

Theory and Algorithms for Finding Optimal Regression Designs

by

Yue Yin

B.Sc., Beijing Institute of Technology, 2011

M.Sc., University of Victoria, 2013

A Dissertation Submitted in Partial Fulfillment of the  
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mathematics and Statistics

© Yue Yin, 2017

University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

Theory and Algorithms for Finding Optimal Regression Designs

by

Yue Yin

B.Sc., Beijing Institute of Technology, 2011

M.Sc., University of Victoria, 2013

Supervisory Committee

---

Dr. Julie Zhou., Supervisor  
(Department of Mathematics and Statistics)

---

Dr. Farouk Nathoo., Departmental Member  
(Department of Mathematics and Statistics)

---

Dr. Boualem Khouider, Departmental Member  
(Department of Mathematics and Statistics)

---

Dr. Stan Dosso, Outside Member  
(School of Earth and Ocean Sciences)

## Supervisory Committee

---

Dr. Julie Zhou., Supervisor  
(Department of Mathematics and Statistics)

---

Dr. Farouk Nathoo., Departmental Member  
(Department of Mathematics and Statistics)

---

Dr. Boualem Khouider, Departmental Member  
(Department of Mathematics and Statistics)

---

Dr. Stan Dosso, Outside Member  
(School of Earth and Ocean Sciences)

---

## ABSTRACT

In this dissertation, we investigate various theoretical properties of optimal regression designs and develop several numerical algorithms for computing them. The results can be applied to linear, nonlinear and generalized linear models.

Our work starts from how to solve the design problems for A-,  $A_s$ -, c-, I- and L-optimality criteria on one-response model. Theoretical results are hard to derive for many regression models and criteria, and existing numerical algorithms can not compute the results efficiently when the number of support points is large. Therefore we consider to solve the design problems based on SeDuMi program in MATLAB. SeDuMi is developed to solve semidefinite programming (SDP) problems in optimization. To apply it, we derive a general transformation to connect the design problems with SDP problems, and propose a numerical algorithm based on SeDuMi to solve these SDP problems. The algorithm is quite general under the least squares estimator (LSE) and weighted least squares estimator (WLSE) and can be applied to both linear and nonlinear regression models.

We continue to study the optimal designs based on one-response model when the error distribution is asymmetric. Since the second-order least squares estimator (SLSE) is more efficient than the LSE when the error distribution is not symmetric, we study optimal designs under the SLSE. We derive expressions to characterize A- and D-optimality criteria and develop a numerical algorithm for finding optimal designs under the SLSE based on SeDuMi and CVX programs in MATLAB. Several theoretical properties are also derived for optimal designs under SLSE. To check the optimality of the numerical results, we establish the Kiefer-Wolfowitz equivalence theorem and apply it to various applications.

Finally, we discuss the optimal design problems for multi-response models. Our algorithm studied here is based on SeDuMi and CVX, and it can be used for linear, nonlinear and generalized linear models. The transformation invariance property and dependence on the covariance matrix of the correlated errors are derived. We also correct the errors in the literature caused by formulation issues.

The results are very useful to construct optimal regression designs on discrete design space. They can be applied to any one-response and multi-response models, various optimality criteria, and several estimators including LSE, maximum likelihood estimator, best linear unbiased estimator, SLSE and WLSE.

# Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	v
List of Tables	viii
List of Figures	x
List of Abbreviations	xi
Acknowledgements	xii
Dedication	xiii
<b>1 Introduction</b>	<b>1</b>
<b>2 Using SeDuMi to compute various optimal designs for regression models</b>	<b>5</b>
2.1 Introduction . . . . .	6
2.2 General results and algorithm . . . . .	7
2.2.1 SDP and SeDuMi . . . . .	9
2.2.2 Computing optimal designs via SeDuMi . . . . .	11
2.2.3 Kiefer-Wolfowitz equivalence theorem . . . . .	16
2.3 Optimal design problems and transformations . . . . .	17
2.3.1 A- and $A_s$ -optimal designs . . . . .	17
2.3.2 c-optimal design . . . . .	18
2.3.3 I-optimal design . . . . .	18
2.3.4 L-optimal design . . . . .	19

2.4	Applications . . . . .	19
2.5	Other optimal design problems . . . . .	24
2.5.1	Optimal design for unequal error variances . . . . .	24
2.5.2	Optimal design with linear constraints on weights . . . . .	25
2.6	Discussion . . . . .	26
<b>3</b>	<b>Optimal designs using the second-order least squares estimator</b>	<b>28</b>
3.1	Introduction . . . . .	29
3.2	A- and D-optimality criteria . . . . .	30
3.3	Properties of $\xi^A(\mathbf{x})$ . . . . .	32
3.4	Numerical algorithms . . . . .	36
3.4.1	Multiplicative algorithms . . . . .	36
3.4.2	Convex optimization algorithms . . . . .	38
3.5	Applications and efficiencies . . . . .	41
3.5.1	Examples . . . . .	41
3.5.2	A-efficiency and D-efficiency . . . . .	47
3.6	Discussion . . . . .	48
<b>4</b>	<b>Optimal designs for multi-response nonlinear regression models with several factors via semi-definite programming</b>	<b>49</b>
4.1	Introduction . . . . .	50
4.2	Optimality criteria for multi-response models . . . . .	51
4.3	Properties of optimal designs . . . . .	54
4.4	SDP for optimal design problems . . . . .	60
4.4.1	A-optimality . . . . .	63
4.4.2	$A_s$ -optimality . . . . .	64
4.4.3	c-optimality . . . . .	64
4.5	Applications . . . . .	65
4.6	Multi-response generalized linear models . . . . .	71
4.7	Conclusion . . . . .	77
<b>5</b>	<b>Discussions</b>	<b>78</b>
	<b>Appendix: MATLAB Codes</b>	<b>82</b>
	<b>My Publications</b>	<b>94</b>

**Bibliography**

# List of Tables

Table 2.1	The I-optimal design for Example 2.4 . . . . .	21
Table 2.2	I-optimal designs for the nonlinear model in Example 2.5 . . . . .	23
Table 2.3	L-optimal designs for the quadratic model in Example 2.6 . . . . .	23
Table 3.1	A- and D-optimal weights, $w_1^A, w_5^A, w_9^A, w_1^D, w_5^D, w_9^D$ , in Example 3.3 . . . . .	42
Table 3.2	A- and D-optimal design points and their weights (in parentheses) for the Michaelis-Menten model with $\alpha = 1$ and $\beta = 1$ . . . . .	45
Table 3.3	D-optimal design support points and their weights (in parentheses) for compartmental model (3.18) with $\alpha_1 = \alpha_2 = 1, \beta_1 = 0.1, \beta_2 = 1.0$ , and $S_N = \{50(i - 1)/(N - 1) : i = 1, \dots, N\}$ . . . . .	46
Table 3.4	A-optimal and D-optimal design points and their weights (in parentheses) for the Emax model with various values of $N$ and $t$ . . . . .	47
Table 3.5	A-efficiency and D-efficiency for the optimal designs in Examples 3.3 and 3.4. . . . .	48
Table 4.1	A-optimal designs in Example 4.4 . . . . .	66
Table 4.2	A- and D-optimal designs in Example 4.5 for various values of $\rho$ . . . . .	67
Table 4.3	A-optimal weights computed from SDP in (4.21) and Atashgah and Seifi (2009) (denoted by ATSE) with $\text{trace}(\mathbf{A}^{-1}(\mathbf{w})) = 17.546$ and 18.012, respectively. . . . .	67
Table 4.4	Locally $D$ -optimal designs for the bivariate Emax model (Example 4.6) when the nominal values are $ED_{50}^* = 1, E_{\max}^* = S_{\max}^* = 1$ and selected values of $SD_{50}^*$ . The support points are $x_1, x_2, x_3$ and 500 with corresponding weights $w_1, w_2, w_3$ and $1 - w_1 - w_2 - w_3$ . If a point has zero weight, then it is not specified in the table. . . . .	69

Table 4.5	Locally $c$ -optimal designs for the bivariate Emax model (Example 4.6) when the nominal values for the model parameters are $ED_{50}^* = 1$ , $E_{\max}^* = S_{\max}^* = 1$ , $k_1 = k_2 = 1$ and selected values of $SD_{50}^*$ . The support points of the optimal designs are $x_1, x_2, x_3$ and 500 with corresponding weights $w_1, w_2, w_3$ and $1 - w_1 - w_2 - w_3$ .	70
Table 4.6	D-optimal designs for the bivariate probit response model . . .	73
Table 4.7	D-efficiency for Case (i) . . . . .	74
Table 4.8	D-optimal designs for different values of $\rho$ for the GLM . . . . .	76

# List of Figures

Figure 1.1 Emax model on a given design space . . . . .	2
Figure 1.2 A-optimal design for Emax model when $\theta = (1, 1, 2)^\top$ . . . . .	3
Figure 3.1 Optimal design results for Example 3.3: (a) A-optimal design on design space $S_{9,1}$ for $t = 0.3$ , (b) A-optimal design on design space $S_{9,1}$ for $t = 0.9$ , (c) D-optimal design on design space $S_{9,2}$ for $t = 0.3$ , (d) D-optimal design on design space $S_{9,2}$ for $t = 0.9$ ,	42
Figure 4.1 The convex sets (shaded areas) and solutions to the: (a) SDP problem in (4.16), (b) SDP problem in (4.17). . . . .	61
Figure 5.1 Various optimal designs studied in this dissertation . . . . .	79

# List of Abbreviations

<b>BLUE</b> .....	Best linear unbiased estimator
<b>CUI</b> .....	Composite utility index
<b>CVX</b> .....	Matlab-based modelling system for convex optimization
<b>GLMs</b> .....	Generalized linear models
<b>LSE</b> .....	Least squares estimator
<b>MLE</b> .....	Maximum likelihood estimator
<b>PSD</b> .....	Positive semi-definite
<b>SeDuMi</b> .....	Self-dual minimization
<b>SLSE</b> .....	Second order least squares estimator
<b>SDP</b> .....	Semidefinite programming
<b>WLSE</b> .....	Weighted least squares estimator

## ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Professor Julie Zhou for her support of my research and study at University of Victoria in the past years, for her patience, inspiration and technical advice. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my supervisor, I would like to thank Professors Stan Dosso, Boualem Khouider and Farouk Nathoo for serving on my thesis committee and for their valuable comments, insights and guidance.

I would also like to thank my significant other, Yi Chen, for his encouragement and accompany along the time. Last and certainly not least, I would like to thank my parents, for their endless love and support throughout my life.

*Yue Yin, Victoria, BC, Canada*

DEDICATION

To my beloved parents,  
My Late grandfather,  
And all my friends,  
For your love, endless support, and encouragement.

# Chapter 1

## Introduction

Optimal design problems have been applied to various regression models, including linear, nonlinear and generalized linear models (GLMs). See, for example, Fedorov (1972), Pukelsheim (1993), Berger and Wong (2009), and Dean et al. (2015). Optimal design problem becomes a more and more important issue in real industry applications, which are across broad disciplines, including agriculture, engineering, chemical, mechanical and pharmaceutical industry. The general form of the regression model considered in this dissertation can be written as

$$y_{ij} = \boldsymbol{\eta}_i(\mathbf{x}_j; \boldsymbol{\theta}_i) + \epsilon_{ij}, \quad j = 1, \dots, n, \quad i = 1, \dots, r, \quad (1.1)$$

where  $y_{ij}$  is the  $j$ th observation on response variable  $y_i$ ,  $\mathbf{x}$  denotes the vector of  $p$  independent variables,  $\boldsymbol{\eta}_i(\mathbf{x}; \boldsymbol{\theta}_i)$  is the  $i$ th known linear or nonlinear function of  $\boldsymbol{\theta}_i$ ,  $\boldsymbol{\theta}_i$  is the unknown vector of regression parameters for  $y_i$ , and  $\epsilon_{ij}$  are random errors with mean zero. Model (1.1) is a one-response model if  $r = 1$ , otherwise it is a multi-response model.

A design problem is to choose the “optimal” design points  $\mathbf{x}_1, \dots, \mathbf{x}_n$  from a user-selected design space and run the experiments such that we can get the most accurate estimation for unknown parameters. For example, Berger and Wong (2009) considered optimal designs for an Emax model with one design variable,

$$y_j = \frac{\alpha x_j^{\beta_2}}{\beta_1 + x_j^{\beta_2}} + \epsilon_j, \quad j = 1, \dots, n, \quad \boldsymbol{\theta} = (\alpha, \beta_1, \beta_2)^\top. \quad (1.2)$$

Figure 1.1 shows the response function  $\eta(\mathbf{x}_j; \boldsymbol{\theta}) = \frac{\alpha x_j^{\beta_2}}{\beta_1 + x_j^{\beta_2}}$  on a given design space with  $\boldsymbol{\theta} = (1, 1, 2)^\top$ , and  $x_1, \dots, x_n$  are design points.

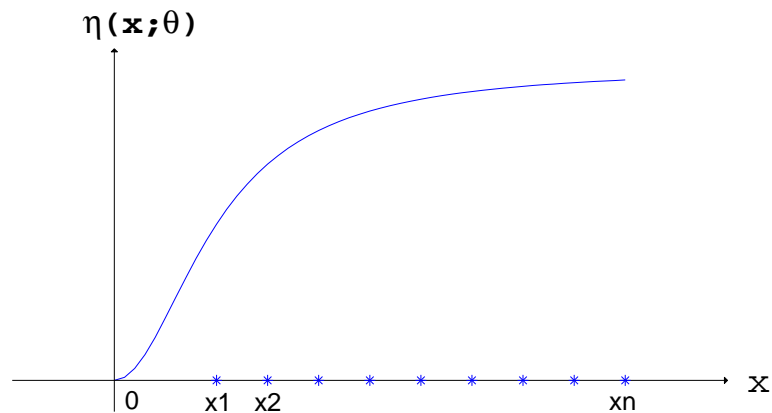


Figure 1.1: Emax model on a given design space

The design problem is to choose “optimal” design points according to some optimality criterion. It is obvious that all the design points can be chosen more than once. One example of optimal designs is shown in Figure 1.2 when the design space is the interval  $[0, 5]$ . From the figure, we know that there are three design points chosen for A-optimal design,  $x_1 = 0.451$ ,  $x_2 = 1.426$  and  $x_3 = 5.000$ , and  $w_1$ ,  $w_2$  and  $w_3$  are the weights (proportions of times) for these three design points selected respectively. More details can be found in Chapter 3.

For some simple regression models, it is easy to obtain the theoretical results for optimal designs. There are challenges to derive the theoretical results for many regression models and design criteria, which motivated researchers to study the optimal designs by using numerical algorithms. However, there are also issues with existing numerical algorithms. Some algorithms either take long time to converge or only work with small number of parameters and design variables.

In this dissertation, we study different types of optimal designs, their properties, and numerical algorithms. In Chapter 2, we derive an efficient algorithm based on

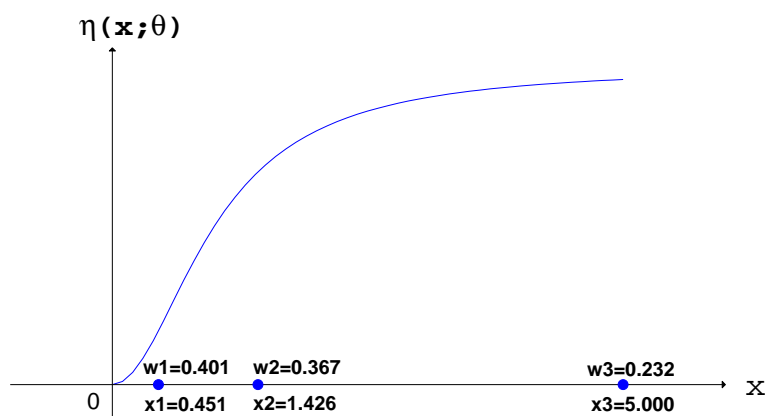


Figure 1.2: A-optimal design for Emax model when  $\theta = (1, 1, 2)^\top$

SeDuMi (self-dual minimization) program of MATLAB to compute A-,  $A_s$ -, c-, I- and L-optimal designs using the least squares estimator (LSE) for one-response model. SeDuMi (Sturm, 1999) is an effective and efficient algorithm to solve semi-definite programming (SDP) problems, which is a class of convex optimization problems (Boyd and Vandenberghe, 2004). In Chapter 3, we discuss the case when the error distribution is not symmetric, and we study the optimal designs based on the second-order least squares estimator (SLSE) for one-response model. The SLSE was developed in Wang and Leblanc (2008), which is a more efficient estimator than the LSE when the error distribution is asymmetric. In Chapter 4, we focus on the multi-response models. Generalized linear models are also discussed in this chapter. In all cases, regression models can be linear or nonlinear, and our algorithms work well. Many theoretical properties of optimal designs are also derived.

The main contributions of this dissertation can be summarized as follows.

1. An efficient algorithm based on SeDuMi is developed to find optimal designs for various criteria.
2. Optimal designs based on the SLSE are investigated. Several theoretical properties are obtained, and an efficient algorithm based on CVX (Boyd and Vandenberghe, 2004) is developed.
3. Optimal design problems for multi-response models are studied. Theoretical properties and efficient algorithms are developed. We also correct some errors in Atashgah and Seifi (2009).

It should be noted that the main result of Chapter 2 has been accepted for publication in *Statistical Papers*, which is a joint paper with Professors Weng Kee Wong and Julie Zhou; see Wong et al. (2017). The main result of Chapter 3 has been accepted for publication in *Statistica Sinica*; see Yin and Zhou (2016). The main result of Chapter 4 has been submitted for publication; see Wong et al. (2016).

## Chapter 2

# Using SeDuMi to compute various optimal designs for regression models

In this chapter, we focus on one-response linear and nonlinear regression models. To find optimal designs defined on a discrete design space, we introduce a powerful optimization tool called SeDuMi (Strum, 1999), which is widely used for solving SDP problems (Boyd and Vandenberghe, 2004). We first formulate A-,  $A_s$ -, c-, I- and L-optimal design problems as SDP problems, and then use SeDuMi in MATLAB to solve them effectively. The optimality of all the SDP-generated designs will be verified using the well known Kiefer-Wolfowitz equivalence theorem (Pukelsheim, 1993).

Chapter 2 is organized as follows. In Section 2.1, we briefly introduce optimal design problems, and give a review of SeDuMi and other algorithms for computing optimal designs. In Section 2.2, we derive a general result to transform optimal design problems into SDP problems and develop a general algorithm for solving optimal design problems via SeDuMi. In Section 2.3, we give details to transform A-,  $A_s$ -, c-, I- and L-optimal design problems into SDP problems. In Section 2.4, we present and compare several applications of optimal designs. In Section 2.5, we show the broad applicability of SDP by finding optimal designs based on the weighted least squares estimator (WLSE) and optimal designs with user-imposed weight constraints. In Section 2.6, we give a conclusion for this chapter. The main result of this chapter has been written in a research paper, Wong et al. (2017), which has been accepted

for publication in *Statistical Papers*.

## 2.1 Introduction

SDP problems form a special class of convex optimization problems. They have a linear objective function with a linear positive semi-definite (PSD) matrix as constraint. As SDP has desirable theoretical properties and computational efficiencies (Boyd and Vandenberghe, 2004) there are increasing applications of SDP to various areas of research, such as traditional convex constrained optimization, control theory, and combinatorial optimization. It is well known that SDP can solve optimization problems with hundreds variables efficiently (Papp, 2012). Therefore we are curious if we can formulate optimal design problems as SDP problems so that we can benefit from the good properties of SDP and efficiently generate various types of optimal designs. Boyd and Vandenberghe (2004) discussed A-, D- and E- optimal design problems as SDP problems briefly. Ye et al. (2015) presented details for A-optimal and E-optimal design problems as SDP problems. Recent work using SDP to find optimal designs includes Duarte and Wong (2015) and Duarte et al (2016). We will derive a general result for transforming design problems into SDP problems and develop a detailed general algorithm for finding optimal designs.

Our setup is that we have a given model with unknown parameters and a user-selected design space  $S$ . We use a discrete design space consisting of finite potential design points. These design points are typically uniformly spaced across the design space. The design problems are to find the weights for all the design points based on specific design criteria so that we can get the most precise inference for the unknown parameters. Those points that receive a positive weight under the design criterion are the support points of the optimal design.

Optimal designs can be constructed analytically for some regression models and optimality criteria. In general, it is very hard to solve optimal design problems analytically. Various numerical methods have been proposed for finding optimal designs. For example, multiplicative algorithm (Silvey et al., 1978; Yu, 2010; Bose and Mukerjee, 2015), semi-infinite programming algorithm (Duarte and Wong, 2014; Duarte et al., 2015), coordinate exchange algorithm (Meyer and Nachtsheim, 1995; Cuervo et al., 2016), interior point method (Lu and Pong, 2013) and CVX program in MATLAB

(Papp, 2012; Gao and Zhou, 2015). Heuristic algorithms have also been studied in the literature. These algorithms are nature-inspired and popular in computer science to solve complex optimization problems, such as genetic algorithm (Broudiscou et al., 1996; Montepiedra et al., 1998; Herrmann 1999; Hamada et al., 2001; Drain et al., 2004; Karami and Wiens, 2014), simulated annealing algorithm (Kirkpatrick et al., 1983; Meyer and Nachtsheim, 1988; Fang and Wiens, 2000; Givens and Hoeting, 2005; Wilmut and Zhou, 2011; Rempel and Zhou, 2014; Yin and Zhou, 2015) and particle swarm optimization (Qiu et al., 2014; Wong et al., 2015; Chen et al., 2015).

In this chapter, we introduce another efficient algorithm, SeDuMi. SeDuMi (Sturm, 1999) is a powerful algorithm to solve SDP problems in MATLAB. We need to firstly convert the optimal design problem into a SDP problem to apply SeDuMi. We have derived innovative ways to transform various types of optimal design problems into SDP problems. In particular, we provide a general algorithm that uses SeDuMi to solve optimal design problems. SeDuMi is both flexible and efficient for finding A-,  $A_s$ , c-, E-, I-, L-optimal designs for any linear or nonlinear regression model using the LSE or WLSE. These criteria are commonly used in practice, see for example, A-optimal designs in Hardin and Sloane (1993) and Gao and Zhou (2014),  $A_s$ -optimal designs in Berger and Wong (2009), c-optimal designs in Han and Chaloner (2003) and Dette et al. (2004), E-optimal designs in Pukelsheim and Studden (1993) and Imhof and Studden (2001), I-optimal designs in Dette and O'Brien (1999) and Gianchandani and Crary (1998), and L-optimal designs in He et al. (1996).

## 2.2 General results and algorithm

Suppose our interests are in studying the relationship between a response variable  $y$  and the vector of independent variables  $\mathbf{x}$ . Assume that if we can run  $n$  times of experiments, and the model is

$$y_i = g(\mathbf{x}_i; \boldsymbol{\theta}) + \epsilon_i, \quad i = 1, \dots, n, \quad (2.1)$$

where  $y_i$  is the observation at  $\mathbf{x}_i$ ,  $\boldsymbol{\theta} \in R^q$  is an unknown regression parameter vector, and  $g(\mathbf{x}; \boldsymbol{\theta})$  is a known linear or nonlinear function of  $\boldsymbol{\theta}$ . The errors  $\epsilon_i$ 's are independent and identically distributed each with mean 0 and constant variance  $\sigma^2$ .

A commonly used technique of estimating the unknown parameter vector is the least squares estimation. Based on  $n$  observations, the LSE of  $\boldsymbol{\theta}$  is defined as

$$\hat{\boldsymbol{\theta}} = \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - g(\mathbf{x}_i; \boldsymbol{\theta}))^2.$$

The covariance matrix of  $\hat{\boldsymbol{\theta}}$  is

$$\operatorname{Cov}(\hat{\boldsymbol{\theta}}) = \frac{\sigma^2}{n} \mathbf{A}^{-1},$$

where  $\mathbf{A}^{-1}$  denotes the inverse of a matrix  $\mathbf{A}$  and

$$\mathbf{A} = \frac{1}{n} \sum_{i=1}^n \frac{\partial g(\mathbf{x}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}} \frac{\partial g(\mathbf{x}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}^\top} \quad (2.2)$$

is the information matrix and  $\boldsymbol{\theta}^*$  is the true but unknown parameter value of  $\boldsymbol{\theta}$ . For linear models, the matrix  $\mathbf{A}$  does not depend on  $\boldsymbol{\theta}^*$  and the covariance matrix of  $\hat{\boldsymbol{\theta}}$  is exact. However, for nonlinear models, the matrix  $\mathbf{A}$  depends on  $\boldsymbol{\theta}^*$  and the covariance matrix is an approximation from its asymptotic distribution (Seber and Wild, 1989, p.24).

In practice, for a given number of observations, say  $n$ , we want to choose  $n$  settings of variable vectors to run  $n$  times of experiments so that we can get the most accurate estimation for the parameters. Optimal design problems aim to select the best choices of settings  $\mathbf{x}_1, \dots, \mathbf{x}_n$  from the design space  $S \subset R^p$  based on specific design criterion to perform experiments and obtain observations  $y_1, \dots, y_n$ . Various optimality criteria have been investigated in the literature, and most of them are based on the covariance matrix of  $\hat{\boldsymbol{\theta}}$ . Since the covariance matrix is not a scalar function, we often consider scalar functions such as trace, determinant, largest eigenvalue and so forth, which will give us A-, D- and E-optimal design criteria. It is clear that optimal designs depend on the optimality criteria and model assumptions; see, for example, Fedorov (1972), Pukelsheim (1993), Dette and Studden (1997), and Berger and Wong (2009).

