Optimal experimental designs for DNA microarray experiments

Moonen, E.J.G.

Award date:
2007

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Master's Thesis

Optimal Experimental Designs for DNA Microarray Experiments

by

E.J.G. Moonen, B.Sc.

Supervisors:
Dr. A. Di Bucchianico

Dr. Ir. E.E.M. van Berkum

Eindhoven, 17th December 2006

Preface

This Master's thesis is the result of 9 months of research at the Department of Mathematics and Computer Science. With this thesis I end my study of Industrial and Applied Mathematics, in the Statistic Probability and Operations Research track, and I will receive the degree of Master of Science. I did my research at the chair of Probability and Statistics at the Eindhoven University of Technology (TU/e). The research topic of this thesis, Optimal Experimental Designs for DNA Microarray Experiments, was introduced to me by Dr. Ir. E.E.M. van Berkum and Dr. A. Di Bucchianico from the Department of Mathematics and Computer Science, who became my supervisors on this project. I would like to take some time to thank everybody who, in one way or another, helped me during my 9 months of research. First of all there are my supervisors. They helped me every time I had a question or problem, always in the most patient and precise way, in spite of their very busy schedules. I would also like to thank Ir. W.I.P.M. Kortsmit for his help with solving some programming problems in *Mathematica*, Ir. M.A.A. Boon for his help with problems in LaTeX and Dr. E.D. Schoen for an insightful discussion on how to model the dye-factor into our model. I would love to thank my parents who made it possible for me to begin (and finish) my study. Not only financially, but also for their support and love. I also want to thank my sister and all my friends, Barbara in particular, for their support and love.

Summary

The subject of this thesis is Optimal Experimental Designs for DNA Microarray Experiments. DNA Microarray Experiments are used to detect the expression levels of many thousands of genes simultaneously. In medical science microarray technology is for example used for finding out which genes play an important role in a specific form of cancer by comparing tissues with and without tumour.

Microarray experiments are carried out on slides. These slides are very expensive. The goal of this thesis was to find out how these slides should be used in a way that as much as possible information can be retrieved from the experiment. Historically this topic is very recent. In Glonek and Solomon (2003) and Kerr and Churchill (2001) different statistical models for microarray experiments are given. In Glonek and Solomon (2003) a new optimality criterion is introduced, admissibility. In Van Berkum (1985) great theoretical background can be found, of which microarray experiments turn out to be an application.

We start the thesis with some mathematical background. We introduce mathematical techniques used throughout this thesis. These techniques include regression analysis, design of experiments (especially optimal design of paired comparison experiments) and calculus. We also give an overview and outline of the proof of the celebrated Equivalence Theorem of optimal design of experiments.

In another part of this thesis we investigated models used so far, and add a commonly used model in regression analysis to the list of possible models. We looked at the new class of admissible designs and gave drawbacks of this criterion.

In Yang and Speed (2002) it is stated that the assignment of the red and green dye could effect the results. Some genes could possibly have a better response with green than with red, or vice versa. We added a dye-factor to the model and did research to optimal designs for different criteria when this dye-factor was added. We noted that we could identify two different admissibility criteria in this new model. One could see the new parameter as a nuisance parameter (called Admissible3), or as another parameter in which we are interested (called Admissible4).

We wrote a program to find optimal designs for different criteria. Using this program, we found optimal designs for different optimality criteria. We also noted that some criteria have drawbacks.

We end this thesis with some ideas for further research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis we want to investigate the design of microarray experiments. Microarrays are used to detect the expression levels of many thousands of genes simultaneously. In medical science microarray technology is for example used for finding out which genes play an important role in a specific form of cancer by comparing tissues with and without tumour.

Before we give a detailed description of how a microarray experiment is conducted we take a basic look at genetics. The information we give is a summary of Gonick and Wheelis (1997). The human body consists of cells. All cells have been formed from division of a pre-existing cell and before this division everything in a cell is doubled, which is called mitosis. In each cell of our body DNA is stored. Threadlike systems of DNA in the center of a cell are called chromosomes, they store our complete genetic information. Hence, more specific DNA (DeoxyriboNucleic Acid) is the molecule inside the cell that carries genetic information. The structural arrangement of DNA looks like a spiral staircase. DNA can yield interesting information on the human body.

Now we have an idea of what DNA is we can take a closer look at a DNA microarray experiment. DNA microarrays consist of thousands of individual DNA sequences printed in a high density array on a glass microscope slide. The experiment typically involves hybridizing two DNA samples. One sample is labeled with a red-fluorescent dye, Cyanine 5 (Cy_5) and the other is labeled with a green-fluorescent dye, Cyanine 3 (Cy_3). The ratio of the returned light intensities after hybridization indicates the relative abundance of the corresponding DNA probe in the two samples and is the quantity used in these experiments.

Microarray experiments generate large and complex multivariate data sets, and some of the greatest challenges lie not in generating these data but in tools to analyze the large amounts of data. Therefore it is a goal to design the experiments so that the reliability and efficiency of the obtained data can be improved. As already said, the emphasis of this thesis will be on investigating the designs historically used with microarray experiments.

In Chapter 2 an introduction to the mathematical techniques for design of experiments will be given. Chapter 3 gives an analysis of the variance model for DNA microarray experiments. In this chapter admissible designs are introduced. The results of these designs are stated in Chapter 4. In Chapter 5 we state our conclusions.

Chapter 2

Mathematical techniques

In this chapter we give an introduction to mathematical techniques used in this thesis. These techniques include regression analysis, (optimal) design of experiments and calculus. In the next section we give results about regression analysis, which plays an important role in designing experiments.

2.1 Regression analysis

In this section some results about linear regression are stated. These results can all be found in Van Berkum (2003) and in Montgomery et al. (2001).

2.1.1 Properties of random vectors

Let Y be a random vector with expectation vector μ and covariance matrix V. Let A be a deterministic matrix and a a deterministic vector. The following holds

$$\begin{aligned} \mathrm{Cov}[Y] &= \mathbb{E}\left[(Y-\mathbb{E}[Y])(Y-\mathbb{E}[Y])'\right], \\ \mathbb{E}[a'Y] &= a'\mu, \\ \mathbb{E}[AY] &= A\mu, \\ \mathrm{Cov}[a'Y] &= a'Va, \\ \mathrm{Cov}[AY] &= AVA', \\ \mathbb{E}[Y'AY] &= \mathrm{tr}[AV]+\mu'A\mu, \end{aligned} \tag{2.1}$$

where $\mathrm{tr}[A]$ of an $n \times n$ matrix A is defined as the sum of its diagonal elements, so

$$\mathrm{tr}[A] = \sum_{i=1}^{n} a_{ii}.$$

2.1.2 Models and estimation

First we introduce the linear regression model. This model is

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon,$$

where $\beta \in \mathbb{R}^{k+1}$ and ε a random variable with $\mathbb{E}[\varepsilon] = 0$ and $\mathrm{Var}[\varepsilon] = \sigma^2$.

We assume that n observations are made. For each of these observations the model holds, so

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \ldots + \beta_k x_{1k} + \varepsilon_1, \\ Y_2 &= \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \ldots + \beta_k x_{2k} + \varepsilon_2, \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \ldots + \beta_k x_{nk} + \varepsilon_n, \end{aligned}$$

where it is assumed that all ε_i are independent and have the same distribution.

We can simplify this by using a different notation, using matrices. In this matrix notation the model becomes

$$Y = X\beta + \varepsilon, \tag{2.2}$$

with

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} \quad \text{and} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

If we want to estimate β using Least Squares estimators, we get

$$\widehat{\beta} = (X'X)^{-1}X'Y. \tag{2.3}$$

This estimator for β is unbiased. This can be shown easily using one of the properties stated in (2.1),

$$\mathbb{E}[\widehat{\beta}] = \mathbb{E}[(X'X)^{-1}X'Y] = (X'X)^{-1}X' \; \mathbb{E}[Y] = (X'X)^{-1}X'X\beta = \beta.$$

In the next section we discuss design of experiments. We will need the covariance matrix of $\widehat{\beta}$. This covariance matrix of $\widehat{\beta}$ is

$$\begin{aligned}
\mathrm{Cov}[\widehat{\beta}] &= \mathbb{E}\left[(\widehat{\beta} - \mathbb{E}[\widehat{\beta}])(\widehat{\beta} - \mathbb{E}[\widehat{\beta}])'\right] \\
&= \mathbb{E}\left[\left((X'X)^{-1}X'Y - \mathbb{E}[(X'X)^{-1}X'Y]\right)\left((X'X)^{-1}X'Y - \mathbb{E}[(X'X)^{-1}X'Y]\right)'\right] \\
&= (X'X)^{-1}X' \cdot \mathbb{E}\left[(Y - \mathbb{E}[Y])(Y - \mathbb{E}[Y])'\right] \cdot ((X'X)^{-1}X')' \\
&= (X'X)^{-1}X' \cdot \sigma^2 I \cdot ((X'X)^{-1}X')' \\
&= (X'X)^{-1}\sigma^2 \cdot X'X(X'X)^{-1} \\
&= (X'X)^{-1}\sigma^2. \qquad (2.4)
\end{aligned}$$

2.2 Design of experiments

In this section some results about design of experiments, in particular optimal design of experiments are presented. These results can also be found in Van Berkum (1985).

In the situation where one wants to investigate a relation between a regressor variable and a response variable a regression model is often used. Before investigating this relation, observations have to be made. Sometimes one can choose for which values of the regressor variable the response variable has to be observed. The choice of these values may determine the quality of the investigation. The theory of design of experiments is developed to be a guide in this choice.

2.2.1 Model choice

The model for design of experiments is usually a linear regression model, but we use a slightly different notation

$$y = f_1(x)\beta_1 + \ldots + f_k(x)\beta_k, \qquad (2.5)$$

where

$$\begin{aligned}
x &\in \mathcal{U}, \\
\mathcal{U} &\subseteq \mathbb{R}^n \\
f_j &: \mathcal{U} \to \mathbb{R}, \text{ continuous on the experimental region } \mathcal{U}.
\end{aligned}$$

Another choice for the definition of the experimental region, as can be found in Silvey (1980), would be choosing a design space $\mathcal{U}$, similar to the experimental region in model (2.5), and defining χ as the induced design space

$$\chi = f(\mathcal{U}), \qquad (2.6)$$

where $f = (f_1, \ldots, f_k)'$.

For example, if $f = (1, u, u^2)'$ and $\mathcal{U} = [-1, 1]$ then the induced design space χ is a subset of $\mathbb{R}^3$. Then $\chi = \left\{ x = \begin{bmatrix} 1 \\ u \\ u^2 \end{bmatrix}; -1 \leq u \leq 1 \right\}$, which is an arc of a parabola in the plane $x_1 = 1$.

This notation simplifies some work that has to be done for the Equivalence Theorem 2.2, in Section 2.3.2.

Now we can describe a design in the following way. The m points were observations will be done are denoted by $u_1, \ldots, u_m$, where $u_i, \in \mathcal{U}$. The number of observations in the point u_i is equal to $n_i \in \mathbb{N}$. The total number of observations is denoted by N, so $\sum_{i=1}^{m} n_i = N$. A design is denoted by $\mathcal{E}(N)$ or just $\mathcal{E}$. In the notation of Fedorov (1972) the design $\mathcal{E}$ of an experiment may be written as a collection of variables

$$\mathcal{E}(N) = [u_1, u_2, \ldots, u_m; n_1, n_2, \ldots, n_m; N] \tag{2.7}$$

A design may be constructed by choosing both the u_i and the n_i. Mostly the objects have been fixed so only the n_i can be chosen. In the construction of a design as in (2.34) both the points u_i -and therefore the objects- and the n_i have to be chosen.

An exact normalized design $\varepsilon(N)$, or ε is a collection of variables

$$\varepsilon(N) = [u_1, u_2, \ldots, u_m; p_1, p_2, \ldots, p_m] \tag{2.8}$$

where

$$p_i = n_i/N,$$

and

$$\sum_{i=1}^{m} p_i = 1. \tag{2.9}$$

If the p_i can take on any nonnegative value satisfying (2.9) the design (2.8) is called a discrete normalized design. Further, a continuous normalized design is characterized by a probability measure ξ on the region $\mathcal{U}$.

For the construction of optimal designs the information matrix (the inverse of the covariance matrix) of a design is needed.

We now show how the information matrix of a design can be derived from the Fisher information matrix. First recall the definition of Fisher information. (An elaborate explanation on Fisher information and more can be found in Hogg et al. (2004)). Fisher information is thought of as the amount of information that an observable random variable X contains about an unobservable parameter θ upon which the probability distribution of X depends. If we call $f(X, \theta)$ the density function of the random variable X, then the Fisher information can be defined as

$$\mathcal{I}(\theta) = \mathbb{E}\left[\frac{\partial}{\partial\theta} \log f(X, \theta)\right]^2. \tag{2.10}$$

In the case where there are p parameters, thus making θ a vector of length p, the Fisher information matrix is defined as a $p \times p$ symmetric matrix

$$\mathcal{I}(\theta) = \mathrm{Cov}(\nabla f(X;\theta)), \tag{2.11}$$

thus the (i,j) element can be notated as

$$\begin{aligned} (\mathcal{I}(\theta))_{ij} &= \mathbb{E}\left[\frac{\partial}{\partial\theta_i}\log f(X,\theta)\frac{\partial}{\partial\theta_j}\log f(X,\theta)\right] &(2.12)\\ &= -\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i\,\partial\theta_j}\log f(X,\theta)\right]. &(2.13)\end{aligned}$$

So if we consider the regression model (2.2), the vector $Y = (Y_1,\ldots,Y_n)$ is multivariate normally distributed with expectation vector $\mu = X\beta$ and covariance matrix $V = \sigma^2 I$. So the joint probability distribution function of Y, where we take $\theta = \beta = (\beta_1,\ldots,\beta_k)'$, has the form

$$\begin{aligned} f(Y;\beta) &= \frac{1}{\sqrt{(2\pi)^n|V|}}\exp\left[-\frac{1}{2}(Y-\mu)'V^{-1}(Y-\mu)\right]\\ &= \frac{1}{\sqrt{(2\pi\sigma^2)^n}}\exp\left[-\frac{1}{2\sigma^2}(Y-X\beta)'(Y-X\beta)\right]. \end{aligned}$$

Hence

$$\log f(Y;\beta) = \log\left[\frac{1}{\sqrt{(2\pi\sigma^2)^n}}\right] - \frac{1}{2\sigma^2}(Y-X\beta)'(Y-X\beta). \tag{2.14}$$

So

$$\begin{aligned} \frac{\partial}{\partial\beta_i}\log f(Y,\beta) &= \frac{2x_i'(Y-X\beta)}{2\sigma^2}\\ &= \frac{x_i'(Y-X\beta)}{\sigma^2}, &(2.15)\end{aligned}$$

where x_i' is the i^{th} row of X, and so

$$\begin{aligned} \frac{\partial^2}{\partial\beta_i\,\partial\beta_j}\log f(Y,\beta) &= \frac{\partial}{\partial\beta_j}\frac{x_i'(Y-X\beta)}{\sigma^2}\\ &= -\frac{x_i'x_j}{\sigma^2}, &(2.16)\end{aligned}$$

where x_j is the j^{th} column of X. So

$$(\mathcal{I}(\beta))_{ij} = \frac{x_i'x_j}{\sigma^2}. \tag{2.17}$$

Therefore the Fisher information matrix for this setting is

$$\mathcal{I}(\beta) = \frac{X'X}{\sigma^2}. \tag{2.18}$$

It can be noted that the information matrix is the inverse of the covariance matrix (2.4). Hence,

$$\mathcal{I}(\beta) = \mathrm{Cov}(\widehat{\beta})^{-1}. \tag{2.19}$$

2.3 Optimal design of experiments

2.3.1 Criteria for optimal designs

Results in this section can be found in Van Berkum (2004). The variances of the estimators and the covariances between the estimators determine properties of those estimators. Criteria depend on the information matrix of the design. Widely used criteria include:

1. The D-criterion. This criterion minimizes the generalized variance or the determinant of the covariance matrix. This is useful because the volume of the confidence interval of the vector of unknown parameters, which is an ellipsoid, is proportional to $(\det(X'X)^{-1})^{\frac{1}{2}}$. So a D-optimal design is a design which has a minimal volume of the confidence region for β and therefore has to maximize the determinant of the information matrix.

