers to the resulting LHD's based on OA's as U-designs.

The largest challenge in constructing a U-design consist of finding an appropriate OA for the experiment. The user is not free in choosing whatever combinations of the variables n, k, p, r he would like, and finding a satisfying OA can be very hard. The construction of OA's is a far from trivial process, which constitutes an entirely own field of study, but this will not be further discussed in this report. Fortunately, several OA's for different combinations of n, k exist in tabular form and can be found in books or on the web. Some software programs have algorithms for constructing OA's implemented.

4.3.1 The Orthogonal Array Sampling Algorithm

An OA based LHS, or a U-design, can be contstructed as follows:

- Choose an OA$(n, k, p, r; \lambda)$, denoted by $\mathbf{A}$ with the desired strength r and number of samples n, factors k and levels p. The p levels are taken as p =1,2,...p, where p$\geq$ 2
- For each column in $\mathbf{A}$, replace each of the λk^{r-1} numbers with entry s by a random permutation of $(s-1)\lambda p^{r-1}+1, (s-1)\lambda p^{r-1}+2, ..., s\lambda p^{r-1}$, where $s = 1, ..., p$.
- The elements of an OA based LHD, denoted by $\mathbf{U}$ and scaled to the hypercube $[0,1]^k$, are found as

$$u_{ij} = \frac{a_{ij} - U^*_{ij}}{n}, \quad \text{for } 1 \leq i \leq n,\ 1 \leq j \leq k,$$

  where each U^*_{ij} is a random value drawn from the uniform distribution U$^*[0,1]$.

Example 2 (U(8,2,2,2;2))**.** *We have an OA(8,2,2,2;2) given as:*

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \\ 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{bmatrix}$$

For each of the four 1's in each column we draw a permutation of $\{1,2,3,4\}$*, and for the four 2's a permutation of* $\{5,6,7,8\}$*. This might give the following array:*

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 3 & 7 \\ 5 & 2 \\ 8 & 6 \\ 1 & 4 \\ 4 & 8 \\ 6 & 3 \\ 7 & 5 \end{bmatrix}$$

We randomize within each bin and scale to $[0,1]^2$ *to obtain the U-design:*

$$\mathbf{U} = \frac{1}{n}(\mathbf{A} - \mathbf{U}^*) = \frac{1}{8}\left(\begin{bmatrix} 2 & 1 \\ 3 & 7 \\ 5 & 2 \\ 8 & 6 \\ 1 & 4 \\ 4 & 8 \\ 6 & 3 \\ 7 & 5 \end{bmatrix} - \begin{bmatrix} 0.156 & 0.860 \\ 0.927 & 0.596 \\ 0.629 & 0.405 \\ 0.655 & 0.452 \\ 0.377 & 0.652 \\ 0.714 & 0.491 \\ 0.204 & 0.391 \\ 0.400 & 0.887 \end{bmatrix} \right) = \begin{bmatrix} 0.230 & 0.017 \\ 0.259 & 0.800 \\ 0.546 & 0.199 \\ 0.918 & 0.693 \\ 0.077 & 0.418 \\ 0.410 & 0.938 \\ 0.724 & 0.326 \\ 0.824 & 0.514 \end{bmatrix}$$

A plot of the design can be seen in figure 4.5. We started off with an OA with $p = 2$ *levels for each factor, so the resulting design can be seen as divided into* $2^2 = 4$ *larger bins. As the index* $\lambda = 2$ *we will therefore find exactly two samples within each of these bins. We also see that the design clearly satifyes the LHD criterion in that there is only one sample in each row and each column.*

As with the LHS algorithm, also the OA sampling algorithm can be adopted to variables having non-uniform distributions. The OA used as the basis for the algorithm need not not have the same number of levels p for each factor. If some factors are expected to have a larger influence on the response, it might be reasonable to divide these into more levels than the other factors.

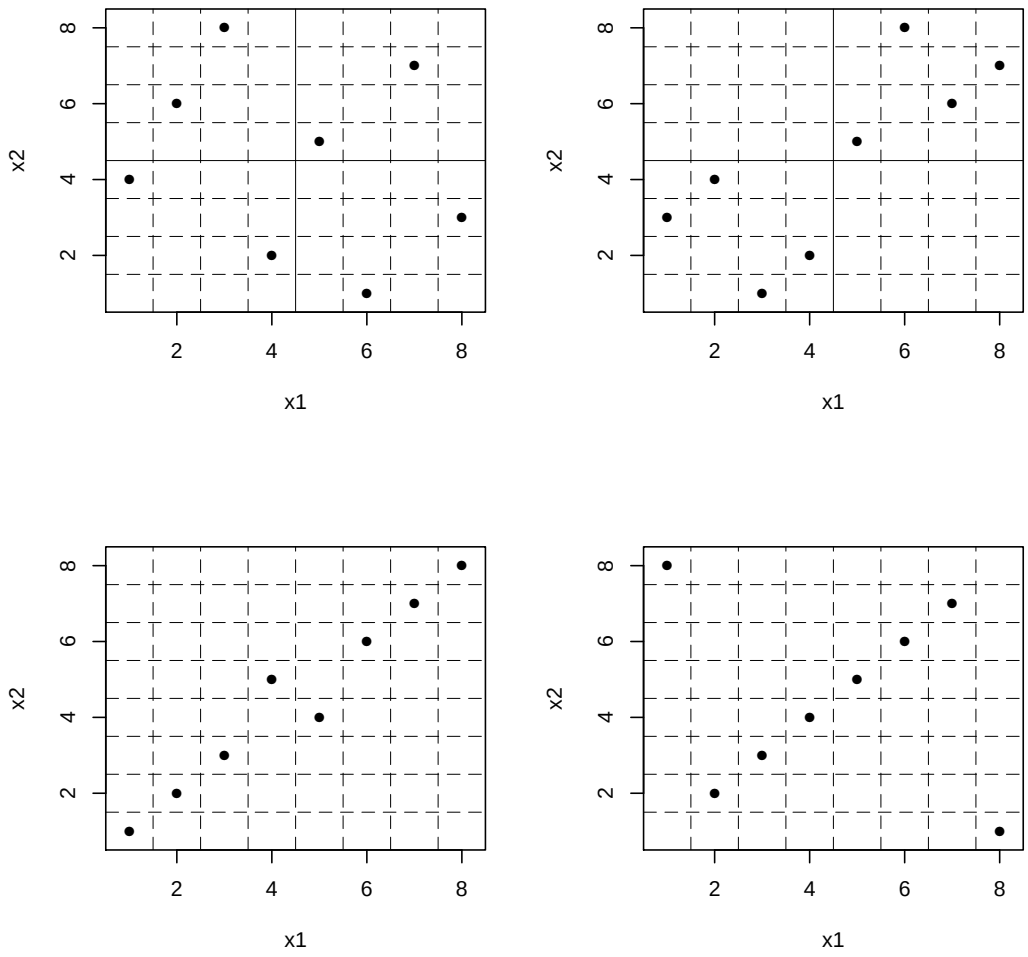

Figure 4.3: LHS plots. Upper left: Good coverage of the design space. Upper right: Only a half-fraction of the design space is covered. Bottom left and right: Samples are co-linear/High spatial correlation.

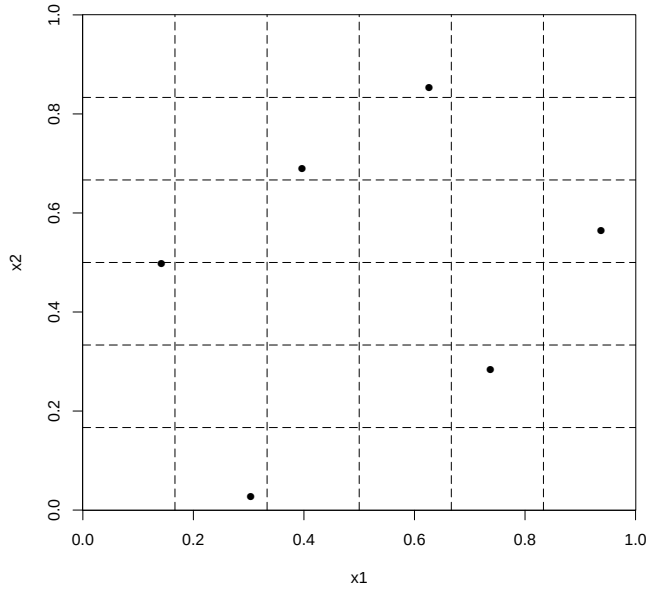

Figure 4.4: Plot of the LHS found in example 1.

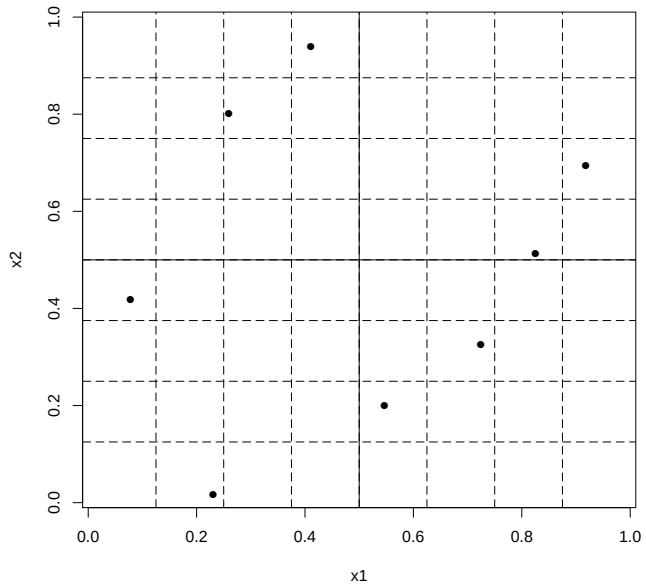

Figure 4.5: Plot of the OA design found in example 2.

Chapter 5

Multi-level Binary Replacement Design

The multi-level binary replacement (MBR) design was developed by Martens et al.[8] for high-dimensional problems in nonlinear systems. Applications include initial range finding, i.e. screening of the experimantal factors, final optimization finding and metamodelling.

When studying a complicated model, for example a system of coupled nonlinear differential equations, it can be hard for the scientist to predict how variations in the input will effect the output. Martens et al.[8] states the following properties for a good experimental design, which they have considered when constructing the MBR design:

- Each factor should be studied at many levels so as to be prepared for nonlinearities in the model behaviour.
- The design should allow testing of many different combinations of the input factors.
- To avoid combinatorial explosion or too high computational cost, it is important to reduce the number of runs as much as possibly, but without missing out on important aspects of the model behaviour. That is, we want to find a design which extracts as much information as possible from as few samples as possible.

Ideally, one would like to test all combinations of all factors at all levels. This will quickly lead to combinatorial explosion; already for an experiment with 5 factors at 8 levels, a total of $8^5 = 32768$ runs are required.

The goal of the MBR design method is as for the LHD and OA design method that they want the experimental samples to be spread evenly over the design space. In a follow-up article, Tøndel et al. [11] compared the MBR design to the CCD and random sampling (MC-sampling). In this report we would like to compare the MBR design with LHD's and OA designs, as the nature of these designs lie closer to the MBR design than does classical designs like the CCD.

In this chapter we will first describe the MBR design contruction approach proposed by Martens et al.[8]. We will then discuss some of the sampling properties of the MBR design in relation to LHD's and OA designs. Finally, we will propose a method do draw permutations of a generated MBR design, so as to construct a LHD based on a MBR design.

5.1 The MBR Design Method

The MBR design method combines binary recording of the factor levels with a fractional factorial design in its binary variables. The approach for constructing the MBR design can be described as follows. Consider an experiment with k factors. Each factor $\kappa \in \{1, 2, \ldots, k\}$ can be assessed at a number of $L(\kappa)$ levels, where $L(\kappa)$ is a multiple of 2. The levels are taken to be $\mathbf{d}_\kappa = (0, 1, \ldots, L(\kappa) - 1)$.

Let $M(\kappa)$ represents the number of binary bits necessary to represent the $L(\kappa)$ levels as binary numbers. We then have the relation $M(\kappa) = \log_2 L(\kappa)$. Let row i in the matrix F_κ represent the binary coding of the corresponding element i in the vector $\mathbf{d}_\kappa$. The elements of F_κ takes the values 0 or 1. When constructing a factorial design, the levels are usually taken to be -1 or +1, so let G_κ be the corresponding matrix to F_κ where all 0-values are shifted to -1. We now have three equivalent representations of the levels of factor κ.