### 2.2.1 SDP and SeDuMi

SeDuMi is a MATLAB toolbox for Optimization. It implements the primal-dual interior-point methods for SDP problems. Primal-dual interior-point methods are very similar to the barrier method, but often more efficient than the barrier method. For a primal-dual interior-point method, the search directions are obtained from Newtons method. Both the primal and dual variables are updated at each iteration, and the duality gap will be calculated as well. The program will stop until the duality gap smaller than the tolerance error.

Before we discuss SDP problems and show how SeDuMi program in MATLAB can solve them efficiently, we introduce convex optimization problems first. Convex optimization problems have convex objective functions, with or without constraints. Such optimization problems arise in many research fields, such as business, economics, engineering and statistics. Boyd and Vandenberghe (2004) provides many examples, along with research problems in this area.

SDP problems form a special class of convex optimization problems, which have a linear objective function and linear matrix constraints. A general form of SDP problems is given as follows:

$$\begin{cases} \min_{\mathbf{v}} \mathbf{a}^\top \mathbf{v} \\ \text{subject to: } \mathbf{H}_0 + v_1 \mathbf{H}_1 + \cdots + v_m \mathbf{H}_m \succeq 0, \end{cases} \quad (2.3)$$

where  $\mathbf{v} = (v_1, \cdots, v_m)^\top$  is the vector to optimize,  $\mathbf{a}$  is a constant vector,  $\mathbf{H}_i$ 's are  $k \times k$  constant matrices,  $i = 0, 1, \cdots, m$ , and the notation " $\succeq 0$ " means that the matrix to its left is PSD. The vector  $\mathbf{a}$  and the constant matrices  $\mathbf{H}_i, i = 0, \dots, m$  are specific to the optimization problem of interest.

We use Example 2.1 to show how to use SeDuMi program in MATLAB to solve a specific SDP problem. In next section we will derive results that link different optimal design problems with the generic SDP problem in (2.3) and develop a general algorithm using SeDuMi to solve optimal design problems.

**Example 2.1.** Consider a SDP problem defined by

$$\begin{cases} \min_{\mathbf{v}} \mathbf{a}^\top \mathbf{v}, \\ \text{subject to: } \mathbf{H}_0 + v_1 \mathbf{H}_1 + v_2 \mathbf{H}_2 + v_3 \mathbf{H}_3 \succeq 0, \end{cases}$$

where  $\mathbf{a} = (0, 0, -1)^\top$ ,  $\mathbf{v} = (v_1, v_2, v_3)^\top$ , and

$$\mathbf{H}_0 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{H}_1 = \begin{pmatrix} 0 & -2 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix},$$

$$\mathbf{H}_2 = \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \quad \mathbf{H}_3 = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The MATLAB code listed below uses SeDuMi to solve the problem. The minimizer obtained from the algorithm is  $\mathbf{v}^* = (0.5, 0, 1)^\top$ .

### MATLAB program for Example 2.1

---

```
H0=[1 1 0 0 0; 1 1 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 0 0 0 0 1];
H1=[0 -2 0 0 0; -2 0 0 0 0; 0 0 1 0 0; 0 0 0 0 0; 0 0 0 0 -1];
H2=[0 -1 0 0 0; -1 -1 0 0 0; 0 0 0 0 0; 0 0 0 1 0; 0 0 0 0 -1];
H3=[-1 0 0 0 0; 0 -1 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 0 0 0 0 0];
bt=[0 0 -1];
ct=vec(H0);
At=-[vec(H1) vec(H2) vec(H3)];
K.s=size(H0,1);
[u,v,info]=sedumi(At,bt,ct,K);
v
```

---

□

### 2.2.2 Computing optimal designs via SeDuMi

Optimal design problems for regression models are often convex optimization problems. Here we first show that many optimal design problems can be transformed into SDP problems as in (2.3) before we apply the SeDuMi algorithm to find various optimal designs.

Suppose the discrete design space  $S$  consists of  $N$  points denoted by  $S_N = \{\mathbf{u}_1, \dots, \mathbf{u}_N\} \subset R^p$ . These points  $\mathbf{u}_1, \dots, \mathbf{u}_N$  represent candidate design points of the optimal design from  $S$ . The distinguishing feature of an approximate design is that only the proportion  $w_i$  of the total observations to be taken at support point  $\mathbf{u}_i$  has to be determined, and not the number of observations at each of the candidate points. We denote such an approximate design by

$$\xi = \begin{pmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_N \\ w_1 & \cdots & w_N \end{pmatrix},$$

where  $w_1, \dots, w_N$  are weights at the points  $\mathbf{u}_1, \dots, \mathbf{u}_N$ , respectively, and they satisfy

$$w_i \geq 0, \quad i = 1, \dots, N, \quad \text{and} \quad \sum_{i=1}^N w_i = 1. \quad (2.4)$$

We need to find “optimal” weights according to various optimal design criteria. Points with positive weights after the optimization become the support points of the optimal design. If  $n$  observations are to be taken for the experiment, an approximate design is implemented by taking  $nw_i$  observations at each of its support point  $\mathbf{u}_i$  subject to each  $nw_i$  is rounded to an integer and they sum to  $n$ . For approximate designs, the information matrix  $\mathbf{A}$  in (2.2) becomes

$$\mathbf{A}(\mathbf{w}) = \sum_{i=1}^N w_i \frac{\partial g(\mathbf{u}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}} \frac{\partial g(\mathbf{u}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}^\top}, \quad (2.5)$$

where  $\mathbf{w} = (w_1, \dots, w_N)^\top$  is a weight vector.

We notice that the covariance matrix of  $\hat{\boldsymbol{\theta}}$  is proportional to  $\mathbf{A}^{-1}$ . Since the matrix  $\mathbf{A}$  is not a scalar function, we consider some forms of measures of the information matrix to construct optimal designs.

Let  $\phi(\mathbf{w}) = \text{trace}(\mathbf{T}(\mathbf{A}(\mathbf{w}))^{-1}\mathbf{T}^\top)$ , where  $\mathbf{T}$  is a  $r \times q$  constant matrix with  $\text{rank}(\mathbf{T}) = r \leq q$ . For singular  $\mathbf{A}(\mathbf{w})$ , we define  $\phi(\mathbf{w})$  to be  $\infty$ . Accordingly, in what is to follow, we are concerned with non-singular information matrices only. A class of approximate optimal design problems can be stated as

$$\begin{cases} \min_{\mathbf{w}} \phi(\mathbf{w}) \\ \text{subject to: } w_i \geq 0, i = 1, \dots, N, \text{ and } \sum_{i=1}^N w_i = 1. \end{cases} \quad (2.6)$$

This class of design problems includes many commonly used optimality criteria, and examples will be given in Sections 2.3 and 2.4. For nonlinear regression models, the optimal designs usually depend on the true parameter values  $\boldsymbol{\theta}^*$ , and they are called locally optimal designs. For simplicity, we just use the term optimal design instead of locally optimal design in this dissertation.

Let  $\mathbf{W}_N = \text{diag}(w_1, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i)$  be an  $N \times N$  diagonal matrix and let  $\mathbf{e}_i$  be the  $i$ th unit vector in  $R^q$ ,  $i = 1, \dots, q$ . The following theorem gives a general result for transforming the design problem in (2.6) into a SDP problem in (2.3).

**Theorem 2.1.** *For the design problem in (2.6), there exists a  $(q - r) \times q$  matrix  $\mathbf{U}$  such that the matrix*

$$\mathbf{D} = \begin{pmatrix} \mathbf{T} \\ \mathbf{U} \end{pmatrix}_{q \times q} \quad (2.7)$$

has rank  $q$ , and the following two steps transform (2.6) into a SDP problem:

(i) *The design problem in (2.6) is equivalent to the problem*

$$\begin{cases} \min_{\mathbf{w}} \text{trace}(\mathbf{T}_r \mathbf{D}(\mathbf{A}(\mathbf{w}))^{-1} \mathbf{D}^\top \mathbf{T}_r^\top) \\ \text{subject to: } w_i \geq 0, i = 1, \dots, N, \text{ and } \sum_{i=1}^N w_i = 1, \end{cases} \quad (2.8)$$

where  $\mathbf{T}_r = (\mathbf{I}_r, \mathbf{0})$  is a  $r \times q$  matrix,  $\mathbf{I}_r$  is the  $r \times r$  identity matrix, and  $\mathbf{0}$  is an  $r \times (q - r)$  matrix of zeros.

(ii) Let  $\mathbf{B}(\mathbf{w}) = \mathbf{D}^{-\top} \mathbf{A}(\mathbf{w}) \mathbf{D}^{-1}$ , let  $\mathbf{v} = (w_1, \dots, w_{N-1}, v_N, \dots, v_{N+r-1})^\top$  and let

$$\mathbf{B}_i = \begin{pmatrix} \mathbf{B}(\mathbf{w}) & \mathbf{e}_i \\ \mathbf{e}_i^\top & v_{N+i-1} \end{pmatrix}, \quad (2.9)$$

$i = 1, \dots, r$ . The design problem in (2.8) can be transformed into a SDP problem as follows:

$$\begin{cases} \min_{\mathbf{v}} v_N + \dots + v_{N+r-1} \\ \text{subject to: } \mathbf{B}_1 \oplus \dots \oplus \mathbf{B}_r \oplus \mathbf{W}_N \succeq 0, \end{cases} \quad (2.10)$$

where the notation “ $\oplus$ ” denotes matrix direct sum. This implies that if  $\mathbf{v}^* = (w_1^*, \dots, w_{N-1}^*, v_N^*, \dots, v_{N+r-1}^*)^\top$  is a solution to problem (2.10) then  $\mathbf{w}^* = (w_1^*, \dots, w_{N-1}^*, 1 - \sum_{i=1}^{N-1} w_i^*)^\top$  is a solution to problem (2.8).

**Proof of Theorem 2.1:**

(i) From (2.7) and  $\mathbf{T}_r = (\mathbf{I}_r, \mathbf{0})$ , it is clear that  $\mathbf{T}_r \mathbf{D} = \mathbf{T}$ . Thus, problem (2.8) is the same as problem (2.6).

(ii) We first show that problem (2.10) is a SDP problem. By (2.5), all the elements of  $\mathbf{A}(\mathbf{w})$  are linear functions of weights  $w_1, \dots, w_{N-1}$ , so are the elements of  $\mathbf{B}(\mathbf{w})$ . From (2.9) and  $\mathbf{W}_N$ , all the elements of  $\mathbf{B}_1, \dots, \mathbf{B}_r$  and  $\mathbf{W}_N$  are linear functions of  $\mathbf{v} = (w_1, \dots, w_{N-1}, v_N, \dots, v_{N+r-1})^\top$ , so the constraint in (2.10) is a linear matrix constraint. It is obvious that the objective function in (2.10) is a linear function of  $\mathbf{v}$ . Thus, problem (2.10) is a SDP problem.

Now we show that a solution to problem (2.10) provides a solution to problem (2.8). Since  $\mathbf{B}(\mathbf{w}) = \mathbf{D}^{-\top} \mathbf{A}(\mathbf{w}) \mathbf{D}^{-1}$ , it is easy to verify that  $\mathbf{T}_r \mathbf{D} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{D}^\top \mathbf{T}_r^\top = \mathbf{T}_r (\mathbf{B}(\mathbf{w}))^{-1} \mathbf{T}_r^\top$ . Let  $b_{ii}$  ( $i = 1, \dots, q$ ) be the diagonal elements of  $(\mathbf{B}(\mathbf{w}))^{-1}$ . Then we have

$$\text{trace} (\mathbf{T}_r \mathbf{D} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{D}^\top \mathbf{T}_r^\top) = \text{trace} (\mathbf{T}_r (\mathbf{B}(\mathbf{w}))^{-1} \mathbf{T}_r^\top) = \sum_{i=1}^r b_{ii}. \quad (2.11)$$

The constraints in (2.8) is equivalent to have  $\mathbf{W}_N \succeq 0$ . Thus, problem (2.8) is to minimize  $\sum_{i=1}^r b_{ii}$  over the design weights subject to  $\mathbf{W}_N \succeq 0$ .

By (2.9) and  $\mathbf{B}_i \succeq 0$ , we get

$$v_{N+i-1} \geq \mathbf{e}_i^\top \mathbf{B}(\mathbf{w})^{-1} \mathbf{e}_i = b_{ii}, \quad i = 1, \dots, r. \quad (2.12)$$

Since we minimize  $\sum_{i=1}^r v_{N+i-1}$  in (2.10), a solution to (2.10) must have  $v_{N+i-1}^* = b_{ii}$  from (2.12) and  $\sum_{i=1}^r b_{ii}$  is minimized. Also the solution satisfies  $\mathbf{W}_N \succeq 0$  from the constraint in (2.10). Therefore, if  $\mathbf{v}^* = (w_1^*, \dots, w_{N-1}^*, v_N^*, \dots, v_{N+r-1}^*)^\top$  is a solution to problem (2.10) then  $\mathbf{w}^* = (w_1^*, \dots, w_{N-1}^*, 1 - \sum_{i=1}^{N-1} w_i^*)^\top$  is a solution to problem (2.8).  $\square$

The formulation in (2.8) is straightforward and is a useful step to check if an optimal design problem can be transformed into a SDP problem. The matrix  $\mathbf{D}$  is important in the formation of the SDP problem in (2.10), but it may not be unique. Any  $\mathbf{D}$  with rank  $q$  works in Theorem 2.1. We also notice that the problem in (2.10) is not in the same form as in (2.3). To find matrices  $\mathbf{H}_0, \dots, \mathbf{H}_{N+r-1}$  so that the constraint in (2.10) can be written in the form as in (2.3), we define vectors

$$\mathbf{f}(\mathbf{u}_i) = \frac{\partial g(\mathbf{u}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}}, \quad i = 1, \dots, N, \quad (2.13)$$

and  $q \times q$  matrices,

$$\mathbf{C}_i = \mathbf{D}^{-\top} \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top \mathbf{D}^{-1}, \quad i = 1, \dots, N. \quad (2.14)$$

Then we can express the constraint in (2.10) as

$$\mathbf{B}_1 \oplus \dots \oplus \mathbf{B}_r \oplus \mathbf{W}_N = \mathbf{H}_0 + w_1 \mathbf{H}_1 + \dots + w_{N-1} \mathbf{H}_{N-1} + v_N \mathbf{H}_N + \dots + v_{N+r-1} \mathbf{H}_{N+r-1},$$

where  $\mathbf{H}_0, \dots, \mathbf{H}_{N+r-1}$  are  $(r(q+1) + N) \times (r(q+1) + N)$  constant symmetric matrices

and given by

$$\begin{aligned}
\mathbf{H}_0 &= \left( \begin{array}{cc} \mathbf{C}_N & \mathbf{e}_1 \\ \mathbf{e}_1^\top & 0 \end{array} \right) \oplus \left( \begin{array}{cc} \mathbf{C}_N & \mathbf{e}_2 \\ \mathbf{e}_2^\top & 0 \end{array} \right) \oplus \cdots \oplus \left( \begin{array}{cc} \mathbf{C}_N & \mathbf{e}_r \\ \mathbf{e}_r^\top & 0 \end{array} \right) \oplus \text{diag}(\underbrace{0, \dots, 0}_{N-1}, 1), \\
\mathbf{H}_1 &= \underbrace{\left( \begin{array}{cc} \mathbf{C}_1 - \mathbf{C}_N & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{array} \right) \oplus \cdots \oplus \left( \begin{array}{cc} \mathbf{C}_1 - \mathbf{C}_N & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{array} \right)}_r \oplus \text{diag}(1, \underbrace{0, \dots, 0}_{N-2}, -1), \\
&\vdots \\
\mathbf{H}_i &= \underbrace{\left( \begin{array}{cc} \mathbf{C}_i - \mathbf{C}_N & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{array} \right) \oplus \cdots \oplus \left( \begin{array}{cc} \mathbf{C}_i - \mathbf{C}_N & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{array} \right)}_r \oplus \text{diag}(\underbrace{0, \dots, 0}_{i-1}, 1, \underbrace{0, \dots, 0}_{N-i-1}, -1), \\
&\vdots \\
\mathbf{H}_{N-1} &= \underbrace{\left( \begin{array}{cc} \mathbf{C}_{N-1} - \mathbf{C}_N & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{array} \right) \oplus \cdots \oplus \left( \begin{array}{cc} \mathbf{C}_{N-1} - \mathbf{C}_N & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{array} \right)}_r \oplus \text{diag}(\underbrace{0, \dots, 0}_{N-2}, 1, -1), \\
\mathbf{H}_{N+j} &= \text{diag}(\underbrace{0, \dots, 0}_{(j+1)(q+1)-1}, 1, 0, \dots, 0), \quad j = 0, 1, \dots, r-1. \tag{2.15}
\end{aligned}$$

For a given regression model, a design space  $S_N$  and a design criterion, we can compute vectors  $\mathbf{f}(\mathbf{u}_i)$  in (2.13) and specify matrix  $\mathbf{D}$  in (2.7), so the SDP problem in (2.10) or (2.3) is well defined. All the  $\mathbf{H}_i$ 's in (2.3) are given in (2.15). Here is a general algorithm to apply SeDuMi for computing the optimal design.

---

**Algorithm 2.1: Use SeDuMi for computing optimal designs**

---

Step 1: For a given regression model and a discrete design space  $S_N$ , write down the information matrix  $\mathbf{A}(\mathbf{w})$  as in (2.5).

Step 2: Find a matrix  $\mathbf{T}$  of rank  $r$  so that the design criterion can be written as in (2.6), and construct the nonsingular matrix  $\mathbf{D}$  in (2.7).

Step 3: Let  $\mathbf{B}(\mathbf{w}) = \mathbf{D}^{-\top} \mathbf{A}(\mathbf{w}) \mathbf{D}^{-1}$  and  $\mathbf{W}_N = \text{diag}(w_1, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i)$ .

Step 4: Use (2.9) and the linear matrix constraint in (2.10) to find matrices  $\mathbf{H}_0, \dots, \mathbf{H}_{N+r-1}$  so that the constraint in (2.10) is written in the form as in (2.3). The details are given in (2.13), (2.14) and (2.15).

Step 5: Follow **MATLAB program for Example 2.1** to apply SeDuMi for finding a solution to problem (2.10).

---

Section 2.3 will discuss various optimality criteria and provide transformations to form SDP problems.

### 2.2.3 Kiefer-Wolfowitz equivalence theorem

Since the problems in (2.8) and (2.10) are (strictly) convex optimization problems, the solution should be unique and is globally optimal. However, we still want to make sure that the numerical solution from Algorithm 2.1 gives the globally optimal design. The Kiefer-Wolfowitz equivalence theorem (Pukelsheim, 1993) allows us to easily verify whether the generated design is globally optimal when the design criterion is convex or concave. For example, to check optimality for problem (2.6) or (2.8), we first define

$$\phi_{A_i}(\mathbf{w}) = (\mathbf{f}(\mathbf{u}_i))^\top (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{T}^\top \mathbf{T} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{f}(\mathbf{u}_i), \quad i = 1, \dots, N, \quad (2.16)$$

and apply Lemma 2.2 below.

**Lemma 2.1.** *The function  $\phi(\mathbf{w})$  is convex in  $\mathbf{w}$ .*

**Proof of Lemma 2.1:** Let  $\mathbf{w}_0$  and  $\mathbf{w}_1$  be two weight vectors and  $\alpha \in [0, 1]$ , and define  $\mathbf{w}_\alpha = (1 - \alpha)\mathbf{w}_0 + \alpha\mathbf{w}_1$ . Assume  $\mathbf{A}(\mathbf{w}_0)$  and  $\mathbf{A}(\mathbf{w}_1)$  are nonsingular. We need to show that  $\phi(\mathbf{w}_\alpha)$  is a convex function of  $\alpha$ . It is easy to get

$$\begin{aligned} \frac{\partial \phi(\mathbf{w}_\alpha)}{\partial \alpha} &= -\text{trace} \left( \mathbf{T} \mathbf{A}^{-1}(\mathbf{w}_\alpha) (\mathbf{A}(\mathbf{w}_1) - \mathbf{A}(\mathbf{w}_0)) \mathbf{A}^{-1}(\mathbf{w}_\alpha) \mathbf{T}^\top \right), \\ \frac{\partial^2 \phi(\mathbf{w}_\alpha)}{\partial \alpha^2} &= 2 \text{trace} \left( \mathbf{T} \mathbf{A}^{-1}(\mathbf{w}_\alpha) (\mathbf{A}(\mathbf{w}_1) - \mathbf{A}(\mathbf{w}_0)) \mathbf{A}^{-1}(\mathbf{w}_\alpha) (\mathbf{A}(\mathbf{w}_1) - \mathbf{A}(\mathbf{w}_0)) \mathbf{A}^{-1}(\mathbf{w}_\alpha) \mathbf{T}^\top \right). \end{aligned}$$

Since the information matrices  $\mathbf{A}(\mathbf{w}_0)$  and  $\mathbf{A}(\mathbf{w}_1)$  are positive definite,  $\mathbf{A}(\mathbf{w}_\alpha)$  is also positive definite. Then it is clear that  $\frac{\partial^2 \phi(\mathbf{w}_\alpha)}{\partial \alpha^2} \geq 0$ , which implies that  $\phi(\mathbf{w}_\alpha)$  is a convex function of  $\alpha$ .  $\square$

**Lemma 2.2.** *A weight vector  $\hat{\mathbf{w}}$  is an optimal design if and only if matrix the  $\mathbf{A}(\hat{\mathbf{w}})$  is nonsingular and  $\phi_{A_i}(\hat{\mathbf{w}}) \leq \phi(\hat{\mathbf{w}})$  for all  $i = 1, \dots, N$ .*

**Proof of Lemma 2.2:** For any  $\mathbf{w}$ , define  $\mathbf{w}_\alpha = (1 - \alpha)\hat{\mathbf{w}} + \alpha\mathbf{w}$ . If  $\hat{\mathbf{w}}$  is an optimal design, then we must have  $\frac{\partial\phi(\mathbf{w}_\alpha)}{\partial\alpha}|_{\alpha=0} \geq 0$  for any  $\mathbf{w}$ . Similar to the proof of Lemma 2.1, we have

$$\begin{aligned}
\frac{\partial\phi(\mathbf{w}_\alpha)}{\partial\alpha}|_{\alpha=0} &= -\text{trace}(\mathbf{T}\mathbf{A}^{-1}(\hat{\mathbf{w}})(\mathbf{A}(\mathbf{w}) - \mathbf{A}(\hat{\mathbf{w}}))\mathbf{A}^{-1}(\hat{\mathbf{w}})\mathbf{T}^\top), \\
&= -\text{trace}(\mathbf{T}\mathbf{A}^{-1}(\hat{\mathbf{w}})\mathbf{A}(\mathbf{w})\mathbf{A}^{-1}(\hat{\mathbf{w}})\mathbf{T}^\top) + \phi(\hat{\mathbf{w}}) \\
&= -\text{trace}\left(\mathbf{T}\mathbf{A}^{-1}(\hat{\mathbf{w}})\sum_{i=1}^N w_i \mathbf{f}(\mathbf{u}_i)(\mathbf{f}(\mathbf{u}_i))^\top \mathbf{A}^{-1}(\hat{\mathbf{w}})\mathbf{T}^\top\right) + \sum_{i=1}^N w_i \phi(\hat{\mathbf{w}}), \\
&\hspace{15em} \text{by (2.5) and (2.13),} \\
&= -\sum_{i=1}^N w_i (\phi_{A_i}(\hat{\mathbf{w}}) - \phi(\hat{\mathbf{w}})), \quad \text{by (2.16),} \\
&\geq 0, \quad \text{for any } \mathbf{w},
\end{aligned}$$

which leads to  $\phi_{A_i}(\hat{\mathbf{w}}) - \phi(\hat{\mathbf{w}}) \leq 0$ , for all  $i = 1, \dots, N$ .  $\square$

These results are similar to those in Bose and Mukerjee (2015). In practice, let  $\delta$  be a small positive number, say  $\delta = 10^{-5}$ . An optimal design with weight vector  $\hat{\mathbf{w}}$  should satisfy:

$$\phi_{A_i}(\hat{\mathbf{w}}) - \phi(\hat{\mathbf{w}}) \leq \delta, \quad \text{for all } i = 1, \dots, N. \quad (2.17)$$

## 2.3 Optimal design problems and transformations

Many optimal design problems belong to the class of problems in (2.6), and they include A-,  $A_s$ -, c-, I-, and L-optimal design problems. For each problem, we discuss how to obtain matrices  $\mathbf{T}$  and  $\mathbf{D}$ , and  $r$  in Algorithm 2.1.