2. The G-criterion. This criterion minimizes the maximum variance (over $\mathcal{U}$) of the estimated expected response function. The expected response in a point $x \in \mathcal{U}$ is equal to $\mathbb{E}(y_x) = (f(x))'\beta$. The least squares estimator for this expected response is

$$\widehat{\mathbb{E}(y_x)} = (f(x))'\widehat{\beta}. \tag{2.20}$$

   The variance of this estimator is

$$V(\widehat{\mathbb{E}(y_x)}) = (f(x))'(X'X)^{-1}f(x)\sigma^2. \tag{2.21}$$

   The G-criterion considers the maximum

$$\max_{x\in\chi} V(\widehat{\mathbb{E}(y_x)}). \tag{2.22}$$

   A G-optimal design minimizes this maximum. The quantity $V(\widehat{\mathbb{E}(y_x)})/\sigma^2$ may be viewed as a function of x and is called the variance function. The variance function is denoted by

$$d(x, \mathcal{E}) = (f(x))'(X'X)^{-1}(f(x)), \tag{2.23}$$

   where $\mathcal{E}$ is the design.

3. The A-criterion. This criterion minimizes the average of the variance of the unknown parameters or the trace of the covariance matrix. So the sum of the variances of the unknown parameters is minimal in a A-optimal design.

4. The E-criterion. This criterion minimizes the largest eigenvalue of the covariance matrix.

2.3.2 Equivalence Theorem

In this section we show some results about the celebrated Equivalence Theorem of optimal design of experiments. This theorem can be used when finding optimal designs. Under certain conditions pairs of criteria are equivalent. Usually the equivalence between D- and G-optimality is used. So one can use properties of both D- and G-optimal designs.

We give results as stated in Fedorov (1972) and Silvey (1980). Both provide excellent material considering this theorem and its proof, but differ slightly. Our goal is to give a unification of these two approaches.

First we recall model (2.5) and let $f = (f_1, \ldots, f_k)'$. Also we recall the induced design space, $\chi = f(U)$.

The Equivalence Theorem states that for continuous normalized designs, under assumptions we explain later, some of the optimality criteria (see Section 2.3.1) are equivalent. The most import equivalence is the equivalence between D- and G-optimality.

The Equivalence Theorem holds under the following assumptions. This is the part in which Fedorov (1972) and Silvey (1980) differ slightly.

Fedorov (1972) states that

1. The experimental region $\mathcal{U}$ has to be a compact set,

2. The function f as defined in model (2.5) has to be continuous on $\mathcal{U}$.

Silvey (1980) states that the induced design space χ has to be a compact set. These conditions are closely related, as we show now. We show that the condition of Fedorov (1972) implies the condition of Silvey (1980). We use the following theorem.

Theorem 2.1 *Let $X \subseteq \mathbb{R}^n$, $K \subseteq X$ and $f : K \to \mathbb{R}^n$ continuous on K. If K is a compact set, then $f(K)$ is a compact set.*

To prove whether a set K is compact, we need to show that every sequence (b_k) in K has a converging subsequence with its limit in K.

Proof: Let $(b_k)_{k\in\mathbb{N}}$ be a sequence in $f(K)$. By definition, there exist points $a_k \in K$ such that $f(a_k) = b_k$. Because K is a compact set, the sequence $(a_k)_{k\in\mathbb{N}}$ has a convergent subsequence, say $(a_{\ell_k})_{k\in\mathbb{N}}$ with $\lim_{k\to\infty} a_{\ell_k} = a^\star \in K$. Because f is continuous on K is follows that the sequence $(b_{\ell_k})_{k\in\mathbb{N}}$ converges to $b^\star := f(a^\star) \in f(K)$. So we have found a convergent subsequence of the sequence $(b_k)_{k\in\mathbb{N}}$ with a limit in $f(K)$. Therefore $f(K)$ is a compact set.

□

We now take the set K to be the experimental region $\mathcal{U}$ and the function f as defined in model (2.5). The experimental region mostly is a compact set. Most common are the possibilities that this set is a finite set (for instance if we have qualitative factors), or a closed interval (for instance if we have quantitative factors). The functions f_j are most commonly polynomial functions, which are continuous on any experimental region. So the set $\chi = f(\mathcal{U})$ is a compact set.

In the remainder of this section, we proceed with giving information about the Equivalence Theorem, using the conditions of Fedorov (1972). Note that the condition of Silvey (1980) provides is more general. But the conditions are almost similar, and the proof of Fedorov (1972) is easier to understand. So we decided to follow the proof of Fedorov (1972).

In Fedorov (1972) two theorems and three lemmas are used in the proof of the Equivalence Theorem, first stated and proved in Kiefer and Wolfowitz (1960). We first give a version of the Equivalence Theorem, and give the outline of the proof. For an extensive proof, we refer to p. 72 and further of Fedorov (1972). In Pukelsheim (1993) a more general version of the Equivalence Theorem can be found (including the equivalence of A- and E-optimality).

Theorem 2.2 (Kiefer-Wolfowitz Equivalence Theorem) *If we have*

$$y = f_1(x)\beta_1 + \ldots + f_k(x)\beta_k,$$

where

$$\begin{array}{rcl} x & \in & \mathcal{U}, \\ \mathcal{U} & \subseteq & \mathbb{R}^n \\ f_j & : & \mathcal{U} \to \mathbb{R}, \text{ continuous on the experimental region } \mathcal{U}, \end{array}$$

the experimental region $\mathcal{U}$ is a compact set, and the function $f = (f_1, \ldots, f_k)'$ is continuous on $\mathcal{U}$, then, for normalized designs

1. *The following assertions are equivalent:*
   - *(a) the design $\varepsilon^\star$ maximizes $\det(M(\varepsilon))$, where $M(\varepsilon) = \int_{\mathcal{U}} p(x)\ f(x)\ f(x)'\ \mathrm{d}x$, and $\int_{\mathcal{U}} p(x)\ \mathrm{d}x = 1$, (D-optimality),*
   - *(b) the design $\varepsilon^\star$ minimizes $\max_{x \in \mathcal{U}}\ f(x)'\ M^{-1}(\varepsilon)\ f(x)$, (G-optimality),*
   - *(c) $\max_{x \in \mathcal{U}}\ f(x)'\ M^{-1}(\varepsilon^\star)\ f(x) = k$, where k is the number of parameters.*
2. *If $\varepsilon^\star$ and $\widetilde{\varepsilon}$ satisfy (a)-(c), then $M(\varepsilon^\star) = M(\widetilde{\varepsilon})$.*

3. If $\varepsilon^\star$ and $\widetilde{\varepsilon}$ satisfy (a)-(c), then a linear combination of $\varepsilon^\star$ and $\widetilde{\varepsilon}$ satisfy (a)-(c).

Outline of the proof: The proof consists of several steps. First of all the equivalence of (a) and (b) has to be proven. This part uses three lemmas and two theorems. We give these lemmas and theorems, and show how they are used in the proof. We do not give proofs of these lemmas and theorems, proofs can be found on pages 69-71 of Fedorov (1972).

Lemma 2.1 *Let the information matrix of an arbitrary design ε be continuous; then*

1. *The weighted sum of the variances of the estimates of the response surface $d(x, \varepsilon)$, taken over all points of the design ε, is equal to the number of unknown parameters k:*

$$\sum_{i=1}^{n} p_i d(x_i, \varepsilon) = k; \tag{2.24}$$

in the more general case,

$$\int_{\mathcal{U}} d(x_i, \varepsilon) \mathrm{d}\xi(x) = k. \tag{2.25}$$

2. *The minimal value of $\max_{x \in \mathcal{U}} d(x, \varepsilon)$ cannot be less than k:*

$$\max_{x \in \mathcal{U}} d(x, \varepsilon) \geq k. \tag{2.26}$$

Lemma 2.2 *The function $\log det\,(M(\varepsilon))$, where $M(\varepsilon)$ is the information matrix of the design ε, is a strictly concave function.*

Lemma 2.3 *Let there be two designs ε_1 and ε_2 with information matrices $M(\varepsilon_1)$ and $M(\varepsilon_2)$. Then*

$$\frac{d}{d\alpha} \log det\,(M(\varepsilon)) = tr\; M^{-1}(\varepsilon)\left[M(\varepsilon_2) - M(\varepsilon_1)\right], \tag{2.27}$$

where $M(\varepsilon)$ is the information matrix of the design

$$\varepsilon = (1-\alpha)\varepsilon_1 + \alpha\varepsilon_2, \qquad 0 < \alpha < 1.$$

Theorem 2.3 (Carathéodory's Theorem) *Each point $s^\star$ in the convex hull $S^\star$ of any subset S, of the n-dimensional space, can be represented in the form*

$$s^\star = \sum_{i=1}^{n+1} \alpha_i s_i, \tag{2.28}$$

where

$$\alpha_i \geq 0, \qquad \sum_{i=1}^{n+1} \alpha_i = 1, \qquad s_i \in S.$$

Theorem 2.4 *1. For any design ε the information matrix $M(\varepsilon)$ is a symmetric positive-semidefinite matrix.*

2. The matrix $M(\varepsilon)$ is degenerate (det $(M(\varepsilon)) = 0$), if the support of the design ε contains less than k points (where k is the number of unknown parameters).

3. The family of matrices $M(\varepsilon)$, corresponding to all possible normalized designs, is convex. If the function $f(x)$ is continuous on the experimental region $\mathcal{U}$ and $\mathcal{U}$ is compact, then the set of information matrices is compact.

4. For any design ε the matrix $M(\varepsilon)$ can be represented in the form

$$M(\varepsilon) = \sum_{i=1}^{n} p_i f(x_i) f(x_i)', \tag{2.29}$$

where

$$n \leq \frac{m(m+1)}{2} + 1 \qquad 0 \leq p_i \leq 1, \qquad \sum_{i=1}^{n} p_i = 1. \tag{2.30}$$

In the proof Theorem 2.4, Theorem 2.3 is used. We do not give the proofs of these lemmas and theorems. For proofs we refer to Fedorov (1972). We now continue with the outline of the proof of the Equivalence Theorem.

In the part where it is proven that (b) follows from (a), Lemma 2.3 is used to show that for a D-optimal design $\varepsilon^\star$, $d(x, \varepsilon^\star) \leq k$, but from Lemma 2.1 it follows that $\max_{x \in \mathcal{U}} d(x, \varepsilon) \geq k$. So (b) follows from (a). The proof continuous with proving that (a) follows from (b). This is done by contradiction, with the use Lemma 2.3 and Theorem 2.4.

The proof continues with proving the equivalence of (a) and (c), and (b) and (c). These equivalences follow immediately from Lemma 2.1. We complete the proof by giving the last steps of the proof in Fedorov (1972).

Let the designs ε_1 and ε_2 with information matrices $M(\varepsilon_1)$ and $M(\varepsilon_2)$ be D-optimal and $M(\varepsilon_1) \neq M(\varepsilon_2)$.

We consider the information matrix corresponding to the composition of designs ε_1 and ε_2: $M(\varepsilon) = (1-\alpha)M(\varepsilon_1) + \alpha M(\varepsilon_2)$. Corresponding to Lemma 2.2,

$$\log \det (M(\varepsilon)) > (1-\alpha) \log \det (M(\varepsilon_1)) + \alpha \log \det (M(\varepsilon_2)). \tag{2.31}$$

But according to the definition of a D-optimal design

$$\det (M(\varepsilon_1)) = \det (M(\varepsilon_2)) \geq \det (M(\varepsilon)). \tag{2.32}$$

From (2.31) and (2.32) it follows that

$$M(\varepsilon_1) = M(\varepsilon_2) = M(\varepsilon). \tag{2.33}$$

Considering the results that (a)-(c) are equivalent and (2.33), it is not difficult to obtain the validity of the concluding part of the theorem. The theorem is proved.

□

2.4 Paired Comparison Designs

We now switch to the setting of paired comparison designs. In paired comparison experiments observations are made by presenting pairs of objects to one or more judges. The judge decides which object he prefers. This method may be used in cases where objects can be judged only subjectively. Now we can describe a paired comparison design in the following way. The m pairs in which the comparisons are made denoted by $(u_1, v_1), \ldots, (u_m, v_m)$, where $u_i,\ v_i \in \chi$. The number of comparisons in the pair (u_i, v_i) is equal to $n_i \in \mathbb{N}$. The total number of comparisons is denoted by N, so $\sum_{i=1}^{m} n_i = N$. A design is denoted by $\mathcal{E}(N)$ or just $\mathcal{E}$. In the notation of Fedorov (1972) the design $\mathcal{E}$ of a paired comparison experiment may be written as a collection of variables

$$\mathcal{E}(N) = [(u_1, v_1), (u_2, v_2), \ldots, (u_m, v_m); n_1, n_2, \ldots, n_m; N] \tag{2.34}$$

A design may be constructed by choosing both the (u_i, v_i) and the n_i. Mostly the objects have been fixed so only the n_i can be chosen. In the construction of a design as in (2.34) both the pairs -and therefore the objects- and the n_i have to be chosen.

An exact normalized design $\varepsilon(N)$, or ε is a collection of variables

$$\varepsilon(N) = [(u_1, v_1), (u_2, v_2), \ldots, (u_m, v_m); p(u_1, v_1), p(u_2, v_2), \ldots, p(u_m, v_m)] \tag{2.35}$$

where

$$p(u_i, v_i) = n_i/N,$$

and

$$\sum_{i=1}^{m} p(u_i, v_i) = 1. \tag{2.36}$$

If the $p(u_i, v_i)$ can take on any nonnegative value satisfying (2.36) the design (2.35) is called a discrete normalized design. Further, a continuous normalized design is characterized by a probability measure ξ on the region $\mathcal{U}$.

If we assume model (2.5) and an ordinary least squares method, then we can define a matrix $M(\varepsilon)$ as

$$M(\varepsilon) = \int_{\mathcal{U}} \int_{\mathcal{U}} p(x, y)\ (f(x) - f(y))\ (f(x) - f(y))'\ \mathrm{d}x\ \mathrm{d}y \tag{2.37}$$

where

$$f(u_i) \quad = \quad (f_1(u_i), f_2(u_i), \ldots, f_k(u_i))' \tag{2.38}$$

and

$$\int_{\mathcal{U}} \int_{\mathcal{U}} p(x, y)\ \mathrm{d}x\ \mathrm{d}y \quad = \quad 1.$$

This matrix $M(\varepsilon)$ is not the information matrix of the design ε. The actual information matrix would be $M(\varepsilon)/\sigma^2$. Also we introduce the variance function of an estimated response difference between points x and y

$$d(x, y, \varepsilon) = (f(x) - f(y))' \; M^{-1}(\varepsilon)(f(x) - f(y)), \tag{2.39}$$

This is not the actual variance, the variance of an estimated response difference between points x and y is $(f(x) - f(y))' \; M^{-1}(\varepsilon)(f(x) - f(y))\sigma^2$.

Now only the experimental region and the function $f(x)$ have to be specified.

2.5 Calculus

In this section some standard techniques from calculus are presented. These techniques are useful for solving optimization problems, which appear in the design of optimal experiments. This information can be found in Adams (1999) .

2.5.1 Lagrange multipliers

A technique for finding extreme values of a differentiable function $f(x, y)$ subject to the equality constraint $g(x, y) = 0$ is the technique of Lagrange multipliers. In our case the $f(x, y)$ is typically polynomial function. For this technique we introduce the Lagrangian function

$$L(x, y, \lambda) = f(x, y) + \lambda g(x, y). \tag{2.40}$$

At any critical point of L we must have

$$\begin{aligned} \frac{\partial L}{\partial x} &= 0 \\ \frac{\partial L}{\partial y} &= 0 \\ \frac{\partial L}{\partial \lambda} &= 0. \end{aligned}$$

Note first that if one wants to find extreme values of a function on a specific bounded area, the method slightly changes. The extreme values are boundary points, or internal critical points. For the boundary points the method of Lagrange multipliers can be used, for the internal critical points it holds that $\nabla f = 0$.

This definition of Lagrange multipliers can be expanded to n variables with $m \leq n - 1$ constraints.