Example 3 (Representation of the factor levels for the MBR design)**.** *Assume that the factor κ takes on $L(\kappa) = 4$ different levels, which can be represented by $M(\kappa) = 2$ factor bits. We then have the equivalent representation of the factor levels:*

$$\mathbf{d}_\kappa = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} \Leftrightarrow F_\kappa = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \Leftrightarrow G_\kappa = \begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}$$

The total number of binary factors necessary to represent the original k factors in

a factorial experiment is given by

$$M_{tot} = \sum_{\kappa=1}^{k} M(\kappa).$$

A full factorial experiment in G would then require a total of $2^{M_{tot}}$ runs. We want to reduce the number of runs to $n = 2^{M_{tot}-M_{conf}}$ in a fractional factorial design, where M_{conf} represents the number of design generators necessary for the desired reduction in sample size. For a chosen confoundation pattern, we can now construct a $2^{M_{tot}-M_{conf}}$ fractional factorial design in G. The design G is then transformed back to the corresponding representation of F. The MBR design is found by transforming F back to the original level coding for each factor.

5.1.1 Choosing a Confounding Pattern for the MBR Design

The crucial step in generating the MBR design lies in choosing the confounding pattern. Each factor κ in the original design is represented by $M(\kappa)$ binary factors in G. For a fractional factorial design one would normally choose the design generators that results in a design of the highest possible resolution, but Martens et al.[8] argues that this will not necessarily give the best MBR-design. They suggest that a simple way to choose a confounding pattern is to generate several patterns, and then choose the design which by graphical inspection seems to give the best spatial coverage.

For a given set of design generators, the alias structure of the design should be checked. Tøndel et al.[11] considered a simple example with two design factors, denoted A and B, investigated at 8 and 4 levels respectively. They chose two different confounding patterns, which gave a optimized and a non-optimized MBR design. The two designs can be seen in figure 5.1. For the non-optimized design, they found that the binary factors for B where confounded only with binary factors from A. They also suggest an optimized MBR design method. Shortly described, an optimal MBR design is found by generating a number of different confounding patterns, and then choose the design which gives the best score with respect to some space-spanning criterion.

5.1.2 The MBR Design Algorithm

A short stepwise summary of the MBR design algorithm can be described as follows.

- Consider an experiment with k factors, each measured on $L(\kappa)$ levels with decimal coding $\mathbf{d}_\kappa = (1, 2, \ldots, L(\kappa) - 1)$, and let $M_{tot} = \sum_{\kappa=1}^{k} M(\kappa)$ be the number of binary factors necessary to represent all factor levels.
- Choose a confounding pattern which gives the desired reduction in design size.
- Construct a fractional factorial design G according to the chosen confounding pattern.
- Transform G to binary representation F, i.e. shift all -1 values to 0.
- Transform the binary factors in F to the corresponding decimal coding to obtain the MBR design, denoted D.

Example 4 (Two factor MBR design)**.** *We would like to reconstruct the example given by Tøndel et al. [11]. Two factors, A and B, are measured on $L(A) = 8$ and $L(B) = 4$ levels. This gives $M(A) = log_2\ 8 = 3$ and $M(B) = log_2\ 4 = 2$. We denote the binary factors for A by a_1, a_2 and a_3, and for B by b_1 and b_2. This gives the relationship between decimal coding and binary coding:*

$$A = a_1 + 2a_2 + 4a_3, \qquad B = b_1 + 2b_2$$

For the optimal design we have the design generators $a_3 = a_1 b_2$ and $b_1 = a_2 b_2$. The 2^{5-2} fractional design G, the corresponding binary representation F and the MBR-design D_{opt} are then found as:

$$\mathbf{G} = \begin{array}{c} \begin{matrix} a_1 & a_2 & a_3 & b_1 & b_2 \end{matrix} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \Rightarrow \mathbf{F} = \begin{array}{c} \begin{matrix} a_1 & a_2 & a_3 & b_1 & b_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \Rightarrow \mathbf{D}_{opt} = \begin{bmatrix} 0 & 2 \\ 4 & 1 \\ 2 & 3 \\ 6 & 0 \\ 1 & 1 \\ 5 & 2 \\ 3 & 0 \\ 7 & 3 \end{bmatrix}$$

For the non-optimal design we have the design generators $b_1 = a_1a_2$ *and* $a_3 = a_2b_2$*. We find* D_{nonopt} *as:*

$$\mathbf{G} = \begin{array}{c} \begin{matrix} a_1 & a_2 & a_3 & b_1 & b_2 \end{matrix} \\ \begin{bmatrix} -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \Rightarrow \mathbf{F} = \begin{array}{c} \begin{matrix} a_1 & a_2 & a_3 & b_1 & b_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \Rightarrow \mathbf{D}_{nonopt} = \begin{bmatrix} 0 & 3 \\ 4 & 1 \\ 2 & 0 \\ 6 & 2 \\ 1 & 2 \\ 5 & 0 \\ 3 & 1 \\ 7 & 3 \end{bmatrix}$$

A plot of the optimal and the non-optimal MBR design can be seen in figure 5.1. The optimal design seems to have better spatial coverage.

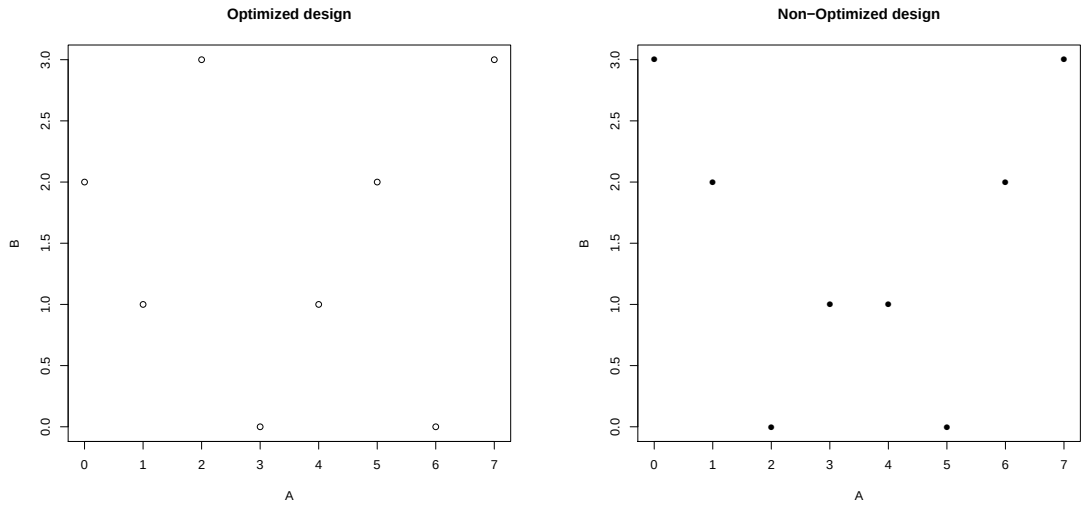

Figure 5.1: Two-factor MBR designs based on different confounding patterns. Left: Optimal MBR design. Right: Non-optimal MBR design.

5.2 Sampling Properties of the MBR Design

There are both advantages and disadvantages to the MBR design compared to LHS and OA sampling. The MBR method offer good possibilities to generate designs with few samples, but many factors, while maintaining control over the factor interactions. However, the number of samples we can choose are restricted to be multiples of 2. For the LHD, no such restrictions apply. OA designs require that an OA exists and can be found for the given number of samples, factors and levels.

Compared to the OA and MBR design methods, the LHD algorithm is easier to implement. The MBR design algorithm requires that we choose a confounding pattern, which can strongly affect the designs space-filling properties. Optimized MBR designs can be generated, but require that some criteria is chosen as basis for comparison.

An advantage of the MBR design method is that known or anticipated information about the factor correlations can be taken into consideration when choosing the design generators. If some factors are believed to be more influential than the others, we can choose to assess these at a higher number of levels. The LHD algorithm offers no possibilities to incorporate parameter correlation when constructing a design, nor to assess the factors at a different number of levels. For the OA-sampling algorithm one might choose an OA which has more levels for some of the factors.

An important aspect of the MBR design is that the design levels need not represent experimental levels which are equally spanned, or even continuous. If a factor can only be assessed at certain predetermined levels, randomization within bins will not be possible. The LHS and OA algorithms considers the design space to be continuous, and are therfore not directly applicable if the factor levels are discrete.

5.3 MBR based Latin Hypercube Designs

For comparisons between the LHD, OA design and MBR design, the first two design methods will have a clear advantage as they are not restricted to only certain levels when assessing the factors. We will therfore suggest an extension to the current MBR design algorithm, such that also the MBR designs are free to assess the factors at all levels.

The LHS algorithm divides the span of each factor into a number of bins equal to

the number of design samples, and then randomize within each bin. For the MBR design, the number of samples will usually be higher or equal to the number of levels for which each factor is investigated.

If one or more of the factors in the MBR design are assessed at a number of levels lower than the number of samples, we may draw permutations within these factors in a similar fashion to the OA sampling algorithm.

Consider for example a MBR design generated with n sample points and k factors, where each factor $\kappa \in \{1, 2, \ldots, k\}$ is measured at $L(\kappa)$ levels. Assume that $n > L(\kappa)\ \forall\ \kappa$. As both n and $L(\kappa)$ are multiples of 2, the value $\lambda(\kappa) = \frac{n}{L(\kappa)}$ will be an integer value representing the number of samples measured at each level for factor κ. By replacing the $\lambda(\kappa)$ samples with entry $s = 0, 1, 2, ..., L(\kappa) - 1$ with a random permutation of the numbers $s\lambda(\kappa) + 1, s\lambda(\kappa) + 2, ..., (s + 1)\lambda(\kappa)$, we will obtain a design which is a LHD, but is based on a MBR design. By randomizing within in each bin and scaling the design to the hypercube $[0, 1]^k$, we have a MBR based LHD which can be compared to convential LHD's and OA designs.

5.3.1 Algorithm for Constructing a MBR based LHD

An algorithm for generating a LHD based on a MBR design can be described as follows.

- Generate a MBR design, denoted D, with n samples and k factors, according to the algorithm described in section 5.1.2. Each factor κ is measured at $L(\kappa)$ levels.
- For each column in D with $\lambda(\kappa) = \frac{n}{L(\kappa)} > 1$, replace each of the $L(\kappa)$ numbers with entry $s = 0, 1, 2, ..., L(\kappa) - 1$ by a random permutation of $s\lambda(\kappa) + 1, s\lambda(\kappa) + 2, ..., (s + 1)\lambda(\kappa)$.
- The elements of a MBR based LHD, denoted by D* and scaled to the hypercube $[0, 1]^k$, can be found as

$$d^*_{ij} = \frac{d_{ij} - U_{ij}}{n}, \quad \text{for } 1 \leq i \leq n,\ 1 \leq j \leq k,$$

  where each U_{ij} is a random value drawn from the uniform distribution $\mathrm{U}^*[0, 1]$.