### 2.3.1 A- and $A_s$ -optimal designs

For an A-optimal design problem, we minimize the average variance of  $\hat{\boldsymbol{\theta}}$ , i.e.,

$$\min_{\mathbf{w}} \text{trace}\left(\text{Var}(\hat{\boldsymbol{\theta}})\right),$$

which is equivalent to  $\min_{\mathbf{w}} \text{trace}(\mathbf{A}(\mathbf{w})^{-1})$ . Hence, it is obvious that the A-optimal design problem belongs to (2.6) with matrix  $\mathbf{T} = \mathbf{I}_q$ . Since this  $\mathbf{T}$  is already full rank,

we can just let  $\mathbf{D} = \mathbf{T} = \mathbf{I}_q$ , which leads to  $\mathbf{B}(\mathbf{w}) = \mathbf{A}(\mathbf{w})$  and  $r = q$  in Algorithm 2.1.

For an  $A_s$ -optimal design problem, we minimize the average variance of a subset of  $\hat{\boldsymbol{\theta}}$ . For example, we minimize the average variance of  $(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3)^\top$ , assuming  $q > 3$ . Without loss of generality (which can be done by permuting the order of regression parameters if needed), let  $\hat{\boldsymbol{\theta}}_s = (\hat{\theta}_1, \dots, \hat{\theta}_s)^\top$  be the subset of interest, where  $s < q$ . The  $A_s$ -optimal design problem can be written as

$$\min_{\mathbf{w}} \text{trace} \left( \text{Var}(\hat{\boldsymbol{\theta}}_s) \right),$$

which is equivalent to  $\min_{\mathbf{w}} \text{trace} \left( \mathbf{T} \mathbf{A}(\mathbf{w})^{-1} \mathbf{T}^\top \right)$  with  $\mathbf{T} = (\mathbf{I}_s, \mathbf{0}_{s \times (q-s)})$ . In this case, we choose  $\mathbf{D} = \mathbf{I}_q$ , so  $\mathbf{B}(\mathbf{w}) = \mathbf{A}(\mathbf{w})$  and we have  $r = s$  in Algorithm 2.1.

### 2.3.2 c-optimal design

For a c-optimal design problem, we minimize the variance of  $\mathbf{c}^\top \hat{\boldsymbol{\theta}}$ , i.e.,

$$\min_{\mathbf{w}} \mathbf{c}^\top \text{Var}(\hat{\boldsymbol{\theta}}) \mathbf{c} = \min_{\mathbf{w}} \sigma^2 \mathbf{c}^\top \mathbf{A}(\mathbf{w})^{-1} \mathbf{c},$$

where  $\mathbf{c} = (c_1, \dots, c_q)^\top \in R^q$  is a user-selected constant vector. Without loss of generality, we assume that  $c_1 \neq 0$ . For this criterion, we have  $\mathbf{T} = \mathbf{c}^\top$ , a  $1 \times q$  matrix, and we choose

$$\mathbf{D} = \begin{pmatrix} c_1 & c_2 \cdots c_q \\ \mathbf{0}_{(q-1) \times 1} & \mathbf{I}_{q-1} \end{pmatrix}_{q \times q}.$$

In Algorithm 2.1, we have  $r = 1$ .

### 2.3.3 I-optimal design

Consider a linear regression model,  $y_i = \boldsymbol{\theta}^\top \mathbf{f}(\mathbf{x}_i) + \epsilon_i$ ,  $i = 1, \dots, n$ . For an I-optimal design problem, we minimize the average predicted variance over the design space, i.e.,

$$\min_{\mathbf{w}} \text{trace} \left( \mathbf{A}(\mathbf{w})^{-1} \mathbf{M} \right),$$

where  $\mathbf{M} = \int_S \mathbf{f}(\mathbf{x}) (\mathbf{f}(\mathbf{x}))^\top d\mathbf{x}$  or  $\mathbf{M} = \frac{1}{N} \sum_{i=1}^N \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top$ . Matrix  $\mathbf{M}$  is positive definite, so its rank is  $q$ . Let  $\mathbf{M}^{1/2}$  be the symmetric square root of  $\mathbf{M}$ , i.e.,  $(\mathbf{M}^{1/2})^\top = \mathbf{M}^{1/2}$ , and  $\mathbf{M}^{1/2} \mathbf{M}^{1/2} = \mathbf{M}$ . For this criterion, we have  $\mathbf{T} = \mathbf{M}^{1/2}$ , which is a  $q \times q$

matrix with rank  $q$ , and so we can choose  $\mathbf{D} = \mathbf{T}$ . In Algorithm 2.1, we have  $r = q$ .

### 2.3.4 L-optimal design

For an L-optimal design problem, we minimize the average variance of some function of  $\hat{\boldsymbol{\theta}}$ , i.e.,

$$\min_{\mathbf{w}} \text{trace}(\mathbf{L}\mathbf{A}(\mathbf{w})^{-1}),$$

where  $\mathbf{L}$  is a  $q \times q$  researcher-selected matrix, which reflects the interest of the researcher. The matrix  $\mathbf{L}$  can often be written as  $\mathbf{L} = \mathbf{L}_1\mathbf{L}_1^\top$ , where  $\mathbf{L}_1$  is a  $q \times r$  matrix with  $q \geq r$ . Berger and Wong (2009, p. 242) provides examples. Then, we have

$$\min_{\mathbf{w}} \text{trace}(\mathbf{L}\mathbf{A}(\mathbf{w})^{-1}) = \min_{\mathbf{w}} \text{trace}(\mathbf{L}_1\mathbf{L}_1^\top\mathbf{A}(\mathbf{w})^{-1}) = \min_{\mathbf{w}} \text{trace}(\mathbf{L}_1^\top\mathbf{A}(\mathbf{w})^{-1}\mathbf{L}_1).$$

Thus, for this criterion, we have  $\mathbf{T} = \mathbf{L}_1^\top$ , and we choose  $\mathbf{D}$  as indicated in (2.7).

## 2.4 Applications

In this section, we provide several examples to find optimal designs using Algorithm 2.1 and show that it is both efficient and flexible. Example 2.2 constructs c-optimal designs, while Example 2.3 and 2.6 find L-optimal designs. Example 2.4 and 2.5 give two examples for I-optimal designs for linear and nonlinear model, respectively. We choose different  $N$  values for different examples. The candidate design points on a discretized design space will be varying for different values of  $N$ . Therefore the calculated optimal designs would be not all the same, especially for relevant small values of  $N$ , which can be noticed from several examples in this dissertation. However, as the value of  $N$  goes larger, the calculated support points for optimal designs are clustered to the theoretical support points, as calculated optimal designs converge. Two representative MATLAB programs are also given in the Appendix.

**Example 2.2.** Consider the quadratic model,  $y_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \epsilon_i$ , with design space  $S = [-1, 1]$ . If we are interested in the average response at  $x = 2$ , which is  $\theta_0 + 2\theta_1 + 4\theta_2$ , and we want to minimize the variance of the LSE of  $\theta_0 + 2\theta_1 + 4\theta_2$ , then the corresponding design problem is a c-optimal design problem with  $\mathbf{c}^\top = (1, 2, 4)$ . This design problem is studied in Berger and Wong (2009, page 240).

Suppose we discretize the design space  $S = [-1, 1]$  into equally spaced points to form the discrete design space  $S_N = \{u_i = -1 + \frac{2(i-1)}{N-1} : i = 1, \dots, N\}$ . To use Algorithm 2.1 to find the  $c$ -optimal design for  $N = 501$ , we first calculate the matrix  $\mathbf{D}$  from Section 2.3.2 and get

$$\mathbf{D} = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Our MATLAB program given in the Appendix finds the optimal design and it is very fast. The  $c$ -optimal design is

$$\xi = \begin{pmatrix} -1 & 0 & 1 \\ 0.1428572 & 0.4285714 & 0.4285714 \end{pmatrix}.$$

The points with zero weights are not listed as support points of  $\xi$ , which can be directly verified to be optimal using the condition in (2.17) with  $\delta = 10^{-5}$ . This design result also agrees with the optimal design reported in Berger and Wong (2009, page 240).  $\square$

**Example 2.3.** Consider the cubic model  $y_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \theta_3 x_i^3 + \epsilon_i$  on the design space  $S = [-1, 1]$ .

Suppose we are interested in the inference for  $\mathbf{T}\boldsymbol{\theta} = (\theta_0 - \theta_1, \theta_0 - \theta_2, \theta_0 - \theta_3)^\top$ . We choose  $N = 201$ , and the design space is  $S_N = \{u_i = -1 + \frac{2(i-1)}{N-1} : i = 1, \dots, N\}$ . The matrix  $\mathbf{T}$  is

$$\mathbf{T} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{pmatrix}.$$

To solve this problem in SeDuMi, we let

$$\mathbf{D} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

It is clear that  $\mathbf{D}$  is a full rank matrix. The optimal design is minimizing trace  $(\text{Cov}(\mathbf{T}\hat{\boldsymbol{\theta}}))$ ,

and we get

$$\xi = \begin{pmatrix} -1 & -0.42 & 0.41 & 1 \\ 0.1291 & 0.3644 & 0.3638 & 0.1427 \end{pmatrix}.$$

This result is directly verified to be optimal using the condition in (2.17) with  $\delta = 10^{-5}$ .  $\square$

**Example 2.4.** Consider a linear model with three regressors,

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_{12} x_1 x_2 + \theta_{13} x_1 x_3 + \theta_{23} x_2 x_3 + \epsilon,$$

with a cubic design space. For each design variable we choose three values  $-1$ ,  $0$  and  $1$ . Therefore, we have  $N = 27$  support points in  $S_N$ , including 8 vertices, one point at the center of the cube, and 18 others. For the I-optimal design problem, we use  $\mathbf{M} = \frac{1}{N} \sum_{i=1}^N \mathbf{f}(\mathbf{u}_i)(\mathbf{f}(\mathbf{u}_i))^\top$ . Using Algorithm 2.1 we find the design in Table 2.1 in less than 1 second. The design has nonzero weights at the 8 vertices and it is easy to verify that this  $2^3$  factorial design satisfies the condition in (2.17) with  $\delta = 10^{-10}$ . We also computed the I-optimal design for five regressors with  $N = 243$  and got the  $2^5$  factorial design.  $\square$

Table 2.1: The I-optimal design for Example 2.4

$(x_1, x_2, x_3)$	$w_i$
$(-1, -1, -1)$	0.125
$(-1, -1, +1)$	0.125
$(-1, +1, -1)$	0.125
$(-1, +1, +1)$	0.125
$(+1, -1, -1)$	0.125
$(+1, -1, +1)$	0.125
$(+1, +1, -1)$	0.125
$(+1, +1, +1)$	0.125

**Example 2.5.** Consider a nonlinear regression model,

$$y_i = \frac{\theta_1}{\theta_1 - \theta_2} (e^{-\theta_2 x_i} - e^{-\theta_1 x_i}) + \epsilon_i,$$

where  $\theta_1 > \theta_2 > 0$ , and  $x_i \in S = [0, 20]$ . The initial parameter estimates are  $\theta_1^* = 0.70$

and  $\theta_2^* = 0.20$ . Optimal designs have been studied, for example, in Dette and O'Brien (1999). We choose  $N = 501$  and the design space is  $S_N = \{u_i = 20(i-1)/(N-1), i = 1, \dots, N\}$ . For the I-optimal design problem, we use  $\mathbf{M} = \frac{1}{N} \sum_{i=1}^N \mathbf{f}(u_i)(\mathbf{f}(u_i))^\top$ . Using Algorithm 2.1 we can find the I-optimal design in about 85.8 seconds. The I-optimal design has two support points

$$\xi = \begin{pmatrix} 1.32 & 6.76 \\ 0.32798 & 0.67202 \end{pmatrix}.$$

It is easy to verify that this optimal design satisfies the condition in (2.17) with  $\delta = 10^{-6}$ , and it is also consistent with the result in Dette and O'Brien (1999). The MATLAB code for this example is given in the Appendix.

We have also computed optimal designs for various design space  $S$  and initial estimates of parameters, and representative optimal designs are shown in Table 2.2. The design space is  $S_N = \{u_i = a + (b-a)(i-1)/(N-1), i = 1, \dots, N\} \subset S = [a, b]$  with  $N = 501$ . Notice that all the I-optimal designs in Table 2.2 have two support points, and the designs depend on the initial estimates of parameters and the design space.  $\square$

**Example 2.6.** Consider the same quadratic model as in Example 2.2,  $y_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \epsilon_i$  on the design space  $[-1, 1]$ . However this time the goal is to estimate the turning point in the mean response curve (Berger and Wong, 2009). We differentiate the mean function and set it equal to zero, the turning point occurs when the value of  $x$  is equal to  $-\frac{\theta_1}{2\theta_2}$ . The asymptotic variance of the maximum likelihood estimator of the turning point is approximately equal to

$$\text{trace}(\mathbf{L}\mathbf{A}(\mathbf{w})^{-1}) = \text{trace}(\mathbf{L}_1\mathbf{L}_1^\top\mathbf{A}(\mathbf{w})^{-1})$$

where  $\mathbf{L}_1^\top = (0, -1/(2\theta_2), \theta_1/(2\theta_2^2))$  is the derivative vector of turning point with respect to the three parameters. Representative L-optimal designs for different values of parameters are shown in Table 2.3, when  $N = 201$ .  $\square$

Table 2.2: I-optimal designs for the nonlinear model in Example 2.5

Initial estimates of $\theta_1$ and $\theta_2$ and $S$	Optimal designs
$\theta_1^* = 0.9, \theta_2^* = 0.3, S = [0, 20]$	$\xi = \begin{pmatrix} 1.00 & 4.76 \\ 0.3374 & 0.6626 \end{pmatrix}$
$\theta_1^* = 1.2, \theta_2^* = 0.5, S = [0, 20]$	$\xi = \begin{pmatrix} 0.72 & 3.12 \\ 0.3528 & 0.6472 \end{pmatrix}$
$\theta_1^* = 1.8, \theta_2^* = 1.2, S = [0, 20]$	$\xi = \begin{pmatrix} 0.40 & 1.60 \\ 0.3798 & 0.6202 \end{pmatrix}$
$\theta_1^* = 0.5, \theta_2^* = 0.05, S = [0, 20]$	$\xi = \begin{pmatrix} 1.88 & 20.0 \\ 0.3641 & 0.6359 \end{pmatrix}$
$\theta_1^* = 0.5, \theta_2^* = 0.05, S = [0, 25]$	$\xi = \begin{pmatrix} 1.85 & 22.10 \\ 0.3189 & 0.6811 \end{pmatrix}$
$\theta_1^* = 0.09, \theta_2^* = 0.04, S = [0, 20]$	$\xi = \begin{pmatrix} 7.56 & 20.00 \\ 0.6026 & 0.3974 \end{pmatrix}$
$\theta_1^* = 0.09, \theta_2^* = 0.04, S = [0, 30]$	$\xi = \begin{pmatrix} 9.24 & 30.00 \\ 0.5524 & 0.4476 \end{pmatrix}$
$\theta_1^* = 0.09, \theta_2^* = 0.04, S = [0, 50]$	$\xi = \begin{pmatrix} 9.70 & 39.30 \\ 0.4318 & 0.5682 \end{pmatrix}$
$\theta_1^* = 0.8, \theta_2^* = 0.08, S = [0, 10]$	$\xi = \begin{pmatrix} 1.20 & 10.00 \\ 0.4111 & 0.5889 \end{pmatrix}$
$\theta_1^* = 0.8, \theta_2^* = 0.08, S = [0, 15]$	$\xi = \begin{pmatrix} 1.17 & 13.83 \\ 0.3265 & 0.6735 \end{pmatrix}$

Table 2.3: L-optimal designs for the quadratic model in Example 2.6

Initial estimates of $\theta_1$ and $\theta_2$ and $-\frac{\theta_1}{2\theta_2}$	Optimal designs
$\theta_1^* = -2, \theta_2^* = 1, -\frac{\theta_1}{2\theta_2} = 1$	$\xi = \begin{pmatrix} -1.00 & 0 & 1.00 \\ 0.1250 & 0.5000 & 0.3750 \end{pmatrix}$
$\theta_1^* = -1, \theta_2^* = 2, -\frac{\theta_1}{2\theta_2} = \frac{1}{4}$	$\xi = \begin{pmatrix} -0.50 & 1.00 \\ 0.5000 & 0.5000 \end{pmatrix}$
$\theta_1^* = 1, \theta_2^* = 2, -\frac{\theta_1}{2\theta_2} = -\frac{1}{4}$	$\xi = \begin{pmatrix} -1.00 & 0.50 \\ 0.5000 & 0.5000 \end{pmatrix}$

## 2.5 Other optimal design problems

There are other design problems that can be solved via SeDuMi. In this section, we will show two classes of design problems that can be transformed into SDP problems and solved by SeDuMi. One class includes design problems based on the WLSE, and another includes design problems with linear constraints on the design weights.

### 2.5.1 Optimal design for unequal error variances

When the error variance in model (2.1) is not constant, say  $Var(\epsilon_i) = \sigma^2/\lambda(\mathbf{x}_i)$ , and  $\lambda(\mathbf{x})$  is a given positive function, the WLSE is more efficient than the LSE. For simplicity, let us focus on linear models here,  $y_i = \boldsymbol{\theta}^\top \mathbf{f}(\mathbf{x}_i) + \epsilon_i$ . The WLSE of  $\boldsymbol{\theta}$  is given by

$$\hat{\boldsymbol{\theta}}_\lambda = \left( \sum_{i=1}^n \lambda(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i) (\mathbf{f}(\mathbf{x}_i))^\top \right)^{-1} \sum_{i=1}^n \lambda(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i) y_i,$$

and  $Var(\hat{\boldsymbol{\theta}}_\lambda) = \frac{\sigma^2}{n} \left( \frac{1}{n} \sum_{i=1}^n \lambda(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i) (\mathbf{f}(\mathbf{x}_i))^\top \right)^{-1}$ . Following the discussion in Section 2.2, we modify the information matrix to study optimal designs based on the WLSE as,

$$\mathbf{A}_\lambda(\mathbf{w}) = \sum_{i=1}^N w_i \lambda(\mathbf{u}_i) \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top = \sum_{i=1}^N w_i \sqrt{\lambda(\mathbf{u}_i)} \mathbf{f}(\mathbf{u}_i) (\sqrt{\lambda(\mathbf{u}_i)} \mathbf{f}(\mathbf{u}_i))^\top. \quad (2.18)$$

Thus, design problems based on the WLSE can also be written as

$$\begin{cases} \min_{\mathbf{w}} \text{trace}(\mathbf{T}(\mathbf{A}_\lambda(\mathbf{w}))^{-1} \mathbf{T}^\top) \\ \text{subject to: } w_i \geq 0, \quad i = 1, \dots, N, \quad \text{and} \quad \sum_{i=1}^N w_i = 1, \end{cases} \quad (2.19)$$

which are similar to those in (2.6), and they include A-,  $A_s$ -, c-, I-, and L-optimal design problems. The transformations discussed in Section 2.3 to form SDP problems can also be applied to (2.19). We can still use Algorithm 2.1 for computing the optimal designs based on the WLSE, but we need to make one change in equation (2.14). The change is

$$\mathbf{C}_i = \mathbf{D}^{-\top} \lambda(\mathbf{u}_i) \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top \mathbf{D}^{-1}, \quad i = 1, \dots, N,$$

which is due to the change in the information matrix in (2.18).

The condition to check for an optimal design  $\hat{\mathbf{w}}$  becomes,

$$\phi_{Ai\lambda}(\hat{\mathbf{w}}) - \phi_{\lambda}(\hat{\mathbf{w}}) \leq \delta, \text{ for all } i = 1, \dots, N,$$

where

$$\begin{aligned} \phi_{\lambda}(\mathbf{w}) &= \text{trace}(\mathbf{T}(\mathbf{A}_{\lambda}(\mathbf{w}))^{-1}\mathbf{T}^{\top}), \\ \phi_{Ai\lambda}(\mathbf{w}) &= \lambda(\mathbf{u}_i)(\mathbf{f}(\mathbf{u}_i))^{\top}(\mathbf{A}_{\lambda}(\mathbf{w}))^{-1}\mathbf{T}^{\top}\mathbf{T}(\mathbf{A}_{\lambda}(\mathbf{w}))^{-1}\mathbf{f}(\mathbf{u}_i), \quad i = 1, \dots, N. \end{aligned}$$

This condition is easily derived from (2.17).

**Example 2.7.** Consider the cubic regression model,

$$y_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \theta_3 x_i^3 + \epsilon_i, \quad i = 1, \dots, n,$$

and the design space is  $S = [-1, 1]$ . The random errors  $\epsilon_i$ 's are independent, each with mean 0 and variance that depends where the observation is taken. We model the heteroscedasticity by letting  $Var(\epsilon_i) = \sigma^2/\lambda(x_i)$ , and for specificity, assume  $\lambda(x_i) = (1 + x_i^2)^{-4}$  is a positive function of  $x_i$ . This and similar variance functions have been used in optimal designs in the literature, for example, Dette et al. (1999). The discrete design space is  $S_N = \{-1 + 2(i-1)/(N-1), i = 1, \dots, N\}$  with  $N = 501$ . Using Algorithm 2.1, the generated design is

$$\xi = \begin{pmatrix} -1.000 & -0.328 & 0.328 & 1.000 \\ 0.25273 & 0.24727 & 0.24727 & 0.25273 \end{pmatrix},$$

which can be verified to be A-optimal using the Kiefer-Wolfowitz's equivalence theorem. Our choice of the form of  $\lambda(x)$  is arbitrary and the method works for other forms of heteroscedasticity as well.  $\square$

## 2.5.2 Optimal design with linear constraints on weights

Sometimes there are linear constraints on the design weights to ensure that the optimal designs have certain structure or properties. For example, it may be desirable that the optimal designs be symmetric or rotatable. For linear equality constraints on  $\mathbf{w}$ , we can easily deal with them by reducing the number of independent weights in the design problem. For linear inequality constraints, we can put them in the

matrix constraint  $\mathbf{W}_N \succeq 0$  by modifying  $\mathbf{W}_N$ . Let us use the following two simple constraints to illustrate these ideas. Suppose we want the weights to satisfy

$$w_1 = w_N, \quad w_2 \geq w_3.$$

One is an equality constraint, and the other is an inequality constraint. Since  $\sum_{i=1}^N w_i = 1$  and  $w_1 = w_N$ , there are only  $N - 2$  independent weights. Thus, we can write the information matrix  $\mathbf{A}(\mathbf{w})$  as a linear function of  $w_1, w_2, \dots, w_{N-2}$  by replacing  $w_N$  and  $w_{N-1}$  by  $w_1$  and  $1 - w_1 - \sum_{i=1}^{N-2} w_i$ , respectively. The matrix  $\mathbf{W}_N$  becomes

$$\mathbf{W}'_N = \text{diag} \left( w_1, w_2, \dots, w_{N-2}, 1 - w_1 - \sum_{i=1}^{N-2} w_i, w_2 - w_3 \right),$$

which is positive-semidefinite and includes both constraints  $w_1 = w_N$ ,  $w_2 \geq w_3$ . The results in Theorem 2.1 are still valid after replacing  $\mathbf{W}_N$  with  $\mathbf{W}'_N$ .

## 2.6 Discussion

We have shown that many optimal design problems can be transformed into SDP problems and can be solved by SeDuMi in MATLAB. The approach is very useful for finding optimal designs if we have a given discrete design space. When the number of points in the design space is large, it is challenging for some existing algorithms to find optimal designs. SeDuMi is fast and can handle situations when the design space is discretized using a large number of points. For our examples, the average computation times required to find the optimal designs in Examples 2.2, 2.5 and 2.7 are about 97.2, 90 and 100 seconds, respectively, when  $N = 501$ . For Examples 2.3 and 2.6, it took about 16 and 12 seconds to get the results, respectively, when  $N = 201$ . For Example 2.4, it took less than 1 second for the 3 x-variable problem and about 80 seconds for the 5 x-variable problem. We are particularly pleased with the speed of SeDuMi and its capability for finding different types of optimal designs over a fine set of grid points with  $N=501$ ; other algorithms, such as the multiplicative algorithm may not work or will take a much longer time to find the optimal design. Additionally, we have derived theoretical results useful for using SeDuMi, and also presented a condition to check if the SeDuMi-generated design is optimal.

In Ye et al. (2015), two criteria (A-optimality and E-optimality) were discussed. A-optimality criterion is a special case of (2.6), but E-optimality criterion does not belong to (2.6). E-optimality is a minimax type of criterion, not differentiable but still E-optimal design problems can also be transformed into SDP problems.

We close by emphasizing that the methodology presented here is quite general. It is applicable to find several types of optimal designs for linear and nonlinear regression models defined on a discretized design space, including when there are linear constraints on the weight distribution of the sought optimal design.

## Chapter 3

# Optimal designs using the second-order least squares estimator

In this chapter, we investigate properties and numerical algorithms for A- and D-optimal regression designs based on the SLSE in Wang and Leblanc (2008). Usually, we use the LSE to estimate the unknown parameters as in Chapter 2. Many optimal designs constructed based on the LSE are presented in Chapter 2. However, when the error distribution is highly skewed, our results indicate that the optimal designs based on the SLSE are more efficient than those based on the LSE. In this chapter, we still focus on one-response model and compare the results with the optimal designs based on the LSE. First we discuss the SLSE and its properties. To compute A-optimal designs based on the SLSE, we derive a characterization of the A-optimality criterion, and formulate the optimal design problems under the SLSE as a semidefinite programming problem. We then apply the SeDuMi and CVX programs in MATLAB to compute A- and D-optimal designs under the SLSE. We find that the resulting algorithms for both A- and D-optimal designs under the SLSE can be faster than more conventional multiplicative algorithms, especially in nonlinear models.