We give an example to illustrate the use of Lagrange multipliers. This example is Exercise 13.3.1 of Adams (1999).

Example 2.1 *Maximize x^3y^5 subject to the constraint $x + y = 8$.*

First we define:

$$L(x, y, \lambda) = x^3y^5 + \lambda(x + y - 8). \tag{2.41}$$

We now search for critical points of L:

$$\begin{aligned} \frac{\partial L}{\partial x} &= 3x^2y^5 + \lambda = 0 \\ \frac{\partial L}{\partial y} &= 5x^3y^4 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= x + y - 8 = 0. \end{aligned}$$

By eliminating λ from the first two equations we get

$$3x^2y^5 = 5x^3y^4. \tag{2.42}$$

So

$$y = \frac{5}{3}x. \tag{2.43}$$

Substituting this into the third equation yields

$$\begin{aligned} x &= 3 \\ y &= 5 \\ f(3, 5) &= 84375. \end{aligned}$$

This is a maximum, it is the only critical point, and for instance the pair $(x, y) = (8, 0)$ yields $f(8, 0) = 0$ which is smaller then $f(3, 5)$.

This concludes our example.

2.5.2 Hessian matrix

The Hessian matrix $H(f)$ is the square matrix of second partial derivatives of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. If all partial derivatives of order two of f exist, then

$$H(f)_{ij}(x) = D_iD_jf(x)$$

where $x = (x_1, \ldots, x_n)$. So in matrix form

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \tag{2.44}$$

The Hessian matrix can be used to determine the nature of a critical point x.

The following test can be applied to a critical point x. If the matrix $H(f)$ is positive definite at x, then f attains a local minimum at x. An $n \times n$ real matrix A is positive definite if and only if for all $x \in \mathbb{R}^n$ it holds that $x'Ax > 0$. For a symmetric $n \times n$ real matrix A it also holds that this matrix is positive definite if and only if all eigenvalues are positive. Note that the Hessian matrix in the case of polynomial functions f is symmetric.

If the matrix $H(f)$ is negative definite (which has a similar definition as positive definite) at x, then f attains a local maximum at x. If the Hessian has both positive and negative eigenvalues then x is a saddle point for f. The test is inconclusive if the Hessian is positive semidefinite or negative semidefinite.

An example where this test is inconclusive is if we take $f(x, y) = x^2 - y^2$. This results in one critical point, $(0, 0)$. The matrix $H(f)$ is

$$H(f) = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix},$$

which has eigenvalues -2 and 2. So this critical point is a saddle point. The reason why this is called a saddle point, is illustrated in Figure 2.1.

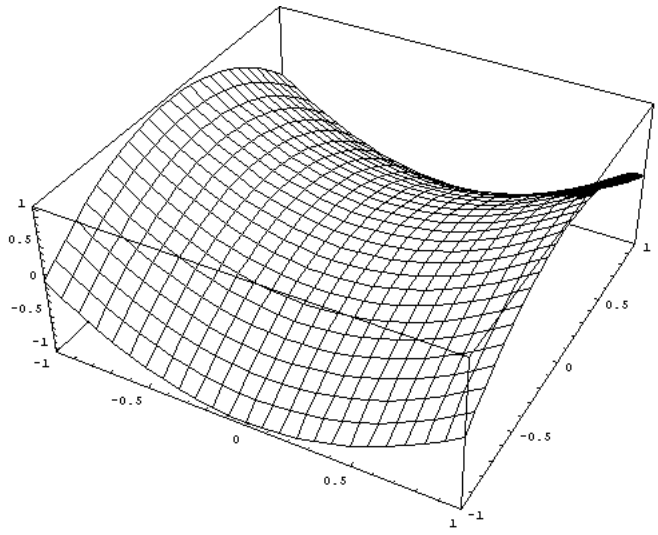

Figure 2.1: Saddle point (0,0) of the function $f(x, y) = x^2 - y^2$.

The application where we use Lagrange Multipliers, or a similar technique, is the following.

Example 2.2 *We assume we would like to investigate a problem where we have two factors, say x_1 and x_2, where $|x_i| \leq 1$, $i = 1, 2$. We would like to find a D-optimal normalized comparison design where we are interested in the main effects, and first order interactions. So our model is*

$$\log \pi = \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2.$$

The number of parameters in this case is $k = 3$. So according to Theorem 2.4, part 4, there is a D-optimal design with no more than 7 pairs.

According to Theorem 3.2.4 of Van Berkum (1985) a D-optimal design ε exists with the pairs of $S(0,2)$ and the pairs of $S(1,1)$ all with weight $\frac{1}{6}$.

$S(0,2)$ contains the pairs

((1,1),(-1,-1))
((1,-1),(-1,1)),

$S(1,1)$ contains the pairs

((1,1),(1,-1))
((1,1),(-1,1))
((-1,-1),(1,-1))
((-1,-1),(-1,1))

A graphical interpretation of these pairs can be found in Figure 2.2 and Figure 2.3. An arrow indicates which two points are compared.

So $\varepsilon = [((1,1),(-1,-1)),((1,-1),(-1,1)),((1,1),(1,-1)),((1,1),(-1,1)),$
$((-1,-1),(1,-1)),((-1,-1),(-1,1));1/6,1/6,1/6,1/6,1/6,1/6]$. *We note that the number of pairs is 6, which is smaller than 7. There could be a design with a smaller number of observations made, but we are satisfied with the result.*

The D-optimality of this design can be verified by verifying the G-optimality of the design, using the Theorem 2.2. Therefore we need the information matrix of ε.

By Theorem 2.4, part 4, we can construct the matrix $M(\varepsilon)$ with the use of the vector

$$(f(x) - f(y)) = \begin{pmatrix} x_1 - y_1 \\ x_2 - y_2 \\ x_1x_2 - y_1y_2 \end{pmatrix}.$$

So we have

$$M(\varepsilon) = \sum_{i=1}^{6} p(u_i, v_i)(f(u_i) - f(v_i))(f(u_i) - f(v_i))',$$

where (u_i, v_i) is the i^{th} pair of ε. Therefore we get

$$M(\varepsilon) = \begin{pmatrix} 8/3 & 0 & 0 \\ 0 & 8/3 & 0 \\ 0 & 0 & 8/3 \end{pmatrix}.$$

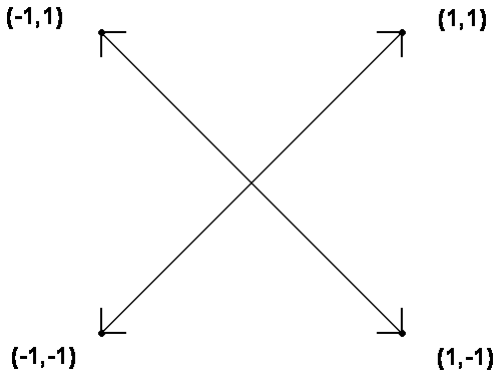

Figure 2.2: The pairs of $S(0, 2)$.

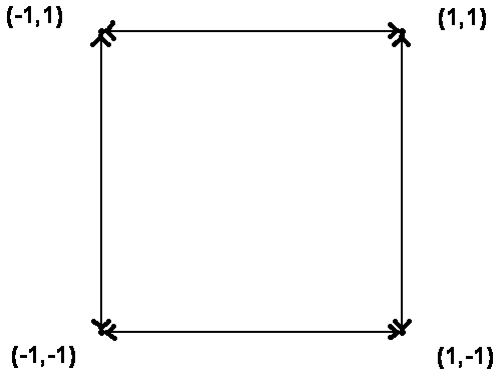

Figure 2.3: The pairs of $S(1, 1)$.

This leads to

$$\begin{aligned} d(x,y,\varepsilon) &= (f(x)-f(y))'M^{-1}(\varepsilon)(f(x)-f(y)) \\ &= (x_1-y_1, x_2-y_2, x_1x_2-y_1y_2)\cdot \begin{pmatrix} 3/8 & 0 & 0 \\ 0 & 3/8 & 0 \\ 0 & 0 & 3/8 \end{pmatrix} \cdot \begin{pmatrix} x_1-y_1 \\ x_2-y_2 \\ x_1x_2-y_1y_2 \end{pmatrix} \\ &= \frac{3}{8}(x_1-y_1)^2+\frac{3}{8}(x_2-y_2)^2+\frac{3}{8}(x_1x_2-y_1y_2)^2. \end{aligned} \tag{2.45}$$

So we have to prove that (2.45) is maximized in the pairs of ε. This is not so obvious. We can use the method of Lagrange Multipliers to do this. The problem with that is that we have to check 3^k-1, so in this example $3^4-1=80$, faces of the hypercube, and the interior of the hypercube. We also note that we are not able to use the properties of the Hessian. This is because the function usually does not have its critical points in this region. The simplify the work that has to be done to check the optimality of the pairs of ε a computer program was written in Mathematica*. The code of this program, and the remaining proof of optimality, can be found in Appendix B.*

This concludes our example.

In the next chapter we discuss an analysis of variance model for DNA microarray experiments.

Chapter 3

Analysis of variance model for DNA microarray experiments

3.1 Introduction to microarray experiments

Microarrays are used for surveying the expression levels of many thousands of genes simultaneously. A microarray experiment typically involves hybridizing two DNA samples (which are converted RNA samples). One sample is labeled with a red-fluorescent dye, Cyanine 5 (Cy_5) and the other is labeled with a green-fluorescent dye, Cyanine 3 (Cy_3). The ratio of the returned light intensities is the quantity used in these experiments.

So this is a case in which the experiments are paired comparison experiments, as we discussed in Section 2.4. The different sources of variance in these experiments are firstly the varieties used, for instance two types of treatment, or two types of tissue. Secondly, some types of genes react better to the green-fluorescent dye than to the red-fluorescent dye, and the other way around. This results in different ratio's when two settings are compared, and the dye is switched between the settings. This swapping of the dye assignment is called a *dye-swap*. In this dye-swap lies our greatest interest. For instance, does a dye-swap result in more information about the parameters?

A third possible source of variation is the use of different arrays.

Before we go on and answer questions about the assignments of the dyes, we give some more information about possible models, or parameterizations, which can be found in the Sections 3.2 and 3.3. We also give possible criteria for optimal paired comparison designs in the setting of DNA microarray experiments. This can be found in the Sections 3.4, 3.5 and 3.6.

In the next sections, we concentrate on finding parameterizations for designs of DNA microarray experiments. These designs can be split up in two types, the one-factor designs (discussed in Section 3.2), and the two-factor designs (discussed in Section 3.3). For the one-factor designs we can use the model proposed in Kerr and Churchill (2001). We discuss this model in the next section.

3.2 One-factor designs

First of all, the name of these designs is confusing. The number of factors in these designs is not equal to one. The name one-factor designs refers to the number of varieties. Examples of these varieties are type of drugs, area of the brain, healthy or infected cell tissue etc.

Other factors that occur beside the varieties are the array on which the experiment is conducted, the dye assigned to a variety and the genes which are compared. We call these factors A (for array), D (dye), V (variety) and G (genes), and call the baselevel effect μ. Then, in Kerr and Churchill (2001) the following simple model is proposed

$$y_{ijkg} = \mu + A_i + D_j + V_k + G_g + (VG)_{kg} + \varepsilon_{ijkg}. \tag{3.1}$$

Here it is assumed that there exists a transformation of microarray data on which the effects are additive (for instance a log scale). Using this scale, y_{ijkg} represents the intensity from array i and dye j of variety k and gene g. This model includes the main effects, and the effect that is usually of interest, the interaction of V and G. Other plausible interactions are (AG) and (DG). The model then becomes

$$y_{ijkg} = \mu + A_i + D_j + V_k + G_g + (VG)_{kg} + (AG)_{ig} + (DG)_{jg} + \varepsilon_{ijkg}. \tag{3.2}$$

Further, independency is assumed, and the error ε_{ijkg} is assumed to be distributed with mean μ and variance σ^2.

If we now take a look at the examples in Yang and Speed (2002), we find nine different one-factor designs. We discuss two different designs, namely a direct design and a design that includes dye-swap. We give the analysis of variance (ANOVA) models of these examples. If two settings are compared, the graphical representation of the design is a "multi-digraph", where an arrow indicates that two settings are compared. A multi-digraph is a set of vertices (points, representing a certain setting) and edges (in this case the edges are in a certain direction, so they are represented by arrows). The tail of the arrow is the setting labeled with the green dye, and the head of the arrow is the setting labeled with the red dye.

Finally, before looking at the examples, note that we need restrictions on the parameters. The number of observations is lower than the total number of parameters. So there are some dependencies between parameters. These dependencies are compensated for by the restrictions.

Example 3.1 *This example can be found on p. 580 of Yang and Speed (2002). There is one factor in this setting. This factor has two possible levels, say A and B. These levels are compared five times, on different arrays, and every time A is labeled with green dye, and B with the red dye. A graphical representation can be found in Figure 3.1. If we now use model (3.2) we find that* $i = 1, \ldots 5$*, representing the five used arrays,* $j = 1, 2$*, where we define* D_1 *as green,* $k = 1, 2$*, where we define* $V_1 = A$*, and* $g = 1, \ldots, n$*.*

So there are five comparisons, all yielding the following relation

$$\begin{aligned} \mathbb{E}\left[y_{A\to B}, G_g\right] &= \mathbb{E}\left[y_{i11g} - y_{i22g}\right] \\ &= D_1 - D_2 + V_1 - V_2 + (VG)_{1g} - (VG)_{2g} + (DG)_{1g} - (DG)_{2g}. \end{aligned}$$

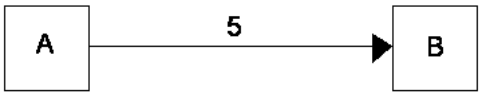

Figure 3.1: Design on p. 580 of Yang and Speed (2002) .

We also have some restrictions on the model. These restrictions are

$$\begin{aligned}
\sum_{i=1}^{5} A_i &= 0, \qquad D_1 + D_2 = 0, \qquad V_1 + V_2 = 0, \\
\sum_{g=1}^{n} G_g &= 0, \qquad \sum_{g=1}^{n} (VG)_{kg} = 0, \quad (VG)_{1g} + (VG)_{2g} = 0, \\
\sum_{g=1}^{n} (AG)_{ig} &= 0, \qquad (AG)_{1g} + (AG)_{2g} = 0, \quad \sum_{g=1}^{n} (DG)_{jg} = 0, \\
(DG)_{1g} &+ (DG)_{2g} = 0.
\end{aligned}$$

Example 3.2 *This example is an adapted version of the example that can be found on the top of p. 582 of Yang and Speed (2002). There is one factor in this setting. This factor has two possible levels, say A and B. These levels are compared two times, on different arrays.*

The first time A is labeled with green dye, and B with the red dye. The second time the dies are swapped. A graphical representation can be found in Figure 3.2. If we now use model (3.2) we find that $i = 1, \ldots 2$*, representing the two used arrays,* $j = 1, 2$*, where we define* D_1 *as green,* $k = 1, 2$*, where we define* $V_1 = A$*, and* $g = 1, \ldots, n$*.*

So there are two comparisons, yielding the following relations

$$\begin{aligned}
\mathbb{E}\left[y_{A \to B}, G_g\right] &= \mathbb{E}\left[y_{i11g} - y_{i22g}\right] \\
&= D_1 - D_2 + V_1 - V_2 + (VG)_{1g} - (VG)_{2g} + (DG)_{1g} - (DG)_{2g}.
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left[y_{A \leftarrow B}, G_g\right] &= \mathbb{E}\left[y_{i22g} - y_{i11g}\right] \\
&= D_2 - D_1 + V_2 - V_1 + (VG)_{2g} - (VG)_{1g} + (DG)_{2g} - (DG)_{1g}.
\end{aligned}$$

We also have some restrictions on the model. These restrictions are

$$\begin{aligned}
A_1 + A_2 &= 0, \qquad D_1 + D_2 = 0, \qquad V_1 + V_2 = 0, \\
\sum_{g=1}^{n} G_g &= 0, \qquad \sum_{g=1}^{n} (VG)_{kg} = 0, \quad (VG)_{1g} + (VG)_{2g} = 0, \\
\sum_{g=1}^{n} (AG)_{ig} &= 0, \qquad (AG)_{1g} + (AG)_{2g} = 0, \quad \sum_{g=1}^{n} (DG)_{jg} = 0, \\
(DG)_{1g} &+ (DG)_{2g} = 0.
\end{aligned}$$

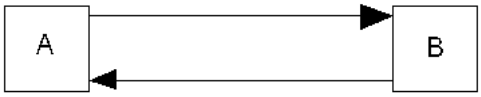

Figure 3.2: Design on p. 582 of Yang and Speed (2002) .