Example 5 (MBR based LHD)**.** *For $n = 16$ samples and $k = 2$ factors, with each factor measured at $L(\kappa) = 8$ levels we have generated a MBR design using*

the design generators $b_2 = a_1\ a_2\ a_3$ *and* $b_3 = a_1\ a_2\ b_1$*. The design is given as*

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 4 & 4 & 2 & 2 & 6 & 6 & 1 & 1 & 5 & 5 & 3 & 3 & 7 & 7 \\ 0 & 5 & 6 & 3 & 6 & 3 & 0 & 5 & 2 & 7 & 4 & 1 & 4 & 1 & 2 & 7 \end{bmatrix}^T.$$

A plot of the design can be seen to the left in figure 5.2. We draw permutations according to the algorithm described, and obtain:

$$\mathbf{D} = \begin{bmatrix} 1 & 2 & 9 & 10 & 5 & 6 & 13 & 14 & 4 & 3 & 12 & 11 & 7 & 8 & 15 & 16 \\ 2 & 11 & 14 & 8 & 13 & 7 & 1 & 12 & 5 & 16 & 9 & 3 & 10 & 4 & 6 & 15 \end{bmatrix}^T.$$

We randomize within each bin and scale to $[0,1]^k$*, and get the MBR based LHD:*

$$D^* = \frac{1}{n}(D - U) = \frac{1}{16}\left(\begin{bmatrix} 1 & 2 \\ 2 & 11 \\ 9 & 14 \\ 10 & 8 \\ 5 & 13 \\ 6 & 7 \\ 13 & 1 \\ 14 & 12 \\ 4 & 5 \\ 3 & 16 \\ 12 & 9 \\ 11 & 3 \\ 7 & 10 \\ 8 & 4 \\ 15 & 6 \\ 16 & 15 \end{bmatrix} - \begin{bmatrix} 0.139 & 0.501 \\ 0.916 & 0.191 \\ 0.162 & 0.333 \\ 0.322 & 0.628 \\ 0.567 & 0.776 \\ 0.102 & 0.134 \\ 0.113 & 0.372 \\ 0.846 & 0.738 \\ 0.747 & 0.318 \\ 0.634 & 0.721 \\ 0.664 & 0.833 \\ 0.106 & 0.054 \\ 0.342 & 0.954 \\ 0.156 & 0.686 \\ 0.095 & 0.809 \\ 0.818 & 0.393 \end{bmatrix}\right) = \begin{bmatrix} 0.054 & 0.094 \\ 0.068 & 0.676 \\ 0.552 & 0.854 \\ 0.605 & 0.461 \\ 0.277 & 0.764 \\ 0.369 & 0.429 \\ 0.805 & 0.039 \\ 0.822 & 0.704 \\ 0.203 & 0.293 \\ 0.148 & 0.955 \\ 0.708 & 0.510 \\ 0.681 & 0.184 \\ 0.416 & 0.565 \\ 0.490 & 0.207 \\ 0.932 & 0.324 \\ 0.949 & 0.913 \end{bmatrix}$$

A plot of the design can be seen to the right in figure 5.2. The MBR based LHD, D*, are marked by the solid dots, while the original MBR design D scaled to $[0,1]^k$ is represented by the crosses.

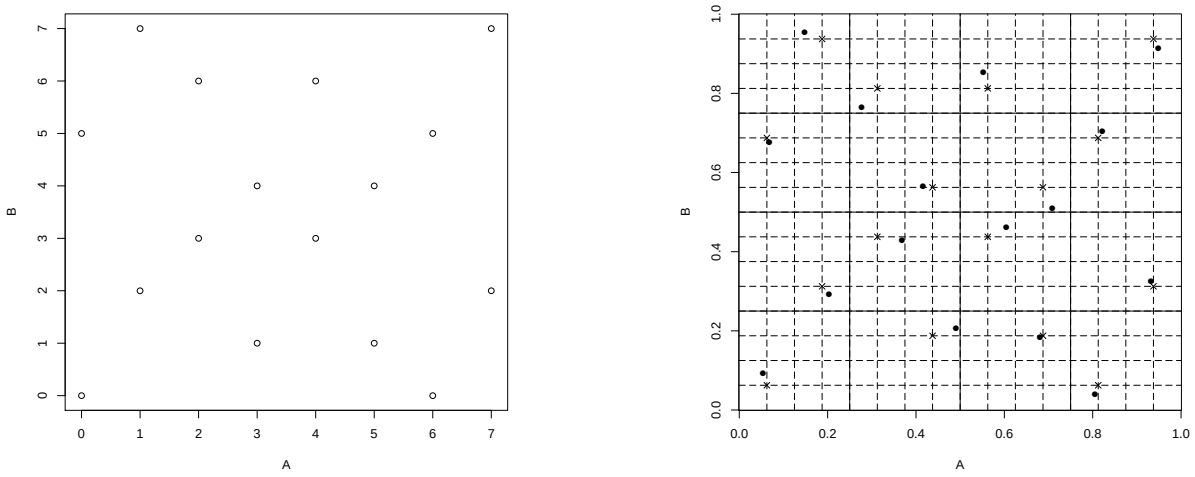

Figure 5.2: Left: The MBR design D from example 5. Right: An MBR based LHD, D*, originating from the design to the left. D* is marked by the filled dots, the original design D is marked by the crosses.

Chapter 6

Comparison Criteria for Evaluation of Designs

There are several intuitive goals that are important when evaluating experimental designs. Classical physical experiments are mainly concerned about reducing the variance error. In modern computer experiments there is no random error and the main focus is to reduce the bias error. One tend to favour designs which are space filling, and natural requirements for an efficient design include good coverage of the design space, ability to fit complex models, many levels for each factor and good projection properties.

For low factor experiments the coverage and projection properties can be checked manually by graphical representation, but this might not be so easy when the number of factors is large. One is interested in finding mathematical criteria which reflect a designs space-filling properties and ability to fit metamodels, and whom can be used to compare different DoE methods.

In this chapter we will first present some of the most popular classical design criteria, namely the D-optimality and A-optimality criterion. We will give a short explanation as to why these, and other classical criteria, are less applicable for use in computer experiments. We will then present some criteria that are more suitable when evaluating modern design methods: The maximin distance criteria, which is often used when constructing optimal designs. The integrated mean squared error (IMSE), and the closely related root mean squared error (RMSE) and maximum absolute error (Max). Finally the alias sum of squares criterion will be presented.

6.1 Classical Comparison Criteria

The goal of classical DoE is to reduce the variance error, and classical criteria are typically related to properties of the variance-covariance matrix or the Fisher information matrix.

6.1.1 D-criterion

Given a design $\mathcal{D}$ with an associated design matrix X, a D-optimal design seeks to maximize the determinant of the information matrix $M = (X'X)$. The information matrix is proportional to the inverse of the covariance matrix for the least squares estimator, so by maximizing M we seek to minimize the variance.

The D-criterion, denoted D_{crit}, is defined by Montgomery et al. [10] as:

$$D_{crit} = \frac{1}{|M^{-1}|} = \frac{1}{|(X'X)^{-1}|}$$

where one would prefer the design with the highest value of D_{crit}.

6.1.2 A-criterion

The A-optimality criterion is also based on the information matrix M, and is defined by Montgomery et al. [10] as:

$$A_{crit} = tr(M^{-1}) = tr[(X'X)^{-1}]$$

where tr is the trace of the matrix, i. e. the sum of the diagonal elements. An optimal design would seek the lowest value possible of A_{crit}.

Whereas the D-optmality criterion is based on the whole covariance matrix, the A-optimality criterion is only the sum of the variances of the estimated parameters.

6.1.3 Other Classical Criteria

Other popular classical design criteria include V-optimality [1], the average prediction variance, G-optimality [10], the maximum prediction variance, and the condition number [3], which evaluates the sphericity and symmetry of a design.

6.1.4 Classical Criteria for Evaluating Modern DoE Methods

An important aspect of modern design methods is to reduce the bias error by seeking space-filling designs. But placing several points in the interior of the

design space will naturally increase the correlation between the samples. Design criteria which are concerned about minimizing the variance may therefore be bad measures when comparing modern design methods, as the criteria generally will prefer designs that are non space-filling.

6.2 Maximin Criterion

The maximin criterion for a design $\mathcal{D}$, in a slightly modified form presented by Bursztyn et al.[2], is given as:

$$d_{maximin} = \min_{x_i, x_j \in \mathcal{D}} d(x_i, x_j) = \min_{x_i, x_j \in \mathcal{D}} \parallel x_i - x_j \parallel \tag{6.1}$$

i. e. $d_{maximin}$ is the minimum distance between two design points found for all design points in $\mathcal{D}$. As mentioned in section 4.2.2, LHD's can be constructed which are optimal with respect to the maximin criterion.

A large value of the maximin criterion will indicate that the points are spread out as far apart as possible, therby indicating good space filling properties. Caution however has to be taken if the number of samples n is less than 2^k, as an optimal design in this case will be a full or fractional design. But a high maximin criterion value paired with another requirement, for example that the design satisfies the LHS-criterion, indicates a design with good coverage of the design space.

6.3 The IMSE, RMSE and Max Criteria

The integrated mean squared error (IMSE), root mean squared error (RMSE) and maximum absolute error (Max) criteria are measures as to how well an estimated metamodel $\hat{y}(x)$ based on a design $\mathcal{D}$ fits the true underlying model $y(x)$. More information on finding these metamodels are presented in chapter 7.

After fitting a metamodel we want to measure the deviance from the true model. The IMSE criterion is defined by Bursztyn et al. [2] as follows:

$$IMSE = \int E\{y(x) - \hat{y}(x)\}^2 dx.$$

For complex functions $y(x)$ the exact value of the IMSE criterion can be hard to calulate, and numerical approximations might be necessary.

The RMSE criteriom is defined by Simpson et al.[12] as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n_{error}} (y_i - \hat{y}_i)^2}{n_{error}}}.$$

Here n_{error} represents the number of validation points. These can be placed along a uniformly spaced grid or chosen randomly within the design space. The RMSE value therfore represents an approximative value of the IMSE, but it is computationally much cheaper and will therfore be preferred when evaluating different design methods in chapter 8.

The maximum absolute error (Max) criterion is defined by Simpson et al.[12] as:

$$\text{Max} = \max_{i=1,..,n_{error}} (|y_i - \hat{y}_i|).$$

The maximum is taken over a set of n_{error} validation points. If using the same set of validation points for the RMSE and Max criterion, the two criteria values can be calculated simultaneously. The RMSE represents an average value of the deviance between the true model and the metamodel, whilst the Max criterion represents the maximum deviance.

In addition to the application of comparing different design methods, the RMSE and Max values are often used as measures as to how good different metamodels perform.

6.4 Alias Sum of Squares Criterion

The alias sum of squares criterion (ASSC) was originally proposed by Bursztyn et al. [2]. They found that the criteria had an overall good agreement with the IMSE criteria, but with the advantage of beeing computationally much cheaper. The basic idea was to find a criteria that reflects both a designs efficiency in factor screening and its flexibility to fit more complex models in those factors that are active. We will here give a short presentation of how the criterion value is derived. For further details, see the original article by Bursztyn et al. [2].

Suppose we start by using the ordinary least squares estimator to fit an approximative first-order regression model:

$$\mathbf{y}_i = \beta_0 + \sum_{j=1}^{k} \beta_j \mathbf{x}_{ij} = \mathbf{f}(\mathbf{x}_i)\boldsymbol{\beta}.$$

This will be a natural first step in factor screening. The assumption now is that the first order model will not give a sufficiently good description of the output data. We fit a new model with added higher order terms:

$$\mathbf{y}_i = \beta_0 + \sum_{j=1}^{k} \beta_j \mathbf{x}_{ij} + \sum_{j=k+1} \beta_j \mathbf{f}_j(\mathbf{x}_i) = \mathbf{f}'(\mathbf{x}_i)\boldsymbol{\beta} + \mathbf{f}_2'(\mathbf{x}_i)\boldsymbol{\beta}_2.$$

The full model can be written in matrix form as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{f}'(\mathbf{x}_1) \\ \vdots \\ \mathbf{f}'(\mathbf{x}_n) \end{bmatrix} \boldsymbol{\beta} + \begin{bmatrix} \mathbf{f}_2'(\mathbf{x}_1) \\ \vdots \\ \mathbf{f}_2'(\mathbf{x}_n) \end{bmatrix} \boldsymbol{\beta}_2 = X\boldsymbol{\beta} + X_2\boldsymbol{\beta}_2.$$

The expression of the least square estimator is thereby found as:

$$\hat{\boldsymbol{\beta}} = (X'X)^{-1}X'\mathbf{y} = \boldsymbol{\beta} + (X'X)^{-1}X'X_2\boldsymbol{\beta}_2 = \boldsymbol{\beta} + A\boldsymbol{\beta}_2,$$

where A is called the alias matrix. From the alias matrix we can get an idea as to how much the estimates of the first order model are biased by the extra terms added.

By assuming that $\boldsymbol{\beta}_2$ is random with normal distribution $\mathrm{N}(\mathbf{0}, \sigma_\beta^2)$, the variance of $\hat{\boldsymbol{\beta}}$ is given as:

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \mathrm{Var}(\boldsymbol{\beta} + A\boldsymbol{\beta}_2) = A\ \mathrm{Var}(\boldsymbol{\beta}_2)\ A' = \sigma_\beta^2\ AA'\ \propto\ AA'.$$

We can now apply a classical variance-based criteria, like A-optimality or D-optimality, to measure the extent of how higher-order bias affects the simple approximation. (ref Bursztyn) suggests using the A-optimality criterion for $\mathrm{Var}(\hat{\boldsymbol{\beta}})$. This gives us the the A-optimality criterion for the alias matrix, denoted ASSC(a), as:

$$\mathrm{ASSC(a)} = tr(\mathrm{Var}(\hat{\boldsymbol{\beta}}))\ \propto\ tr(AA') = \sum_{i,j} a_{ij}^2,$$

where a_{ij} represents the ij'th element of the alias matrix. As the criterion is used to compare different design methods, we need not include the proportionality constant σ_β^2 in the calculations.