Chapter 3 is organized as follows. In Section 3.1, we give a brief review for the SLSE and introduce the design problem under the SLSE. In Section 3.2, we derive several properties of optimal designs under the SLSE and an expression of the A-optimality criterion. In Section 3.3, we develop numerical algorithms for computing

optimal designs. In Section 3.4, we present several applications, and compare numerical algorithms and efficiencies of optimal designs. In Section 3.5, we give a conclusion for this chapter. The main work of this chapter has been accepted for publication in *Statistica Sinica*; see Yin and Zhou (2016).

### 3.1 Introduction

Optimal design criteria are usually based on the LSE as in Chapter 2. Recently, Wang and Leblanc (2008) proposed the SLSE, a new method of estimation for unknown parameters in regression models. The SLSE takes the distribution of errors into consideration, and it turns out that the SLSE is more efficient than the LSE when the error distribution is not symmetric. Using this result, Gao and Zhou (2014) proposed new optimality criteria under the SLSE and obtained several results. Bose and Mukerjee (2015) and Gao and Zhou (2015) made further developments, including the convexity results for the criteria and numerical algorithms. Bose and Mukerjee (2015) applied the multiplicative algorithms in Zhang and Mukerjee (2013) for computing the optimal designs, while Gao and Zhou (2015) used the CVX program in MATLAB (Grant and Boyd, (2013)). The previous work focused more on the D-optimal designs than on the A-optimal designs, which motivated us to investigate more properties of A-optimal designs based on the SLSE and develop an efficient algorithm for computing A-optimal designs.

In model (2.1), the SLSE  $\hat{\gamma}_{SLSE}$  of  $\boldsymbol{\gamma} = (\boldsymbol{\theta}^\top, \sigma^2)^\top$  minimizes

$$Q(\boldsymbol{\gamma}) = \sum_{i=1}^n \rho_i^\top(\boldsymbol{\gamma}) \tilde{\mathbf{W}}_i \rho_i(\boldsymbol{\gamma}),$$

where vector  $\rho_i(\boldsymbol{\gamma}) = (y_i - g(\mathbf{x}_i; \boldsymbol{\theta}), y_i^2 - g^2(\mathbf{x}_i; \boldsymbol{\theta}) - \sigma^2)^\top$  includes the differences between the observed and expected first and second moments of  $y$ , and  $\tilde{\mathbf{W}}_i = \tilde{\mathbf{W}}(\mathbf{x}_i)$  is a  $2 \times 2$  positive semidefinite matrix that may depend on  $\mathbf{x}_i$ . The most efficient SLSE is obtained by choosing optimal matrices  $\tilde{\mathbf{W}}_i$  to minimize the asymptotic covariance matrix of  $\hat{\gamma}_{SLSE}$ , as derived in Wang and Leblanc (2008). For the rest of the chapter, the discussion is about the most efficient SLSE.

Suppose  $\boldsymbol{\theta}_0$  and  $\sigma_0^2$  are the true values of  $\boldsymbol{\theta}$  and  $\sigma^2$ , respectively. Let  $\mu_3 = E(\epsilon_1^3 | \mathbf{x})$ ,

$\mu_4 = E(\epsilon_1^4 | \mathbf{x})$ , and  $t = \mu_3^2 / (\sigma_0^2(\mu_4 - \sigma_0^4))$ . Under some regularity conditions (Wang and Leblanc, (2008)), the asymptotic covariance matrix of  $\hat{\boldsymbol{\gamma}}_{SLS}$  is

$$Cov(\hat{\boldsymbol{\gamma}}_{SLS}) = \begin{pmatrix} Cov(\hat{\boldsymbol{\theta}}_{SLS}) & \frac{\mu_3}{\mu_4 - \sigma_0^4} V(\hat{\sigma}_{SLS}^2) \mathbf{G}_2^{-1} \mathbf{g}_1 \\ \frac{\mu_3}{\mu_4 - \sigma_0^4} V(\hat{\sigma}_{SLS}^2) \mathbf{g}_1^\top \mathbf{G}_2^{-1} & V(\hat{\sigma}_{SLS}^2) \end{pmatrix}, \quad (3.1)$$

where

$$\begin{aligned} Cov(\hat{\boldsymbol{\theta}}_{SLS}) &= (1 - t) \sigma_0^2 (\mathbf{G}_2 - t \mathbf{g}_1 \mathbf{g}_1^\top)^{-1}, \\ V(\hat{\sigma}_{SLS}^2) &= \frac{(\mu_4 - \sigma_0^4)(1 - t)}{1 - t \mathbf{g}_1^\top \mathbf{G}_2^{-1} \mathbf{g}_1}, \end{aligned} \quad (3.2)$$

$$\begin{aligned} \mathbf{g}_1 &= E \left[ \frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \right], \\ \mathbf{G}_2 &= E \left[ \frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\top} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \right]. \end{aligned} \quad (3.3)$$

The expectation in (3.3) is taken with respect to the distribution of  $\mathbf{x}$ . The asymptotic covariance matrix of the LSE,  $\hat{\boldsymbol{\gamma}}_{OLS} = (\hat{\boldsymbol{\theta}}_{OLS}^\top, \hat{\sigma}_{OLS}^2)^\top$ , is

$$Cov(\hat{\boldsymbol{\gamma}}_{OLS}) = \begin{pmatrix} Cov(\hat{\boldsymbol{\theta}}_{OLS}) & \mu_3 \mathbf{G}_2^{-1} \mathbf{g}_1 \\ \mu_3 \mathbf{g}_1^\top \mathbf{G}_2^{-1} & V(\hat{\sigma}_{OLS}^2) \end{pmatrix} = \begin{pmatrix} \sigma_0^2 \mathbf{G}_2^{-1} & \mu_3 \mathbf{G}_2^{-1} \mathbf{g}_1 \\ \mu_3 \mathbf{g}_1^\top \mathbf{G}_2^{-1} & \mu_4 - \sigma_0^4 \end{pmatrix}. \quad (3.4)$$

If the error distribution is symmetric, then  $\mu_3 = 0$ ,  $t = 0$ , and the covariance matrices in (3.1) and (3.4) are the same. For asymmetric errors, we have  $0 < t < 1$  (Gao and Zhou, (2014)) and  $Cov(\hat{\boldsymbol{\gamma}}_{OLS}) - Cov(\hat{\boldsymbol{\gamma}}_{SLS}) \succeq 0$  from Wang and Leblanc (2008), so the SLSE is more efficient than the LSE.

## 3.2 A- and D-optimality criteria

In Gao and Zhou (2014), the A- and D-optimal designs based on the SLSE are defined to minimize  $\text{tr}(Cov(\hat{\boldsymbol{\theta}}_{SLS}))$  and  $\det(Cov(\hat{\boldsymbol{\theta}}_{SLS}))$ , respectively. For any distribution  $\xi(\mathbf{x}) \in \Xi_N$  of  $\mathbf{x}$ , let  $\xi(\mathbf{x}) = \{(\mathbf{u}_i, w_i) \mid w_i = P(\mathbf{x} = \mathbf{u}_i), \mathbf{u}_i \in S_N, i = 1, \dots, N\}$ , where

$$\sum_{i=1}^N w_i = 1, \quad \text{and } w_i \geq 0, \quad \text{for } i = 1, \dots, N. \quad (3.5)$$

Define  $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) = \partial g(\mathbf{x}; \boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ , and write  $\mathbf{g}_1$  and  $\mathbf{G}_2$  in (3.3) as

$$\begin{aligned}\mathbf{g}_1(\mathbf{w}) &= \mathbf{g}_1(\mathbf{w}; \boldsymbol{\theta}_0) = \sum_{i=1}^N w_i \mathbf{f}(\mathbf{u}_i; \boldsymbol{\theta}_0), \\ \mathbf{G}_2(\mathbf{w}) &= \mathbf{G}_2(\mathbf{w}; \boldsymbol{\theta}_0) = \sum_{i=1}^N w_i \mathbf{f}(\mathbf{u}_i; \boldsymbol{\theta}_0) \mathbf{f}^\top(\mathbf{u}_i; \boldsymbol{\theta}_0),\end{aligned}\tag{3.6}$$

where weight vector  $\mathbf{w} = (w_1, \dots, w_N)^\top$ .

Let  $\mathbf{A}(\mathbf{w}) = \mathbf{A}(\mathbf{w}; \boldsymbol{\theta}_0) = \mathbf{G}_2(\mathbf{w}) - t\mathbf{g}_1(\mathbf{w})\mathbf{g}_1^\top(\mathbf{w})$ . By (3.2), the A- and D-optimal designs minimize the loss functions

$$\phi_1(\mathbf{w}) = \text{tr}((\mathbf{A}(\mathbf{w}))^{-1}) \quad \text{and} \quad \phi_2(\mathbf{w}) = \det((\mathbf{A}(\mathbf{w}))^{-1})\tag{3.7}$$

over  $\mathbf{w}$  satisfying the conditions in (3.5), respectively. If  $\mathbf{A}(\mathbf{w})$  is singular,  $\phi_1(\mathbf{w})$  and  $\phi_2(\mathbf{w})$  are defined to be  $+\infty$ . The A- and D-optimal designs are denoted by  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$ , respectively. For nonlinear models, since optimal designs often depend on the unknown parameter  $\boldsymbol{\theta}_0$ , they are called locally optimal designs. Since all the elements of  $\mathbf{g}_1$  and  $\mathbf{G}_2$  in (3.6) are linear functions of  $\mathbf{w}$ , we have the following results from Bose and Mukerjee (2015).

**Lemma 3.1.**  *$\phi_1(\mathbf{w})$  and  $\log(\phi_2(\mathbf{w}))$  are convex functions of  $\mathbf{w}$ .*

For a discrete distribution on  $S_N$ , define

$$\mathbf{B}(\mathbf{w}) = \begin{pmatrix} 1 & \sqrt{t} \mathbf{g}_1^\top(\mathbf{w}) \\ \sqrt{t} \mathbf{g}_1(\mathbf{w}) & \mathbf{G}_2(\mathbf{w}) \end{pmatrix},\tag{3.8}$$

so all the elements of  $\mathbf{B}(\mathbf{w})$  are linear functions of  $\mathbf{w}$ . Gao and Zhou (2015) derived an alternative expression for the D-optimality criterion in Lemma 3.2.

**Lemma 3.2.** *The D-optimal design based on the SLSE minimizes  $1/\det(\mathbf{B}(\mathbf{w}))$ , and  $-\log(\det(\mathbf{B}(\mathbf{w})))$  and  $-(\det(\mathbf{B}(\mathbf{w})))^{1/(q+1)}$  are convex functions of  $\mathbf{w}$ .*

From (3.7) and Lemma 3.2,  $\xi^D(\mathbf{x})$  minimizes  $1/\det(\mathbf{A}(\mathbf{w}))$  or  $1/\det(\mathbf{B}(\mathbf{w}))$ . In fact,  $\det(\mathbf{A}(\mathbf{w})) = \det(\mathbf{B}(\mathbf{w}))$ . It is easier to use  $\mathbf{B}(\mathbf{w})$  to develop numerical algorithms for computing the optimal designs in Section 3.4.

### 3.3 Properties of $\xi^A(\mathbf{x})$

Let  $T$  be a one-to-one transformation defined on  $S_N$  with  $T^2\mathbf{u} = \mathbf{u}$  for any  $\mathbf{u} \in S_N$ . We say that the design space  $S_N$  is invariant under the transformation  $T$  or that  $S_N$  is  $T$ -invariant. If a distribution  $\xi(\mathbf{x})$  on a  $T$ -invariant  $S_N$  satisfies  $P(\mathbf{x} = \mathbf{u}_i) = P(\mathbf{x} = T\mathbf{u}_i)$ , for  $i = 1, \dots, N$ , we say the distribution is invariant under the transformation  $T$ , or  $T$ -invariant. To derive transformation invariance properties of  $\xi^A(\mathbf{x})$ , we order the points in  $S_N$  such that  $T\mathbf{u}_i = \mathbf{u}_{N-i+1}$  for  $i = 1, \dots, m$  and, if  $m < N/2$ ,  $T\mathbf{u}_i = \mathbf{u}_i$  for  $i = m+1, \dots, N-m$ . Here the points  $\mathbf{u}_{m+1}, \dots, \mathbf{u}_{N-m}$  are fixed under  $T$ . If  $m = N/2$ , then there are no fixed points in  $S_N$ . For  $T$ -invariant  $\xi(\mathbf{x})$ , the weights satisfy  $w_i = w_{N-i+1}$  for  $i = 1, \dots, m$ . To partially reverse the order of the elements in  $\mathbf{w}$ , set

$$\text{rev}(\mathbf{w}) = \left( \underbrace{w_N, \dots, w_{N-m+1}}_{\text{reversed}}, \underbrace{w_{m+1}, \dots, w_{N-m}}_{\text{fixed}}, \underbrace{w_m, \dots, w_1}_{\text{reversed}} \right)^\top.$$

**Theorem 3.1.** *Suppose  $\xi^A(\mathbf{x})$  is an  $A$ -optimal design for a regression model on a  $T$ -invariant  $S_N$ . If the weight vector of  $\xi^A(\mathbf{x})$ ,  $\mathbf{w}^A = (w_1^A, \dots, w_N^A)$ , satisfies*

$$\text{tr} \left( (\mathbf{A}(\mathbf{w}^A))^{-1} \right) = \text{tr} \left( (\mathbf{A}(\text{rev}(\mathbf{w}^A)))^{-1} \right), \quad (3.9)$$

*then there exists an  $A$ -optimal design that is invariant under the transformation  $T$ .*

**Proof of Theorem 3.1:** Let  $I_1 = \{1, \dots, m, (N-m+1), \dots, N\}$  and  $I_2 = \{(m+1), \dots, (N-m)\}$ . If  $m = N/2$ ,  $I_2$  is an empty set. Using  $\xi^A(\mathbf{x})$ , we define a distribution  $\xi^\lambda(\mathbf{x})$  having weight vector  $\mathbf{w}(\lambda)$  with elements  $w_i(\lambda) = (1-\lambda)w_i^A + \lambda w_{N+1-i}^A$  for  $i \in I_1$  and  $w_i(\lambda) = w_i^A$  for  $i \in I_2$ , for  $\lambda \in [0, 1]$ . Since  $S_N$  is  $T$ -invariant, it is obvious that distribution  $\xi^{0.5}(\mathbf{x})$  is  $T$ -invariant.

We show that  $\phi_1(\mathbf{w}(\lambda)) \leq \phi_1(\mathbf{w}^A)$ , where  $\phi_1$  is defined in (3.7). For fixed weight  $\mathbf{w}^A$ , the elements of  $\mathbf{w}(\lambda)$  are linear functions of  $\lambda$ . From Lemma 3.1,  $\phi_1(\mathbf{w})$  is a convex function of  $\mathbf{w}$ , so  $\phi_1(\mathbf{w}(\lambda))$  is a convex function of  $\lambda$ . Notice that  $\mathbf{w}(0) = \mathbf{w}^A$  and  $\mathbf{w}(1) = \text{rev}(\mathbf{w}^A)$ . By (3.7) and (3.9), we have  $\phi_1(\mathbf{w}(0)) = \phi_1(\mathbf{w}(1))$ . Using the convex property, we get

$$\phi_1(\mathbf{w}(\lambda)) \leq (1-\lambda)\phi_1(\mathbf{w}(0)) + \lambda\phi_1(\mathbf{w}(1)) = \phi_1(\mathbf{w}(0)) = \phi_1(\mathbf{w}^A).$$

Since  $\xi^A(\mathbf{x})$  minimizes  $\phi_1(\mathbf{w})$ , we must have  $\phi_1(\mathbf{w}(\lambda)) = \phi_1(\mathbf{w}^A)$ , for all  $\lambda \in [0, 1]$ . This implies that  $\xi^\lambda(\mathbf{x})$  is also an A-optimal design. Thus, there exists an A-optimal design  $\xi^{0.5}(\mathbf{x})$  that is  $T$ -invariant.  $\square$

The condition in (3.9) requires that one know the weights of an A-optimal design that can be hard to derive analytically. The next theorem gives two sufficient conditions to check for the condition.

**Theorem 3.2.** *The condition in (3.9) holds if one of the following conditions hold:*

- (i) *there exists a  $q \times q$  constant matrix  $\mathbf{Q}$  with  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_q$  (identity matrix) such that  $\mathbf{f}(T\mathbf{x}; \boldsymbol{\theta}_0) = \mathbf{Q} \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}_0)$  for all  $\mathbf{x} \in S_N$ ;*
- (ii) *there exists a  $q \times q$  matrix  $\mathbf{U}$  satisfying  $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_q$  such that  $\mathbf{g}_1(\text{rev}(\mathbf{w})) = \mathbf{U} \mathbf{g}_1(\mathbf{w})$  and  $\mathbf{G}_2(\text{rev}(\mathbf{w})) = \mathbf{U} \mathbf{G}_2(\mathbf{w}) \mathbf{U}^\top$  for any  $\mathbf{w}$ .*

**Proof of Theorem 3.2:** (i) For  $T$ -invariant  $S_N$ , we have  $T\mathbf{u}_i = \mathbf{u}_{N+1-i}$  for  $i \in I_1$  and  $T\mathbf{u}_i = \mathbf{u}_i$  for  $i \in I_2$ . If there exists a  $q \times q$  constant matrix  $\mathbf{Q}$  with  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_q$  such that  $\mathbf{f}(T\mathbf{x}; \boldsymbol{\theta}_0) = \mathbf{Q} \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}_0)$  for all  $\mathbf{x} \in S_N$ , we have, from (3.6),

$$\begin{aligned} \mathbf{g}_1(\text{rev}(\mathbf{w})) &= \sum_{i \in I_1} w_{N+1-i} \mathbf{f}(\mathbf{u}_i; \boldsymbol{\theta}_0) + \sum_{i \in I_2} w_i \mathbf{f}(\mathbf{u}_i; \boldsymbol{\theta}_0) \\ &= \sum_{i \in I_1} w_i \mathbf{f}(\mathbf{u}_{N+1-i}; \boldsymbol{\theta}_0) + \sum_{i \in I_2} w_i \mathbf{f}(\mathbf{u}_i; \boldsymbol{\theta}_0) \\ &= \sum_{i=1}^N w_i \mathbf{f}(T\mathbf{u}_i; \boldsymbol{\theta}_0) = \sum_{i=1}^N w_i \mathbf{Q} \mathbf{f}(\mathbf{u}_i; \boldsymbol{\theta}_0) = \mathbf{Q} \mathbf{g}_1(\mathbf{w}), \end{aligned}$$

and similarly,  $\mathbf{G}_2(\text{rev}(\mathbf{w})) = \mathbf{Q} \mathbf{G}_2(\mathbf{w}) \mathbf{Q}^\top$ . Since  $\mathbf{A}(\mathbf{w}) = \mathbf{G}_2(\mathbf{w}) - t\mathbf{g}_1(\mathbf{w})\mathbf{g}_1^\top(\mathbf{w})$ , it is clear that  $\mathbf{A}(\text{rev}(\mathbf{w})) = \mathbf{Q} \mathbf{A}(\mathbf{w}) \mathbf{Q}^\top$ . Thus,

$$\begin{aligned} \text{tr}((\mathbf{A}(\text{rev}(\mathbf{w})))^{-1}) &= \text{tr}\left((\mathbf{Q} \mathbf{A}(\mathbf{w}) \mathbf{Q}^\top)^{-1}\right) = \text{tr}\left((\mathbf{Q}^\top)^{-1} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{Q}^{-1}\right) \\ &= \text{tr}((\mathbf{A}(\mathbf{w}))^{-1}), \text{ from } \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_q, \end{aligned}$$

which implies that the condition in (3.9) holds.

- (ii) The proof is similar to that in part (i) and is omitted.  $\square$

The conditions in Theorem 3.2 are easy to verify, especially condition (i). The results in Theorems 3.1 and 3.2 can be applied to both linear and nonlinear models.

For some regression models, the transformation invariance property implies the symmetry of  $\xi^A(\mathbf{x})$ .

**Example 3.1.** For the second-order regression model with independent variables  $x_1$  and  $x_2$ ,  $y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \epsilon$ , we study the symmetry of A-optimal designs for the design spaces

$$S_{9,1} = \{(1, 0), (-1, 0), (0, 1), (0, -1), (1, 1), (-1, 1), (1, -1), (-1, -1), (0, 0)\},$$

$$S_{9,2} = \{(\sqrt{2}, 0), (-\sqrt{2}, 0), (0, \sqrt{2}), (0, -\sqrt{2}), (1, 1), (-1, 1), (1, -1), (-1, -1), (0, 0)\}.$$

Except for the center point  $(0, 0)$ , the points in  $S_{9,1}$  are located on the edges of a square while the points in  $S_{9,2}$  are on a circle with radius  $\sqrt{2}$ . These spaces are invariant under several transformations, including

$$\begin{aligned} T_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix}, & T_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} x_1 \\ -x_2 \end{pmatrix}, \\ T_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} -x_1 \\ -x_2 \end{pmatrix}, & T_4 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}. \end{aligned}$$

Transformation  $T_3$  can be viewed as the combination of  $T_1$  and  $T_2$ . It is easy to show that there exists an A-optimal design that is invariant under  $T_1$ ,  $T_2$ , or  $T_3$ . For  $T_4$ , we have  $\mathbf{f}(T_4 \mathbf{x}; \boldsymbol{\theta}) = \mathbf{Q} \mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$  with

$$\mathbf{Q} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \oplus 1,$$

where  $\oplus$  is the matrix direct sum. It is clear that  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_5$ . Thus, there exists an A-optimal design that is invariant under  $T_4$ . If we apply the four transformations sequentially and use the results in Theorems 3.1 and 3.2, there exists an A-optimal design that is invariant under all the four transformations. This implies that there exists an A-optimal design  $\xi^A(\mathbf{x})$  on  $S_{9,1}$  (or  $S_{9,2}$ ) having  $w_1^A = w_2^A = w_3^A = w_4^A$  and  $w_5^A = w_6^A = w_7^A = w_8^A$ .  $\square$

**Example 3.2.** Consider a nonlinear model,  $y_i = \theta_1 x / (x^2 + \theta_2) + \epsilon$ ,  $\theta_1 \neq 0$ ,  $\theta_2 \neq 0$ , on the design space  $S_N \subset [-a, a]$ , invariant under transformation  $Tx = -x$ . Here  $\mathbf{f}(x; \boldsymbol{\theta}) = (x / (x^2 + \theta_2), -\theta_1 x / ((x^2 + \theta_2)^2))^\top$ , and it is easy to verify that  $\mathbf{f}(Tx; \boldsymbol{\theta}_0) = \mathbf{Q} \mathbf{f}(x; \boldsymbol{\theta}_0)$  with  $\mathbf{Q} = \text{diag}(-1, -1)$  and  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_2$ . Thus, there exists an A-optimal design that is symmetric about zero.  $\square$

The results in Theorems 3.1 and 3.2 can be extended easily to D-optimal designs  $\xi^D(\mathbf{x})$  by changing  $\text{tr}()$  to  $\det()$  in (3.9). By applying the result in Theorem 3.1, we can reduce the number of unknown weights in the loss functions  $\phi_1(\mathbf{w})$  and  $\phi_2(\mathbf{w})$  in (3.7). For instance, in Example 3.1, the number of unknown weights is reduced to 3.

From Lemma 3.2, an alternative expression for  $\phi_2(\mathbf{w})$  is  $\phi_2(\mathbf{w}) = \det((\mathbf{B}(\mathbf{w}))^{-1})$ , since  $\det((\mathbf{A}(\mathbf{w}))^{-1}) = \det((\mathbf{B}(\mathbf{w}))^{-1})$ . For  $\phi_1(\mathbf{w})$ , we do not have  $\text{tr}((\mathbf{A}(\mathbf{w}))^{-1}) = \text{tr}((\mathbf{B}(\mathbf{w}))^{-1})$ , but we can also characterize the A-optimality criterion using  $\mathbf{B}(\mathbf{w})$ , in Theorem 3.3.