3.3 Two-factor designs

If two types of varieties are involved, for instance one variety is the type of tissue (healthy or tumorous) and the other variety is time, we can not use the model discussed in Section 3.2. In this case not only the main-effects are of interest, but also the interaction between the varieties. The models (3.1) and (3.2) are not useful for this type of interaction between varieties.

In Glonek and Solomon (2003) and Yang and Speed (2002) examples can be found of parameterizations that are useful in this situation. These parameterizations are different from the standard ANOVA parameterization for a design with two factors. We firstly give the standard parameterization, with all conditions, for the situation where two different factors, A and B occur, both with two levels, and the interaction of these factors. We secondly give a parameterization used in Glonek and Solomon (2003). We take τ_1 to be the effect of A at a high level and δ_1 to be the effect of B at a high level.

$$\begin{aligned} Y_{ijk} &= \nu + \tau_i + \delta_j + (\tau\delta)_{ij} + \varepsilon_{ijk} \qquad \begin{cases} i &= 0,1 \\ j &= 0,1 \\ k &= 1,\ldots,n \end{cases} \qquad (3.3) \\ \sum_i \tau_i &= 0, \qquad \sum_j \delta_j = 0, \qquad \sum_i (\tau\delta)_{ij} = 0, \qquad \sum_j (\tau\delta)_{ij} = 0, \end{aligned}$$

where k is the number of replicates. So if we take $k = 1$ (so no replicates), we would have 9 parameters, but only 4 observations. But we also have some restrictions on the parameters. The first two restrictions state that

$$\begin{aligned} \tau_0 + \tau_1 &= 0 \\ \delta_0 + \delta_1 &= 0, \end{aligned}$$

so there are a total of two dependencies in these restrictions. The other four restrictions ($(\tau\delta)_{00} + (\tau\delta)_{10} = 0$, $(\tau\delta)_{01} + (\tau\delta)_{11} = 0$, $(\tau\delta)_{00} + (\tau\delta)_{01} = 0$ and $(\tau\delta)_{10} + (\tau\delta)_{11} = 0$) have a total of three dependencies, so there are five dependencies. This is exactly what we needed, because now there are $9 - 5 = 4$ independent parameters, and also 4 observations.

In Glonek and Solomon (2003) and Yang and Speed (2002) the restrictions are chosen in a different way. They set five parameters to 0. So the parameterization becomes

$$
\begin{aligned}
Y_{ijk} &= \mu + \widetilde{\tau}_i + \widetilde{\delta}_j + \widetilde{(\tau\delta)}_{ij} + \varepsilon_{ijk} \qquad \left\{ \begin{array}{rcl} i &=& 0,1 \\ j &=& 0,1 \\ k &=& 1,\ldots,n \end{array} \right. \qquad (3.4) \\
\widetilde{\tau}_0 &= 0, \qquad \widetilde{\delta}_0 = 0, \qquad \widetilde{(\tau\delta)}_{00} = 0, \qquad \widetilde{(\tau\delta)}_{10} = 0, \qquad \widetilde{(\tau\delta)}_{10} = 0
\end{aligned}
$$

In the following discussion of different parameterizations, we take $k = 1$. So now we can leave the index k out of the parameterization.

Firstly, in Glonek and Solomon (2003) the effects of the parameters at a high level are indicated with α for A, β for B and $(\alpha\beta)$ for the interaction AB. So if we would translate that to parameterization (3.4) this results in

$$
\begin{aligned}
\alpha &= \widetilde{\tau}_1 \\
\beta &= \widetilde{\delta}_1 \\
(\alpha\beta) &= \widetilde{(\tau\delta)}_{11}.
\end{aligned}
$$

The claim in Glonek and Solomon (2003) is that the choice of parameterization has no effect on the contrasts of the factors.

We now derive the contrasts of A, B and AB, for both parameterizations.

If we use parameterization (3.3) the contrasts are

$$
\begin{aligned}
\text{Contrast}(A) &= y_{10} - y_{00} + y_{11} - y_{01} = \tau_1 + (\tau\delta)_{10} - \tau_0 - (\tau\delta)_{00} \\
&= \tau_1 - \tau_0 = -2\tau_0 \\
\text{Contrast}(B) &= \delta_1 - \delta_0 = -2\delta_0 \\
\text{Contrast}(AB) &= y_{11} - y_{10} + y_{00} - y_{01} \\
&= \delta_1 - \delta_0 + (\tau\delta)_{11} - (\tau\delta)_{10} + \delta_0 - \delta_1 + (\tau\delta)_{00} - (\tau\delta)_{01} \\
&= (\tau\delta)_{11} - (\tau\delta)_{10} + (\tau\delta)_{00} - (\tau\delta)_{01} \\
&= -2(\tau\delta)_{10} + 2(\tau\delta)_{00} \\
&= 4(\tau\delta)_{00}.
\end{aligned}
$$

If we use parameterization (3.4) the contrasts are

$$
\begin{aligned}
\text{Contrast}(A) &= y_{10} - y_{00} + y_{11} - y_{01} = \alpha + \alpha + (\alpha\beta) = 2\alpha + (\alpha\beta) \\
\text{Contrast}(B) &= 2\beta + (\alpha\beta) \\
\text{Contrast}(AB) &= y_{11} - y_{10} + y_{00} - y_{01} = \beta + (\alpha\beta) - \beta = (\alpha\beta).
\end{aligned}
$$

So if we use parameterization (3.4) the parameter $(\alpha\beta)$ appears in the contrast of A and B, which is clearly a disadvantage of parameterization (3.4). This is the case because $\mathbb{E}[y_{11}]$ contains the parameters α and β, so if these parameters have to be estimated it is clear that $\mathbb{E}[y_{11}]$ contributes to this estimation.

We note that in the parameterizations with two factors, no other effects are taken into account. For instance, the effect of a dye-swap can not be estimated in these parameterizations. This is at least strange, because this parameterization is used in Yang and Speed (2002), where the use of a dye-swap is advised.

In the next section we discuss some other results of Glonek and Solomon (2003).

3.4 Admissible designs

In this section we discuss some other results of Glonek and Solomon (2003).

The main result in Glonek and Solomon (2003) is the introduction of a new class of designs. These designs are called admissible.

Before we discuss these designs, first we recall parameterization (3.4), and define the vector of parameters

$$\gamma = (\alpha, \beta, (\alpha\beta))'.$$

We now give the definition of admissible designs, as can be found in Glonek and Solomon (2003).

Definition 3.1 (Admissible designs) *A design with a total of n observations and design matrix X is said to be* admissible *if there is no other design with n observations and design matrix $X_\star$, such that*

$$c_i \geq c_i^\star$$

for all i, with strict inequality for at least one i, where $c_i, c_i^\star$ are the diagonal elements of $(X'X)^{-1}$ and $(X_\star' X_\star)^{-1}$ respectively. A design that is not admissible is said to be inadmissable.

In Glonek and Solomon (2003) the parameterization (3.4) is used to find the admissible designs when 6 slides may be used. We now concentrate on the admissible designs in parameterization (3.3) when 6 slides may be used.

To find this set of admissible designs a computer program was written in *Mathematica*. The code of this program can be found in Appendix C. An extensive description of the algorithm used in this program can also be found in Appendix C. We give a brief description here.

Algorithm 3.1 (Admissible designs)

1. ***Input:*** *number of slides that can be used, difference vectors for the chosen parameterization.*
2. *The set of all possible combinations of assignments of dyes to the slides is generated.*
3. *For every assignment, the covariance matrix is constructed. Note that if the design is degenerated the covariance matrix can not be constructed (because det $X'X = 0$). This is saved as a list with at every entry a covariance matrix and the corresponding design matrix.*

4. *The designs in the list are replaced by their notation as in Glonek and Solomon (2003).*

5. *The admissible designs are chosen by the criterion in Glonek and Solomon (2003). This is done with the use of two steps to speed up the process.*

   (a) *All designs with the same values of the diagonal elements of the covariance matrix are viewed as one design (so they are combined in classes).*

   (b) *After checking the admissibility criterion with one design, compared to all other designs, the list of designs is updated, such that inadmissable designs are disposed off.*

6. ***Output:*** *The list of admissible designs (and the diagonal elements of the covariance matrix) is returned. Note that all elements of the admissible classes have to be returned.*

The admissible designs in parameterization (3.3) when 6 slides may be used are given in Tables A.1. The numbers of the configuration correspond with the numbers of the arrows in Figure 3.3.

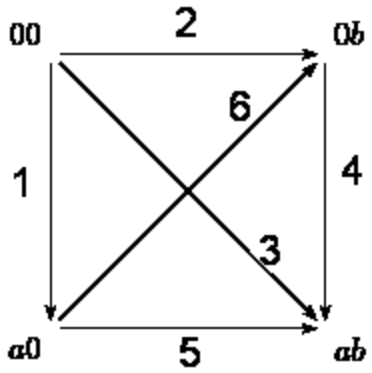

Figure 3.3: Labeling of the arrows of a two factor experiment.

We also use the numbers of the arrows in Figure 3.3 to show how to construct a design matrix X for the parameterizations (3.4) and (3.3). If n slides are available then we assign one of the 6 arrows to each of the slides. Each assignment corresponds with a row in X. So there are six different possible rows for each parameterization. We call these possible rows *design vectors*. Each design vector has three entries, corresponding with the different parameters of interest in the parameterizations.

If parameterization (3.4) is used, the parameters of interest are α, β and $(\alpha\beta)$. The design vectors are:

arrow	calculation			corresponding design vector
1 :	$y_{10} - y_{00} =$	$\mu + \alpha - \mu$	$= \alpha$	$(1,0,0)$
2 :	$y_{01} - y_{00} =$	$\mu + \beta - \mu$	$= \beta$	$(0,1,0)$
3 :	$y_{11} - y_{00} =$	$\mu + \alpha + \beta + (\alpha\beta) - \mu$	$= \alpha + \beta + (\alpha\beta)$	$(1,1,1)$
4 :	$y_{11} - y_{01} =$	$\mu + \alpha + \beta + (\alpha\beta) - (\mu + \beta)$	$= \alpha + (\alpha\beta)$	$(1,0,1)$
5 :	$y_{11} - y_{10} =$	$\mu + \alpha + \beta + (\alpha\beta) - (\mu + \alpha)$	$= \beta + (\alpha\beta)$	$(0,1,1)$
6 :	$y_{01} - y_{10} =$	$\mu + \beta - (\mu + \alpha)$	$= -\alpha + \beta$	$(-1,1,0)$

If parameterization (3.3) is used, we use the representation of the vectors in $\mathbb{R}^2$, with an added entry. The third entry is the product of the two other entries. So 00 becomes $(-1,1,-1)$, $a0$ becomes $(-1,-1,1)$, $0b$ becomes $(1,1,1)$ and ab becomes $(1,-1,-1)$. The design vectors are:

arrow	calculation			corresponding design vector
1 :	$y_{00} - y_{10} =$	$(-1,1,-1) - (-1,-1,1)$	$=$	$(0,2,-2)$
2 :	$y_{00} - y_{01} =$	$(-1,1,-1) - (1,1,1)$	$=$	$(-2,0,-2)$
3 :	$y_{00} - y_{11} =$	$(-1,1,-1) - (1,-1,-1)$	$=$	$(-2,2,0)$
4 :	$y_{01} - y_{11} =$	$(1,1,1) - (1,-1,-1)$	$=$	$(0,2,2)$
5 :	$y_{10} - y_{11} =$	$(-1,-1,1) - (1,-1,-1)$	$=$	$(-2,0,2)$
6 :	$y_{10} - y_{01} =$	$(-1,-1,1) - (1,1,1)$	$=$	$(-2,-2,0)$

We derived these design vectors but noted that we switched the tail and the head of the arrow in the substraction. This has no effect on the estimation of the parameters. In the design vectors this produces an extra $-$, but it is canceled out in the multiplication $X'X$.

We can also generate the admissible designs, when parameterization (3.4) is used. We give these results in Table A.2. The results in this table are the same as in Glonek and Solomon (2003).

If we compare Table A.1 with the admissible designs found in Table A.2, we see that the sets of admissible designs differ greatly. Some admissible designs of Table A.1 are not admissible when parameterization (3.4) is used (for instance the design $\{1,1,1,1,1,1\}$), and vice versa (for instance the design $\{3,1,0,1,1,0\}$).

So now the question arises whether or not the criterion of admissibility is a good one.

We think that there are two possible ways at looking at this. Firstly, if we only look at one fixed parametrization, admissibility would be a nice criterion.

On the other hand, mathematicians would want a criterion to be invariant to transformations on the parameters in the parameterization. In Van Berkum (2004) it is stated that the D- and G-criterion are invariant to a transformation on the parameters of the form $\Theta = A\beta$.

In this case, the transformation is of this form. We prove that in the next section.

In Section 2.3.1 we saw some standard criteria for optimal designs. The D-criterion is usually used, so we take a closer look at this criterion in the next section.

3.5 D_N-efficiency of designs

In the previous section, we saw that the criterion of admissibility was not invariant for the parameterizations (3.3) and (3.4). We now discuss the D-criterion (see Section 2.3.1), especially the D_N-efficiency of a design. We also prove that this D_N-efficiency is invariant under a transformation on the parameters of the form $\Theta = A\beta$. First we define N, the number of observations, and k the number of parameters in the parameterization. Now we can define the D_N-efficiency of a design.

Definition 3.2 (D_N-efficiency) *The D_N-efficiency of a design ε, with design matrix X is defined as*

$$D_N - \mathit{eff} = \left(\frac{\det\,(X'X)}{\det\,(X_D'X_D)} \right)^{\frac{1}{k}},$$

where X_D is the design matrix of a D_N-optimal design.

We show that the D_N-efficiency is invariant when comparing designs when six slides may be used, for the two parameterizations. To show this, we need the D-optimal design. In Van Berkum (1985) this D-optimal design can be found. This is the design we used in Example 2.2.

We give two examples in which we illustrate that the D_N-efficiency is invariant for the two parameterizations (3.3) and (3.4). In these examples we give designs in the notation of Glonek and Solomon (2003). The corresponding labeling of the arrows can be found in Figure 3.3. We refer to design matrices in parameterization (3.3) with X, and in parameterization (3.4) with Y. We only discuss designs where six slides (observations) may be used.

Example 3.3 *The first design we use is one of the three designs that are considered good in Glonek and Solomon (2003). This design is $A = \{2, 2, 0, 1, 1, 0\}$. Before we take a look at the design matrices in the two parameterizations, we need to know the determinant of the information matrix of the D_6-optimal design in both parameterizations.*

So first, the design-, and information matrix of the D_6-optimal design in parameterization (3.3) are

$$X_D = \begin{pmatrix} 0 & 2 & -2 \\ -2 & 0 & -2 \\ -2 & 2 & 0 \\ 0 & 2 & 2 \\ -2 & 0 & 2 \\ -2 & -2 & 0 \end{pmatrix} \qquad X_D'X_D = \begin{pmatrix} 16 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 16 \end{pmatrix}.$$

So $\det\,(X_D'X_D) = 16^3 = 4096$.

The design-, and information matrix of the D_6-optimal design in parameterization (3.4) are

$$Y_D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 1 & 0 \end{pmatrix} \qquad Y_D'Y_D = \begin{pmatrix} 4 & 0 & 2 \\ 0 & 4 & 2 \\ 2 & 2 & 3 \end{pmatrix}.$$

So $\det\ (Y_D'Y_D) = 16$.