Chapter 7

Metamodelling

An important objective in experiments is to estimate how the process is affected by the experimental factors. An approximation model predicting how the output variables depend on the input variables is called a metamodel. It provides a link between the function parameters and the response over the entire experimental domain.

In computer experiments, the true underlying function might be complex, and it is desirable to fit a simpler surrogate model so as to better understand which variables or factors have a significant effect on the response, and how the variables are correlated. A variety of different metamodelling techniques exist for constructing such models.

In this chapter, response surface models and kriging models will be shortly presented. Other popular approximation models for computer experiments include multivariate adaptive regression splines [6] and radial basis function approximations [4], but these will not be discussed in this report.

7.1 Responce Surface Models

Response surface models have a wide variety of applications, and are popular in both classical and computer experiments. A first order response surface model has the form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k, \tag{7.1}$$

where $\hat{y}$ represents the estimated response for the input parameters $x_1, x_2, ..., x_k$. The parameters β are a set of unknown parameters which needs to be estimated.

A second order polynomial response surface model is given by:

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_i \sum_j \beta_{ij} x_i x_j. \tag{7.2}$$

The parameters β are found by least squares regression, and are given as:

$$\hat{\boldsymbol{\beta}} = (X'X)^{-1} X'\mathbf{y}.$$

Here X is the design matrix of the experimental sample points, and $\mathbf{y}$ the vector of responses at each sample point.

First order models are useful for screening experiments or if only a small number of samples can be afforded. If nonlinear behaviour is expected one should try to fit a higher order response surface model, e.g. a second or third order model, according to budget, need and number of experimental factors. Response surface models can easily be constructed, but for functions with highly nonlinear or irregular behaviour, the models might be to simple. For more information on response surface methodology, we refer to Montgomery and Myers [10].

7.2 Kriging Models

The kriging model is defined as:

$$y(\mathbf{x}) = \sum_{j=1}^{k} \beta_j f_j(\mathbf{x}) + z(\mathbf{x}),$$

where $f_j(x)$ is a chosen basis over the design space, often taken as a constant term. $z(x)$ is assumed to be a stochastic process with mean zero and covariance function

$$\mathrm{Cov}(z(\mathbf{x}_i), z(\mathbf{x}_j)) = \sigma^2 \mathrm{R}(\mathbf{x}_i, \mathbf{x}_j),$$

where σ^2 is the process variance and R is the correlation function. A typical choice of R is the Gaussian correlation function:

$$\mathrm{R}(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j) = \exp\{-\sum_{l=1}^{k} \theta_j (x_{i,l} - x_{j,l})^2\}.$$

The parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$ needs to be estimated. The estimation of these parameters can be computationally very demanding. The kriging approximation model $\hat{y}$ is found by the best linear unbiased predictor (BLUP) of $y(\mathbf{x})$. For further

information on Kriging models and details on the BLUP and the estimation of $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$, we refer to Fang et al.[5]

Although kriging models are computationally much more demanding than response surface models, they are much better adapted to provide accurate predictions of highly nonlinear behaviour.

Chapter 8

Comparison of Modern DoE Methods

We have performed a comparison study of the LHD, the OA design and the MBR design. We have considered different design criteria and evaluated the score of the design methods. MC-sampling or random sampling designs, hereby referred to as random designs, have also been included for comparison. We have first conducted an initial experiment with only two factors and 8 or 16 samples. We have then considered a higher-order experiment with five design factors and 16, 32, 64 and 128 design samples. All computations and implementation of designs have been done using the statistical software R.

Our main focus in the comparison have been the RMSE, Max and alias criteria, as these criteria reflect important characteristics of modern DoE methods. The maximin criterion have also been computed for the high-factor experiments. Whereas the maximin criterion can be computed directly from a design, the RMSE, Max and alias criteria demand that some assumptions are made.

The alias criterion measures the skewness of a design if a simple first- or second-order response surface model is to be fitted, but the true model is that of one or two degrees higher. The order of the true and estimated model needs to be decided before the alias criterion value can be computed.

The RMSE criterion measures a designs ability to fit metamodels directly, in the sense that it computes the average deviation between the true model and the estimated metamodel over a set of validation points. The Max-criterion returns the maximal deviation found over the set of validation points. To obtain the RMSE and Max criteria values we need a computer model that generates responses based on the design samples. We need also decide what type of metamodel is to be

fitted. In this report we have only considered response surface models, where the polynomial order of the fitted models are adjusted according to the number of factors and experimental samples. The underlying computer models are chosen as polynomials with interactions of one or two degrees higher than the fitted model.

8.1 Two Factor Designs

8.1.1 Designs with n=8 Experimental Points

We have first considered experimental designs with two factors and 8 sample points. For the MBR designs, we have considered the two factors on 8 and 4 levels respectively. The design generators were chosen on the form given in example 4, giving designs of resolution III. In total we found 12 different sets of design generators that yielded acceptable MBR designs. For some choices of design generators, the resulting MBR designs were only measured on 4 levels for the 8-level factor, and these designs were therfore rejected.

For comparison we constructed 12 different LHD's, OA designs and random designs. Whereas the MBR design is restricted to 8 and 4 levels in the experimental region, the other designs have no such restrictions, and consider the design space to be continuous. All designs were scaled down to the design space $[0, 1]^2$.

The results for the RMSE and Max criteria can be seen in table 8.1 and table 8.2 respectively. The criteria were computed using a set of 100 validation points spread evenly over a grid in the design space. The exact representation of the true underlying function and the metamodels used are given in section A.1 of the appendix. The tables contains the minimum, maximum, median and mean value of the criteria found for the 12 designs generated for each method. We have also included the criteria values found for the optimal and non-optimal MBR designs given in example 4.

For the RMSE criterion the MBR method gave better mean values than both the random designs and the LHD's, whilst the OA designs gave the lowest scores for the mean value. The best random design actually obtained a lower RMSE score than the best MBR design, but the worst random design gave a very high score compared to all the other design methods. Also for the LHD the worst design with respect to the RMSE criterion gave a substantially higher value than both the OA designs and MBR designs, which had the lowest variance in the criterion score.

The same general trend can be seen for the Max criterion. The OA designs ob-

tained the best mean criterion score. The minimum score was actually obtained for a random design, but the random designs still gave the highest mean and maximum score. The MBR designs performed worse than the OA-design, but slightly better than the LHD's.

Compared to all the 12 MBR designs evaluated, the MBR design found as optimal by Tøndel et al.[11] obtained about average scores with respect to both criteria, whilst the non-optimal MBR design gave the worst score for the RMSE criterion, but an average score for the Max criterion.

Figure 8.1 shows plots of three experimental designs for each of the design methods. For the MBR designs at the top, we see that the space-filling properties vary quite a lot for the different choices of design generators. The MBR design to the left looks good, the MBR design in the middle leaves some regions unexplored while the MBR design to the right shows clear linear trends. The LHD's and OA designs have quite good space filling properties, but some of the designs leave unexplored areas. For the random designs, both the design in the middle and to the left are clearly non space-filling, while the design to the right seems to cover the design space well.

	LHS	OA	Random	MBRD	MBRD(opt)	MBRD(non-opt)
min	0.13070	0.11540	0.13587	0.14607		
max	0.63739	0.23717	1.13069	0.27496		
median	0.15760	0.17393	0.22638	0.17129		
mean	0.20798	0.17067	0.29700	0.18563	0.17270	0.27496

Table 8.1: Two factor designs with $n = 8$ samples: Results for the RMSE criterion.

	LHS	OA	Random	MBRD	MBRD(opt)	MBRD(non-opt)
min	0.53970	0.53567	0.42657	0.46980		
max	1.40151	1.02316	2.68916	0.99741		
median	0.80243	0.64808	0.75250	0.77877		
mean	0.81244	0.70061	0.95642	0.79002	0.92245	0.77877

Table 8.2: Two factor designs with $n = 8$ samples: Results for the Max criterion.

8.1.2 Designs with n=16 Experimental Points

In the previous section the MBR designs were restricted to certain levels, which gave it a disadvantage when compared to the other designs methods. We would like to conduct a comparison where all the design methods have the same constraints with respect to available levels. We have therfore considered experimental designs with two factors and 16 runs, where each of the factors can be assessed at 8 levels. The design generators for construction of the MBR design are given in section A.2 in the appendix. A total of 45 sets of design generators which yielded unique designs were found and used in the comparison.

For the LHD and OA algorithms, the designs were first constructed regularly, but without randomization within bins. The designs were than collapsed from the original 16 levels to only 8 levels. This might result in replicated experimental points, which is of no use in computer experiments without random error. If a design had replicated points, it was therefore rejected and a new sample was drawn. OA designs were constructed based on OA's with both $p = 2$ and $p = 4$ levels. For the random designs, each of the samples was randomly chosen from one of the 8 levels.

The results for the RMSE, Max and alias criteria are given in table 8.3, table 8.4 and table 8.5 respectively. 45 different designs for each design method was generated. For both the RMSE and Max criteria the MBR designs gave lower scores than all the other designs for the minimum, maximum, mean and median values. The LHD's and OA(p=2) designs gave very similar scores. As expected, the OA(p=4) designs scored better than the OA(p=2) designs, and the random designs gave the worst scores. The maximum and minimum scores for the MBR design varied very little, which indicates that the choice of the design generators was less important for this experiment, as all the designs performed almost equally good.

For the alias criterion, all 45 MBR designs actually obtained the same criterion value. On the average, the LHD's, OA designs and MBR designs obtained very similar scores. The random designs had both the lowest minimum and the highest maximum score, but performed worse than the other designs with respect to the mean criterion value.

	LHS	OA(p=2)	OA(p=4)	Random	MBRD
min	0.11002	0.11114	0.10862	0.11088	0.10655
max	0.17977	0.17766	0.13923	0.36790	0.11215
median	0.12912	0.13100	0.12060	0.13829	0.10783
mean	0.13750	0.13567	0.12112	0.15519	0.10910

Table 8.3: Two factor designs with $n = 16$ samples: Results for the RMSE criterion.

	LHS	OA(p=2)	OA(p=4)	Random	MBRD
min	0.35856	0.36877	0.47589	0.36504	0.32385
max	0.98439	0.93095	0.81323	1.10884	0.38932
median	0.62294	0.62242	0.60178	0.65074	0.37295
mean	0.63095	0.62857	0.59816	0.68151	0.36551

Table 8.4: Two factor designs with $n = 16$ samples: Results for the Max criterion.

8.2 Five Factor Designs

We have conducted two comparison studies of five factor designs. In the first comparison, all DoE methods are restricted to 8 levels for each factor. These levels might represent factor levels which are unevenly spaced. In the second study, the factor levels are assumed to be continuous and assessible over the entire design space $[0, 1]^5$. We will hereafter refer to the two studies as non-randomized and randomized respectively. For both studies we have considered designs with $n = 16, 32, 64$ and 128 experimental samples. For the randomized comparison study, also the maximin LHD is included.

The different designs have been constructed in the following manner.

- Random design. Non-randomized: All samples points were drawn randomly from the set $\{1, 2, ..., 8\}$, and scaled to $[0, 1]^5$.
  Randomized: According to the description in section 4.1.1.
- LHD. Randomized: According to the algorithm described in section 4.2.1.
  Non-randomized: As above, but without randomization within bins. The design levels were then merged such that the resulting design was measured at exactly 8 levels.
- OA design. Randomized: According to the algorithm described in section 4.3.1. OA's with $p = 2$ or $p = 4$ and the highest possible strength according

	LHS	OA(p=2)	OA(p=4)	Random	MBRD
min	2.36007	2.36323	2.45267	1.97414	2.61893
max	3.16226	3.03686	2.79613	3.68306	2.61893
median	2.60600	2.62571	2.61663	2.69055	2.61893
mean	2.65505	2.66068	2.61795	2.76187	2.61893

Table 8.5: Two factor designs with $n = 16$ samples: Results for the ASSC.

to the number of levels and samples was chosen as basis for the algorithm. Non-randomized: As above, but without randomization within bins. The design levels were merged such that the resulting design was measured at exactly 8 levels.

- MBR design. Non-randomized: According to the algorithm described in section 5.1.2, with 8 levels for each factor. The choice of design generators for the different sample sizes are given in section A.2 of the appendix. Randomized: For all n, we generated MBR-designs as described above with 8 levels for each factor, and used these as a basis for drawing permutations according the algorithm described in section 5.3.1.