**Theorem 3.3.** *If  $\mathbf{G}_2(\mathbf{w})$  in (3.8) is nonsingular, then  $\phi_1(\mathbf{w}) = \text{tr}((\mathbf{A}(\mathbf{w}))^{-1}) = \text{tr}(\mathbf{C}(\mathbf{B}(\mathbf{w}))^{-1})$ , where  $\mathbf{C} = 0 \oplus \mathbf{I}_q$  is a  $(q+1) \times (q+1)$  matrix.*

**Proof of Theorem 3.3:** From equation (3.8),

$$\mathbf{B}(\mathbf{w}) = \begin{pmatrix} 1 & \sqrt{t} \mathbf{g}_1^\top(\mathbf{w}) \\ \sqrt{t} \mathbf{g}_1(\mathbf{w}) & \mathbf{G}_2(\mathbf{w}) \end{pmatrix},$$

it is easy to verify that, for nonsingular  $\mathbf{G}_2(\mathbf{w})$ ,

$$(\mathbf{B}(\mathbf{w}))^{-1} = \begin{pmatrix} 1/b & -\frac{\sqrt{t}}{b} \mathbf{g}_1^\top(\mathbf{w}) (\mathbf{G}_2(\mathbf{w}))^{-1} \\ -\frac{\sqrt{t}}{b} (\mathbf{G}_2(\mathbf{w}))^{-1} \mathbf{g}_1(\mathbf{w}) & (\mathbf{G}_2(\mathbf{w}) - t \mathbf{g}_1(\mathbf{w}) \mathbf{g}_1^\top(\mathbf{w}))^{-1} \end{pmatrix},$$

where  $b = 1 - t \mathbf{g}_1^\top(\mathbf{w}) (\mathbf{G}_2(\mathbf{w}))^{-1} \mathbf{g}_1(\mathbf{w})$ . With  $\mathbf{C} = 0 \oplus \mathbf{I}_q$ , we have

$$\text{tr}(\mathbf{C}(\mathbf{B}(\mathbf{w}))^{-1}) = \text{tr}\left(\left(\mathbf{G}_2(\mathbf{w}) - t \mathbf{g}_1(\mathbf{w}) \mathbf{g}_1^\top(\mathbf{w})\right)^{-1}\right) = \text{tr}((\mathbf{A}(\mathbf{w}))^{-1}).$$

$\square$

This characterization of the A-optimality criterion is useful for developing an efficient algorithm for computing A-optimal designs. If we are interested in a subset of the model parameters, the criterion can be easily modified. Let  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_2^\top)^\top$ , where  $\boldsymbol{\theta}_1 \in R^{q_1}$  and  $\boldsymbol{\theta}_2 \in R^{q_2}$  with  $q_1 + q_2 = q$ . The A-optimal design based on the SLSE of  $\boldsymbol{\theta}_2$  minimizes  $\phi_3(\mathbf{w}) = \text{tr}(\mathbf{C}_1(\mathbf{B}(\mathbf{w}))^{-1})$ , where  $\mathbf{C}_1 = \mathbf{0}_{q_1+1} \oplus \mathbf{I}_{q_2}$  and  $\mathbf{0}_{q_1+1}$  is a  $(q_1 + 1) \times (q_1 + 1)$  matrix of zeros.

## 3.4 Numerical algorithms

For some regression models,  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$  can be constructed analytically. Examples are given in Gao and Zhou (2014) and Bose and Mukerjee (2015). In general, it is hard to find the optimal designs analytically, so numerical algorithms are developed. After reviewing the algorithms in Bose and Mukerjee (2015), we propose efficient algorithms for computing  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$ . These algorithms do not use the derivatives of the loss functions. Yang *et al.* (2013) proposed another efficient algorithm for computing optimal designs, which needs the calculation of the first and second derivatives of objective functions.

### 3.4.1 Multiplicative algorithms

Bose and Mukerjee (2015) proposed multiplicative algorithms to compute  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$ . For simplicity, we write  $\mathbf{f}(\mathbf{u})$  for  $\mathbf{f}(\mathbf{u}; \boldsymbol{\theta}_0)$ . Define, for  $i = 1, \dots, N$ ,

$$\begin{aligned}\psi_{Ai}(\mathbf{w}) &= (1-t)\mathbf{f}^\top(\mathbf{u}_i)\mathbf{A}^{-2}\mathbf{f}(\mathbf{u}_i) + t(\mathbf{f}(\mathbf{u}_i) - \mathbf{g}_1(\mathbf{w}))^\top \mathbf{A}^{-2}(\mathbf{f}(\mathbf{u}_i) - \mathbf{g}_1(\mathbf{w})), \\ \psi_{Di}(\mathbf{w}) &= (1-t)\mathbf{f}^\top(\mathbf{u}_i)\mathbf{A}^{-1}\mathbf{f}(\mathbf{u}_i) + t(\mathbf{f}(\mathbf{u}_i) - \mathbf{g}_1(\mathbf{w}))^\top \mathbf{A}^{-1}(\mathbf{f}(\mathbf{u}_i) - \mathbf{g}_1(\mathbf{w})),\end{aligned}\quad (3.10)$$

where  $\mathbf{A}^{-1} = (\mathbf{A}(\mathbf{w}))^{-1}$  and  $\mathbf{A}^{-2} = (\mathbf{A}(\mathbf{w}))^{-1}(\mathbf{A}(\mathbf{w}))^{-1}$ .

Start with the uniform weight vector,  $\mathbf{w}^{(0)} = (1/N, \dots, 1/N)^\top$ . For  $\xi^A(\mathbf{x})$ , the multiplicative algorithm finds  $\mathbf{w}^{(j)}$ ,  $j = 1, 2, \dots$ , iteratively as

$$w_i^{(j)} = w_i^{(j-1)} \psi_{Ai}(\mathbf{w}^{(j-1)}) / \text{tr}(\mathbf{A}(\mathbf{w}^{(j-1)}))^{-1},$$

for  $i = 1, \dots, N$ , till  $\mathbf{w}^{(j)}$  satisfies

$$\psi_{A_i}(\mathbf{w}^{(j)}) - \text{tr}(\mathbf{A}(\mathbf{w}^{(j)}))^{-1} \leq \delta, \quad \text{for } i = 1, \dots, N, \quad (3.11)$$

for some prespecified small  $\delta (> 0)$ . Similarly, for  $\xi^D(\mathbf{x})$ , the algorithm finds  $\mathbf{w}^{(j)}$  iteratively as  $w_i^{(j)} = w_i^{(j-1)} \psi_{D_i}(\mathbf{w}^{(j-1)})/q$ , for  $i = 1, \dots, N$ , till  $\mathbf{w}^{(j)}$  satisfies

$$\psi_{D_i}(\mathbf{w}^{(j)}) - q \leq \delta, \quad \text{for } i = 1, \dots, N. \quad (3.12)$$

Conditions in (3.11) and (3.12) are approximated from necessary and sufficient conditions for the  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$  in Bose and Mukerjee (2015), which are stated below.

**Lemma 3.3.** *The weight vector  $\mathbf{w}$  is*

- (a) *A-optimal if and only if  $\mathbf{A}(\mathbf{w})$  is nonsingular and  $\psi_{A_i}(\mathbf{w}) \leq \text{tr}(\mathbf{A}(\mathbf{w}))^{-1}$ , for  $i = 1, \dots, N$ ,*
- (b) *D-optimal if and only if  $\mathbf{A}(\mathbf{w})$  is nonsingular and  $\psi_{D_i}(\mathbf{w}) \leq q$ , for  $i = 1, \dots, N$ .*

These algorithms can preserve the transformation invariance property for the weights at each iteration, if there exist transformation invariant  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$ .

**Theorem 3.4.** *Suppose the design space  $S_N$  is invariant under a transformation  $T$ . If there exists a  $q \times q$  constant matrix  $\mathbf{Q}$  with  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_q$  such that  $\mathbf{f}(T\mathbf{x}; \boldsymbol{\theta}_0) = \mathbf{Q} \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}_0)$  for all  $\mathbf{x} \in S_N$ , then the weights from the multiplicative algorithms satisfy  $rev(\mathbf{w}^{(j)}) = \mathbf{w}^{(j)}$  for all  $j = 0, 1, 2, \dots$ .*

**Proof of Theorem 3.4:** We use mathematical induction to prove the result. Since the design space  $S_N$  is  $T$ -invariant, the points in  $S_N$  are ordered such that  $T\mathbf{u}_{N-i+1} = \mathbf{u}_i$  for  $i = 1, \dots, m, (N - m + 1), \dots, N$  and  $T\mathbf{u}_i = \mathbf{u}_i$  for  $i = (m + 1), \dots, (N - m)$  (if  $m < N/2$ ).

(1) In the algorithms  $\mathbf{w}^{(0)} = (1/N, \dots, 1/N)^\top$ , so it is obvious that  $rev(\mathbf{w}^{(0)}) = \mathbf{w}^{(0)}$ .

(2) Assume  $rev(\mathbf{w}^{(j-1)}) = \mathbf{w}^{(j-1)}$ . We want to show that  $rev(\mathbf{w}^{(j)}) = \mathbf{w}^{(j)}$ . From

the proof of Theorem 3.3, if  $\mathbf{f}(T\mathbf{x}; \boldsymbol{\theta}_0) = \mathbf{Q} \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}_0)$  for all  $\mathbf{x} \in S_N$ , then

$$\begin{aligned} \mathbf{g}_1(\mathbf{w}^{(j-1)}) &= \mathbf{g}_1(\text{rev}(\mathbf{w}^{(j-1)})) = \mathbf{Q} \mathbf{g}_1(\mathbf{w}^{(j-1)}), \\ \mathbf{G}_2(\mathbf{w}^{(j-1)}) &= \mathbf{G}_2(\text{rev}(\mathbf{w}^{(j-1)})) = \mathbf{Q} \mathbf{G}_2(\mathbf{w}^{(j-1)}) \mathbf{Q}^\top, \\ \mathbf{A}(\mathbf{w}^{(j-1)}) &= \mathbf{A}(\text{rev}(\mathbf{w}^{(j-1)})) = \mathbf{Q} \mathbf{A}(\mathbf{w}^{(j-1)}) \mathbf{Q}^\top. \end{aligned}$$

Thus, from equation (3.10), we have  $\psi_{A_i}(\mathbf{w}^{(j-1)}) = \psi_{A(N-i+1)}(\mathbf{w}^{(j-1)})$  and  $\psi_{D_i}(\mathbf{w}^{(j-1)}) = \psi_{D(N-i+1)}(\mathbf{w}^{(j-1)})$  for  $i = 1, \dots, m$ . For the A-optimal design, we get

$$\begin{aligned} w_i^{(j)} &= w_i^{(j-1)} \psi_{A_i}(\mathbf{w}^{(j-1)}) / \text{tr}(\mathbf{A}(\mathbf{w}^{(j-1)}))^{-1} \\ &= w_{N-i+1}^{(j-1)} \psi_{A(N-i+1)}(\mathbf{w}^{(j-1)}) / \text{tr}(\mathbf{A}(\mathbf{w}^{(j-1)}))^{-1} \\ &= w_{N-i+1}^{(j)}, \quad \text{for } i = 1, \dots, m. \end{aligned}$$

Therefore  $\text{rev}(\mathbf{w}^{(j)}) = \mathbf{w}^{(j)}$ . Similarly, it is true for the D-optimal design.  $\square$

This result depends on the fact that the initial weight vector satisfies  $\text{rev}(\mathbf{w}^{(0)}) = \mathbf{w}^{(0)}$ .

### 3.4.2 Convex optimization algorithms

The CVX program in MATLAB (Grant and Boyd, (2013)) is powerful and widely used to solve convex optimization problems. Gao and Zhou (2015) applied the CVX program to find the D-optimal designs based on the SLSE through the moments of distribution  $\xi(\mathbf{x})$ . The optimal design problems for  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$  on a discrete design space can be formulated as convex optimization problems, differently from those in Gao and Zhou (2015). Using  $w_N = 1 - \sum_{i=1}^{N-1} w_i$ , we define a weight vector having  $N - 1$  weights as  $\tilde{\mathbf{w}} = \left( w_1, w_2, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i \right)^\top$ . Let  $\mathbf{D}(\tilde{\mathbf{w}}) = \text{diag} \left( w_1, w_2, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i \right)$  be a diagonal matrix. One has  $\phi_1(\mathbf{w}) = \phi_1(\tilde{\mathbf{w}})$  and  $\phi_2(\mathbf{w}) = \phi_2(\tilde{\mathbf{w}})$ , and  $\phi_1(\tilde{\mathbf{w}})$  and  $\log(\phi_2(\tilde{\mathbf{w}}))$  are convex functions of  $\tilde{\mathbf{w}}$ . The conditions in (3.5) are equivalent to that  $\mathbf{D}(\tilde{\mathbf{w}}) \succeq 0$ . Thus, the A- and D-optimal design problems become, respectively,

$$\begin{cases} \min_{\tilde{\mathbf{w}}} \phi_1(\tilde{\mathbf{w}}), \\ \text{subject to: } \mathbf{D}(\tilde{\mathbf{w}}) \succeq 0, \end{cases} \quad (3.13)$$

$$\begin{cases} \min_{\tilde{\mathbf{w}}} \log(\phi_2(\tilde{\mathbf{w}})), \\ \text{subject to: } \mathbf{D}(\tilde{\mathbf{w}}) \succeq 0. \end{cases} \quad (3.14)$$

The CVX program in MATLAB has some technical issues. In (3.14), we need to use  $\mathbf{B}(\tilde{\mathbf{w}})$  in  $\phi_2(\tilde{\mathbf{w}})$ , and the CVX program works well to solve  $\min_{\tilde{\mathbf{w}}} -\log(\det(\mathbf{B}(\tilde{\mathbf{w}})))$  or  $\min_{\tilde{\mathbf{w}}} -(\det(\mathbf{B}(\tilde{\mathbf{w}})))^{1/(q+1)}$ . In (3.13), however, it does not work to use  $\mathbf{A}(\tilde{\mathbf{w}})$  in  $\phi_1(\tilde{\mathbf{w}})$ , and it is not straightforward to use  $\mathbf{B}(\tilde{\mathbf{w}})$ . We develop a novel formulation of the A-optimal design problem with a linear objective function and linear matrix inequality constraints that is an SDP problem.

Let  $\mathbf{e}_i$  be the  $i$ th unit vector in  $R^{q+1}$ ,  $i = 1, \dots, q+1$ ,  $\mathbf{v} = (v_2, \dots, v_{q+1})^\top$ , and

$$\begin{aligned} \mathbf{B}_i &= \begin{pmatrix} \mathbf{B}(\tilde{\mathbf{w}}) & \mathbf{e}_i \\ \mathbf{e}_i^\top & v_i \end{pmatrix}, \quad \text{for } i = 2, \dots, q+1, \\ \mathbf{H}(\tilde{\mathbf{w}}, \mathbf{v}) &= \mathbf{B}_2 \oplus \dots \oplus \mathbf{B}_{q+1} \oplus \mathbf{D}(\tilde{\mathbf{w}}). \end{aligned} \quad (3.15)$$

Since  $\mathbf{B}(\tilde{\mathbf{w}})$  and  $\mathbf{D}(\tilde{\mathbf{w}})$  are linear matrices in  $\tilde{\mathbf{w}}$ ,  $\mathbf{H}(\tilde{\mathbf{w}}, \mathbf{v})$  is a linear matrix in  $\tilde{\mathbf{w}}$  and  $\mathbf{v}$ . Then  $\xi^A(\mathbf{x})$  can be solved through

$$\begin{cases} \min_{\tilde{\mathbf{w}}, \mathbf{v}} \sum_{i=2}^{q+1} v_i, \\ \text{subject to: } \mathbf{H}(\tilde{\mathbf{w}}, \mathbf{v}) \succeq 0, \end{cases} \quad (3.16)$$

**Theorem 3.5.** *The solutions to the optimization problems (3.13) and (3.16) satisfy*

(i) *if  $\tilde{\mathbf{w}}^*$  is a solution to (3.13), then  $(\tilde{\mathbf{w}}^*, \mathbf{v}^*)$  is a solution to (3.16) with*

$$\mathbf{v}^* = (\mathbf{e}_2^\top (\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1} \mathbf{e}_2, \dots, \mathbf{e}_{q+1}^\top (\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1} \mathbf{e}_{q+1})^\top,$$

(ii) *if  $(\tilde{\mathbf{w}}^*, \mathbf{v}^*)$  is a solution to (3.16), then  $\tilde{\mathbf{w}}^*$  is a solution to (3.13).*

**Proof of Theorem 3.5:** (i) If  $\tilde{\mathbf{w}}^*$  is a solution to (3.13), then  $\mathbf{A}(\tilde{\mathbf{w}}^*) \succ 0$  (positive definite) and  $\mathbf{B}(\tilde{\mathbf{w}}^*) \succ 0$  by (3.7) and Theorem 3.3. Let

$$\mathbf{v}^* = (v_2^*, \dots, v_{q+1}^*)^\top = (\mathbf{e}_2^\top (\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1} \mathbf{e}_2, \dots, \mathbf{e}_{q+1}^\top (\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1} \mathbf{e}_{q+1})^\top.$$

Then, from (3.15)  $\mathbf{B}_i \succeq 0$ , for  $i = 2, \dots, q+1$  and the constraint in (3.16) is satisfied.

For any  $\tilde{\mathbf{w}}$  satisfying  $\mathbf{D}(\tilde{\mathbf{w}}) \succeq 0$  and  $\mathbf{B}(\tilde{\mathbf{w}}) \succ 0$ , we get

$$\begin{aligned}
\sum_{i=2}^{q+1} v_i^* &= \sum_{i=2}^{q+1} \mathbf{e}_i^\top (\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1} \mathbf{e}_i \\
&= \text{tr}(\mathbf{C}(\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1}) \\
&= \phi_1(\tilde{\mathbf{w}}^*), \quad \text{from Theorem 3.3,} \\
&\leq \phi_1(\tilde{\mathbf{w}}), \quad \text{from } \tilde{\mathbf{w}}^* \text{ being a solution to problem (3.13),} \\
&= \sum_{i=2}^{q+1} \mathbf{e}_i^\top (\mathbf{B}(\tilde{\mathbf{w}}))^{-1} \mathbf{e}_i \\
&\leq \sum_{i=2}^{q+1} v_i, \quad \text{from } \mathbf{B}_i \succeq 0,
\end{aligned}$$

which implies that  $(\tilde{\mathbf{w}}^*, \mathbf{v}^*)$  is a solution to (3.16).

(ii) Suppose that  $(\tilde{\mathbf{w}}^*, \mathbf{v}^*)$  is a solution to (3.16). Since  $\mathbf{B}_i \succeq 0$ , we must have  $\mathbf{B}(\tilde{\mathbf{w}}^*) \succ 0$ . For any  $\tilde{\mathbf{w}}$  satisfying  $\mathbf{D}(\tilde{\mathbf{w}}) \succeq 0$  and  $\mathbf{B}(\tilde{\mathbf{w}}) \succ 0$ , we have

$$\begin{aligned}
\phi_1(\tilde{\mathbf{w}}^*) &= \text{tr}(\mathbf{C}(\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1}), \text{ from Theorem 3.3,} \\
&= \sum_{i=2}^{q+1} \mathbf{e}_i^\top (\mathbf{B}(\tilde{\mathbf{w}}^*))^{-1} \mathbf{e}_i \\
&\leq \sum_{i=2}^{q+1} v_i^*, \quad \text{from } \mathbf{B}_i \succeq 0, \\
&\leq \sum_{i=2}^{q+1} v_i, \quad \text{from } \tilde{\mathbf{v}}^* \text{ being a solution to problem (3.16),} \\
&= \sum_{i=2}^{q+1} \mathbf{e}_i^\top (\mathbf{B}(\tilde{\mathbf{w}}))^{-1} \mathbf{e}_i, \quad \text{by choosing } v_i = \mathbf{e}_i^\top (\mathbf{B}(\tilde{\mathbf{w}}))^{-1} \mathbf{e}_i, \\
&= \phi_1(\tilde{\mathbf{w}}).
\end{aligned}$$

Thus,  $\tilde{\mathbf{w}}^*$  is a solution to (3.13). □

To solve (3.16), the SeDuMi program in MATLAB can be used. See Sturm (1999) for a user's guide. Using the multiplicative algorithms and the CVX and SeDuMi programs, we can compute  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$  for linear and nonlinear models with any discrete design space  $S_N$ . Conditions in (3.11) and (3.12) are also useful to verify that the numerical solutions from the CVX and SeDuMi programs are A- and D-optimal

designs.

## 3.5 Applications and efficiencies

We compute  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$  for various linear and nonlinear models and give representative results. The A-optimal designs are computed by the multiplicative algorithm and the SeDuMi program, while the D-optimal designs are computed by the multiplicative algorithm and the CVX program. MATLAB codes are provided for some examples in the Appendix. The conditions in (3.11) and (3.12) are used to verify that the numerical solutions are A- and D-optimal designs, respectively. Numerical algorithms are compared, and efficiencies of the SLSE and its optimal designs are discussed. A property of locally optimal designs is also derived for nonlinear models.

### 3.5.1 Examples

**Example 3.3.** Consider the regression model and design spaces in Example 3.1 and compute  $\xi^A(\mathbf{x})$  and  $\xi^D(\mathbf{x})$  for various values of  $t$ . Since the number of points in design spaces  $S_{9,1}$  and  $S_{9,2}$  is small, all the algorithms work well and quickly. We computed the weights,  $w_1, \dots, w_9$ . The results from the multiplicative algorithms are the same as those from the CVX and SeDuMi programs, and the weights have the transformation invariance property discussed in Example 3.1. Representative results are given in Table 3.1, where only three weights,  $w_1, w_5, w_9$ , are listed due to the invariance property. Figure 3.1 shows some A- and D-optimal designs for this example. The results indicate that the optimal designs depend on the value of  $t$ . For small  $t$  the center point has weight zero for all the optimal designs, but for  $t = 0.9$  the center point has a positive weight for three optimal designs.  $\square$

For linear models, the optimal designs do not depend on  $\boldsymbol{\theta}$ . If there is an intercept term in the model, the optimal designs  $\xi^A$  and  $\xi^D$  are the same as those based on the LSE (Gao and Zhou, (2014)). For nonlinear models, the optimal designs usually depend on the true value,  $\boldsymbol{\theta}_0$ , and are called locally optimal designs. In practice an estimate of  $\boldsymbol{\theta}_0$  is used to construct the optimal designs. However, if a nonlinear model is linear in a subset of parameters, then optimal design  $\xi^D$  does not depend on the true value of the subset.

Table 3.1: A- and D-optimal weights,  $w_1^A, w_5^A, w_9^A, w_1^D, w_5^D, w_9^D$ , in Example 3.3

$t$	$w_1^A$	$w_5^A$	$w_9^A$	$w_1^D$	$w_5^D$	$w_9^D$
Design space $S_{9,1}$						
0	0.131	0.119	0.000	0.071	0.179	0.000
0.3	0.130	0.120	0.000	0.072	0.178	0.000
0.5	0.128	0.122	0.000	0.074	0.176	0.000
0.9	0.118	0.121	0.044	0.088	0.162	0.000
Design space $S_{9,2}$						
0	0.104	0.146	0.000	0.125	0.125	0.000
0.3	0.104	0.146	0.000	0.125	0.125	0.000
0.5	0.104	0.146	0.000	0.125	0.125	0.000
0.9	0.088	0.125	0.148	0.116	0.116	0.072

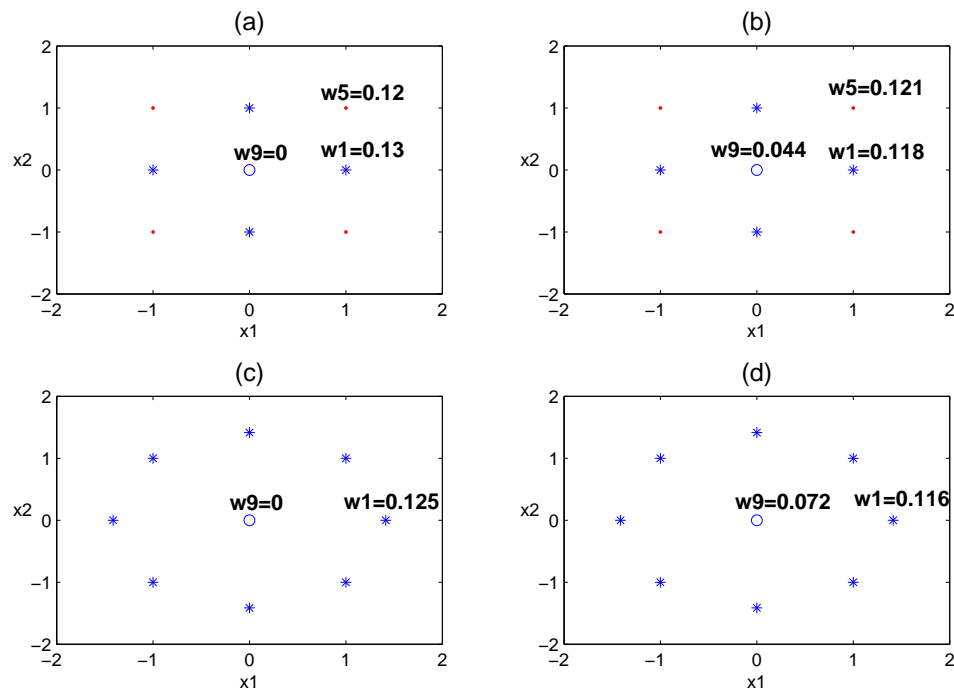


Figure 3.1: Optimal design results for Example 3.3: (a) A-optimal design on design space  $S_{9,1}$  for  $t = 0.3$ , (b) A-optimal design on design space  $S_{9,1}$  for  $t = 0.9$ , (c) D-optimal design on design space  $S_{9,2}$  for  $t = 0.3$ , (d) D-optimal design on design space  $S_{9,2}$  for  $t = 0.9$ ,

**Theorem 3.6.** Let  $\boldsymbol{\theta} = (\boldsymbol{\alpha}^\top, \boldsymbol{\beta}^\top)^\top$ , where  $\boldsymbol{\alpha} \in R^a$  and  $\boldsymbol{\beta} \in R^b$  with  $a + b = q$ . For a nonlinear model

$$g(\mathbf{x}; \boldsymbol{\theta}) = \sum_{i=1}^a \alpha_i h_i(\mathbf{x}; \boldsymbol{\beta}_i), \quad (3.17)$$

where  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_a)^\top$ ,  $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \dots, \boldsymbol{\beta}_a^\top)^\top$  with  $\boldsymbol{\beta}_i \in R^{q_i}$ , and  $h_i(\mathbf{x}; \boldsymbol{\beta}_i)$  can be linear or nonlinear in  $\boldsymbol{\beta}_i$ , then  $\xi^D(\mathbf{x})$  does not depend on  $\boldsymbol{\alpha}$ .