We now look at the design-, and information matrices of design A.

$$X_A = \begin{pmatrix} 0 & 2 & -2 \\ 0 & 2 & -2 \\ -2 & 0 & -2 \\ -2 & 0 & -2 \\ 0 & 2 & 2 \\ -2 & 0 & 2 \end{pmatrix} \qquad X_A'X_A = \begin{pmatrix} 12 & 0 & 4 \\ 0 & 12 & -4 \\ 4 & -4 & 24 \end{pmatrix}.$$

So $\det\ (X_A'X_A) = 3072$.

$$Y_A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad Y_A'Y_A = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

So $\det\ (Y_A'Y_A) = 12$.

So to conclude this example, the D_6-efficiencies are for parameterization (3.3) and (3.4) respectively,

$$\begin{aligned} D_6 - \textit{eff} &= \left(\frac{\det\ (X_A'X_A)}{\det\ (X_D'X_D)}\right)^{\frac{1}{3}} = \left(\frac{3072}{4096}\right)^{\frac{1}{3}} = \left(\frac{3}{4}\right)^{\frac{1}{3}} \\ D_6 - \textit{eff} &= \left(\frac{\det\ (Y_A'Y_A)}{\det\ (Y_D'Y_D)}\right)^{\frac{1}{3}} = \left(\frac{12}{16}\right)^{\frac{1}{3}} = \left(\frac{3}{4}\right)^{\frac{1}{3}}. \end{aligned}$$

So the D_6-efficiency is the same for these parameterizations and design A.

Example 3.4 *In this second example we use another of the three designs that are considered good in Glonek and Solomon (2003). This design is* $B = \{2, 2, 0, 1, 0, 1\}$.

We now look at the design-, and information matrices of design B.

$$X_B = \begin{pmatrix} 0 & 2 & -2 \\ 0 & 2 & -2 \\ -2 & 0 & -2 \\ -2 & 0 & -2 \\ 0 & 2 & 2 \\ -2 & -2 & 0 \end{pmatrix} \qquad X_B'X_B = \begin{pmatrix} 12 & 4 & 8 \\ 4 & 16 & -4 \\ 8 & -4 & 20 \end{pmatrix}.$$

So $\det\ (X_B'X_B) = 2048$.

$$Y_B = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix} \qquad Y_B'Y_B = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

So $\det\ (Y_B'Y_B) = 8$.

So to conclude this example, the D_6-efficiencies are for parameterization (3.3) and (3.4) respectively,

$$\begin{aligned} D_6 - \textit{eff} &= \left(\frac{\det\ (X_B'X_B)}{\det\ (X_D'X_D)}\right)^{\frac{1}{3}} = \left(\frac{2048}{4096}\right)^{\frac{1}{3}} = \left(\frac{1}{2}\right)^{\frac{1}{3}} \\ D_6 - \textit{eff} &= \left(\frac{\det\ (Y_B'Y_B)}{\det\ (Y_D'Y_D)}\right)^{\frac{1}{3}} = \left(\frac{8}{16}\right)^{\frac{1}{3}} = \left(\frac{1}{2}\right)^{\frac{1}{3}}. \end{aligned}$$

So the D_6-efficiency is also the same for these parameterizations and design B.

From these examples, the question arises whether this invariance might always be present. We now prove that the D_N-efficiency is invariant under a transformation on the parameters of the form $\Theta = A\beta$.

Lemma 3.1 (Invariance of D_N-efficiency) *Let X be the design matrix of a given design ε. If a transformation on the parameters exists, such that the design matrix for the transformed design, ϵ, is $Y = X \cdot W$, for a given transformation matrix W, then the D_N-efficiencies of ε and ϵ are the same.*

Proof:

Let $\varepsilon^\star$ be the D_N-optimal design. The corresponding design matrices (corresponding to the parameterizations of ε and ϵ) are $X_\star$ and $Y_\star = X_\star \cdot W$. Then the D_N-efficiency of ϵ is

$$\begin{aligned}\left(\frac{\det(Y'Y)}{\det(Y_\star'Y_\star)}\right)^{\frac{1}{k}} &= \left(\frac{\det((XW)'XW)}{\det((X_\star W)'(X_\star W))}\right)^{\frac{1}{k}} \\ &= \left(\frac{\det(W'X'XW)}{\det(W'X_\star'X_\star W)}\right)^{\frac{1}{k}} \\ &= \left(\frac{\det(W')\cdot\det(X'X)\cdot\det(W)}{\det(W')\cdot\det(X_\star'X_\star)\cdot\det(W)}\right)^{\frac{1}{k}} = \left(\frac{\det(X'X)}{\det(X_\star'X_\star)}\right)^{\frac{1}{k}}.\end{aligned}$$

This is, by Definition 3.2, the D_N-efficiency of ε.

□

We now show that for the parameterizations (3.3) and (3.4) such a matrix W exists. Every design ε consist of only six possible different assignments of colors to the slides. These six different assignments result in six different possible rows in the design matrices. This results in the following six equations:

$$\begin{aligned}(0,2,-2)\cdot W &= (1,0,0)\\ (-2,0,-2)\cdot W &= (0,1,0)\\ (-2,2,0)\cdot W &= (1,1,1)\\ (0,2,2)\cdot W &= (1,0,1)\\ (-2,0,2)\cdot W &= (0,1,1)\\ (-2,-2,0)\cdot W &= (-1,1,0).\end{aligned}$$

We only need three independent equations to solve this system. This results in

$$W = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} \end{pmatrix}. \tag{3.5}$$

One could also want to transform the other way round, so we give the matrix W^{-1}

$$W^{-1} = \begin{pmatrix} 0 & 2 & -2 \\ -2 & 0 & -2 \\ 0 & 0 & 4 \end{pmatrix}. \tag{3.6}$$

This implies that the D_N-efficiency is indeed invariant for the parameterizations (3.3) and (3.4), as we saw in Examples 3.3 and 3.4. We also saw that the determinants of the information matrices for corresponding designs in Examples 3.3 and 3.4 differed a factor $\left(\frac{1}{16}\right)^2$. This can be explained from the proof of Lemma 3.1:

$$\det(W'W) = \det\left[\begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} \end{pmatrix}' \cdot \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} \end{pmatrix}\right] = \left(\frac{1}{16}\right)^2.$$

To see why the invariance does not hold for the admissibility criterion, we now take a look at the covariance matrix of an arbitrarily design, and the corresponding transformed covariance matrix. Firstly, note this arbitrarily covariance matrix is symmetric.

$$\begin{aligned} (X'X)^{-1} &= \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \\ ((XW)'(XW))^{-1} &= (W'X'XW)^{-1} = W^{-1}(X'X)^{-1}(W')^{-1} \\ &= 4 \cdot \begin{pmatrix} d+f-2e & c+f-b-e & 2(e-f) \\ c+f-b-e & a+2c+f & -2(c+f) \\ 2(e-f) & -2(c+f) & 4f \end{pmatrix}. \end{aligned} \tag{3.7}$$

From this we see that variance of the interaction parameter only differs a factor 16 between the two parameterizations. The problem lies with the values for the variances of the two main parameters. There is no one to one mapping of the values for the variances of those two parameters. In the transformed covariance matrix, the covariances of B and AB, and A and AB are involved for in the value for the variances of A and B respectively.

We use the D-criterion in the remainder of this project, because of the invariance of the D_N-efficiency for the parameterizations (3.3) and (3.4).

Before we go on, and model the possibility of a dye-swap, we first take a brief look at some other possible criteria, that were taken into consideration.

3.6 Other criteria

We now discuss some other possible criteria. We give optimal designs, and information about possible invariance for these criteria. We also give our conclusion about which criterion we choose. We refer to Appendix C for a *Mathematica* program to find the optimal designs for the different criteria.

Variance of the interaction parameter

In Glonek and Solomon (2003) it is stated that the most important parameter is the interaction parameter, $(\alpha\beta)$. So another criterion that could be interesting is to only look at the value of the variance for this term, and select the design with the least value.

For parameterization (3.3) it turned out that there were two possible optimal designs, both with value 1/24 for the variance of the interaction parameter. Those designs are $\{1, 2, 0, 1, 2, 0\}$ and $\{2, 1, 0, 2, 1, 0\}$.

For parameterization (3.4) it turned out that the optimal designs were the same as for parameterization (3.3). The values for the variance of the interaction parameter are in this parameterization 2/3.

So this criterion seems to be invariant under this transformation. We refer to the transformed covariance matrix (3.7). This invariance can be seen directly from the transformed covariance matrix. The transformed covariance matrix shows a difference of a factor 16 between the variances of the interaction parameter for the two parameterizations. This factor is the same as the difference in the variances of the interaction parameter that we found between the optimal designs for the two parameterizations.

A-criterion

In Section 2.3.1 several other criteria are mentioned. We now look at the A-criterion.

For parameterization (3.3) it turned out that the optimal design is $\{1, 1, 1, 1, 1, 1\}$. For parameterization (3.4) it turned out that the optimal design is $\{2, 2, 0, 1, 1, 0\}$.

So the A-criterion is not invariant under the transformation.

E-criterion

Another criterion mentioned in Section 2.3.1 is the E-criterion.

For parameterization (3.3) it turned out that the optimal design is $\{1, 1, 1, 1, 1, 1\}$. For parameterization (3.4) it turned out that there are three optimal design, $\{1, 3, 0, 0, 2, 0\}$, $\{2, 2, 0, 1, 1, 0\}$ and $\{3, 1, 0, 2, 0, 0\}$.

So the E-criterion is not invariant under the transformation.

Conclusion

We prefer a criterion to be invariant under the transformation. So we prefer the D-criterion.

In the next chapter we give results obtained from the admissible design program. We also model the dye-swap and give optimal designs when a dye-swap is included in the model.

Chapter 4

Results about admissible designs, and including a dye-swap.

In the previous chapter we saw that being admissible was not invariant under a transformation on the parameters of the form $\Theta = A\beta$. In this chapter we give the some other results about optimal designs, which give other drawbacks of admissibility. We obtained these results with the admissible design program of Appendix C. The results about computing time are all obtained with the *Timing*-function in *Mathematica*. These results can be found in Section 4.1. In Section 4.3 we add a dye-swap possibility to our model. In Section 4.4 we give an introduction to the program we wrote. This program can find optimal designs for different criteria. In Section 4.5 we give results about optimal designs when a dye-swap is added to the model. We will also give results about optimal designs without a dye-swap.

In the next section, we give the results about admissible designs.

4.1 Results about admissible designs

In this section we present the results we obtained about admissible designs, using the admissible design program of Appendix C. More specific, we give results about the number of admissible designs for different numbers of slides (n), and we give results about the computing time our program needed. These results contribute to our opinion that admissibility is not a good criterion. Not only is this criterion not invariant under a transformation on the parameters of the form $\Theta = A\beta$, some other drawbacks also occur, which we illustrate. We will not give all tables and figures in this section. We will only give a few of them to illustrate our findings. All other tables and figures of this section can be found in Appendix A.

The first result is about the amount of admissible designs and the time our program needed to find these admissible designs. We did this calculations on an Intel ® Celeron ® 1.4 Ghz laptop, with 512 Mb RAM. We added 4 Gb of virtual memory to speed up the proces and to make it possible to do the computations for larger values of n. We give tables with the amounts of admissible designs, and the time the program took to find these

admissible designs, for both parameterization (3.3) (Table 4.1) and (3.4) (Table A.3).

n	1	2	3	4	5	6	7
admissible designs	0	0	16	39	42	79	78
runtime (in seconds)	0	0	0	0	0	1	1
n	8	9	10	11	12	13	14
admissible designs	180	124	294	180	433	294	597
runtime (in seconds)	2	2	5	6	13	16	33
n	15	16	17	18	19	20	21
admissible designs	430	786	600	1000	792	1239	1006
runtime (in seconds)	43	96	148	268	408	702	1037
n	22	23					
admissible designs	1515	1242					
runtime (in seconds)	1452	2179					

Table 4.1: Number of admissible designs for different numbers of slides (n), and time the program needed to find these designs, using parameterization (3.3)

We also give a graphical interpretation of these results. Firstly we give plots of the amount of admissible designs for the two parameterizations. For parameterization (3.3) this can be found in Figure 4.1. For parameterization (3.4) this can be found in Figure A.1.

Secondly we give plots of the time the program took to find the admissible designs for the two parameterizations. For parameterization (3.3) this can be found in Figure A.4. For parameterization (3.4) this can be found in Figure A.5.

The plot in Figure A.1 looks like an exponential function. The plot in Figure 4.1 looks like two different exponential functions in one plot. To demonstrate this, we plotted the same numbers, but in two different figures. Figure A.2 shows the lengths of the sets of admissible designs for an even number of slides (n). Figure A.3 shows the lengths of the sets of admissible designs for an odd number of slides (n).

We think that the fact that the number of admissible designs in parameterization (3.4) is always increasing, and the number of admissible designs in parameterization (3.3) is alternate increasing and decreasing is a large drawback of admissibility.

The numbers of admissible designs do not grow exponentially. The ratio of the successive numbers decrease slightly as n increases. This holds for both parameterizations, as can be seen in Table A.4 (for an even number of slides, and parameterization (3.3)), Table A.5 (for an odd number of slides, and parameterization (3.3)) and Table A.6 (for parameterization (3.4)).

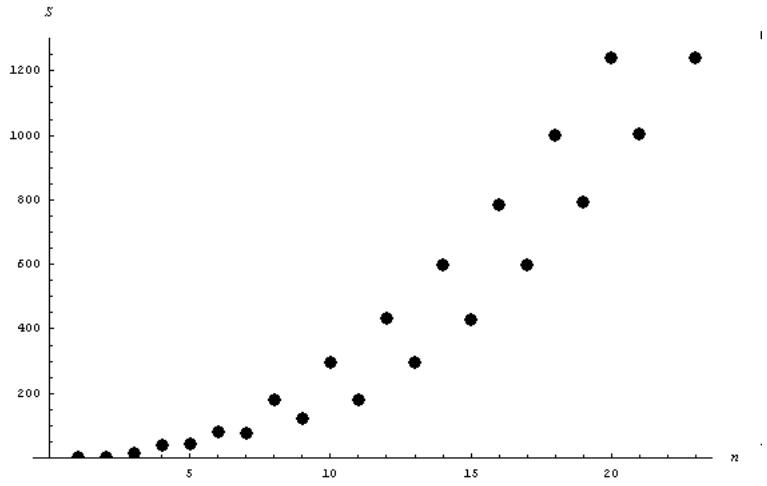

Figure 4.1: Plot of the amount of admissible designs (S) for different numbers of slides (n), using parameterization (3.3)

Finally, we also saw a decrease in the relative values of the number of admissible designs for a certain number of slides, related to the total number of possible combinations. These results van be found in Table A.7 for parameterization (3.3) and Table A.8 for parameterization (3.4).

In the next section we give results about the number of comparisons made, more precisely how we reduced this number.

4.2 Result about the number of comparisons made

Finding admissible designs is not hard, but it uses a lot of computing time. This was the main problem when we wrote the program to find admissible designs, that can be found in Appendix C.

We thought of two things to speed up that process. These thoughts can be found in Algorithm 3.1, but we give them some more attention here. First we want to give a worst case scenario. Without improvements, the number of comparisons that have to be made, when n slides can be used is:

$$\sum_{i=1}^{\binom{n+5}{5}} i - 1 = \frac{1}{2}\binom{n+5}{5}\left(\binom{n+5}{5} - 1\right), \tag{4.1}$$

where the number $\binom{n+5}{5}$ is the total number of assignments (this is a standard combinatoric problem, with solution $\binom{n+6-1}{6-1} = \binom{n+5}{5}$).

For large numbers of slides (values of n) the number of (4.1) is very large, which makes it very time consuming.

We now give the two steps we used to speed up the process, and some explanation about them. After this explanation we give an example to illustrate the effect of these improvements.

Divide the designs in to classes: In stead of comparing all designs (or more precisely, in the case of admissibility, the diagonal elements of the covariance matrices of all designs), we divide the designs in classes. Some designs turned out to have the same diagonal elements in their respectively covariance matrices. So the results of the comparison with other designs is the same for each design in one class.

Update the list with classes of designs after each iteration: Updating the list with classes of designs is done by removing the classes of designs that turned out to be inadmissible. This results in that the number of iterations that have to be done decreases from the length of the set of classes of designs, to the length of the set of classes of admissible designs.

Now we give an example in which we illustrate the effect of these two improvements.