- Maximin LHS. Randomized: The design was created using the function maximinLHS from the lhs-package in R. The design seeks to find a LHD which is optimal with respect to the maximin distance criteria given in definition 3.

For each design method and each $n = 16, 32, 64$ and 128, we generated 100 designs for comparison. For the LHD's, OA designs and random designs, replicated sample points might occur for the non-randomized designs. If a design did not have n unique samples, the design was rejected and a new design generated.

The OA's used as basis for the OA algorithm were generated using the function oa.design available in the DoE.base-package in R, and then checked for correctness. We were not able to find OA's of strength $r = 3$ for the OA-designs on $p = 4$ levels with $n = 64$ and 128 samples, which should teoretically exist according to the relationship $n = \lambda p^r$. Instead, OA's of strength $r = 2$ with index $\lambda = 4$ and 8 for $n = 64$ and 128 respectively were generated.

The criteria were computed under the following assumptions:

- RMSE/Max. True underlying function: a polynomial with squared terms terms and interaction terms up to third order.

Metamodel: Response surface model. For $n = 16$, a simple first order model was fitted. For $n = 32, 64$ and 128 a second order model was fitted. The exact representation of the true underlying function and the metamodels are given in section A.1 of the appendix.
Validation points: 1000 points drawn randomly within the design space.

- Alias sum of squares criteria (ASSC). For $n = 16$ we used a first-order model as the simple model and a full second order model as the extra terms. For $n = 32, 64$ and 128 we used a second order model as the simple model and extra terms of order three.

- Maximin criterion: Given directly from equation 6.1.

8.2.1 Non-randomized

In the non-randomized comparison study all designs are measured on exactly 8 levels for each factor.

In figure 8.2 and 8.3 the results for the different design methods and criteria have been plotted against each other. Each point in the plot is the mean criterion value of 100 designs for each design method and each number of samples $n = 16, 32, 64$ and 128. The notation OA($p = 2$) and OA($p = 4$) denotes OA designs based on OA's with 2 and 4 levels respectively. We start by looking at the plot for the RMSE criterion at the top of figure 8.2. For $n = 16$ and 32, the general trend is that OA($p = 4$) performs the best, followed by OA($p = 2$), LHD and at last random designs. The MBR design scores somewhere between the OA($p = 2$) and the random design. Especially for $n = 32$, the MBR design performs only slightly better than the random design. For $n = 64$ and 128 the OA($p = 4$) gets a very high score, and performs even worse than the random designs. The MBR design gets the lowest RMSE scores for $n = 64$ and 128.

The Max criterion, plotted at the bottom of figure 8.2, shows the same trend as the RMSE criterion. This is not surprising as the two criteria are closely related. However, for $n = 16$ all design methods perform almost equally good, except for the random design. For $n = 32$, the MBR design gets about the same score as the LHD. The result for the ASSC are plotted at the top of figure 8.3. The highest criterion value for all sample sizes are found for the random design, followed by the LHD and the OA($p = 2$) design. The best scores were obtained for OA($p = 4$). The MBR design gets the same score as the LHD for $n = 16$, but performs the second best after OA($p = 4$) for $n = 32, 64$ and 128.

The maximin distance criterion values are plotted at the bottom of figure 8.3. The value of the criterion indicates the distance of the two closest points in the design, so a high value should indicate that the design samples are evenly spread out. Except for $n = 16$, the MBR design gets the clearly highest scores, followed by the OA($p = 2$) design, LHD and random design. The scores of OA($p = 4$) were somewhat variable. it scored good for $n = 16$, and then substantially worse for increasing n. For $n = 128$, the criterion value were not computed.

For analysis of the variability of the different design methods, the minimum, maximum, mean and median value found from 100 designs were computed for each criterion. The results for $n = 16$ sample points are presented in table 8.6, 8.7, 8.8 and 8.9 for the RMSE, Max, ASSC and maximin criteria respectively. For $n = 32, 64$ and 128 the same general trend with respect to variability can be seen, and the results are given in table A.1, A.2, A.3 and A.4 of the appendix.

	LHS	Random	OA(p=2)	OA(p=4)	MBR
min	0.2653	0.2867	0.2685	0.2798	0.2726
max	0.4590	0.5257	0.3923	0.3625	0.4026
median	0.3246	0.3444	0.3054	0.3054	0.3132
mean	0.3323	0.3565	0.3117	0.3067	0.3181

Table 8.6: Non-randomized designs, $n = 16$ samples. Results for the RMSE criterion.

	LHS	Random	OA(p=2)	OA(p=4)	MBR
min	0.8672	0.8822	0.8873	0.9015	0.9267
max	1.8075	2.0369	1.7937	1.9743	1.6808
median	1.2248	1.2880	1.2003	1.1841	1.2265
mean	1.2541	1.3382	1.2228	1.2467	1.2386

Table 8.7: Non-randomized designs, $n = 16$ samples. Results for the Max criterion.

The random design generally gets the worst scores for the maximum and mean value. However, the random designs sometimes perform very good, for example as for seen in table 8.8, a random design gets the lowest ASSC value. This is not unexpected. If several random designs are generated, we would expect some of these to be good space-filling designs, but there is no certainty that a randomly chosen random design will perform good.

	LHS	Random	OA(p=2)	OA(p=4)	MBR
min	5.1102	4.6419	4.9462	5.1826	4.8214
max	7.2599	9.6373	7.1564	6.3363	8.1297
median	6.1655	6.4556	5.9200	5.6375	6.0362
mean	6.1370	6.5596	5.9072	5.6585	6.1381

Table 8.8: Non-randomized designs, $n = 16$ samples. Results for the ASSC.

	LHS	Random	OA(p=2)	OA(p=4)	MBR
min	0.1250	0.1250	0.2500	0.2500	0.2165
max	0.5000	0.4677	0.5303	0.5000	0.5303
median	0.3307	0.2795	0.3953	0.4146	0.3953
mean	0.3226	0.2765	0.3857	0.4016	0.3759

Table 8.9: Non-randomized designs, $n = 16$ samples. Results for the maximin criterion.

The LHD generally gets low minimum values for all criteria, but also high maximum values compared to the MBR design and OA designs. The LHD gets slightly higher mean values than all the other methods except for the random design.

The MBR design shows little variation of the design scores for $n = 64$ and 128, and obtains the best minimum, maximum and mean score for most criteria. This might imply that for the cases where the number of samples are relatively high, the choice of which binary factors are confounded is less important, as all designs gets a generally good score.

The exception for the MBR design is the designs with $n = 32$ experimental samples. With respect to both the RMSE and Max criteria, the MBR design scored a little better or even worse than the random design. The set of design generators chosen for these designs were probably a poor choice.

The OA($p = 4$) design got the best overall scores for $n = 16$ and 32 samples. For the designs with $n = 64$ and 128 we were not able to generate OA's of strength $r = 3$, and the designs based on the alternative OA's generated in R performed very unsatisfactory.

8.2.2 Randomized

In the randomized comparison study, the design factor levels were assumed to be measured continuously over the design space $[0, 1]^5$.

Figure 8.4 and 8.5 show plots of the results for the different criteria and methods. The main trends are the same as found for the non-randomized designs.

For the RMSE and Max criteria in figure 8.4, the MBR design still gets poor scores for $n = 32$ samples, but substantially better scores than all other methods for $n = 64$ samples and also the best score for $n = 128$ samples. The maximin LHD scores about as good as or better than the conventional LHD, except for the case of $n = 16$, where it performs worse than all other design methods.

For the ASSC seen at top of figure 8.5, the maximin LHD scores very good. The other design methods gets about the same scores as in the non-randomized setting.

Results for the maximin criterion are plotted at the bottom of figure 8.5. The MBR design scores about as good as the maximin LHD. For $n = 16$ and 32, the best criterion scores are obtained for the OA($p = 2$) and OA($p = 4$) designs respectively.

The full summary tables for the design criteria for $n = 16, 32, 64$ and 128 design samples can be found in table A.5, A.6, A.7 and A.8 of the appendix. All design methods show about the same variability in scores as for the non-randomized designs. The maximin LHD has about the same variability as the conventional LHD.

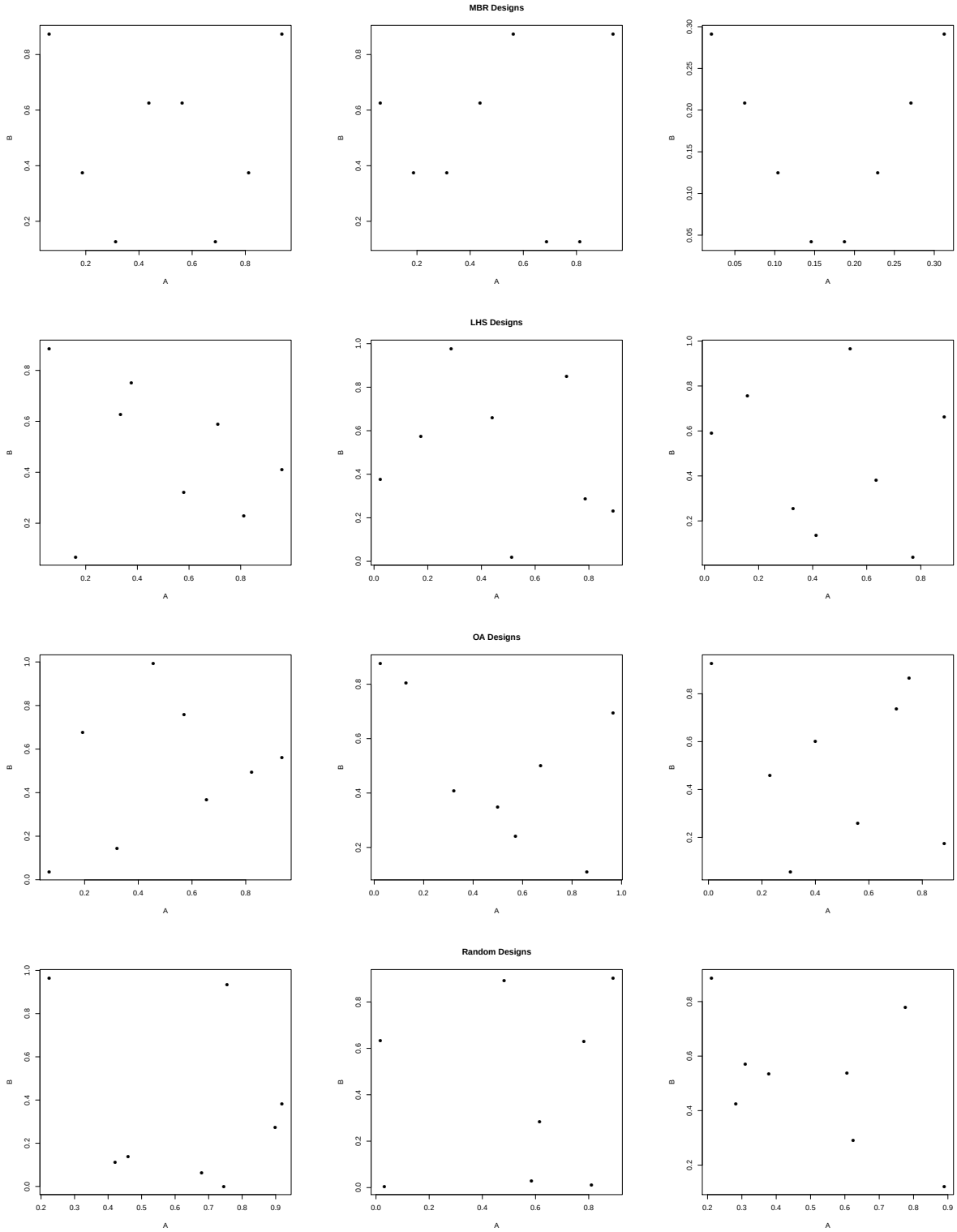

Figure 8.1: Plots of different designs methods with $k = 2$ factors and $n = 8$ samples. Top: MBR design. Second from top: LHD. Second from the bottom: OA design. Bottom: Random design.