**Proof of Theorem 3.6:** From the equation (3.17), we have

$$\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) = \frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial g(\mathbf{x}; \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \\ \frac{\partial g(\mathbf{x}; \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \end{pmatrix} = \mathbf{Q}_\alpha \mathbf{h}(\mathbf{x}; \boldsymbol{\beta}),$$

where  $\mathbf{Q}_\alpha$  is a  $q \times q$  diagonal matrix and  $\mathbf{h}(\mathbf{x}; \boldsymbol{\beta})$  is a  $q \times 1$  vector given by,

$$\begin{aligned} \mathbf{Q}_\alpha &= \text{diag}(\underbrace{1, \dots, 1}_a, \underbrace{\alpha_1, \dots, \alpha_1}_{q_1}, \dots, \underbrace{\alpha_a, \dots, \alpha_a}_{q_a}), \\ \mathbf{h}(\mathbf{x}; \boldsymbol{\beta}) &= \left( h_1(\mathbf{x}; \boldsymbol{\beta}_1), \dots, h_a(\mathbf{x}; \boldsymbol{\beta}_a), \frac{\partial h_1(\mathbf{x}; \boldsymbol{\beta}_1)}{\partial \boldsymbol{\beta}_1^\top}, \dots, \frac{\partial h_a(\mathbf{x}; \boldsymbol{\beta}_a)}{\partial \boldsymbol{\beta}_a^\top} \right)^\top, \end{aligned}$$

respectively. By (3.6), it is clear that

$$\begin{aligned} \mathbf{g}_1(\mathbf{w}; \boldsymbol{\theta}) &= \mathbf{Q}_\alpha \mathbf{g}_1(\mathbf{w}; \boldsymbol{\theta}^1), \\ \mathbf{G}_2(\mathbf{w}; \boldsymbol{\theta}) &= \mathbf{Q}_\alpha \mathbf{G}_2(\mathbf{w}; \boldsymbol{\theta}^1) \mathbf{Q}_\alpha, \\ \mathbf{A}(\mathbf{w}; \boldsymbol{\theta}) &= \mathbf{Q}_\alpha \mathbf{A}(\mathbf{w}; \boldsymbol{\theta}^1) \mathbf{Q}_\alpha, \end{aligned}$$

where  $\boldsymbol{\theta}^1 = (\underbrace{1, \dots, 1}_a, \boldsymbol{\beta}^\top)^\top$ . Thus,  $\det(\mathbf{A}(\mathbf{w}; \boldsymbol{\theta})) = \alpha_1^{2q_1} \dots \alpha_a^{2q_a} \det(\mathbf{A}(\mathbf{w}; \boldsymbol{\theta}^1))$ , which implies that the D-optimal design does not depend on the true value of  $\boldsymbol{\alpha}$ .  $\square$

A similar result for D-optimal designs based on the LSE is in Dette *et al.* (2006). For  $\xi^A(\mathbf{x})$ , this result is not true in general. More discussion about other approaches for locally optimal designs can be found in Yang and Stufken (2012).

**Example 3.4.** The Michaelis-Menten model (Michaelis and Menten, (1913)), one of the best-known models in biochemistry, is used to study enzyme kinetics. Enzyme-kinetics studies the chemical reactions for substrate that are catalyzed by enzymes.

The relationship between the reaction rate and the concentration of the substrate can be described as  $y = \alpha x / (\beta + x) + \epsilon$ ,  $x \geq 0$ , where  $y$  represents the speed of reaction, and  $x$  is the substrate concentration. Optimal designs for this model have been studied by many authors, including Dette *et al.* (2005) and Yang and Stufken (2009). Table 3.2 lists representative results of  $\xi^A$  and  $\xi^D$  for various values of  $t$  and  $N$  for the model with  $\alpha = 1$ ,  $\beta = 1$ , and  $S_N = \{4(i-1)/(N-1) : i = 1, \dots, N\} \subset [0, 4]$ . By Theorem 3.6 the D-optimal designs do not depend on the value of  $\alpha$ , but the A-optimal designs depend on  $\alpha$  from the numerical results. For large  $N$ , the CVX and SeDuMi programs were much faster than the multiplicative algorithms. In fact, the A-optimal designs in Table 3.2 were all calculated by the SeDuMi program. The A-optimal and D-optimal designs depend on  $t$ . For small  $t$  there are two support points, while for large  $t$  there are three support points. When  $t = 0$ , the results give the optimal designs based on the LSE and they match the results shown in the website (<http://optimal-design.biostat.ucla.edu/optimal/OptimalDesign.aspx>). The number of support points for  $t = 0$  also agrees with the result in Yang and Stufken (2009).

□

**Example 3.5.** Compartmental models are nonlinear models which are widely used in pharmacokinetic research (Wise, 1985). Pharmacokinetics studies the drug performance after administration through organs. One compartment model describing the retention curve and the chemical change after the administration can be written as  $y = \alpha_1 \exp(-\beta_1 x) + \epsilon$ , where  $y$  represents the amount of drug in an organ at time  $x \geq 0$ . This is a special case of the model in Theorem 3.6 with  $a = b = 1$ . The parameters  $\alpha_1$  and  $\beta_1$  are affected by the processes of drug distribution and elimination (Weiss, 1983). A model with two homogeneous compartments can be described as (Wise, 1985; Faddy, 1993),

$$y_l = \alpha_1 e^{-\beta_1 x_l} + \alpha_2 e^{-\beta_2 x_l} + \epsilon_l, \quad l = 1, \dots, n, \quad (3.18)$$

which is a case of the model in Theorem 3.6 with  $a = b = 2$ . Since  $\alpha_1$  and  $\alpha_2$  are linear parameters, the D-optimal design does not depend on the true values of  $\alpha_1$  and  $\alpha_2$ . Dette *et al.* (2006) and Yang and Stufken (2012), among others, studied D-optimal designs for these models using the LSE. Here we construct the D-optimal design using the SLSE. MATLAB codes of both CVX program and multiplicative program to compute D-optimal designs are provided in the Appendix. Representative results are given in Table 3.3 for model (3.18) with  $\alpha_1 = \alpha_2 = 1$ ,  $\beta_1 = 0.1$ ,  $\beta_2 = 1.0$ , and

Table 3.2: A- and D-optimal design points and their weights (in parentheses) for the Michaelis-Menten model with  $\alpha = 1$  and  $\beta = 1$

$N = 101$				
$t = 0, 0.3$	$\xi^A$ :	0.520 (0.666)	4.000 (0.334)	
	$\xi^D$ :	0.680 (0.500)	4.000 (0.500)	
$t = 0.7$	$\xi^A$ :	0.640 (0.641)	4.000 (0.359)	
	$\xi^D$ :	0.000 (0.048)	0.680 (0.476)	4.000 (0.476)
$t = 0.9$	$\xi^A$ :	0.000 (0.154)	0.680 (0.536)	4.000 (0.310)
	$\xi^D$ :	0.000 (0.260)	0.680 (0.370)	4.000 (0.370)
$N = 201$				
$t = 0$	$\xi^A$ :	0.500 (0.671)	4.000 (0.329)	
	$\xi^D$ :	0.660 (0.500)	4.000 (0.500)	
$t = 0.3$	$\xi^A$ :	0.540 (0.661)	4.000 (0.339)	
	$\xi^D$ :	0.660 (0.500)	4.000 (0.500)	
$t = 0.7$	$\xi^A$ :	0.640 (0.641)	4.000 (0.359)	
	$\xi^D$ :	0.000 (0.048)	0.660 (0.476)	4.000 (0.476)
$t = 0.9$	$\xi^A$ :	0.000 (0.159)	0.660 (0.536)	4.000 (0.305)
	$\xi^D$ :	0.000 (0.260)	0.660 (0.370)	4.000 (0.370)
$N = 501$				
$t = 0$	$\xi^A$ :	0.504 (0.670)	4.000 (0.330)	
	$\xi^D$ :	0.664 (0.500)	4.000 (0.500)	
$t = 0.3$	$\xi^A$ :	0.536 (0.662)	4.000 (0.338)	
	$\xi^D$ :	0.664 (0.500)	4.000 (0.500)	
$t = 0.7$	$\xi^A$ :	0.632 (0.642)	4.000 (0.358)	
	$\xi^D$ :	0.000 (0.048)	0.664 (0.476)	4.000 (0.476)
$t = 0.9$	$\xi^A$ :	0.000 (0.158)	0.664 (0.536)	4.000 (0.306)
	$\xi^D$ :	0.000 (0.260)	0.664 (0.370)	4.000 (0.370)

$S_N = \{50(i-1)/(N-1) : i = 1, \dots, N\} \subset [0, 50]$ . For large  $N$ , the CVX program is faster than the multiplicative algorithm. For different values of linear parameters  $\alpha_1$  and  $\alpha_2$ , the D-optimal designs are all the same. This shows that the D-optimal designs do not depend on the linear parameters, which is consistent with the result in Theorem 6. When  $t = 0$ , the results give the D-optimal designs based on the LSE, and they are consistent with the result in Dette *et al.* (2006) and Yang and Stufken (2012). The support points of the D-optimal designs are clustered around four points including three interior points and the boundary point at zero, and their weights are equal. When  $t = 0.9$ , the optimal designs have five support points including the two boundary points, and their weights are not equal.

□

Table 3.3: D-optimal design support points and their weights (in parentheses) for compartmental model (3.18) with  $\alpha_1 = \alpha_2 = 1$ ,  $\beta_1 = 0.1$ ,  $\beta_2 = 1.0$ , and  $S_N = \{50(i-1)/(N-1) : i = 1, \dots, N\}$

$N = 101$						
$t = 0$	0.000 (0.250)	1.000 (0.250)	4.000 (0.250)	14.500 (0.250)		
$t = 0.9$	0.000 (0.227)	1.000 (0.227)	4.000 (0.227)	14.500 (0.226)	50.000 (0.093)	
$N = 501$						
$t = 0$	0.000 (0.250)	0.900 (0.250)	3.800 (0.250)	14.400 (0.250)		
$t = 0.9$	0.000 (0.227)	0.900 (0.227)	3.800 (0.227)	14.200 (0.226)	50.000 (0.093)	

**Example 3.6.** Emax model is another nonlinear model in pharmacodynamics to study the dose-response relationship for a new drug (Dette *et al.*, 2005; Kirby *et al.*, 2011). The form of three-parameter Emax model is given by

$$y = \frac{\alpha x^{\beta_2}}{\beta_1 + x^{\beta_2}} + \epsilon, \quad x > 0,$$

where  $y$  is a response variable for a new drug, and  $x$  is the concentration of the drug. The linear parameter  $\alpha$  is the maximum achievable effect, parameter  $\beta_1$  is the dose which produces half of the maximum effect, and  $\beta_2$  is the shape/slope factor. It is clear that this model includes the Michaelis-Menten model as a special case when  $\beta_2 = 1$ . Table 3.4 presents representative A-optimal and D-optimal designs for the Emax model with  $\alpha = 1$ ,  $\beta_1 = 1$ ,  $\beta_2 = 2$ , and  $S_N = \{(5 - 0.001)(i - 1)/(N - 1) + 0.001 : i = 1, \dots, N\} \subset [0.001, 5.0]$ . The CVX and SeDuMi programs are much faster than the multiplicative algorithms for large  $N$ . The D-optimal

designs for  $t = 0$  are consistent with the results shown in the website (<http://optimal-design.biostat.ucla.edu/optimal/OptimalDesign.aspx>). The optimal designs depend on  $t$ . The support points are clustered around three points for small  $t$ , while they are clustered around four points for  $t = 0.9$ . The boundary point 5.0 is a support point in all the optimal designs. For  $t = 0.9$ , the optimal designs include both boundary points 0.001 and 5.0. The results also indicate that the A-optimal designs are more sensitive to  $t$  than the D-optimal designs.  $\square$

Table 3.4: A-optimal and D-optimal design points and their weights (in parentheses) for the Emax model with various values of  $N$  and  $t$

$N = 101$											
$t = 0$	$\xi^A$ :	0.451	(0.403)	1.401	(0.267)	1.451	(0.099)	5.000	(0.231)		
	$\xi^D$ :	0.551	(0.333)	1.451	(0.128)	1.501	(0.206)	5.000	(0.333)		
$t = 0.3$	$\xi^A$ :	0.451	(0.401)	1.401	(0.167)	1.451	(0.199)	5.000	(0.233)		
	$\xi^D$ :	0.551	(0.333)	1.451	(0.128)	1.501	(0.206)	5.000	(0.333)		
$t = 0.7$	$\xi^A$ :	0.501	(0.378)	1.451	(0.286)	1.501	(0.096)	5.000	(0.240)		
	$\xi^D$ :	0.551	(0.333)	1.451	(0.130)	1.501	(0.204)	5.000	(0.333)		
$t = 0.9$	$\xi^A$ :	0.001	(0.099)	0.501	(0.083)	0.551	(0.246)	1.501	(0.351)	5.000	(0.221)
	$\xi^D$ :	0.001	(0.166)	0.551	(0.278)	1.451	(0.108)	1.501	(0.170)	5.000	(0.278)
$N = 201$											
$t = 0$	$\xi^A$ :	0.451	(0.401)	1.426	(0.367)	5.000	(0.232)				
	$\xi^D$ :	0.551	(0.333)	1.476	(0.334)	5.000	(0.333)				
$t = 0.3$	$\xi^A$ :	0.451	(0.401)	1.426	(0.367)	5.000	(0.232)				
	$\xi^D$ :	0.551	(0.333)	1.476	(0.334)	5.000	(0.333)				
$t = 0.7$	$\xi^A$ :	0.501	(0.376)	1.476	(0.381)	5.000	(0.243)				
	$\xi^D$ :	0.551	(0.333)	1.476	(0.334)	5.000	(0.333)				
$t = 0.9$	$\xi^A$ :	0.001	(0.103)	0.531	(0.328)	1.501	(0.348)	5.000	(0.221)		
	$\xi^D$ :	0.001	(0.166)	0.551	(0.278)	1.476	(0.278)	5.000	(0.278)		

### 3.5.2 A-efficiency and D-efficiency

To compare optimal designs based on the SLSE and the LSE, we define A-efficiency and D-efficiency measures as follows. Let

$$a_1(\xi) = \text{tr} \left( \text{Cov}(\hat{\boldsymbol{\theta}}_{SLS}) \right) = \text{tr} \left( (1-t) \sigma_0^2 (\mathbf{G}_2 - t \mathbf{g}_1 \mathbf{g}_1^\top)^{-1} \right),$$

$$d_1(\xi) = \det \left( \text{Cov}(\hat{\boldsymbol{\theta}}_{SLS}) \right) = \det \left( (1-t) \sigma_0^2 (\mathbf{G}_2 - t \mathbf{g}_1 \mathbf{g}_1^\top)^{-1} \right),$$

and take the A-efficiency and D-efficiency measures as,

$$\text{Eff}_A = \frac{a_1(\xi_{SLS}^A)}{a_1(\xi_{OLS}^A)}, \quad \text{Eff}_D = \left( \frac{d_1(\xi_{SLS}^D)}{d_1(\xi_{OLS}^D)} \right)^{1/q},$$

where  $\xi_{SLS}^A$  and  $\xi_{SLS}^D$  are the A- and D-optimal designs based on the SLSE, and  $\xi_{OLS}^A$  and  $\xi_{OLS}^D$  are based on the LSE. Since the SLSE is more efficient than the LSE, all the measures are evaluated using the covariance of the SLSE. If  $\text{Eff}_A < 1$  ( $\text{Eff}_D < 1$ ), then  $\xi_{SLS}^A$  ( $\xi_{SLS}^D$ ) is more A-efficient (D-efficient) than  $\xi_{OLS}^A$  ( $\xi_{OLS}^D$ ).

We computed the efficiency measures for the examples in Section 3.5.1, and representative results are given in Table 3.5. For small  $t$  the optimal designs based on the SLSE and the LSE are similar, which is consistent with the results in Tables 3.1 and 3.2. For large  $t$ , the optimal designs based on the SLSE can be much more efficient than those based on the LSE for some models.

Table 3.5: A-efficiency and D-efficiency for the optimal designs in Examples 3.3 and 3.4.

$t$	Example 3.3 with design space $S_{9,2}$		Example 3.4 with $N = 501$	
	$\text{Eff}_A$	$\text{Eff}_D$	$\text{Eff}_A$	$\text{Eff}_D$
0.0	1.000	1.000	1.000	1.000
0.3	1.000	1.000	0.997	0.999
0.7	1.000	1.000	0.963	0.996
0.9	0.836	0.975	0.704	0.739

## 3.6 Discussion

If the design space  $S$  is not discrete, such as a closed interval, we can discretize it and then use the methods in this paper to construct optimal designs. Gao and Zhou (2014, 2017) obtained some results for the D-optimal designs on closed interval design spaces. Several results here for A-optimal designs can also be extended to closed interval design spaces. For any  $\xi(\mathbf{x})$  on a closed interval, we can define matrix  $\mathbf{B}(\xi)$  similarly to the one in (3.8), but using  $\mathbf{g}_1$  and  $\mathbf{G}_2$  in (3.3). The expression for the A-optimality criterion in Theorem 3.3 can be easily modified using  $\mathbf{B}(\xi)$ . The definition of transformation invariance can be changed slightly by including all discrete and continuous distributions, and the invariance property can be studied for the A-optimal designs.

## Chapter 4

# Optimal designs for multi-response nonlinear regression models with several factors via semi-definite programming

In this chapter, we find A-,  $A_s$ -, c- and D-optimal designs for multi-response regression models with correlated errors by using SDP. For the first time, we extend the methodology to find optimal designs for multi-response nonlinear models and GLMs with multiple factors. Several theoretical results are derived for invariance properties of optimal designs and their dependence on the covariance matrix of the correlated errors. Our contribution in this chapter includes constructing transformations of the A-optimality to connect the optimal design problems with SDP problems; extending SDP methodology to find optimal designs from linear models to nonlinear multi-response models with multiple factors; correcting erroneously reported optimal designs in the literature caused by formulation issues.

Chapter 4 is organized as follows. In section 4.1, we provide a brief literature review for design problems for multi-response regression models. In section 4.2, we discuss the best linear unbiased estimator (BLUE) for multi-response linear models and design optimality criteria. In section 4.3, we derive theoretical properties for A-,  $A_s$ - and D-optimal designs. In section 4.4, we transform A-,  $A_s$ - and c-optimal design problems into SDP problems and use a SDP algorithm to find optimal designs.

In section 4.5, we present optimal designs of A-,  $A_s$ -, c- and D-optimal designs for multi-response multi-factor polynomial models, locally c- and D-optimal designs for a bivariate Emax response model from our algorithm and compare them with those found by other methods. In section 4.6, we find optimal designs for multi-response GLMs of a bivariate Probit model. In section 4.7, we give a conclusion for this chapter. The main work of this chapter is submitted for publication as a research paper; see Wong et al. (2016).

## 4.1 Introduction

In the last two chapters, we studied the optimal designs for one-response linear and nonlinear models using various optimality criteria based on the LSE, WLSE or SLSE. However, there are many experiments with multiple response variables, and we need multi-response models to examine the relationships between the response variables and the design variables.

In the last two decades there are some developments and results for optimal designs for multi-response models. Chang (1997) developed an algorithm to generate near D-optimal designs, and the models include mainly the first-order or second-order polynomial model for each response. Using iterated estimators, Fedorov et al. (2001) studied optimal designs for multiple responses with variance depending on unknown parameters, and they used convex design theory for finding optimal designs. Chang et al. (2001) derived D-optimal and  $D_s$ -optimal designs for polynomial regression models with two response variables and one independent variable. Atashgah and Seifi (2007, 2009) applied convex optimization algorithms for semidefinite programming (SDP) problems to construct A-, D- and E-optimal designs. Liu et al. (2011) proposed new optimality criteria based on predictive ellipsoids, which are extensions to those in Dette and O'Brien (1999) for one response case. Dette et al. (2013) constructed optimal designs for multi-response generalized linear models using a Bayesian D-optimality criterion. Magnusdottir (2013) derived results for c-optimal designs for bivariate Emax models with one independent variable.

In this chapter we explore theoretical properties of optimal designs for multi-response models. In particular, we are able to generalize results for A-,  $A_s$ - and D-optimal designs, and some results are new for A- and  $A_s$ -optimal designs. The

results include:

- (i) transformation invariance properties,
- (ii) the dependence on the covariance matrix of the correlated errors.

In addition, we examine the SDP algorithms in Atashgah and Seifi (2009) and make an improvement for A-optimal designs. We also develop SDP algorithms for finding  $A_s$ - and c-optimal designs, and they are highly efficient and can be applied for linear and nonlinear models, and generalized linear models as well.

## 4.2 Optimality criteria for multi-response models

Suppose we have  $r$  response variables,  $y_1, \dots, y_r$ , and that are related to  $p$  independent variables  $x_1, \dots, x_p$ . Their relationship can be studied through an experiment with  $n$  runs. To fix ideas, we first consider linear models and note that the approach can be applied to nonlinear models and GLMs with several independent factors. The linear models are given by

$$y_{ij} = \mathbf{f}_i^\top(\mathbf{x}_j)\boldsymbol{\theta}_i + \epsilon_{ij}, \quad j = 1, \dots, n, \quad i = 1, \dots, r, \quad (4.1)$$

where  $y_{ij}$  is the  $j$ th observation on the response variable  $y_i$ ,  $\mathbf{x}$  denotes the vector of the  $p$  independent variables, i.e.,  $\mathbf{x} = (x_1, \dots, x_p)^\top$ ,  $\mathbf{x}_j = (x_{1j}, \dots, x_{pj})^\top$  is the  $j$ th run input (design point),  $\mathbf{f}_i(\mathbf{x})$  is a known vector of regressors for variable  $y_i$ ,  $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{iq_i})^\top$  is the vector of regression parameters for  $y_i$ , and  $\epsilon_{ij}$  are random errors with mean zero. Further, we assume that

$$\begin{aligned} Cov(\epsilon_{ij}, \epsilon_{lk}) &= 0, \quad \text{for } j \neq k, \quad i, l = 1, \dots, r, \\ Cov(\epsilon_{ij}, \epsilon_{lj}) &= \sigma_{il}, \quad \text{for } j = 1, \dots, n, \quad i, l = 1, \dots, r. \end{aligned} \quad (4.2)$$

This means that the observations for the  $r$  response variables are correlated in the same run and observations from different runs are independent.