Example 4.1 *In this example, we take a look at the number of comparisons that have to be made, when searching for admissible designs, when* 8 *slides may be used. In this example we use parameterization (3.3).*

If we would not use the improvements, the number of comparisons would be

$$\sum_{i=1}^{\binom{13}{5}} i - 1 = 827,541.$$

If we would leave out the degenerate designs, this would improve the number with about 230,000 comparisons. If these degenerate designs are left out, only 1092 *(in stead of* $\binom{13}{5} = 1287$*) designs are left. So the number of comparisons would be*

$$\sum_{i=1}^{1092} i - 1 = 595,686.$$

In Table 4.2 the lengths of the list of classes of designs, after each iteration can be found.

Using this table we can calculate the number of comparisons that had to be made, after implementing the improvements. This turned out to be $3,887$*, which is almost a factor* 200 *less than without the improvements.*

This concludes our example.

Before we give information about modeling the dye-swap, and start looking for optimal designs with dye-swaps, we make one final remark. The improvements we made in the time the program needs to find sets of admissible designs is helpful when the number of

i	C
0	215
1	192
2	136
3	128
4	123
5-11	118

i	C
12	116
13	114
14	112
15-21	110
22	109
23-25	94

i	C
26-28	79
29-39	78
40-63	70
64	68
65-66	66

Table 4.2: Lengths of the list of classes of designs C, after updating the list i times.

slides that may be used is not larger than 20. After this the numbers of classes of designs are also huge. The computer needs too much memory to find the sets of admissible designs quickly. This can be explained by the fact that before the first update of the list, the first element of the list of classes of designs is compared with all other elements in that list. This alone takes up a lot of memory.

This is not a real problem. The fact is that slides are fairly expensive, so the number of slides usually lies between 6 and 12.

In the next section, we give a model where a dye effect is present.

4.3 Dye-swap

In this section we concentrate on finding optimal designs when a dye-swap is called for. In the next section we model the dye-swap possibility.

4.3.1 Modeling the dye-swap

In this section we give our approach to modeling the dye-swap. There are some difficulties when modeling the dye-swap. Firstly, when adding a new factor to the model, a lot of interactions appear, when a full factorial model is chosen. Secondly, when adding a new factor to the model, a choice has to be made how the information has to be modeled in the design vectors.

We leave out all interactions for this new factor. The reason why we do this is quite simple. We expect that the dye-factor has no interpretable interactions with other parameters. Another, less important reason is that the more interactions we include in the model, the more observations are needed to get an estimation for all parameters. Microarray experiments are expensive, so doing less observations is preferable.

For the second problem we had some insightful discussions with dr. Eric D. Schoen. When an arrow is reversed (for instance in Figure 3.3) the information about the three parameters corresponding to the varieties (and the interaction) is reversed, but the information about the dye stays the same. The latter is because the information that is retrieved from an experiment is always a ratio of red and green. So we model this by

doubling the number of possible vectors. This is done by taking the original vectors, and for every vector v we also add $-v$. After this, we add a 1 to each vector (because the information about the dye is the same, also for the reversed arrows).

We give an example of this procedure.

Example 4.2 *If parameterization (3.4) is used, the arrows of Figure 3.3 correspond to the following vectors*

$$\{1\ ,0\ ,0\}$$
$$\{0\ ,1\ ,0\}$$
$$\{1\ ,1\ ,1\}$$
$$\{1\ ,0\ ,1\}$$
$$\{0\ ,1\ ,1\}$$
$$\{-1\ ,1\ ,0\}.$$

Now we also add the reversed arrows, and add a 1 *to each vector. So for the dye-swap model in parameterization (3.4) the vectors are*

$$\begin{array}{ll}\{1\ ,0\ ,0\ ,1\} & \{-1\ ,0\ ,0\ ,1\}\\ \{0\ ,1\ ,0\ ,1\} & \{0\ ,-1\ ,0\ ,1\}\\ \{1\ ,1\ ,1\ ,1\} & \{-1\ ,-1\ ,-1\ ,1\}\\ \{1\ ,0\ ,1\ ,1\} & \{-1\ ,0\ ,-1\ ,1\}\\ \{0\ ,1\ ,1\ ,1\} & \{0\ ,-1\ ,-1\ ,1\}\\ \{-1\ ,1\ ,0\ ,1\} & \{1\ ,-1\ ,0\ ,1\}.\end{array}$$

This concludes our example.

In the next section we give an introduction to our optimal design program.

4.4 Optimal design program

In this section we give an introduction to our optimal design program. We give some background information on how the program works. We also give some examples of which output is generated from given input.

We discuss the possibilities of the program, and we give a brief overview which functions we used in *Mathematica* to find the optimal designs. For further mathematical background we refer to the different corresponding sections of this thesis.

The program was first designed to find sets of admissible designs for different settings. Later we expanded it to find optimal designs for the majority of the criteria we used during this project. We give an overview of the criteria that can be chosen.

1. D-criterion,
2. A-criterion,
3. E-criterion,
4. Admissible designs
   (a) two-factor designs with interaction (called Admissible3),
   (b) three-factor designs where one factor is the dye-factor, which has no interactions (called Admissible4).

For all of these criteria the information matrices of all designs are needed. So the first step is to generate all possible information matrices.

To find all possible designs we used the function *Compositions*. This function does not generate the designs, but it generates a set which we can easily use to generate the designs. We then generate all corresponding covariance matrices, if they exist (otherwise the design is degenerate). We then use some functions to get a nicer representation of these designs. This representation corresponds with the notation of Glonek and Solomon (2003) for designs.

Now we have the information matrices of all possible non-degenerate designs. We then use the different criteria to decide which designs are optimal. For the admissible designs we use Algorithm 3.1 to pick the optimal designs. For the other criteria the process is not that complex. We generate a list where we can pick the optimal designs instantly. For instance, for the D-criterion we generate a list with the determinant of the covariance matrix for each design, and pick the designs with the smallest determinant.

A list with optimal designs is returned. This list consists of the value of the criterion, and the design, in the notation of Glonek and Solomon (2003).

We give two examples of the output of the program for given input. More detailed information on which input can be given to the program can be found in Appendix C.1.

Example 4.3 *Suppose we are allowed to use* 8 *slides for an experiment. We know that there are two factors of interest, and we model them with parameterization (3.3). We are not interested in a dye-effect, and want a* D_8*-optimal design.*

So our input in Mathematica *is:*

optimaldesigns[8,{{0,2,-2},{-2,0,-2},{-2,2,0},{0,2,2},{-2,0,2},{-2,-2,0}},1],

and the program returns the results:

$$\begin{pmatrix} \frac{1}{9216} & \{1,1,2,1,1,2\} \\ \frac{1}{9216} & \{1,2,1,1,2,1\} \\ \frac{1}{9216} & \{2,1,1,2,1,1\} \end{pmatrix}.$$

So we have to choose one of these three designs. This concludes our example.

Example 4.4 *Suppose we are allowed to use* 7 *slides for an experiment. We know that there are two factors of interest, and we model them with parameterization (3.4). We are interested in a dye-effect, and want an A-optimal design.*

So our input in Mathematica *is:*

```
optimaldesigns[7,
                {{1, 0, 0, 1}, {-1, 0, 0, 1}, {0, 1, 0, 1}, {0, -1, 0, 1}, {1, 1, 1, 1},
                {-1, -1, -1, 1}, {1, 0, 1, 1}, {-1, 0, -1, 1}, {0, 1, 1, 1}, {0, -1, -1, 1},
                {-1, 1, 0, 1}, {1, -1, 0, 1}},
              2],
```

and the program returns the results:

$$\begin{pmatrix} \frac{223}{140} & \{0,2,2,0,0,0,1,0,0,1,0,1\} \\ \frac{223}{140} & \{1,1,1,1,0,0,0,1,1,0,0,1\} \\ \frac{223}{140} & \{1,1,1,1,0,0,1,0,0,1,1,0\} \\ \frac{223}{140} & \{2,0,0,2,0,0,0,1,1,0,1,0\} \end{pmatrix}.$$

So we have to choose one of these four designs. This concludes our example.

In the next section we give optimal designs when a dye-swap is added to the model.

4.5 Optimal designs

In this section we give optimal designs when a dye-swap is added to the model. We chose only to give the number of optimal designs. The reason for this is that for larger number of slides the number of optimal design can also be large. We did generate all optimal designs. We will put them online in *Mathematica* files. These files can be found through http://www.win.tue.nl/bs/software.html.

We give results about number of optimal designs and computing time in seconds for the 5 possible criteria for the two parameterizations, with and without a dye-factor, for different numbers of slides. In Section 4.6 we state some interesting or surprising facts about the optimal designs we found.

All results are found using the program discussed in the previous section. The results about computing time are generated in *Mathematica* using the *Timing* function.

We start with D_N-optimal designs.

D_N-optimal

In Tables 4.3 and 4.4 the results about D_N-optimal designs can be found. The number of slides varies between 1 and 12.

number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	1	3	14	41	106	247	515	1126	1942
number of optimal designs	0	0	0	6	36	24	60	18	32	48	132	15
number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	1	4	14	42	105	228	486	1002	1895
number of optimal designs	0	0	0	6	36	24	60	18	32	48	132	15

Table 4.3: Computing time and number of D_N-optimal designs for different number of slides, when a dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	0	1	1	2	3	5	7
number of optimal designs	0	0	16	3	6	1	6	3	12	3	6	1
number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	0	1	1	2	3	4	6
number of optimal designs	0	0	16	3	6	1	6	3	12	3	6	1

Table 4.4: Computing time and number of D_N-optimal designs for different number of slides, when no dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

A-optimal

In Tables 4.5 and 4.6 the results about A-optimal designs can be found. The number of slides varies between 1 and 12.

number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	1	3	11	30	75	169	391	1149	2498
number of optimal designs	0	0	0	2	8	4	4	8	8	6	12	3
number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	1	6	19	52	128	290	428	1006	2825
number of optimal designs	0	0	0	6	36	24	60	18	32	216	132	15

Table 4.5: Computing time and number of A-optimal designs for different number of slides, when a dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	1	1	2	3	4	5	5
number of optimal designs	0	0	2	1	2	1	2	2	2	1	2	1
number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	1	1	1	2	2	5	10
number of optimal designs	0	0	4	3	6	1	6	3	12	3	6	1

Table 4.6: Computing time and number of A-optimal designs for different number of slides, when no dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

E-optimal

In Tables 4.7 and 4.8 the results about E-optimal designs can be found. The number of slides varies between 1 and 12.

number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	1	7	26	75	189	435	983	3657	5649
number of optimal designs	0	0	0	2	4	8	8	8	4	8	8	89
number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	2	7	26	75	194	453	1003	2307	4843
number of optimal designs	0	0	0	6	24	52	96	255	344	486	744	1501

Table 4.7: Computing time and number of E-optimal designs for different number of slides, when a dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	1	1	2	3	5	8	12
number of optimal designs	0	0	2	3	2	3	2	2	2	2	2	5
number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	1	1	2	3	5	8	11
number of optimal designs	0	0	4	3	18	1	6	24	4	3	18	1

Table 4.8: Computing time and number of E-optimal designs for different number of slides, when no dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

Admissible3

In Tables 4.9 and 4.10 the results about Admissible3 designs can be found. The number of slides varies between 1 and 12 for the designs without a dye-factor, and between 1 and 9 for the designs with a dye-factor.

number of slides	1	2	3	4	5	6	7	8	9
time in seconds	0	0	0	1	5	34	324	2163	10944
number of optimal designs	0	0	0	22	56	98	252	409	602
number of slides	1	2	3	4	5	6	7	8	9
time in seconds	0	0	1	1	4	28	191	507	1617
number of optimal designs	0	0	0	6	132	636	1812	1719	1940

Table 4.9: Computing time and number of Admissible3 designs for different number of slides, when a dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	1	1	3	8	21	50	121
number of optimal designs	0	0	2	5	12	21	38	50	66	97	135	175
number of slides	1	2	3	4	5	6	7	8	9	10	11	12
time in seconds	0	0	0	0	0	1	1	3	3	7	8	16
number of optimal designs	0	0	16	39	42	79	78	180	124	294	180	433

Table 4.10: Computing time and number of Admissible3 designs for different number of slides, when no dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

Admissible4

In Table 4.11 the results about Admissible4 designs can be found. The number of slides varies between between 1 and 9. Note that the dye-factor is necessary for this criterion, so there are no results about Admissible4 designs without the dye-factor.

number of slides	1	2	3	4	5	6	7	8	9
time in seconds	0	0	0	1	5	34	325	2168	11314
number of optimal designs	0	0	0	22	68	116	260	433	750
number of slides	1	2	3	4	5	6	7	8	9
time in seconds	0	0	0	1	4	37	200	327	1091
number of optimal designs	0	0	0	6	132	792	1980	1719	1940

Table 4.11: Computing time and number of Admissible4 designs for different number of slides, when a dye-factor is added. The top three rows are for parameterization (3.4), the bottom three rows are for parameterization (3.3).

In the next section we present some surprising and interesting facts about the optimal designs we found.

4.6 Facts about the optimal designs

In this section we present some surprising and interesting facts about the optimal designs we found.

Before we give some facts about the optimal designs, we want to give our main conclusion. In Section 3.6 and 4.1 we stated some facts why we prefer the D-criterion (or, in this case, where we have exact designs only, the D_N criterion). We think that the results in the previous section also contribute to our point of view.

Firstly we saw that the number of optimal designs was large for admissible designs. We already saw that in Section 4.1, but now we have some more material to compare these numbers. The drawback of a large set of optimal (or in this case admissible) designs is that you only want to choose one. In this large set, some designs have better estimations for specific parameters. For instance, for 9 slides, parameterization (3.3) and the admissible4 criterion the design $\{0,0,0,1,1,2,0,0,0,1,4,0\}$ has a better estimation for τ_1, but the design $\{3,1,1,0,0,0,3,0,0,1,0,0\}$ has a better estimation for the interaction parameter. Both are admissible4, so which one to choose depends on the problem. But if there would be a parameter of special interest, one could consider to choose the design with the most information on that parameter. This could be a design that would not be admissible4.

Secondly, for more than 9 slides the computer could not complete the computation for both admissibility criteria, when a dye-factor was added to the model. The computation needed to much memory. So combining this with our first argument, we advice not to choose designs according to one of the admissibility criteria.

Thirdly we saw that the computing time for E-optimal designs was much larger than for the A- and D_N-optimal designs. This time difference can be more than a factor 2. We also saw that the number of E-optimal designs is very fluctuating for parameterization (3.3) for different numbers of slides. The number of E-optimal designs with a dye-factor added, using parameterization (3.3) is very large. Also, there is a lot of difference in the number of optimal designs between the two parameterizations (3.3) and (3.4). Mostly because of our first argument, and the fact that the E-criterion is not invariant under the two parameterizations, we advice not to choose designs according to the E-criterion.

Fourthly, for the A-criterion we also saw there is a lot of difference (at least a factor 3) in the number of optimal designs between the two parameterizations (3.3) and (3.4). This can be explained by the fact that the A-criterion is not invariant under the two parameterizations.

Finally, due to the results of this thesis, we would advice choosing optimal designs under the D_N-criterion. The main reason to do this is the invariance of the D_N-efficiency, as proven in Section 3.5. In this chapter we added some drawbacks to the other criteria we discussed in this thesis. So if we add those things up, our advice for an optimality criterion is the D_N-criterion.

In the next chapter we give the final conclusions of this thesis.

Chapter 5

Conclusion

In a DNA microarray experiment, design of experiments can be used to optimize the estimations of the results of the experiment. This can be done by allocating the right settings of the parameters to different microarray slides.

The aim of this thesis was to find optimal designs for microarray experiments. First we looked in the literature to find what was done already on this topic. In Van Berkum (1985) we found useful tools to model the problem. Also in that PhD thesis were solutions for some of the well known criteria, like D-optimality. In Glonek and Solomon (2003) a new class of optimal designs were introduced. This class is called admissible designs. In Yang and Speed (2002) it is stated that the assignment of the red and green dye could effect the results. Some genes could possibly have a better response with green than with red, or vice versa. Finding optimal designs for microarray experiments with an added dye-factor was our second goal.