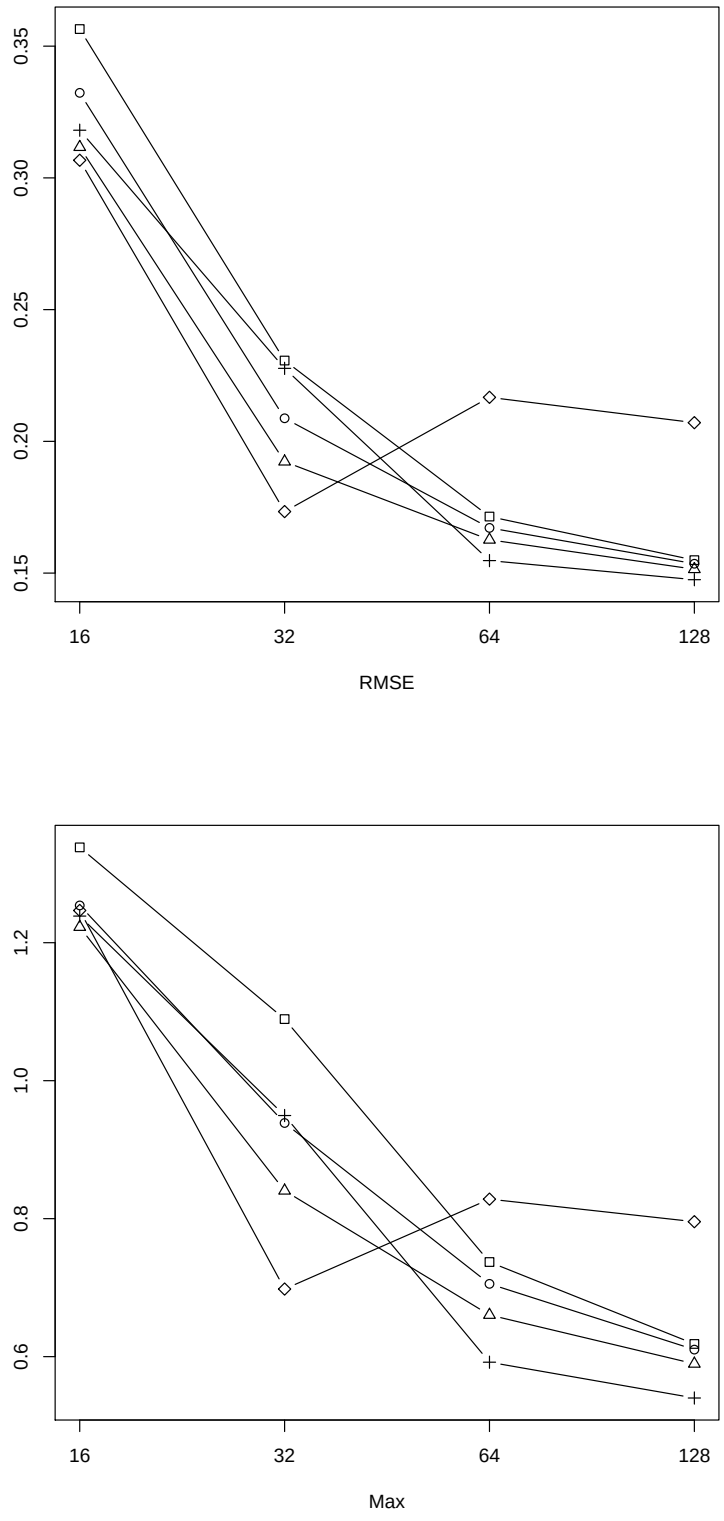

Figure 8.2: Results for the RMSE and Max criteria for five factor non-randomized designs. Design methods: LHS: circles (○), Random: squares (□), OA(p=2): triangles (△), OA(p=4) diamonds: (◇), MBR: pluss (+).

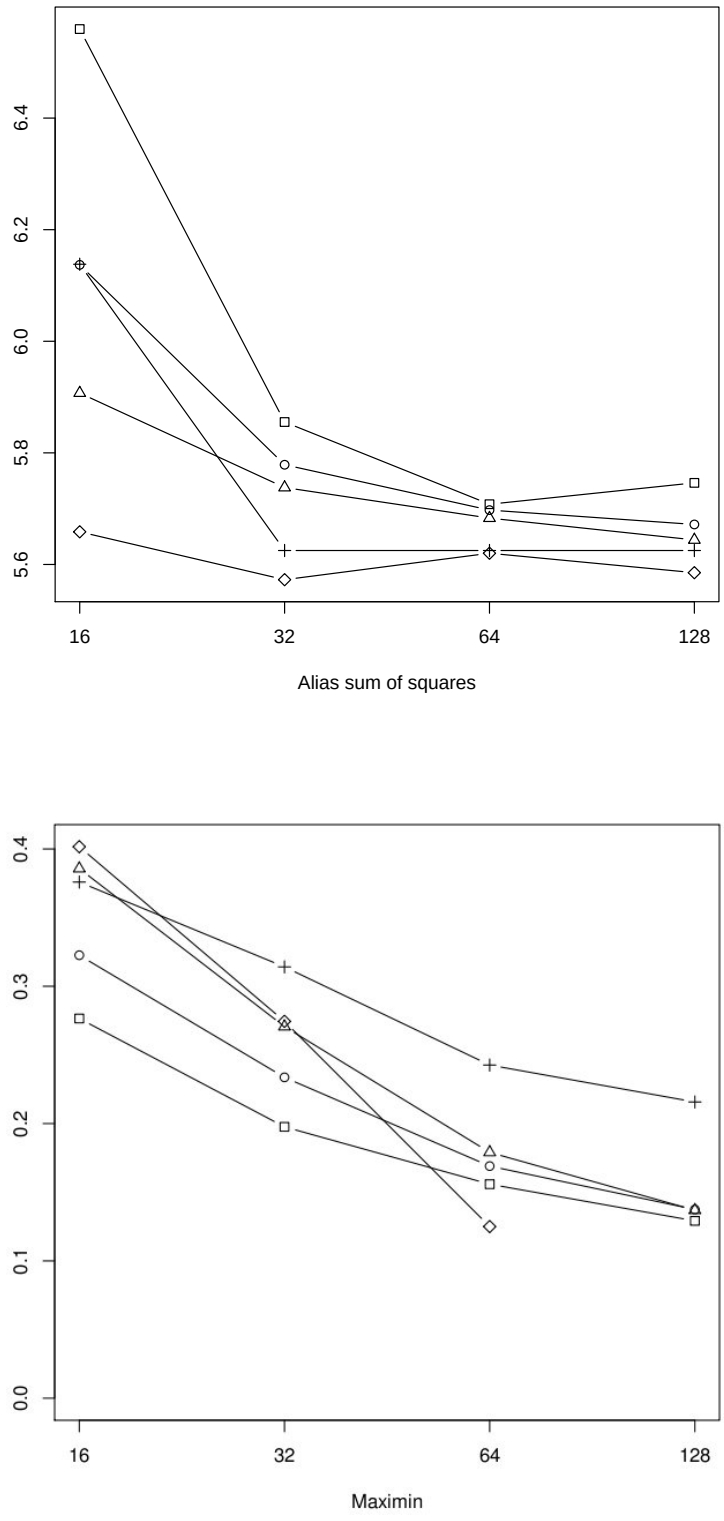

Figure 8.3: Results for the Alias sum of squares and Maximin distance criteria for five factor non-randomized designs. Design methods: LHS: circles (○), Random: squares (□), OA(p=2): triangles (△), OA(p=4) diamonds: (◇), MBR: pluss (+).

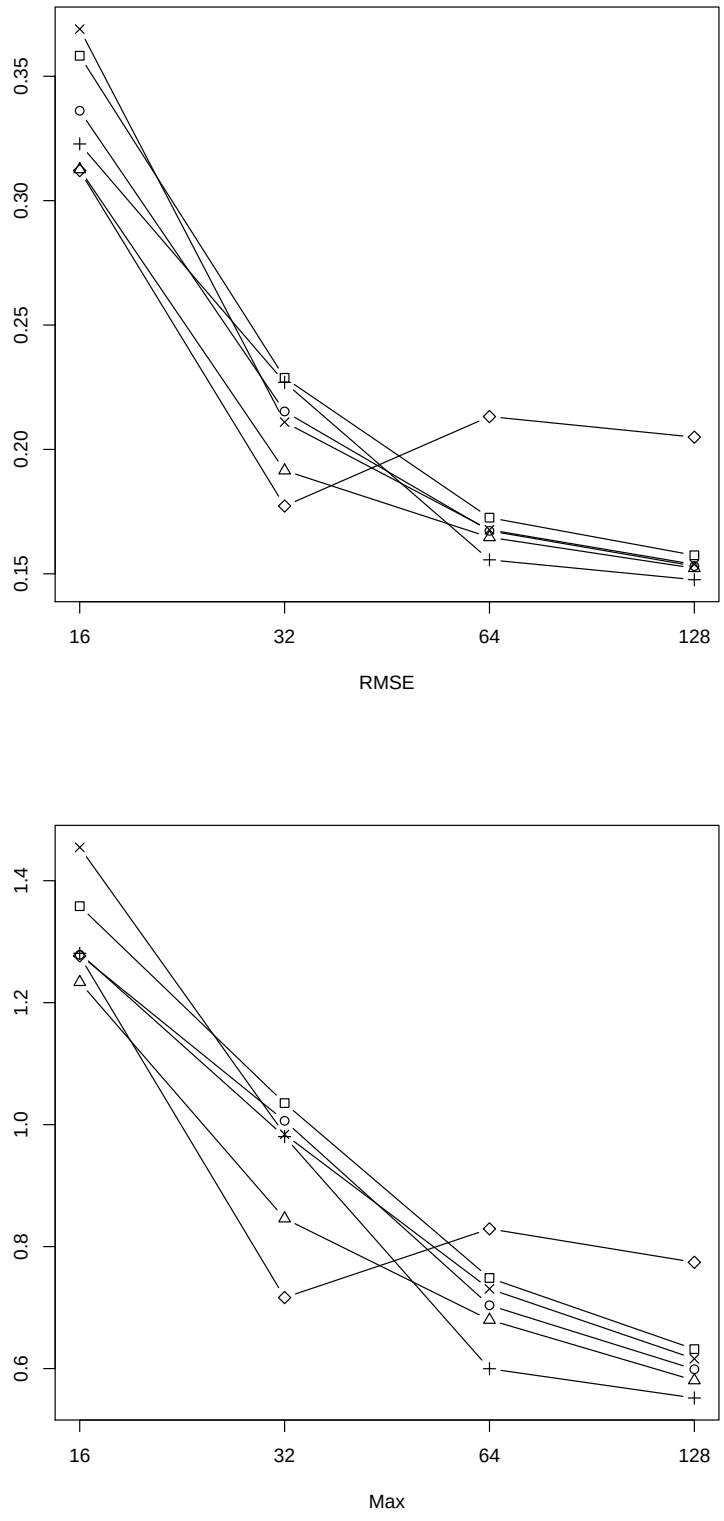

Figure 8.4: Results for the RMSE and Max criteria for five factor randomized designs. Design methods: LHS: circles (○), Random: squares (□), OA(p=2): triangles (△), OA(p=4) diamonds: (◇), MBR: pluss (+), Maximin LHS: cross(×).