Let  $q = q_1 + \cdots + q_r$  and for  $j = 1, \dots, n$ , define

$$\mathbf{y}_j = \begin{pmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{rj} \end{pmatrix}_{r \times 1}, \quad \boldsymbol{\epsilon}_j = \begin{pmatrix} \epsilon_{1j} \\ \epsilon_{2j} \\ \vdots \\ \epsilon_{rj} \end{pmatrix}_{r \times 1}, \quad \mathbf{Z}_j = \begin{pmatrix} \mathbf{f}_1^\top(\mathbf{x}_j) & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{f}_2^\top(\mathbf{x}_j) & 0 & \cdots & 0 \\ \vdots & \vdots & & & \\ 0 & 0 & 0 & \cdots & \mathbf{f}_r^\top(\mathbf{x}_j) \end{pmatrix}_{r \times q}.$$

Put

$$\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{pmatrix}_{nr \times 1}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_n \end{pmatrix}_{nr \times 1}, \quad \mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_n \end{pmatrix}_{nr \times q}, \quad \boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_r \end{pmatrix}_{q \times 1}.$$

We can rewrite model (4.1) in matrix form as

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\theta} + \boldsymbol{\epsilon}, \quad (4.3)$$

where the covariance matrix of the error term  $\boldsymbol{\epsilon}$  is

$$\boldsymbol{\Sigma} = \text{Cov}(\boldsymbol{\epsilon}) = \boldsymbol{\Sigma}_0 \oplus \boldsymbol{\Sigma}_0 \oplus \cdots \oplus \boldsymbol{\Sigma}_0, \quad \text{with } \boldsymbol{\Sigma}_0 = \text{Cov}(\boldsymbol{\epsilon}_1) = (\sigma_{il})_{r \times r}. \quad (4.4)$$

Here  $\boldsymbol{\Sigma}_0$  is assumed to be a known positive definite matrix. The BLUE  $\hat{\boldsymbol{\theta}}$  of  $\boldsymbol{\theta}$  is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{Z}^\top \boldsymbol{\Sigma}^{-1} \mathbf{Z})^{-1} \mathbf{Z}^\top \boldsymbol{\Sigma}^{-1} \mathbf{Y},$$

and its covariance matrix is

$$\text{Cov}(\hat{\boldsymbol{\theta}}) = (\mathbf{Z}^\top \boldsymbol{\Sigma}^{-1} \mathbf{Z})^{-1} = \left( \sum_{j=1}^n \mathbf{Z}_j^\top \boldsymbol{\Sigma}_0^{-1} \mathbf{Z}_j \right)^{-1}. \quad (4.5)$$

For a predetermined sample size  $n$ , we want to choose ‘‘optimal’’ design points  $\mathbf{x}_j$ ,  $j = 1, \dots, n$ , from the given design space to minimize some loss function  $\phi(\text{Cov}(\hat{\boldsymbol{\theta}}))$  for some scalar function  $\phi$ . Various optimality criteria can be used, such as A-,  $A_s$ -, c-, and D-optimality criteria. Throughout, we assume that we are willing to discretize the design space into  $N$  user-selected points. Let this discretized design space be  $S_N \subset R^p$ , let its points be  $\mathbf{u}_1, \dots, \mathbf{u}_N$  and let  $\mathbf{U}_j$  be the  $r \times q$  matrix  $\mathbf{Z}_j$  evaluated at

$\mathbf{u}_j, j = 1, \dots, N$ , such as

$$\mathbf{U}_j = \begin{pmatrix} \mathbf{f}_1^\top(\mathbf{u}_j) & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{f}_2^\top(\mathbf{u}_j) & 0 & \cdots & 0 \\ \vdots & \vdots & & & \\ 0 & 0 & 0 & \cdots & \mathbf{f}_r^\top(\mathbf{u}_j) \end{pmatrix}_{r \times q}. \quad (4.6)$$

A design measure  $\xi(\mathbf{x})$  of  $\mathbf{x}$  is denoted by

$$\xi(\mathbf{x}) = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N \\ w_1 & w_2 & \cdots & w_N \end{pmatrix}, \quad (4.7)$$

where  $w_j$  is the proportion of times that  $\mathbf{u}_j$  is selected as a design point of  $\xi(\mathbf{x})$ , with  $w_j \geq 0$  and  $\sum_{j=1}^N w_j = 1$ . Let  $\mathbf{w} = (w_1, \dots, w_N)$  be the weight vector and let

$$\mathbf{A}(\mathbf{w}) = \sum_{j=1}^N w_j \mathbf{U}_j^\top \Sigma_0^{-1} \mathbf{U}_j, \quad (4.8)$$

so that the covariance matrix in (4.5) is proportional to  $\mathbf{C}(\mathbf{w}) = \mathbf{A}^{-1}(\mathbf{w})$ .

An optimal exact design problem is defined by the optimization problem

$$\begin{cases} \min_{\mathbf{w}} \phi(\mathbf{C}(\mathbf{w})) \\ \text{s.t. } w_j \in \{0, 1/n, 2/n, \dots, 1\}, j = 1, \dots, N, \quad \sum_{j=1}^N w_j = 1. \end{cases}$$

When  $\mathbf{A}(\mathbf{w})$  is singular,  $\phi(\mathbf{C}(\mathbf{w}))$  is defined to be  $+\infty$ . Since it is hard to solve exact design problems in general, we relax the constraints on  $\mathbf{w}$  to find approximate optimal designs defined by:

$$\begin{cases} \min_{\mathbf{w}} \phi(\mathbf{C}(\mathbf{w})) \\ \text{s.t. } w_j \geq 0, j = 1, \dots, N, \quad \sum_{j=1}^N w_j = 1. \end{cases} \quad (4.9)$$

For the D-optimality criterion, we have  $\phi(\mathbf{C}(\mathbf{w})) = \det(\mathbf{C}(\mathbf{w}))$ . For A-,  $A_s$ -, and c-optimality, the loss function  $\phi(\mathbf{C}(\mathbf{w}))$  has a general form given by

$$h(\mathbf{w}, \mathbf{L}) = \text{trace}(\mathbf{L}^\top \mathbf{C}(\mathbf{w}) \mathbf{L}) = \text{trace}(\mathbf{L}^\top \mathbf{A}^{-1}(\mathbf{w}) \mathbf{L}), \quad (4.10)$$

where

$$\mathbf{L} = \begin{cases} \mathbf{I}_q, & \text{for A-optimality,} \\ \text{diag}(\mathbf{a}), & \text{for } A_s\text{-optimality,} \\ \mathbf{c}_{q \times 1} & \text{for c-optimality.} \end{cases} \quad (4.11)$$

Here  $\mathbf{I}_q$  is the  $q \times q$  identity matrix,  $\text{diag}(\mathbf{a})$  is a user-selected diagonal matrix with components of the vector  $\mathbf{a}$  equal to 0 or 1 on the diagonal and  $\mathbf{c}$  is the given vector of coefficients for estimating  $\mathbf{c}^\top \boldsymbol{\theta}$ . An A-optimal design minimizes the average of the variances of  $\hat{\boldsymbol{\theta}}$ , an  $A_s$ -optimal design minimizes the average of the variances of a subset of  $\hat{\boldsymbol{\theta}}$  and the subset is defined by vector  $\mathbf{a}$ , and a c-optimal design minimizes the variance of a linear function of  $\hat{\boldsymbol{\theta}}$ ,  $\mathbf{c}^\top \hat{\boldsymbol{\theta}}$ . Since  $\mathbf{A}(\mathbf{w})$  is linear in  $\mathbf{w}$ ,  $h(\mathbf{w}, \mathbf{L})$  in (4.10) is a convex function of  $\mathbf{w}$  for each fixed  $\mathbf{L}$ . Consequently, using convex analysis results, we can check optimality of a design for multi-response models using Lemma 4.1 as follows.

**Lemma 4.1.** *Let  $h_j(\mathbf{w}, \mathbf{L}) = \text{trace}(\Sigma_0^{-1} \mathbf{U}_j \mathbf{A}^{-1}(\mathbf{w}) \mathbf{L} \mathbf{L}^\top \mathbf{A}^{-1}(\mathbf{w}) \mathbf{U}_j^\top)$ ,  $j = 1, \dots, N$ . Then  $\hat{\mathbf{w}}$  solves problem (4.9) with the loss function (4.10) if and only if  $\mathbf{A}(\hat{\mathbf{w}})$  is non-singular and*

$$h_j(\hat{\mathbf{w}}, \mathbf{L}) \leq h(\hat{\mathbf{w}}, \mathbf{L}), \quad \text{for all } j = 1, \dots, N.$$

We omit its proof because it is similar to Lemma 2.2. This result is useful to verify if a weight vector  $\mathbf{w}$  is an optimal design.

### 4.3 Properties of optimal designs

Let us write  $\Sigma_0^{-1} = (s_{il})_{r \times r}$  and let  $\mathbf{A}_{il}(\mathbf{w}) = \sum_{j=1}^N w_j \mathbf{f}_i(\mathbf{u}_j) \mathbf{f}_l^\top(\mathbf{u}_j)$ ,  $i, l = 1, \dots, r$ . By (4.8), it is easy to verify that

$$\mathbf{A}(\mathbf{w}) = \begin{pmatrix} s_{11} \mathbf{A}_{11}(\mathbf{w}) & \cdots & s_{1r} \mathbf{A}_{1r}(\mathbf{w}) \\ s_{21} \mathbf{A}_{21}(\mathbf{w}) & \cdots & s_{2r} \mathbf{A}_{2r}(\mathbf{w}) \\ \vdots & \cdots & \vdots \\ s_{r1} \mathbf{A}_{r1}(\mathbf{w}) & \cdots & s_{rr} \mathbf{A}_{rr}(\mathbf{w}) \end{pmatrix}, \quad (4.12)$$

**Lemma 4.2.** *If the vector of regression functions are all the same for the  $r$  response*

variables, i.e.,  $\mathbf{f}_1(\mathbf{x}) = \mathbf{f}_2(\mathbf{x}) = \cdots = \mathbf{f}_r(\mathbf{x})$ , then matrix  $\mathbf{A}(\mathbf{w})$  in (4.12) becomes

$$\mathbf{A}(\mathbf{w}) = \Sigma_0^{-1} \otimes \mathbf{A}_{11}(\mathbf{w}), \quad (4.13)$$

where  $\otimes$  denotes Kronecker product and  $\mathbf{A}_{11}(\mathbf{w}) = \sum_{j=1}^N w_j \mathbf{f}_1(\mathbf{u}_j) \mathbf{f}_1^\top(\mathbf{u}_j)$ .

By properties of Kronecker product of matrices, it is straightforward that

(i)  $\mathbf{A}^{-1}(\mathbf{w}) = \Sigma_0 \otimes \mathbf{A}_{11}^{-1}(\mathbf{w})$ ,

(ii)  $\det(\mathbf{A}^{-1}(\mathbf{w})) = (\det(\Sigma_0))^{q_1} (\det(\mathbf{A}_{11}(\mathbf{w})))^{-r}$ ,

(iii)  $\text{trace}(\mathbf{A}^{-1}(\mathbf{w})) = \text{trace}(\Sigma_0) \cdot \text{trace}(\mathbf{A}_{11}^{-1}(\mathbf{w}))$ ,

which directly leads to the following properties of optimal designs.

**Theorem 4.1.** *Under the assumption in Lemma 4.2,*

(i) *the A-optimal and D-optimal designs for the multi-response model do not depend on the covariance matrix  $\Sigma_0$ ,*

(ii) *the A-optimal and D-optimal designs for the multi-response model are the same as those for one-response model with response function  $\mathbf{f}_1^\top(\mathbf{x})\boldsymbol{\theta}_1$ .*

The above results are more general than those in Krafft and Schaefer (1992) and Chang et al. (2001), where they considered models with one independent variable only. If the  $r$  vectors of regressors  $\mathbf{f}_1(\mathbf{x}), \dots, \mathbf{f}_r(\mathbf{x})$  are not the same, optimal designs may depend on the covariance matrix  $\Sigma_0$ . For example, when  $r = 2$  and

$$\Sigma_0 = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \quad -1 < \rho < 1, \quad (4.14)$$

we have  $s_{11} = s_{22} = 1/(1 - \rho^2)$  and  $s_{12} = s_{21} = -\rho/(1 - \rho^2)$ . For this situation, various optimal designs have been studied, see for example, Chang et al. (2001) and Atashgah and Seifi (2007, 2009). Theorem 4.2 and the results that follow provide further insights into such optimal designs.

**Theorem 4.2.** *For  $r = 2$  and  $\Sigma_0$  in (4.14). The A-,  $A_s$ - and D-optimal designs depend on  $\Sigma_0$  only through  $|\rho|$ . In addition, if the two response functions are nested, say  $\mathbf{f}_2^\top(\mathbf{x}) = (\mathbf{f}_1^\top(\mathbf{x}), \mathbf{g}_1^\top(\mathbf{x}))$  for some vector function  $\mathbf{g}_1^\top(\mathbf{x})$ , then the D-optimal design does not depend on  $\Sigma_0$ .*

**Proof of Theorem 4.2:** By (4.12), when  $r = 2$  we have

$$\mathbf{A}(\mathbf{w}) = \begin{pmatrix} s_{11}\mathbf{A}_{11}(\mathbf{w}) & s_{12}\mathbf{A}_{12}(\mathbf{w}) \\ s_{21}\mathbf{A}_{21}(\mathbf{w}) & s_{22}\mathbf{A}_{22}(\mathbf{w}) \end{pmatrix},$$

where for simplicity, we write  $\mathbf{A}_{ij}(\mathbf{w})$  as  $\mathbf{A}_{ij}$ . Let  $\mathbf{G} = \mathbf{A}_{22} - \rho^2\mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$ , let  $s_{11} = s_{22} = 1/(1 - \rho^2)$  and let  $s_{12} = s_{21} = -\rho/(1 - \rho^2)$ . One can verify that

$$\det(\mathbf{A}(\mathbf{w})) = \left(\frac{1}{1 - \rho^2}\right)^q \cdot \det(\mathbf{A}_{11}) \cdot \det(\mathbf{G}),$$

$$\mathbf{A}^{-1}(\mathbf{w}) = (1 - \rho^2) \cdot \begin{pmatrix} \mathbf{A}_{11}^{-1} + \rho^2\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{G}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \rho\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{G}^{-1} \\ \rho\mathbf{G}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{G}^{-1} \end{pmatrix}$$

$$\text{and trace}(\mathbf{A}^{-1}(\mathbf{w})) = (1 - \rho^2) \cdot \text{trace}(\mathbf{A}_{11}^{-1} + \rho^2\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{G}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} + \mathbf{G}^{-1}).$$

It follows that  $A$ - and  $D$ -optimal designs depend on  $\Sigma_0$  only through  $|\rho|$ , and the same is true for any  $A_s$ -optimal design.

If the two response models are nested, say  $\mathbf{f}_2^\top(\mathbf{x}) = (\mathbf{f}_1^\top(\mathbf{x}), \mathbf{g}_1^\top(\mathbf{x}))$ , we may write

$$\mathbf{A}_{21} = \begin{pmatrix} \mathbf{A}_{11} \\ \tilde{\mathbf{A}}_{21} \end{pmatrix}, \quad \mathbf{A}_{12} = \mathbf{A}_{21}^\top, \quad \mathbf{A}_{22} = \begin{pmatrix} \mathbf{A}_{11} & \tilde{\mathbf{A}}_{21}^\top \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{pmatrix},$$

where  $\tilde{\mathbf{A}}_{21} = \sum_{j=1}^N w_j \mathbf{g}_1(\mathbf{u}_j) \mathbf{f}_1^\top(\mathbf{u}_j)$  and  $\tilde{\mathbf{A}}_{22} = \sum_{j=1}^N w_j \mathbf{g}_1(\mathbf{u}_j) \mathbf{g}_1^\top(\mathbf{u}_j)$ . Consequently,

$$\det(\mathbf{G}) = \det((1 - \rho^2)\mathbf{A}_{11}) \cdot \det(\tilde{\mathbf{A}}_{22} - \tilde{\mathbf{A}}_{21}\mathbf{A}_{11}^{-1}\tilde{\mathbf{A}}_{21}^\top),$$

which implies that the  $D$ -optimal design does not depend on  $\rho$  or  $\Sigma_0$ .  $\square$

If the covariance matrix  $\Sigma_0$  depends on a correlation coefficient  $\rho$ , we write it as  $\Sigma_0(\rho)$ . Also note that if  $\mathbf{Q}$  is a  $q \times q$  diagonal matrix with diagonal elements equal to 1 or  $-1$ , we have (i)  $\mathbf{Q}^{-1} = \mathbf{Q}$ , (ii)  $\det(\mathbf{Q}\mathbf{A}^{-1}(\mathbf{w})\mathbf{Q}) = \det(\mathbf{A}^{-1}(\mathbf{w}))$  and (iii)  $\text{trace}(\mathbf{Q}\mathbf{A}^{-1}(\mathbf{w})\mathbf{Q}) = \text{trace}(\mathbf{A}^{-1}(\mathbf{w}))$ .

**Theorem 4.3.** *In (4.8), if  $\Sigma_0^{-1}(\rho)$  satisfies*

$$\mathbf{A}(\mathbf{w}) = \sum_{j=1}^N w_j \mathbf{U}_j^\top \Sigma_0^{-1}(\rho) \mathbf{U}_j = \mathbf{Q} \sum_{j=1}^N w_j \mathbf{U}_j^\top \Sigma_0^{-1}(|\rho|) \mathbf{U}_j \mathbf{Q},$$

where  $\mathbf{Q} = \mathbf{Q}_1 \oplus \mathbf{Q}_2 \oplus \cdots \oplus \mathbf{Q}_r$  with  $\mathbf{Q}_i = \pm \mathbf{I}_{q_i}$ , then  $A$ -,  $A_s$ - and  $D$ -optimal designs depend on  $\Sigma_0$  only through  $|\rho|$ .

The proof of Theorem 4.3 is straightforward and is omitted. Theorem 4.3 is applicable to many situations and we present two here.

Case I: Suppose  $r = 3$  and  $\Sigma_0(\rho) = \begin{pmatrix} 1 & \rho & 0 \\ \rho & 1 & \rho \\ 0 & \rho & 1 \end{pmatrix}$ , where  $-1/2 \leq \rho \leq 1/2$ . Then

$$\Sigma_0^{-1}(\rho) = \frac{1}{1-2\rho^2} \begin{pmatrix} 1-\rho^2 & -\rho & \rho^2 \\ -\rho & 1 & -\rho \\ \rho^2 & -\rho & 1-\rho^2 \end{pmatrix}$$

and one can show that the condition in Theorem 4.3 is satisfied when  $\mathbf{Q}_1 = \mathbf{I}_{q_1}$ ,  $\mathbf{Q}_3 = \mathbf{I}_{q_3}$ ,  $\mathbf{Q}_2 = \text{sign}(\rho)\mathbf{I}_{q_2}$  for  $\rho \neq 0$  and  $\mathbf{Q}_2 = \mathbf{I}_{q_2}$  otherwise.

Case II: If

$$\Sigma_0(\rho) = \begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \dots & \rho^{r-1} \\ \rho & 1 & \rho & \rho^2 & \dots & \rho^{r-2} \\ \rho^2 & \rho & 1 & \rho & & \rho^{r-3} \\ \vdots & \vdots & & & \vdots & \vdots \\ \rho^{r-2} & \rho^{r-1} & \dots & \rho & 1 & \rho \\ \rho^{r-1} & \rho^{r-2} & \dots & \rho^2 & \rho & 1 \end{pmatrix}_{r \times r}, \quad 0 < |\rho| < 1,$$

then  $\Sigma_0(\rho)$  has the following inverse matrix,

$$\Sigma_0^{-1}(\rho) = \frac{1}{1-\rho^2} \begin{pmatrix} 1 & -\rho & 0 & \dots & 0 \\ -\rho & 1+\rho^2 & -\rho & \dots & 0 \\ 0 & -\rho & 1+\rho^2 & \ddots & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \dots & -\rho & 1+\rho^2 & -\rho \\ 0 & \dots & 0 & -\rho & 1 \end{pmatrix}_{r \times r}, \quad 0 < |\rho| < 1,$$

the condition in Theorem 4.3 is true with  $\mathbf{Q}_i = (\text{sign}(\rho))^i \mathbf{I}_{q_i}$  for  $i = 1, \dots, r$ . Thus, the A-, A<sub>s</sub>- and D-optimal designs depend on  $\Sigma_0$  only through  $|\rho|$ .

We now study invariance properties of optimal designs. Let  $T$  be an one-to-one function:  $S_N \mapsto S_N$  and  $T^2$  is an identity map. For a distribution  $\xi(\mathbf{x})$  defined in

(4.7), let

$$\xi(T(\mathbf{x})) = \begin{pmatrix} T(\mathbf{u}_1) & T(\mathbf{u}_2) & \cdots & T(\mathbf{u}_N) \\ w_1 & w_2 & \cdots & w_N \end{pmatrix}. \quad (4.15)$$

The distribution  $\xi$  is  $T$ -invariant if  $\xi(\mathbf{x}) = \xi(T(\mathbf{x}))$ . A  $T$ -invariant distribution  $\xi$  implies that  $w_i = w_j$  if  $T(\mathbf{u}_i) = \mathbf{u}_j$ .

**Theorem 4.4.** *Suppose model (4.3) is defined on a given discretized design space  $S_N$  and  $T$  is an one-to-one function with  $T^2$  being an identity map. If there exists  $r$  diagonal matrices  $\mathbf{Q}_1, \dots, \mathbf{Q}_r$  with all diagonal elements equal to 1 or  $-1$  and*

$$\mathbf{f}_i(T(\mathbf{x})) = \mathbf{Q}_i \mathbf{f}_i(\mathbf{x}) \quad \text{for all } \mathbf{x} \in S_N, \quad i = 1, \dots, r,$$

then there exists  $T$ -invariant  $A$ -,  $A_s$ - and  $D$ -optimal designs for any  $\Sigma_0$ .

**Proof of Theorem 4.4:** For any distribution  $\xi(\mathbf{x})$ , let  $\xi_1(\mathbf{x}) = \xi(T(\mathbf{x}))$  as in (4.15). After rearranging the columns, we write

$$\xi_1(\mathbf{x}) = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N \\ \tilde{w}_1 & \tilde{w}_2 & \cdots & \tilde{w}_N \end{pmatrix}$$

and note that  $\tilde{w}_j = w_i$  and  $\tilde{w}_i = w_j$  if  $T(\mathbf{u}_i) = \mathbf{u}_j$ . Let  $\tilde{\mathbf{w}} = (\tilde{w}_1, \dots, \tilde{w}_N)$  and let  $\xi_{0.5}(\mathbf{x}) = 0.5\xi(\mathbf{x}) + 0.5\xi_1(\mathbf{x})$ , i.e.,

$$\xi_{0.5}(\mathbf{x}) = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N \\ 0.5(w_1 + \tilde{w}_1) & 0.5(w_2 + \tilde{w}_2) & \cdots & 0.5(w_N + \tilde{w}_N) \end{pmatrix}.$$

If  $T(\mathbf{u}_i) = \mathbf{u}_j$ ,  $\xi_{0.5}(\mathbf{x})$  has the same weight at  $\mathbf{u}_i$  and  $\mathbf{u}_j$  and so  $\xi_{0.5}(\mathbf{x})$  is  $T$ -invariant.

Let  $\mathbf{Q} = \mathbf{Q}_1 \oplus \cdots \oplus \mathbf{Q}_r$ . Since  $\mathbf{f}_i(T(\mathbf{x})) = \mathbf{Q}_i \mathbf{f}_i(\mathbf{x})$ , we have

$$\begin{aligned} \mathbf{A}(\tilde{\mathbf{w}}) &= \sum_{j=1}^N \tilde{w}_j \mathbf{U}_j^\top \Sigma_0^{-1} \mathbf{U}_j \\ &= \sum_{j=1}^N w_j \mathbf{Q} \mathbf{U}_j^\top \Sigma_0^{-1} \mathbf{U}_j \mathbf{Q}^\top \\ &= \mathbf{Q} \mathbf{A}(\mathbf{w}) \mathbf{Q}^\top. \end{aligned}$$

Since  $\det(\mathbf{A}(\tilde{\mathbf{w}})) = \det(\mathbf{A}(\mathbf{w}))$ ,  $\text{trace}(\mathbf{A}^{-1}(\tilde{\mathbf{w}})) = \text{trace}(\mathbf{A}^{-1}(\mathbf{w}))$  and it is clear that

$\text{trace}(\mathbf{L}^\top \mathbf{A}^{-1}(\tilde{\mathbf{w}})\mathbf{L}) = \text{trace}(\mathbf{L}^\top \mathbf{A}^{-1}(\mathbf{w})\mathbf{L})$ , where  $\mathbf{L}$  is defined in (4.11) and they are all convex in  $\mathbf{w}$ , we have

$$\det(\mathbf{A}(0.5(\mathbf{w} + \tilde{\mathbf{w}}))) \leq \det(\mathbf{A}(\mathbf{w})),$$

$$\text{trace}(\mathbf{A}^{-1}(0.5(\mathbf{w} + \tilde{\mathbf{w}}))) \leq \text{trace}(\mathbf{A}^{-1}(\mathbf{w})) \text{ and}$$

$$\text{trace}(\mathbf{L}^\top \mathbf{A}^{-1}(0.5(\mathbf{w} + \tilde{\mathbf{w}}))\mathbf{L}) \leq \text{trace}(\mathbf{L}^\top \mathbf{A}^{-1}(\mathbf{w})\mathbf{L}) \text{ for any } \xi(\mathbf{x}).$$

It follows that the weight vector  $0.5(\mathbf{w} + \tilde{\mathbf{w}})$  corresponds to the  $T$ -invariant distribution  $\xi_{0.5}(\mathbf{x})$ , which implies that there exist  $T$ -invariant A-,  $A_s$ - and D-optimal designs.  $\square$

From the proof of Theorem 4.4 above for A- and D-optimality, one can relax the requirement that  $\mathbf{Q}_i$ 's be diagonal. For a function  $T$ , if the  $\mathbf{Q}_i$ 's satisfy  $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ , where  $\mathbf{Q} = \mathbf{Q}_1 \oplus \cdots \oplus \mathbf{Q}_r$ , one can show that there exists  $T$ -invariant A- and D-optimal designs. Examples 4.1 and 4.2 below show how these theoretical results can be applied for various models to study the properties of optimal designs.