We started with modeling the microarray experiments. In Glonek and Solomon (2003) this was done with a different parameterization that usually. We discussed this parameterization as parameterization (3.4). We saw that for the usual parameterization (parameterization (3.3)) the set of admissible designs changed. This invariance is of course a major drawback of the admissibility criterion. We first showed that there was a transformation of the form $\Theta = A\beta$ between the two parameterizations and secondly proved that the D_N-efficiency of designs is invariant under such a transformation. We also gave examples which showed that the A- and E-criterion are not invariant under a transformation of the form $\Theta = A\beta$.

We found some other drawbacks of admissible designs. Firstly, the computing time needed to find the set of admissible designs grows almost exponentially with the number of slides that may be used. Secondly, the number of admissible designs also grows swiftly as the number of slides that may be used increase.

We then concentrated on adding a dye-factor to the model. We had an insightful discussion with Dr. E.D. Schoen and decided to leave out all interactions with this new factor. So we only added one linear term to the model. This model can be found in Section 4.3.1. We concluded that we could distinguish two possible admissibility criteria. The first one we called admissible3. This criterion threats the new parameter as a nuisance

parameter. So the variance of this parameter is not of interest in the admissible3 criterion. On the other hand, in Glonek and Solomon (2003) it was stated that the dye-factor could be of interest. So we added an admissibility criterion that takes all variances of the parameters and threats them as equal. We call this latter criterion admissible4. We used this to find optimal designs for different criteria and the two different parameterizations. These results can be found in Section 4.5.

A large part of this thesis also consisted of writing and implementing an algorithm to choose optimal designs out of several designs. We have implemented the algorithm in *Mathematica*. The program finds optimal designs for a given number of slides, a given parameterization and a given criterion. In the program some of the standard criteria can be chosen. These build in criteria are the D-, A- and E-criterion and admissible3 and -4. We also improved the program such that the computation time was reduced strongly. Hence, an optimal design can be found fairly quick. An elaborate explanation of this program can be found in Section 4.4 and Appendix C. We used this program to find all the results we stated above.

This concludes our research on this subject. There are some possibilities to do further research on this topic. For instance, one could do research on D_N-optimal designs. Maybe there is a relation between D_N-optimal designs and D_{N-1}-optimal designs. The information in our *Mathematica* notebook files could be a handy guide for this research.

Optimal Experimental Designs for DNA Microarray Experiments is a very interesting subject to work on. With this thesis we hope to have given a insight into the different parameterizations and criteria that can be used to find optimal experimental designs. We also wanted to hand some possible solutions on how the problem can be solved, which can be seen as a start of further research. In this thesis we concentrated on finding these optimal designs in an extensive way, but maybe there is a smarter way to find these optimal designs. Hence, the research on this fascinating subject can certainly be elaborated further.

Appendix A

Tables and figures

In this Appendix the tables and figures referred to in Sections 4.1 and 3.4 can be found.

Table A.1: Admissible designs with 6 slides, when parameterization (3.3) is used.

c_τ	c_δ	$c_{(\tau\delta)}$	1	2	3	4	5	6
5/96	5/96	29/96	0	0	2	0	1	3
			0	0	2	1	0	3
			0	0	3	0	1	2
			0	0	3	1	0	2
			0	1	2	0	0	3
			0	1	3	0	0	2
			1	0	2	0	0	3
			1	0	3	0	0	2
9/160	9/160	5/32	0	0	2	1	1	2
			0	1	2	1	0	2
			1	0	2	0	1	2
			1	1	2	0	0	2
3/63	17/192	17/192	0	1	1	0	2	2
			0	1	2	0	2	1
			0	2	1	0	1	2
			0	2	2	0	1	1
11/208	15/208	19/208	0	1	1	1	1	2
			0	1	2	1	1	1
			1	1	1	0	1	2
			1	1	2	0	1	1
1/8	1/24	1/16	2	0	1	2	0	1
1/8	1/16	1/24	2	1	0	2	1	0
1/24	1/16	1/8	0	1	2	0	1	2

Table A.1: (continued)

11/208	19/208	15/208	0	1	1	1	2	1
			0	2	1	1	1	1
			1	1	1	0	2	1
			1	2	1	0	1	1
5/96	29/96	5/96	0	2	0	0	3	1
			0	2	0	1	3	0
			0	2	1	0	3	0
			0	3	0	0	2	1
			0	3	0	1	2	0
			0	3	1	0	2	0
			1	2	0	0	3	0
			1	3	0	0	2	0
9/160	5/32	9/160	0	2	0	1	2	1
			0	2	1	1	2	0
			1	2	0	0	2	1
			1	2	1	0	2	0
1/24	1/8	1/16	0	2	1	0	2	1
1/16	1/24	1/8	1	0	2	1	0	2
15/208	11/208	19/208	1	0	1	1	1	2
			1	0	2	1	1	1
			1	1	1	1	0	2
			1	1	2	1	0	1
17/192	3/64	17/192	1	0	1	2	0	2
			1	0	2	2	0	1
			2	0	1	1	0	2
			2	0	2	1	0	1
19/208	11/208	15/208	1	0	1	2	1	1
			1	1	1	2	0	1
			2	0	1	1	1	1
			2	1	1	1	0	1
15/208	19/208	11/208	1	1	0	1	2	1
			1	1	1	1	2	0
			1	2	0	1	1	1
			1	2	1	1	1	0
19/208	15/208	11/208	1	1	0	2	1	1
			1	1	1	2	1	0
			2	1	0	1	1	1
			2	1	1	1	1	0

Table A.1: (continued)

1/16	1/16	1/16	1	1	1	1	1	1
1/16	1/8	1/24	1	2	0	1	2	0
17/192	17/192	3/64	1	1	0	2	2	0
			1	2	0	2	1	0
			2	1	0	1	2	0
			2	2	0	1	1	0
5/32	9/160	9/160	2	0	0	2	1	1
			2	0	1	2	1	0
			2	1	0	2	0	1
			2	1	1	2	0	0
29/96	5/96	5/96	2	0	0	3	0	1
			2	0	0	3	1	0
			2	0	1	3	0	0
			2	1	0	3	0	0
			3	0	0	2	0	1
			3	0	0	2	1	0
			3	0	1	2	0	0
			3	1	0	2	0	0
1/16	1/8	1/24	1	2	0	1	2	0

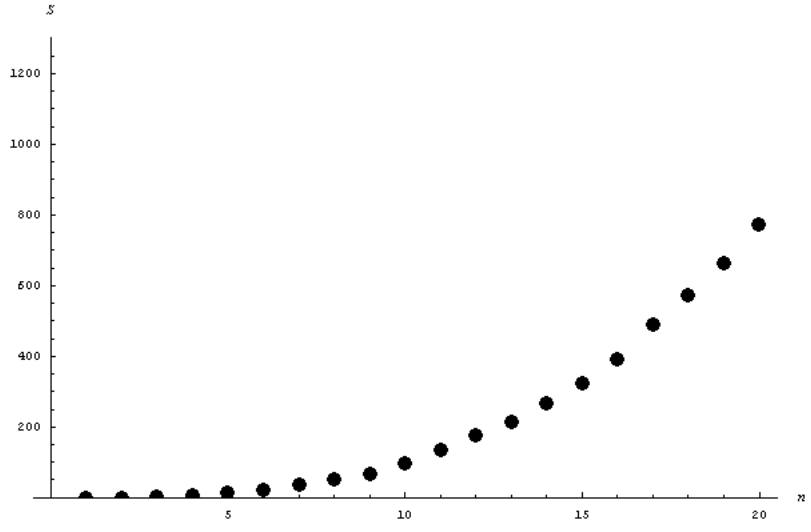

Figure A.1: Plot of the number of admissible designs (S) for different numbers of slides (n), using parameterization (3.4)

c_α	c_β	$c_{(\alpha\beta)}$	1	2	3	4	5	6
7/13	5/13	11/13	1	2	0	1	1	1
2/3	5/12	2/3	1	2	0	1	2	0
6/13	5/13	15/13	1	2	1	0	1	1
4/7	2/7	9/7	1	3	0	0	1	1
7/10	3/10	4/5	1	3	0	1	1	0
2/3	1/3	1	1	3	1	0	1	0
1	1/4	5/4	1	4	0	0	1	0
5/13	7/13	11/13	2	1	0	1	1	1
5/12	2/3	2/3	2	1	0	2	1	0
5/13	6/13	15/13	2	1	1	1	0	1
3/8	3/8	11/8	2	2	0	0	1	1
			2	2	0	1	0	1
5/12	5/12	3/4	2	2	0	1	1	0
2/5	1/2	11/10	2	2	1	0	1	0
1/2	2/5	11/10	2	2	1	1	0	0
1/2	1/3	4/3	2	3	0	0	1	0
2/7	4/7	9/7	3	1	0	1	0	1
3/10	7/10	4/5	3	1	0	1	1	0
1/3	2/3	1	3	1	1	1	0	0
1/3	1/2	4/3	3	2	0	1	0	0
1/4	1	5/4	4	1	0	1	0	0

Table A.2: Admissible designs with 6 slides, when parameterization (3.4) is used.

n	1	2	3	4	5	6	7
admissible designs	0	0	2	5	12	21	38
runtime (in seconds)	0	0	0	0	0	1	1
n	8	9	10	11	12	13	14
admissible designs	50	66	97	135	175	213	267
runtime (in seconds)	3	7	16	38	86	178	368
n	15	16	17	18	19	20	
admissible designs	324	391	488	572	663	774	
runtime (in seconds)	692	1307	2458	3954	6549	10720	

Table A.3: Number of admissible designs for different numbers of slides (n), and time the program needed to find these designs, using parameterization (3.4)

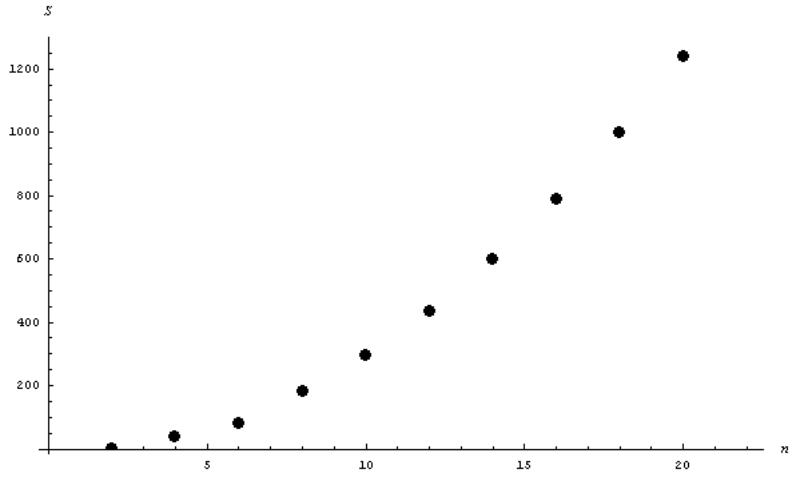

Figure A.2: Plot of the number of admissible designs (S) for different even numbers of slides (n), using parameterization (3.3)

n	3	4	5	6	7	8	9	10	11
	2.03	2.28	1.63	1.47	1.38	1.32	1.27	1.24	1.22

Table A.4: Ratio of the numbers of admissible designs for different values of $2n$ and $2n-2$, when using parametrization (3.3)

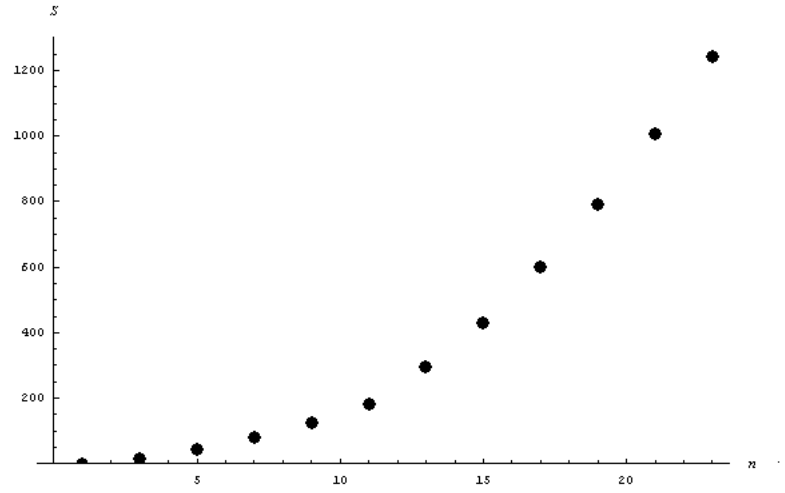

Figure A.3: Plot of the number of admissible designs (S) for different odd numbers of slides (n), using parameterization (3.3)

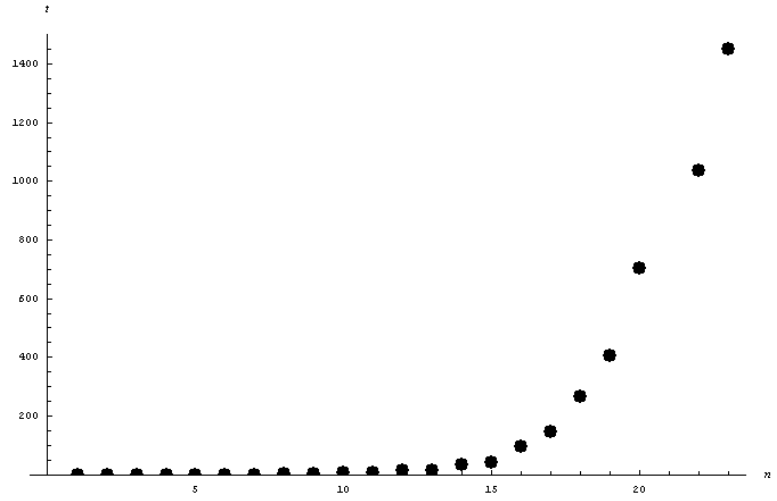

Figure A.4: Plot of the time (in seconds) the program needed to find the admissible designs for different numbers of slides (n), using parameterization (3.3)

n	2	3	4	5	6	7	8	9	10	11
	2.63	1.86	1.59	1.45	1.63	1.46	1.40	1.32	1.27	1.23

Table A.5: Ratio of the numbers of admissible designs for different values of $2n + 1$ and $2n - 1$, when using parametrization (3.3)

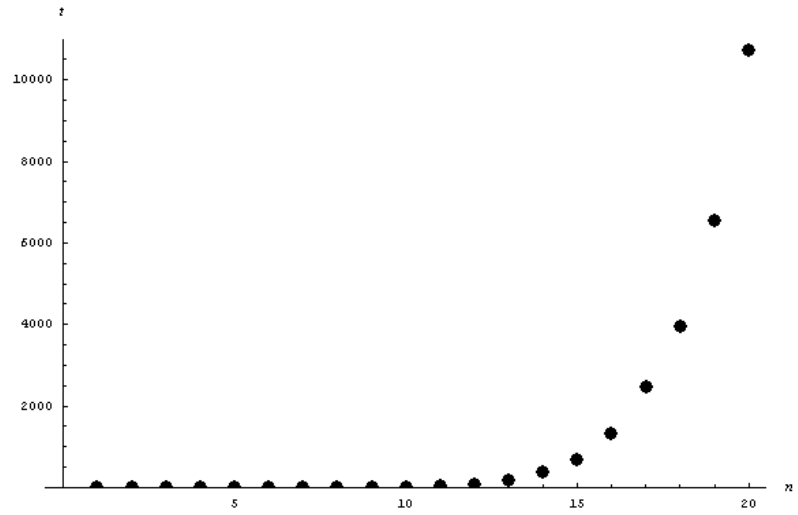

Figure A.5: Plot of the time (in seconds) the program needed to find the admissible designs for different numbers of slides (n), using parameterization (3.4)

n	4	5	6	7	8	9	10	11	12
	2.5	2.4	1.75	1.81	1.32	1.32	1.47	1.39	1.30
n	13	14	15	16	17	18	19	20	
	1.22	1.25	1.21	1.21	1.25	1.17	1.16	1.17	

Table A.6: Ratio of the numbers of admissible designs for different values of n and $n-1$, when using parametrization (3.4)

n	1	2	3	4	5	6	7	8	9	10
$S/\binom{n+5}{5}$	0	0	0.286	0.310	0.167	0.171	0.098	0.140	0.062	0.980
n	11	12	13	14	15	16	17	18	19	20
$S/\binom{n+5}{5}$	0.041	0.070	0.034	0.051	0.028	0.039	0.023	0.030	0.019	0.023
n	21	22	23							
$S/\binom{n+5}{5}$	0.015	0.019	0.013							

Table A.7: Ratio of the number of admissible designs (S) and the total number of possible combinations for different number of slides (n), for parametrization (3.3).

n	1	2	3	4	5	6	7	8	9	10
$S/\binom{n+5}{5}$	0	0	0.036	0.040	0.048	0.045	0.048	0.039	0.033	0.032
n	11	12	13	14	15	16	17	18	19	20
$S/\binom{n+5}{5}$	0.031	0.028	0.025	0.023	0.021	0.019	0.019	0.017	0.016	0.015

Table A.8: Ratio of the number of admissible designs (S) and the total number of possible combinations for different number of slides (n), for parametrization (3.4).