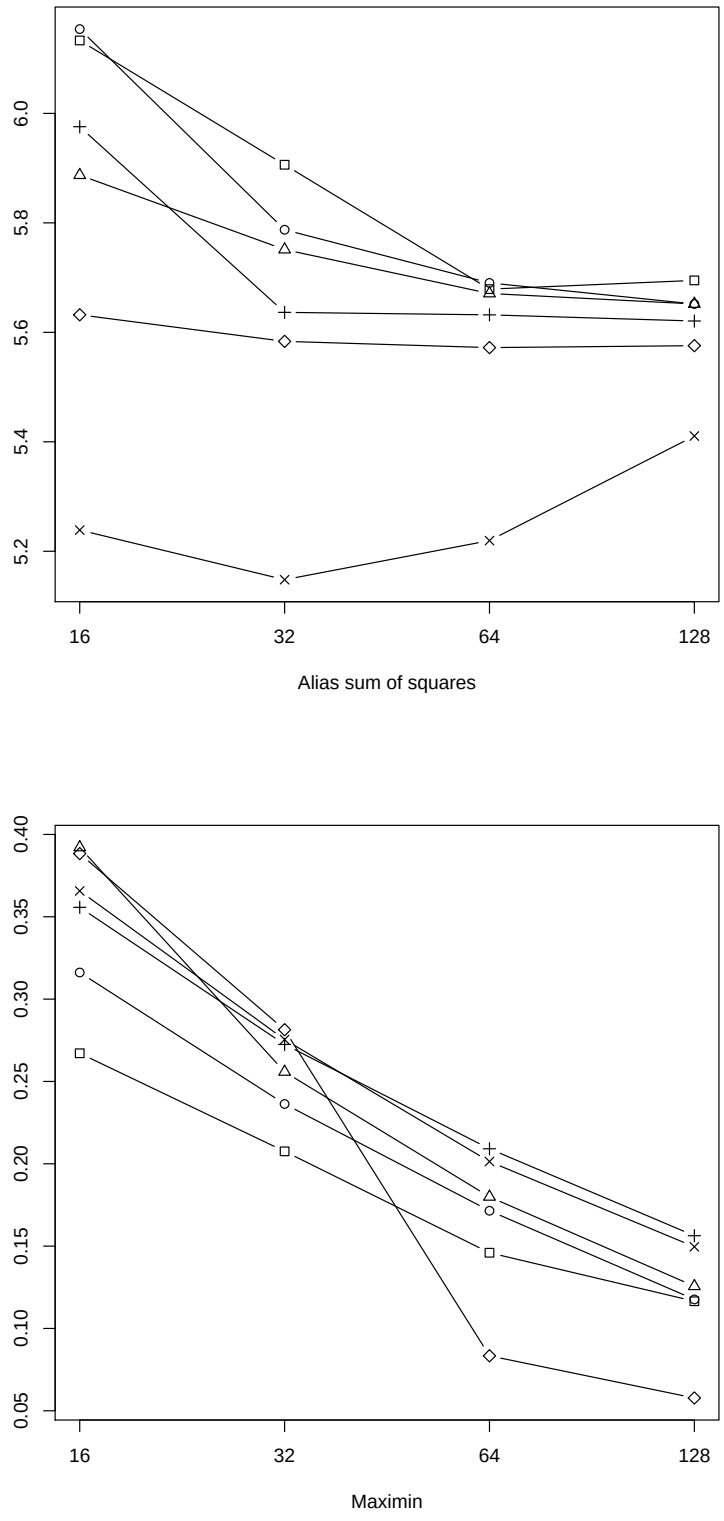

Figure 8.5: Results for the Alias sum of squares and Maximin distance criteria for five factor randomized designs. Design methods: LHS: circles (○), Random: squares (□), OA(p=2): triangles (△), OA(p=4) diamonds: (◇), MBR: pluss (+), Maximin LHS: cross(×).

Chapter 9

Conclusion

We have performed a comparison study of the random design, LHD, OA design and MBR design, where the main focus have been how the MBR design method scored compared to the other designs. We have considered two-factor designs with 8 and 16 design samples, and five-factor designs with 16, 32, 64 and 128 design samples.

The MBR design generally obtained good scores for all criteria. It scored substantially better than the random designs and LHD's in most of the comparisons. Compared to the OA designs, the MBR design scored similarly to the OA design based on OA's with 4 levels, and better than the 2 level OA design, however there were variations with respect to the number of factors and samples.

For the two-factor experiment with 16 design samles, the MBR design scored especially good. It obtained better criterion values than all other design methods, and showed very little variation with respect to how the design factors were confounded.

For the five-factor experiment, the MBR design scored the best for the $n = 64$ and 128 sample point designs. It obtained good mean, maximum and minimum criterion values. For the $n = 32$ sample designs, however, the MBR design performed poorly. We have chosen the set of design generators which would give us the highest possible resolution in a conventional fractional factorial design. In the case of the 32 sample MBR design this proved to be an unfortunate choice.

For a well chosen set of design generators, the MBR design seem to give good scores independent of which binary factors are confounded with each other, especially for the designs with a relatively high number of samples. This is an important property for an experimental design. For example, the LHD's criterion scores seems to vary quite a lot. Although the LHS algorithm may generate good

space-filling designs, there is no absolute certainty that the designs will be balanced in higher dimensions. For experiments with few factors, graphical inspection can be applied to avoid designs with bad coverage or high co-linearity. If the number of factors is high, this will quickly prove to be difficult. A design which ensures overall good coverage will then be desirable, and the MBR design should be well adapted to provide this.

Compared to each other, the random design, LHD and OA design performed much as expected. Random designs generally obtained the worst average scores. The LHD's performed better than the random designs, but obtained higher mean and maximum values than the OA designs. For the OA designs, the design based on OA's with 4 levels scored better than the OA's with 2 levels, but only when we were able to construct OA's of the desirable strength. In experiments where high order interactions are dominating, choosing an OA with fewer levels but higher strength may be more important than an OA on many levels, as the strength decreases with an increasing number of levels.

Although the MBR design method generally performed good in our study, there are some drawbacks to the design. First of all, the restrictions with respect to the sample size of the design are much stronger than for the other methods. The MBR design can only be constructed for sample sizes which are multiples of 2, whereas for random sampling and LHS we are free in choosing whatever number of samples desirable for conducting an experiment. The OA sampling algorithm requires that the relationship $n = \lambda p^r$ is fulfilled, and that an OA can exists and can be found for the desired strength and number of levels. Compared to the other design methods, especially random sampling and LHS, the MBR design is harder to implement on a computer. It also requires that a set of design generators are chosen, and a poor choice might strongly affect the properties of the design.

9.1 Further work

Further investigations as to the properties of the MBR design method should be conducted before making any definite conclusions. In our comparison study, we have only considered low-dimensional experiments, with designs of no more than five factors. The underlying computer models we have considered were polynomials with higher order terms. In computer experiment, the underlying model is usually much more complicated and much harder to assess. If the underlying model has greater local variations over the design space, or show abrupt non-linear behaviour, the response surface models fitted as metamodels in this report will quickly prove to simple. In such cases, more advanced metamodels like the kriging model might

perform better, and this could have a large impact on the choice of the experimental design method.

Bibliography

[1] John J. Borkowski. A comparison of prediction variance criteria for response surface designs. *Journal of Quality Technology*, 2003.

[2] Dizza Bursztyn and David M. Steinberg. Comparison of designs for computer experiments. *Journal of Statistical Planning and Inference*, 2006.

[3] Ward Cheney and David Kincaid. *Numerical Mathematics and Computing*. Brooks/Cole: Cengage Learning, 2008.

[4] Nira Dyn, David Levin, and Samuel Rippa. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM Journal on Scientific Computing*, 1991.

[5] Kai-Tai Fang, Runze Li, and Agus Sudjianto. *Design and Modeling for Computer Experiments*. Chapman & and Hall/CRC, 2006.

[6] Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of statistics*, 1991.

[7] M. Johnson, L. Moore, and D. Ylvisaker. Minimax and maximin distance design. *Journal of Statistical Planning and Inference*, 1990.

[8] Harald Martens, Ingrid Måge, Kristin Tøndel, Julia Isaeva, Martin Høy, and Solve Sæbø. Multi-level binary replacement (mbr) design for computer experiments in high-dimensional nonlinear systems. *Journal of Chemometrics*, 2010.

[9] M.D. McKay, R.J Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 1979.

[10] Dpuglas C. Montgomery and Raymond H. Myers. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, 1995.

[11] Kristin Tøndel, Arne Gjuvsland, Ingrid Måge, and Harald Martens. Screening design for computer experiments: Metamodelling of a deterministic mathematical model of the mammalian circadian clock. *Journal of Chemometrics*, 2010.

[12] Timothy W. Simpson, Dennis K.J. Lin, and Wei Chen. Sampling strategies for computer experiments: Design and analysis. *Journal of the American Statistical Association*, 2001.

[13] Boxin Tang. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 1993.

Appendix A

A.1 Test Functions and Metamodels used in the Comparison Study

For the two factor experiment, as the underlying computer model we chose as our test function a polynomial with squared and cubic terms, and interaction terms up to third order. The exact model is given as:

$$f(x_1, x_2) = x_1 + x_2 + x_1^2 + x_2^2 + x_1x_2 + 3x_1^2x_2 + 2x_1x_2^2 + x_2^3$$

For both $n = 8$ and 16, as metamodels we fitted full second order response surface models given by:

$$\hat{y} = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_{11}x_1^2 + \beta_{22}x_2^2 + \beta_{12}x_1x_2.$$

For the five factor experiment, as the underlying computer model we chose the test function to be a polynomial with squared terms and interaction terms up to third order. The exact model is given as:

$$\begin{aligned} f(x_1, x_2, x_3, x_4, x_5) = {} & x_1 + 3x_2 + 2x_3 - 2x_4 + x_5 + x_1^2 - 0.5x_4^2 + 0.7x_5^2 + 0.4x_1x_2 \\ & - 0.8x_1x_3 + 0.6x_2x_3 - 0.8x_2x_5 + 0.7x_3x_4 + 1.2x_3x_5 - 0.8x_4x_5 \\ & + 0.3x_1^2x_3 + 0.7x_1^2x_4 - 0.4x_1^2x_5 + 0.2x_2^2x_3 + 0.4x_2^2x_4 + 0.2x_2^2x_5 \\ & - 0.7x_3^2x_4 + 0.5x_3^2x_5 + 0.1x_4^2x_5 - 0.4x_1x_2^2 - 0.2x_1x_3^2 + 0.3x_1x_4^2 \\ & + 0.6x_1x_5^2 + 0.9x_2x_3^2 + 0.3x_2x_4^2 + 0.5x_2x_5^2 + 0.2x_3x_4^2 \end{aligned}$$

For $n = 16$, as metamodel we fitted a first order response surface model given by:

$$\hat{y} = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_3x_3 + \beta_4x_4 + \beta_5x_5. \tag{A.1}$$

For $n = 32, 64$ and 128, as metamodels we fitted response surface models with main effect terms and second order interaction terms given by:

$$\begin{aligned}\hat{y} = \beta_0 &+ \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \\ &+ \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{14} x_1 x_4 + \beta_{15} x_1 x_5 + \beta_{23} x_2 x_3 \\ &+ \beta_{24} x_2 x_4 + \beta_{25} x_2 x_5 + \beta_{34} x_3 x_4 + \beta_{35} x_3 x_5 + \beta_{45} x_4 x_5\end{aligned}$$

A.2 Design Generators for Construction of MBR Designs

A.2.1 Two Factor Designs

Two factor MBR design with each factor on 8 levels. We denote the two factors by A and B, and the binary factors for A by a_1, a_2 and a_3 and for B by b_1, b_2 and b_3. To construct a MBR design with 16 levels, we need $2^{6-M_{conf}} = 16 \Rightarrow M_{conf} = 2$ design generators. To obtain a resolution IV 2^{6-2} fractional factorial design we choose design generators of the form:

$$m_1 = c_1\ c_2\ c_3 \qquad m_2 = c_2\ c_3\ c_4.$$

From $\{a_1, a_2, a_3, b_1, b_2, b_3\}$, we choose two of the binary factors to be represented by m_1 and m_2, and the remaining four binary factors to be represented by c_1, c_2, c_3 and c_4. There are several ways to assign the binary factors to the design generators, and out of these we found a set of 45 combinations which yielded unique designs.

A.2.2 Five Factor Designs

Five factor designs with each factor on 8 levels. We denote the factors by {A,B,C,D,E}, and the respective binary factors by $\{a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3, d_1, d_2, d_3, e_1, e_2, e_3\}$. The set of design generators for all five factor MBR designs were chosen by the statistical software Minitab, which suggest the design generators that gives the highest possible resolution.

For all $n = 16, 32, 64$ and 128, we assign M_{conf} of the binary factors $\{a_1, a_2, a_3, b_1, ..., e_2, e_3\}$ to represent $\{m_1, m_2, ..., m_{M_{conf}}\}$ and the remaining $q = 15 - M_{conf}$ binary factors to represent $\{c_1, c_2, ..., c_q\}$. As there are a number of different ways to assign the binary factors to the design generators, for each generated MBR design we randomly chose how the binary factors are assigned to $\{m_1, m_2, ..., m_{M_{conf}}\}$ and $\{c_1, c_2, ..., c_q\}$.

For $n = 16$ we have $M_{conf} = 11$. To obtain a resolution III 2^{15-11} fractional factorial design we choose design generators of the form:

$$\begin{aligned} m_1 &= c_1\ c_2 & m_7 &= c_1\ c_2\ c_3 \\ m_2 &= c_1\ c_3 & m_8 &= c_1\ c_2\ c_4 \\ m_3 &= c_1\ c_4 & m_9 &= c_1\ c_3\ c_4 \\ m_4 &= c_2\ c_3 & m_{10} &= c_2\ c_3\ c_4 \\ m_5 &= c_2\ c_4 & m_{11} &= c_1\ c_2\ c_3\ c_4 \\ m_6 &= c_3\ c_4 \end{aligned}$$

For $n = 32$ we have $M_{conf} = 10$. To obtain a resolution IV 2^{15-10} fractional factorial design we choose design generators of the form:

$$\begin{aligned} m_1 &= c_1\ c_2\ c_3 & m_6 &= c_1\ c_4\ c_5 \\ m_2 &= c_1\ c_2\ c_4 & m_7 &= c_2\ c_3\ c_4 \\ m_3 &= c_1\ c_2\ c_5 & m_8 &= c_2\ c_3\ c_5 \\ m_4 &= c_1\ c_3\ c_4 & m_9 &= c_2\ c_4\ c_5 \\ m_5 &= c_1\ c_3\ c_5 & m_{10} &= c_3\ c_4\ c_5 \end{aligned}$$

For $n = 64$ we have $M_{conf} = 9$. To obtain a resolution IV 2^{15-9} fractional factorial design we choose design generators of the form:

$$\begin{aligned} m_1 &= c_1\ c_2\ c_4 & m_6 &= c_1\ c_3\ c_6 \\ m_2 &= c_1\ c_2\ c_5 & m_7 &= c_2\ c_4\ c_5\ c_6 \\ m_3 &= c_1\ c_2\ c_6 & m_8 &= c_3\ c_4\ c_5\ c_6 \\ m_4 &= c_1\ c_3\ c_4 & m_9 &= c_1\ c_2\ c_3\ c_4\ c_5\ c_6 \\ m_5 &= c_1\ c_3\ c_5 \end{aligned}$$

For $n = 128$ we have $M_{conf} = 8$. To obtain a resolution IV 2^{15-8} fractional factorial design we choose design generators of the form:

$$\begin{aligned} m_1 &= c_2\ c_5\ c_6 & m_5 &= c_1\ c_3\ c_4\ c_5\ c_6 \\ m_2 &= c_5\ c_6\ c_7 & m_6 &= c_1\ c_2\ c_3\ c_5\ c_7 \\ m_3 &= c_1\ c_2\ c_6\ c_7 & m_7 &= c_1\ c_3\ c_4\ c_5\ c_7 \\ m_4 &= c_3\ c_4\ c_6\ c_7 & m_8 &= c_1\ c_2\ c_4\ c_5\ c_6\ c_7 \end{aligned}$$

A.3 Five Factor Design Result Tables

	LHS	Random	OA(p=2)	OA(p=4)	MBR
n=32					
min	0.1555	0.1577	0.1620	0.1497	0.1518
max	0.3520	0.3575	0.2405	0.1960	0.5467
median	0.2026	0.2229	0.1892	0.1727	0.2147
mean	0.2088	0.2307	0.1923	0.1733	0.2277
n=64					
min	0.1455	0.1487	0.1457	0.1871	0.1372
max	0.2007	0.2237	0.1865	0.2585	0.1919
median	0.1654	0.1695	0.1623	0.2157	0.1533
mean	0.1672	0.1715	0.1626	0.2167	0.1547
n=128					
min	0.1411	0.1437	0.1430	0.1861	0.1357
max	0.1818	0.1891	0.1765	0.2275	0.1746
median	0.1529	0.1536	0.1503	0.2070	0.1460
mean	0.1536	0.1549	0.1515	0.2071	0.1475

Table A.1: Non-randomized designs, $n = 32, 64$ and 128 samples. Results for the RMSE criterion.

	LHS	Random	OA(p=2)	OA(p=4)	MBR
n=32					
min	0.5888	0.5549	0.5611	0.5226	0.5350
max	1.6607	2.1793	1.5140	1.1237	2.4226
median	0.8745	1.0614	0.8228	0.6816	0.8644
mean	0.9387	1.0893	0.8405	0.6980	0.9495
n=64					
min	0.4755	0.4721	0.4727	0.5658	0.4705
max	1.0777	1.4877	1.0107	1.2085	0.8712
median	0.6721	0.7189	0.6379	0.8206	0.5876
mean	0.7055	0.7371	0.6604	0.8283	0.5919
n=128					
min	0.4369	0.4813	0.4533	0.6224	0.4305
max	0.8746	0.8768	0.7670	1.0807	0.7495
median	0.5912	0.6183	0.5873	0.7805	0.5377
mean	0.6102	0.6184	0.5896	0.7956	0.5400

Table A.2: Non-randomized designs, $n = 32, 64$ and 128 samples. Results for the Max criterion.

	LHS	Random	OA(p=2)	OA(p=4)	MBR
n=32					
min	4.8874	4.3746	5.1245	5.2296	5.6250
max	6.5548	7.3266	6.5201	6.0304	5.6250
median	5.7756	5.7919	5.7590	5.5926	5.6250
mean	5.7787	5.8553	5.7379	5.5727	5.6250
n=64					
min	5.2651	4.5506	5.1819	5.4019	5.6250
max	6.1653	6.9486	6.1367	5.9272	5.6250
median	5.6967	5.6337	5.6864	5.6157	5.6250
mean	5.6974	5.7080	5.6831	5.6201	5.6250
n=128					
min	5.2083	4.8699	5.2795	5.4227	5.6250
max	6.0266	6.5119	6.0882	5.7846	5.6250
median	5.6575	5.7068	5.6419	5.5877	5.6250
mean	5.6717	5.7462	5.6440	5.5852	5.6250

Table A.3: Non-randomized designs, $n = 32, 64$ and 128 samples. Results for the ASSC.

	LHS	Random	OA(p=2)	OA(p=4)	MBR
n=32					
min	0.1250	0.1250	0.1250	0.1768	0.2165
max	0.3307	0.3062	0.3750	0.3750	0.4146
median	0.2333	0.1898	0.2795	0.2795	0.3307
mean	0.2337	0.1976	0.2706	0.2745	0.3141
n=64					
min	0.1250	0.1250	0.1250	0.1250	0.1768
max	0.2500	0.2165	0.2500	0.1250	0.3307
median	0.1768	0.1768	0.1768	0.1250	0.2333
mean	0.1690	0.1559	0.1790	0.1250	0.2426
n=128					
min	0.1250	0.1250	0.1250	NA	0.1768
max	0.1768	0.1768	0.1768	NA	0.2795
median	0.1250	0.1250	0.1250	NA	0.2165
mean	0.1374	0.1291	0.1369	NA	0.2157

Table A.4: Non-randomized designs, $n = 32, 64$ and 128 samples. Results for the maximin criterion.

	LHS	Random	OA(p=2)	OA(p=4)	MBR	Maximin LHS
n=16						
min	0.2759	0.2725	0.2738	0.2640	0.2751	0.2918
max	0.5255	0.5578	0.3680	0.3599	0.4257	0.6098
median	0.3303	0.3463	0.3097	0.3132	0.3179	0.3572
mean	0.3361	0.3582	0.3126	0.3121	0.3228	0.3690
n=32						
min	0.1573	0.1661	0.1586	0.1528	0.1521	0.1637
max	0.3847	0.3804	0.2563	0.2009	0.4845	0.3073
median	0.2063	0.2185	0.1880	0.1778	0.2219	0.2065
mean	0.2153	0.2288	0.1915	0.1772	0.2270	0.2110
n=64						
min	0.1473	0.1437	0.1464	0.1810	0.1415	0.1427
max	0.1999	0.2206	0.1841	0.2673	0.1956	0.2037
median	0.1663	0.1700	0.1639	0.2114	0.1532	0.1653
mean	0.1673	0.1726	0.1647	0.2132	0.1556	0.1677
n=128						
min	0.1409	0.1437	0.1404	0.1820	0.1385	0.1424
max	0.1725	0.1805	0.1640	0.2428	0.1672	0.1738
median	0.1525	0.1566	0.1524	0.2043	0.1469	0.1522
mean	0.1531	0.1574	0.1523	0.2049	0.1476	0.1538

Table A.5: Randomized designs, $n = 16, 32, 64$ and 128 samples. Results for the RMSE criterion.

	LHS	Random	OA(p=2)	OA(p=4)	MBR	Maximin LHS
n=16						
min	0.9022	0.8940	0.8848	0.8832	0.8830	0.9759
max	2.1040	2.0730	1.8260	2.0584	1.9306	2.1782
median	1.2493	1.3381	1.2143	1.2445	1.2395	1.4164
mean	1.2785	1.3583	1.2337	1.2767	1.2809	1.4547
n=32						
min	0.5567	0.5985	0.5513	0.5016	0.6183	0.5933
max	1.9822	2.4043	1.4213	1.0368	2.0298	1.7704
median	0.9604	0.9684	0.8189	0.7030	0.9699	0.9385
mean	1.0063	1.0353	0.8461	0.7164	0.9804	0.9841
n=64						
min	0.4818	0.4212	0.4778	0.5743	0.4469	0.4757
max	0.9627	1.1798	1.0635	1.1994	0.8406	1.1279
median	0.6859	0.7182	0.6489	0.8119	0.5890	0.7030
mean	0.7037	0.7486	0.6798	0.8291	0.5997	0.7308
n=128						
min	0.4374	0.4592	0.4308	0.5921	0.4354	0.4868
max	0.9575	0.9370	0.7969	1.2645	0.8811	1.0790
median	0.5894	0.6167	0.5826	0.7493	0.5380	0.5936
mean	0.5989	0.6319	0.5806	0.7743	0.5517	0.6160

Table A.6: Randomized designs, $n = 16, 32, 64$ and 128 samples. Results for the Max criterion.

	LHS	Random	OA(p=2)	OA(p=4)	MBR	Maximin LHS
n=16						
min	5.2185	3.8871	4.9911	4.7047	4.6351	4.3442
max	7.1747	8.9799	6.8226	6.3879	7.8018	6.4902
median	6.1494	6.1491	5.9097	5.6011	5.9493	5.1338
mean	6.1541	6.1333	5.8872	5.6320	5.9757	5.2387
n=32						
min	5.0612	4.0930	5.1593	5.1113	5.4112	4.5444
max	6.6224	7.3374	6.3603	6.0250	5.9243	5.9769
median	5.8145	5.9460	5.7271	5.5613	5.6449	5.0838
mean	5.7875	5.9062	5.7512	5.5835	5.6365	5.1481
n=64						
min	5.2470	4.3831	5.2661	5.2729	5.4865	4.7002
max	6.2402	7.1139	6.1416	5.8917	5.7562	5.7424
median	5.6603	5.6731	5.6878	5.5639	5.6310	5.2302
mean	5.6903	5.6789	5.6710	5.5721	5.6320	5.2195
n=128						
min	5.3311	5.0064	5.2234	5.3514	5.4748	5.1199
max	5.9955	6.4227	6.0628	5.8401	5.7117	5.7510
median	5.6529	5.6953	5.6688	5.5888	5.6223	5.3897
mean	5.6518	5.6949	5.6519	5.5756	5.6207	5.4106

Table A.7: Randomized designs, $n = 16, 32, 64$ and 128 samples. Results for the ASSC.

	LHS	Random	OA(p=2)	OA(p=4)	MBR	Maximin LHS
n=16						
min	0.1090	0.0637	0.1752	0.2196	0.2121	0.1980
max	0.4859	0.4332	0.5193	0.4830	0.5034	0.4543
median	0.3218	0.2638	0.3995	0.4032	0.3568	0.3742
mean	0.3162	0.2671	0.3922	0.3884	0.3558	0.3657
n=32						
min	0.1158	0.1055	0.1120	0.1451	0.1392	0.1830
max	0.3508	0.3269	0.3816	0.3811	0.3782	0.3306
median	0.2412	0.2027	0.2573	0.2855	0.2779	0.2791
mean	0.2363	0.2076	0.2558	0.2814	0.2725	0.2755
n=64						
min	0.0600	0.0490	0.0971	0.0468	0.0840	0.0861
max	0.2685	0.2449	0.2542	0.1280	0.2926	0.2362
median	0.1752	0.1479	0.1801	0.0830	0.2172	0.1999
mean	0.1715	0.1460	0.1799	0.0834	0.2091	0.2014
n=128						
min	0.0376	0.0639	0.0586	0.0229	0.0688	0.1236
max	0.1726	0.1769	0.1935	0.0903	0.2246	0.2024
median	0.1192	0.1154	0.1256	0.0583	0.1531	0.1477
mean	0.1176	0.1165	0.1256	0.0578	0.1563	0.1496

Table A.8: Randomized designs, $n = 16, 32, 64$ and 128 samples. Results for the maximin criterion.