Appendix B

Check optimality program

The following program is written to check optimality for a given set of variables, variance function and experimental region. Note that this program is not designed to find optimal designs, but can check the G-optimality of a given paired comparison design.

First, in this section, we give the algorithm we use. Secondly, in Section B.1, we give the code of this program.

Now we give the algorithm we use to check if a given design is optimal. We also explain some of the steps in the algorithm, and how we implemented these steps. The basic idea behind this algorithm is Lagrange multipliers. There is one difference, instead of making a extra variable, the constraint is substituted into the function that has to be optimized.

Algorithm B.1 (Check-optimality)

1. ***Input:*** *the variables, which have to be between {}, the (variance)function that has to be optimized, which has to be between and the experimental region, which has to be between {} and has to be the same for each variable*

2. *Check the critical points of the function in the interior of the hypercube. So check the points where* $\nabla f = 0$. $i = 1$.

3. ***While*** $i < n$***,*** *check the critical points of the faces* S *of the hypercube with dim* $S = n - i$. $i++$

4. *Check the vertices of the hypercube.*

5. *Filter out the optimal solutions by sorting the maximal functionvalues*

6. ***Output:*** *The optimal solutions on the experimental region with their function values.*

We also give an extensive overview of how the algorithm is implemented.

First we read in the input. Instead of making an extra variable the constraint is substituted into the function that has to be optimized. Then the programm checks critical points of the (variance)function (where $\nabla f = 0$). Then one by one the same check is done for the faces of the hypercube.

All possible faces, and the exterior of the hypercube are generated. This is done by taking all possible subsets of the variables, and mapping them on all possible tuples of the extreme values of the experimental region.

For every possible hypercube region the critical points in this region are calculated.

After this, hidden complex solutions (like $(-1)^{\frac{1}{3}}$) are filtered out. This is done with the function

Cases [N [Sort [allsolutions]], _ ? (FreeQ [#, Complex] &)]

We transform our solutions to the situation where one can obviously see which settings are compared.

Finally the optimal solutions are filtered out, by sorting the maximal function values in the *allsolutionssorted* list.

The optimal solutions on the experimental region are returned with their function values.

B.1 Program code

Off[Solve::svars]
g[{a_, variables_}]:={a, Table[variables[[i]], {i, 1, $\frac{\text{Length[variables]}}{2}$}],
Table[variables[[i]], {i, $\frac{\text{Length[variables]}}{2}+1$, Length[variables]}]};

checkoptimality[variables_:{x, v, y, w}, f_:($\frac{3}{8}(x-y)^2+\frac{3}{8}(v-w)^2+\frac{3}{8}(xv-yw)^2$),
experimentalregion_:{$-1, 1$}]:=

Module[
{possiblesets = Subsets[variables], i, possiblehypercuberegions, j, k, allsolutions,
listofsolutions, n, m, o, allsolutionswithoutcomplex, allsolutionssorted, z,
optimalsolutions},

For[$i = 1$; possiblehypercuberegions = {}, $i \leq$ Length[possiblesets], i++,
possiblehypercuberegions = Join[possiblehypercuberegions,
Table[ReplaceAll[variables, Table[Thread[possiblesets[[i]] →
Tuples[experimentalregion, Length[possiblesets[[i]]]][[j]]],
{j, 1, Length[Tuples[experimentalregion,
Length[possiblesets[[i]]]]}][[k]]],
{k, 1, Length[Tuples[experimentalregion, Length[possiblesets[[i]]]]}]]];

For[$n = 1$; allsolutions = {}, $n \leq$ Length[possiblehypercuberegions], n++,
listofsolutions = Cases[Solve[
Table[
D[ReplaceAll[f, Thread[variables → possiblehypercuberegions[[n]]]],

```
Cases[possiblehypercuberegions[[n]], _Symbol][[m]]]==0,
{m, 1, Length[Cases[possiblehypercuberegions[[n]], _Symbol]]}],
Cases[possiblehypercuberegions[[n]], _Symbol]], _?(FreeQ[#, Complex]&)];

allsolutions =
Union[allsolutions,
Table[{ReplaceAll[ReplaceAll[f,
Thread[variables → possiblehypercuberegions[[n]]]], listofsolutions[[o]]],
ReplaceAll[possiblehypercuberegions[[n]], listofsolutions[[o]]]},
{o, 1, Length[listofsolutions]}]]
];

allsolutionswithoutcomplex =
Cases[N[Sort[allsolutions]], _?(FreeQ[#, Complex]&)];

allsolutionssorted = Map[g, allsolutionswithoutcomplex];

For[z = 1; optimalsolutions = {}, z ≤ Length[allsolutionssorted], z++,
If[allsolutionssorted[[z]][[1]] ≥  allsolutionssorted[[Length[allsolutionssorted]]][[1]],
optimalsolutions =
Union[optimalsolutions, {Sort[allsolutionssorted[[z]]]}], optimalsolutions]];

optimalsolutions
]
```

In Example 2.2 the optimality has to be checked of the design ε. So the input we need for this program was

$$checkoptimality[\{x, v, y, w\}, (\frac{3}{8}(x-y)^2 + \frac{3}{8}(v-w)^2 + \frac{3}{8}(xv-yw)^2), \{-1, 1\}],$$

which gave the output

$$\begin{aligned}\{\{&3., \{-1., -1.\}, \{-1., 1.\}\}, \{3., \{-1., -1.\}, \{1., -1.\}\},\\ &\{3., \{-1., -1.\}, \{1., 1.\}\}, \{3., \{-1., 1.\}, \{1., -1.\}\},\\ &\{3., \{-1., 1.\}, \{1., 1.\}\}, \{3., \{1., -1.\}, \{1., 1.\}\}\},\end{aligned}$$

which are indeed the pairs of ε.

Appendix C

Optimal designs program

In this appendix we give a *Mathematica* program that finds optimal designs for different numbers of slides (n), different parameterizations (for instance both parameterization (3.3) and (3.4)). We designed the program such that different criteria of optimality can be chosen.

In Section C.1 we give a short user manual of the program. In Section C.2 we give the program code of the optimal designs program.

C.1 Program documentation

In the next section we give a brief user manual.

C.1.1 User manual

In this section we give a brief user manual. We discuss the input possibilities, but also how one could expand the program with other criteria.

Input

The program needs three input parameters to run. We give and explain these three parameters, and give some standard settings that can be chosen. The parameters have to be typed as a vector, but we give an example after the explanation.

- The number of slides that may be used. We call this parameter n. The larger this number becomes, the longer the program takes to find the optimal designs. For more information on how long the program takes for different values of n we refer to Section 4.1.

- The vectors that can be chosen in the different design matrices. We used four different sets of vectors, the two parameterizations we discussed in Section 3.3, with and

without the dye-swap possibility. For example, if a dye-swap has to be a possibility and we use parameterization (3.4), the vectors are:

$$\{\{1,0,0,1\},\{-1,0,0,1\},\{0,1,0,1\},\{0,-1,0,1\},\{1,1,1,1\},\{-1,-1,-1,1\},$$
$$\{1,0,1,1\},\{-1,0,-1,1\},\{0,1,1,1\},\{0,-1,-1,1\},\{-1,1,0,1\},\{1,-1,0,1\}\}$$

- The optimality criterion. We implemented five different criteria for optimality. The number represents the input that has to be given.

  1. D-criterion,
  2. A-criterion,
  3. E-criterion,
  4. Admissible3 designs,
  5. Admissible4 designs.

In the next section we give some possibilities for expanding the program.

Expanding the program

In this section we give some possibilities for expanding the program. In the program there is one obvious place where one could expand the program. We already mentioned that it was possible to add different criteria. In the program we used a *Which* command to implement the possibility to use different criteria. In this command, it is easy to add a new criteria.

Other expansions are also possible, but this was the one we wanted to discuss briefly.

In the next section we give the program code of the optimal design program.

C.2 Program code

```
<<DiscreteMath‘Combinatorica‘
Off[General::spell1]

optimaldesigns[n_:6, designvectors_:
{{0, 2, −2}, {−2, 0, −2}, {−2, 2, 0}, {0, 2, 2}, {−2, 0, 2}, {−2, −2, 0}}
, crit_:4]:=

Module[
{lijst, hulplijstje, helelijst, covarmatlijst, i, normalenot, j, a, l, m, p, q, tweedelijst, k,
hulplijst, admisslijst, nonadmisslijst, z, y, admisseindlijst, doptimaal, aoptimaal, eoptimaal},
```

```
lijst = Compositions[n, Length[designvectors]];

hulplijstje = designvectors;

f[x_]:=Flatten[Table[Table[hulplijstje[[p]], {q, 1, x[[p]]}], {p, 1, Length[x]}], 1];

helelijst = Map[f, lijst];

covarfunc[x_]:=
If[Det[Transpose[x].x] ≠ 0, {Inverse[Transpose[x].x], x}, 0];

covarmatlijst = DeleteCases[Map[covarfunc, helelijst], _Integer];

s[x_]:={x[[1]], ReplaceAll[x[[2]], Table[hulplijstje[[j]] → j, {j, 1, Length[hulplijstje]}]]};
normalenot = Map[s, covarmatlijst];

Which[
crit == 1,

g[x_]:={Det[x[[1]]], Table[Count[x[[2]], i], {i, 1, Length[designvectors]}]};

a = Sort[Map[g, normalenot]];

doptimaal = Select[a, #[[1]] ≤ First[a][[1]]&]//MatrixForm

, crit == 2,

g[x_]:={Tr[x[[1]]], Table[Count[x[[2]], i], {i, 1, Length[designvectors]}]};

a = Sort[Map[g, normalenot]];

aoptimaal = Select[a, #[[1]] ≤ First[a][[1]]&]//MatrixForm

, crit == 3,

g[x_]:={N[First[Eigenvalues[x[[1]]]]], Table[Count[x[[2]], i], {i, 1, Length[designvectors]}]};

a = Sort[Map[g, normalenot]];

eoptimaal = Select[a, #[[1]] ≤ First[a][[1]]&]//MatrixForm

, crit == 4,
```

g[x_]:={Table[Count[x[[2]], i], {i, 1, Length[designvectors]}],

Table[x[[1]][[i]][[i]], {i, 1, Length[First[normalenot][[1]]]}]};

a = Sort[Map[g, normalenot]];

eerstelijst = a;

tweedelijst = Union[Table[eerstelijst[[i]][[2]], {i, 1, Length[eerstelijst]}]];

nonadmisslijst = {};

admisscheck[x_]:=

$$\text{Which}[x[[1]] \leq y[[1]] \&\& x[[2]] \leq y[[2]] \&\& x[[3]] \leq y[[3]] \&\& \sum_{k=1}^{\text{Length}[x]} x[[k]] < \sum_{k=1}^{\text{Length}[x]} y[[k]],$$

nonadmisslijst = Union[{y}, nonadmisslijst],

$$x[[1]] \geq y[[1]] \&\& x[[2]] \geq y[[2]] \&\& x[[3]] \geq y[[3]] \&\&$$

$$\sum_{k=1}^{\text{Length}[x]} x[[k]] > \sum_{k=1}^{\text{Length}[x]} y[[k]], \text{nonadmisslijst} = \text{Union}[\{x\}, \text{nonadmisslijst}], 1 == 1, 3];$$

hulplijst = tweedelijst;

For[i = 1, $i \leq$ Length[hulplijst], i++, y = hulplijst[[i]];

Map[admisscheck, hulplijst];

hulplijst = Complement[hulplijst, nonadmisslijst];
]
admisslijst = hulplijst;

admisseindlijst = {};

c[x_]:=If[MemberQ[admisslijst, x[[2]]], admisseindlijst = Union[{x}, admisseindlijst]]

Map[c, eerstelijst];

admisseindlijst //MatrixForm

, crit == 5,

g[x_]:={Table[Count[x[[2]], i], {i, 1, Length[designvectors]}],

Table[x[[1]][[i]][[i]], {i, 1, Length[First[normalenot][[1]]]}]};

a = Sort[Map[g, normalenot]];

eerstelijst = a;

tweedelijst = Union[Table[eerstelijst[[i]][[2]], {i, 1, Length[eerstelijst]}]];

nonadmisslijst = {};

admisscheck[x_]:=

Which[x[[1]] $\leq$ y[[1]]&&x[[2]] $\leq$ y[[2]]&&x[[3]] $\leq$ y[[3]]&&x[[4]] $\leq$ y[[4]]&&

$\sum_{k=1}^{\text{Length}[x]} x[[k]] < \sum_{k=1}^{\text{Length}[x]} y[[k]]$, nonadmisslijst = Union[{y}, nonadmisslijst],

x[[1]] $\geq$ y[[1]]&&x[[2]] $\geq$ y[[2]]&&x[[3]] $\geq$ y[[3]]&&x[[4]] $\geq$ y[[4]]&&

$\sum_{k=1}^{\text{Length}[x]} x[[k]] > \sum_{k=1}^{\text{Length}[x]} y[[k]]$, nonadmisslijst = Union[{x}, nonadmisslijst], 1 == 1, 3];

hulplijst = tweedelijst;

For[i = 1, $i \leq$ Length[hulplijst], i++, y = hulplijst[[i]];

Map[admisscheck, hulplijst];

hulplijst = Complement[hulplijst, nonadmisslijst];
]
admisslijst = hulplijst;

admisseindlijst = {};

c[x_]:=If[MemberQ[admisslijst, x[[2]]], admisseindlijst = Union[{x}, admisseindlijst]]

Map[c, eerstelijst];

admisseindlijst //MatrixForm
]
]

Index

Bibliography

R.A. Adams. *Calculus, a Complete Course*. Addison Wesley, fourth edition, 1999.

V.V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York and London, 1972.

G.F.V. Glonek and P.J. Solomon. Factorial and time course designs for cDNA microarray experiments. Technical report, Department of Applied Mathematics, University of Adelaide, Australia, 2003.

L. Gonick and M. Wheelis. The cartoon guide to genetics. HarperPerennial,revised edition, 1997.

R.V. Hogg, J.W. McKean, and A. Craig. *Introduction to Mathematical Statistics*. Prentice Hall, sixth edition, 2004.

M.K. Kerr and G.A. Churchill. Experimental design for gene expression microarrays. *Biostatistics*, 2:183–201, 2001.

J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Cand. J. Math.*, 12:363–366, 1960.

D.C. Montgomery, E.A. Peck, and G.G. Vining. *Introduction to Linear Regression Analysis*. Wiley, 2001.

F. Pukelsheim. *Optimal Design of Experiments*. Wiley, 1993.

S.D. Silvey. *Optimal Design, an Introduction to the Theory for Parameter Estimation*. Chapman and Hall Ltd, New York and London, 1980.

E.E.M. Van Berkum. *Optimal paired comparison designs for factorial experiments*. PhD thesis, Technische Hogeschool Eindhoven, 1985.

E.E.M. Van Berkum. Regressie en variantie analyse. Lecture Notes, 2003.

E.E.M. Van Berkum. Theorie van optimaal proefopzetten. Lecture Notes, 2004.

Y.H. Yang and T. Speed. Design issues for cDNA microarray experiments. *Nature Genetics Reviews*, 3:579–588, August 2002.