**Example 4.1.** Consider the dual-response polynomial model in Chang et al. (2001),  $\mathbf{f}_1^\top(x)\boldsymbol{\theta}_1 = \theta_{10} + \theta_{11}x + \theta_{12}x^2$ ,  $\mathbf{f}_2^\top(x)\boldsymbol{\theta}_2 = \theta_{20} + \theta_{21}x + \theta_{22}x^2 + \theta_{23}x^3$ ,  $S_N = \{u_j \mid u_j = -1 + 2(j-1)/(N-1), j = 1, \dots, N\}$ , and  $\Sigma_0$  in (4.14). Let  $Tx = -x$ . It is clear that  $T$  is an one-to-one function:  $S_N \mapsto S_N$  and  $T^2$  is an identity map. Define  $\mathbf{Q}_1 = \text{diag}(1, -1, 1)$  and  $\mathbf{Q}_2 = \text{diag}(1, -1, 1, -1)$ . Then it is obvious that  $\mathbf{f}_1(T(x)) = \mathbf{Q}_1\mathbf{f}_1(x)$ ,  $\mathbf{f}_2(T(x)) = \mathbf{Q}_2\mathbf{f}_2(x)$ , for all  $x \in S_N$ . By Theorem 4.4, there exists  $T$ -invariant A-,  $A_s$ -, and D-optimal symmetric designs. By Theorem 4.2, the D-optimal designs do not depend on  $\Sigma_0$  since the two response functions are nested with  $\mathbf{f}_2^\top(x) = (\mathbf{f}_1^\top(x), x^3)$ . The A- and  $A_s$ -optimal designs depend only on  $|\rho|$ .  $\square$

**Example 4.2.** Consider a three-response model with two independent variables  $x_1$  and  $x_2$ ,  $\mathbf{f}_1^\top(\mathbf{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$ ,  $\mathbf{f}_2(\mathbf{x}) = \mathbf{f}_1(\mathbf{x})$ ,  $\mathbf{f}_3^\top(\mathbf{x}) = (1, x_1, x_2)$ , and

$$s_9 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -\sqrt{2} \\ 0 \end{pmatrix}, \begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -\sqrt{2} \end{pmatrix}, \begin{pmatrix} 0 \\ \sqrt{2} \end{pmatrix} \right\}.$$

Define three functions

$$T_1(\mathbf{x}) = -\mathbf{x}, \quad T_2(\mathbf{x}) = \begin{pmatrix} x_2 \\ x_1 \end{pmatrix} \quad \text{and} \quad T_3(\mathbf{x}) = \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix}.$$

By Theorem 4.4 for  $T_1$  and  $T_3$  and the comments below Theorem 4.4 for  $T_2$ , we obtain an A-optimal design with equal weights at points  $\mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$  and  $\mathbf{u}_5$  and equal weights at  $\mathbf{u}_6, \mathbf{u}_7, \mathbf{u}_8$  and  $\mathbf{u}_9$ . Similarly, the same result holds for a D-optimal design.  $\square$

## 4.4 SDP for optimal design problems

Although we have obtained several theoretical properties, it is still hard to derive analytical solutions for optimal designs in general. Again numerical methods are applied to construct optimal designs here. One effective method is to use convex optimization algorithms, SeDuMi and CVX, for finding optimal designs as in chapter 2 and 3. SeDuMi and CVX programs are efficient to solve SDP problems, but we need to be careful to transform design problems into SDP problems.

We first give two simple SDP problems in Example 4.3. Then we point out some errors in the literature to compute A-optimal designs.

**Example 4.3.** Define matrices

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 8 & 0 \\ 1 & 0 & v_1 \end{pmatrix}, \mathbf{M}_2 = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 8 & 1 \\ 0 & 1 & v_2 \end{pmatrix}, \mathbf{M}_3 = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 2 & 8 & 0 & 1 \\ 1 & 0 & v_1 & 0 \\ 0 & 1 & 0 & v_2 \end{pmatrix},$$

where  $v_1$  and  $v_2$  are two variables. Consider the following two SDP problems,

$$\begin{cases} \min_{v_1, v_2} v_1 + v_2 \\ \text{s.t. } \mathbf{M}_1 \oplus \mathbf{M}_2 \succeq 0, \end{cases} \quad (4.16)$$

and

$$\begin{cases} \min_{v_1, v_2} v_1 + v_2 \\ \text{s.t. } \mathbf{M}_3 \succeq 0, \end{cases} \quad (4.17)$$

where  $\mathbf{M}_1 \oplus \mathbf{M}_2 \succeq 0$  means that  $\mathbf{M}_1 \oplus \mathbf{M}_2$  is PSD. The objective function  $v_1 + v_2$  in both problems is a linear function of  $v_1$  and  $v_2$ . Since matrices  $\mathbf{M}_1 \oplus \mathbf{M}_2$  and  $\mathbf{M}_3$  are also linear in  $v_1$  and  $v_2$ , the constraints are linear matrices being PSD. Each constraint defines a convex set of  $(v_1, v_2)$ . It is easy to show that the solution to (4.16) is  $(v_1^*, v_2^*) = (2.0, 0.25)$ , while the solution to (4.17) is  $(v_1^*, v_2^*) = (2.5, 0.75)$ . The convex sets and solutions to both problems are plotted in Figure 4.1.  $\square$

In Example 4.3, we note that the matrices  $\mathbf{M}_1, \mathbf{M}_2$  and  $\mathbf{M}_3$  have a common principal submatrix formed by the first two rows and the first two columns. More generally, the SDP problem formulations in (4.16) and (4.17) can be generalized to

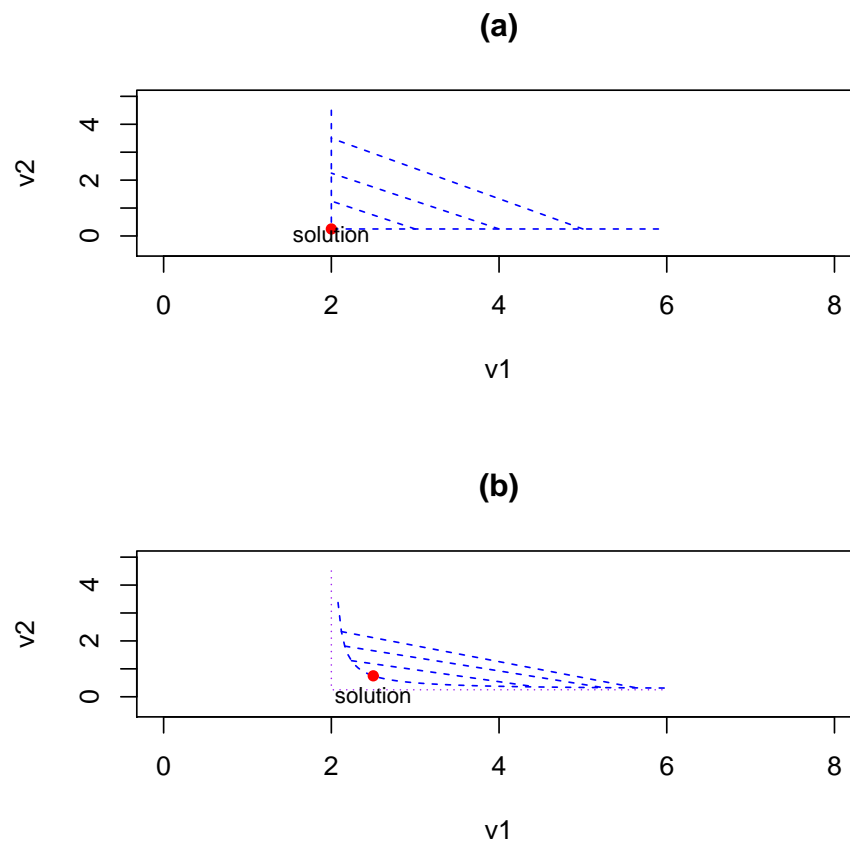


Figure 4.1: The convex sets (shaded areas) and solutions to the: (a) SDP problem in (4.16), (b) SDP problem in (4.17).

find optimal designs as follows. Let  $\mathbf{M}$  be a  $q \times q$  symmetric positive definite matrix and let  $\mathbf{e}_i$  be the  $i$ th unit vector in  $R^q$ ,  $i = 1, \dots, q$ . Define matrices

$$\mathbf{M}_i = \begin{pmatrix} \mathbf{M} & \mathbf{e}_i \\ \mathbf{e}_i^\top & v_i \end{pmatrix}, \quad i = 1, \dots, q, \quad \mathbf{M}_{q+1} = \begin{pmatrix} \mathbf{M} & \mathbf{I}_q \\ \mathbf{I}_q & \mathbf{V} \end{pmatrix},$$

where  $\mathbf{V} = \text{diag}(v_1, \dots, v_q)$  is a diagonal matrix. Then (4.16) and (4.17) become

$$\begin{cases} \min_{v_1, \dots, v_q} v_1 + \dots + v_q \\ \text{s.t. } \mathbf{M}_1 \oplus \dots \oplus \mathbf{M}_q \succeq 0, \end{cases} \quad (4.18)$$

and

$$\begin{cases} \min_{v_1, \dots, v_q} v_1 + \dots + v_q \\ \text{s.t. } \mathbf{M}_{q+1} \succeq 0. \end{cases} \quad (4.19)$$

The solution to problem (4.18) is given in Theorem 4.5 while the solution to problem (4.19) is discussed in Theorem 4.6.

**Theorem 4.5.** *Suppose  $b_{11}, \dots, b_{qq}$  are the diagonal elements of  $\mathbf{M}^{-1}$ . Problem (4.18) has a unique solution given by  $v_1^* = b_{11}, \dots, v_q^* = b_{qq}$ , and  $v_1^* + \dots + v_q^* = \text{trace}(\mathbf{M}^{-1})$ .*

**Proof of Theorem 4.5:** The constraint in problem (4.18) is true if and only if  $\mathbf{M}_i \succeq 0$  for all  $i = 1, \dots, q$ . Since  $\mathbf{M} \succ 0$ ,  $\mathbf{M}_i \succeq 0$  is true if and only if  $v_i - \mathbf{e}_i^\top \mathbf{M}^{-1} \mathbf{e}_i \geq 0$  (Horn and Johnson, 2009, p472). This gives  $v_i - b_{ii} \geq 0$ ,  $i = 1, \dots, q$ , where  $b_{ii}$  is the  $i$ th diagonal element of  $\mathbf{M}^{-1}$  and so  $v_1 + \dots + v_q$  is minimized by  $v_1^* = b_{11}, \dots, v_q^* = b_{qq}$ , and  $v_1^* + \dots + v_q^* = b_{11} + \dots + b_{qq} = \text{trace}(\mathbf{M}^{-1})$ .  $\square$

**Theorem 4.6.** *If  $v_1^*, \dots, v_q^*$  is a solution to problem (4.19), then  $v_1^* + \dots + v_q^* \geq \text{trace}(\mathbf{M}^{-1})$  and the equality holds if and only if  $\mathbf{M}$  is a diagonal matrix.*

**Proof of Theorem 4.6:** From Horn and Johnson (2009, p472), the constraint in problem (4.19) is true if and only if  $\mathbf{V} - \mathbf{I}_q \mathbf{M}^{-1} \mathbf{I}_q \succeq 0$ , which is equivalent to

$$\mathbf{V} - \mathbf{M}^{-1} \succeq 0. \quad (4.20)$$

Therefore  $v_1 + \dots + v_q = \text{trace}(\mathbf{V}) \geq \text{trace}(\mathbf{M}^{-1})$  and the minimizer of  $v_1 + \dots + v_q$  must satisfy  $v_1^* + \dots + v_q^* \geq \text{trace}(\mathbf{M}^{-1})$ .

Let  $\mathbf{M}^{-1} = (b_{ij})_{q \times q}$ . If  $\mathbf{M}$  is a diagonal matrix, so is  $\mathbf{M}^{-1}$  and by (4.20),  $v_i - b_{ii} \geq 0$ ,  $i = 1, \dots, q$  so that the minimizer of  $v_1 + \dots + v_q$  is  $v_1^* = b_{11}, \dots, v_q^* = b_{qq}$  and  $v_1^* + \dots + v_q^* = \text{trace}(\mathbf{M}^{-1})$ .

If  $\mathbf{M}$  is not a diagonal matrix, there exists at least two off-diagonal elements of  $\mathbf{M}^{-1}$  that are nonzero. Without loss of generality, assume that  $b_{12} = b_{21} \neq 0$  since  $\mathbf{M}^{-1}$  is symmetric. By (4.20), we still have  $v_i - b_{ii} \geq 0$ ,  $i = 1, \dots, q$ . However,  $\mathbf{V} = \text{diag}(b_{11}, \dots, b_{qq})$  does not satisfy (4.20) since

$$\mathbf{a}^\top (\mathbf{V} - \mathbf{M}^{-1}) \mathbf{a} = -2|b_{12}| < 0, \text{ with } \mathbf{a} = (1, \text{sign}(b_{12}), 0, \dots, 0)^\top \in R^q.$$

Thus, the minimizer of  $v_1 + \dots + v_q$  must satisfy

$$v_1^* + \dots + v_q^* > b_{11} + \dots + b_{qq} = \text{trace}(\mathbf{M}^{-1}).$$

□

From the proof of Theorem 4.6, we note that the convex set of  $(v_1, \dots, v_q)$  defined by the constraint in (4.19) is smaller than that in (4.18) in general, so the minimum value of  $v_1 + \dots + v_q$  in (4.19) is larger than that in (4.18). The results in Example 4.3 are consistent with Theorems 4.5 and 4.6.

Optimal design problems in (4.9) with loss function in (4.10) can be transformed to SDP problems. Atashgah and Seifi (2009) worked on SDP for multi-response models. However, the transformation for the A-optimality in Atashgah and Seifi (2009) is not correct and needs to be improved,

#### 4.4.1 A-optimality

For A-optimality, the loss function is  $h(\mathbf{w}, \mathbf{I}_q) = \text{trace}(\mathbf{A}^{-1}(\mathbf{w}))$  by (4.10) and (4.11). Define matrices

$$\mathbf{D}_i = \begin{pmatrix} \mathbf{A}(\mathbf{w}) & \mathbf{e}_i \\ \mathbf{e}_i^\top & v_i \end{pmatrix}, \quad i = 1, \dots, q, \quad \mathbf{W}_N = \text{diag} \left( w_1, \dots, w_{N-1}, 1 - \sum_{j=1}^{N-1} w_j \right),$$

where  $v_1, \dots, v_q$  are real variables. Then the constraints in (4.9) are equivalent to  $\mathbf{W}_N \succeq 0$ . By Theorem 4.5, we transform problem (4.9) into a SDP problem as follows:

$$\begin{cases} \min_{v_1, \dots, v_q, w_1, \dots, w_{N-1}} & v_1 + \dots + v_q \\ \text{s.t.} & \mathbf{D}_1 \oplus \dots \oplus \mathbf{D}_q \oplus \mathbf{W}_N \succeq 0 . \end{cases} \quad (4.21)$$

This transformation is an extension from one-response models in Boyd and Vandenberghe (2004) and Ye et al. (2015). The SDP problem for multi-response models in Atashgah and Seifi (2009) is similar to (4.19). From Theorem 4.6, the solution to their problem usually does not minimize trace ( $\mathbf{A}^{-1}(\mathbf{w})$ ).

#### 4.4.2 $\mathbf{A}_s$ -optimality

For  $\mathbf{A}_s$ -optimality, the loss function is  $h(\mathbf{w}, \text{diag}(\mathbf{a})) = \text{trace}(\text{diag}(\mathbf{a}) \mathbf{A}^{-1}(\mathbf{w}) \text{diag}(\mathbf{a}))$  by (4.10) and (4.11). Let  $\mathbf{a} = (a_1, \dots, a_q)^\top$ , where each element  $a_i$  is either 0 or 1. We can transform problem (4.9) into a SDP problem as follows,

$$\begin{cases} \min_{v_1, \dots, v_q, w_1, \dots, w_{N-1}} & a_1 v_1 + \dots + a_q v_q \\ \text{s.t.} & \mathbf{D}_1 \oplus \dots \oplus \mathbf{D}_q \oplus \mathbf{W}_N \succeq 0 . \end{cases} \quad (4.22)$$

The constraints in (4.21) and (4.22) are the same, but the objective functions are different. In (4.22), a subset of the variances is included in the objective function, and the subset is defined by vector  $\mathbf{a}$ . This new transformation for the  $\mathbf{A}_s$ -optimality also leads to a new one for c-optimality.

#### 4.4.3 c-optimality

For c-optimality, the loss function is  $h(\mathbf{w}, \mathbf{c}) = \mathbf{c}^\top \mathbf{A}^{-1}(\mathbf{w}) \mathbf{c}$ . Let  $\mathbf{c} = (c_1, \dots, c_q)^\top$  and  $k$  is the smallest integer such that  $c_k \neq 0$ . Let  $\mathbf{I}_{(k)}$  be the  $q \times (q-1)$  matrix obtained by deleting column  $k$  from  $\mathbf{I}_q$ . Define matrices

$$\mathbf{E} = (\mathbf{c}, \mathbf{I}_{(k)})_{q \times q}, \quad \tilde{\mathbf{D}}_i = \begin{pmatrix} \mathbf{E}^{-1} \mathbf{A}(\mathbf{w}) \mathbf{E}^{-\top} & \mathbf{e}_i \\ \mathbf{e}_i^\top & v_i \end{pmatrix}, \quad i = 1, \dots, q,$$

where  $\mathbf{E}^{-\top} = (\mathbf{E}^\top)^{-1}$  and  $\mathbf{E}$  is non-singular. It is easy to verify that

$$\begin{aligned} h(\mathbf{w}, \mathbf{c}) &= \mathbf{c}^\top \mathbf{A}^{-1}(\mathbf{w}) \mathbf{c} \\ &= \text{trace} \left( \text{diag}(1, 0, \dots, 0) \mathbf{E}^\top \mathbf{A}^{-1}(\mathbf{w}) \mathbf{E} \text{diag}(1, 0, \dots, 0) \right) \\ &= \text{trace} \left( \text{diag}(1, 0, \dots, 0) (\mathbf{E}^{-1} \mathbf{A}(\mathbf{w}) \mathbf{E}^{-\top})^{-1} \text{diag}(1, 0, \dots, 0) \right). \end{aligned}$$

Since  $\mathbf{E}$  is a constant matrix,  $\mathbf{E}^{-1} \mathbf{A}(\mathbf{w}) \mathbf{E}^{-\top}$  is also linear in  $w_1, \dots, w_{N-1}$ . Therefore, we can follow the transformation for  $A_s$ -optimality to get the SDP problem for  $c$ -optimality with  $\mathbf{a} = (1, 0, \dots, 0)^\top$  as

$$\begin{cases} \min_{v_1, \dots, v_q, w_1, \dots, w_{N-1}} & v_1 \\ \text{s.t.} & \tilde{\mathbf{D}}_1 \oplus \dots \oplus \tilde{\mathbf{D}}_q \oplus \mathbf{W}_N \succeq 0. \end{cases} \quad (4.23)$$

SeDuMi or CVX in MATLAB can be applied for finding solutions to (4.21), (4.22), and (4.23). Lemma 4.1 can be used to check for optimal designs. In practice, the following condition is often applied, as it is done similarly in Chapters 2 and 3,

$$h_j(\hat{\mathbf{w}}, \mathbf{L}) - h(\hat{\mathbf{w}}, \mathbf{L}) \leq \delta, \quad \text{for all } j = 1, \dots, N, \quad (4.24)$$

with a small positive  $\delta$ , say  $\delta = 10^{-5}$ .

## 4.5 Applications

We present three examples to show how our approach can find optimal designs for multi-response models, including one nonlinear and two linear models. Example 4.4 has three response variables and two independent variables, while Example 4.5 has two response variables and three independent variables. In Example 4.6, we show the method for finding locally optimal designs for bivariate Emax models. To present the results we have rounded optimal weights to 4 decimal places in the examples, but condition (4.24) should be checked using the numerical results in MATLAB.

**Example 4.4.** A chemical engineer is interested in studying how the two controllable variables, reaction time ( $x_1$ ) and reaction temperature ( $x_2$ ), influence the yield ( $y_1$ ), viscosity ( $y_2$ ) and molecular weight ( $y_3$ ) of a process (Montgomery, 2013, p496). Two quadratic and a linear response models are fitted to study their relationships, as in

Example 4.2, and the covariance matrix is given by

$$\Sigma_0 = \begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{pmatrix}.$$

The design space  $S_N$  includes the 9 points in Example 4.2 and 8 points inside the circle with radius  $\sqrt{2}$ . A-optimal designs are computed using SeDuMi, and representative A-optimal design weights are in Table 4.1. The results show the symmetric property of the optimal designs, which is consistent with the result in Example 4.2. We also notice that the optimal designs are not very sensitive to the values of  $\rho$ 's.  $\square$

Table 4.1: A-optimal designs in Example 4.4

Design points in $S_N$			$\rho_{12} = \rho_{13}$	$\rho_{12} = \rho_{13}$	$\rho_{12} = 0.1, \rho_{13} = 0.2,$
	$x_1$	$x_2$	$= \rho_{23} = 0.1$	$= \rho_{23} = 0.5$	$\rho_{23} = 0.5$
			optimal weights	optimal weights	optimal weights
$\mathbf{u}_1$ :	0	0	0.3252	0.3104	0.3176
$\mathbf{u}_2$ :	-1.0	-1.0	0.0988	0.1010	0.0999
$\mathbf{u}_3$ :	-1.0	1.0	0.0988	0.1010	0.0999
$\mathbf{u}_4$ :	1.0	-1.0	0.0988	0.1010	0.0999
$\mathbf{u}_5$ :	1.0	1.0	0.0988	0.1010	0.0999
$\mathbf{u}_6$ :	$-\sqrt{2}$	0	0.0699	0.0714	0.0707
$\mathbf{u}_7$ :	$\sqrt{2}$	0	0.0699	0.0714	0.0707
$\mathbf{u}_8$ :	0	$-\sqrt{2}$	0.0699	0.0714	0.0707
$\mathbf{u}_9$ :	0	$\sqrt{2}$	0.0699	0.0714	0.0707
$\mathbf{u}_{10}$ :	1.0	0	0	0	0
$\mathbf{u}_{11}$ :	0	1.0	0	0	0
$\mathbf{u}_{12}$ :	-1.0	0	0	0	0
$\mathbf{u}_{13}$ :	0	-1.0	0	0	0
$\mathbf{u}_{14}$ :	$\sqrt{2}/2$	$\sqrt{2}/2$	0	0	0
$\mathbf{u}_{15}$ :	$-\sqrt{2}/2$	$\sqrt{2}/2$	0	0	0
$\mathbf{u}_{16}$ :	$-\sqrt{2}/2$	$-\sqrt{2}/2$	0	0	0
$\mathbf{u}_{17}$ :	$\sqrt{2}/2$	$-\sqrt{2}/2$	0	0	0

**Example 4.5.** Consider a two-response model with three independent variables investigated in Atashgah and Seifi (2007, 2009), where the mean functions have  $\mathbf{f}_1^\top(\mathbf{x}) = (1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_1^2, x_3^2)$  and  $\mathbf{f}_2^\top(\mathbf{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$ . We take a design space  $S_N$  containing 19 possible points in  $R^3$ , which are listed in Table 4.2. For  $\Sigma_0$  in (4.14), we compute A- and D-optimal designs for various values of  $\rho$ . D-

optimal designs are computed using CVX program in MATLAB (Grant and Boyd, 2013). The MATLAB program is given in the Appendix. Representative optimal design weights are reported in Table 4.2. The results indicate that the optimal designs depend on  $\Sigma_0$  only through  $|\rho|$ , which confirms the result in Theorem 4.2. Furthermore, the optimal designs are not sensitive to small changes in  $\rho$ . We also computed D-optimal designs for a design space  $S_N$  with 16 points in Atashgah and Seifi (2007) and our results are the same as theirs.

Table 4.2: A- and D-optimal designs in Example 4.5 for various values of  $\rho$

	Design points in $S_N$			A-optimal design weights			D-optimal design weights		
	$x_1$	$x_2$	$x_3$	$\rho = 0$	$\rho = \pm 0.1$	$\rho = \pm 0.5$	$\rho = 0$	$\rho = \pm 0.1$	$\rho = \pm 0.5$
$\mathbf{u}_1$ :	1.6800	0	0	0.0616	0.0610	0.0441	0.0599	0.0593	0.0469
$\mathbf{u}_2$ :	0	1.6800	0	0.0000	0.0000	0.0276	0.0000	0.0000	0.0009
$\mathbf{u}_3$ :	0	0	0	0.3773	0.3773	0.3640	0.0851	0.0850	0.0822
$\mathbf{u}_4$ :	1.7290	1.7270	-1.7030	0.0000	0.0000	0.0020	0.0000	0.0000	0.0000
$\mathbf{u}_5$ :	1.7280	-1.7290	-1.7200	0.0487	0.0484	0.0401	0.0805	0.0803	0.0757
$\mathbf{u}_6$ :	1.7290	1.7290	1.7290	0.0530	0.0530	0.0527	0.0890	0.0891	0.0896
$\mathbf{u}_7$ :	-1.7250	-1.7230	1.7150	0.0150	0.0146	0.0077	0.0671	0.0670	0.0662
$\mathbf{u}_8$ :	-1.7300	1.7210	1.7290	0.0271	0.0269	0.0269	0.0715	0.0713	0.0674
$\mathbf{u}_9$ :	1.7300	-1.7290	1.7290	0.0369	0.0367	0.0304	0.0748	0.0746	0.0712
$\mathbf{u}_{10}$ :	-1.7300	1.7300	0.0260	0.0578	0.0580	0.0589	0.0805	0.0806	0.0837
$\mathbf{u}_{11}$ :	1.7300	-1.7300	-0.0450	0.0064	0.0070	0.0246	0.0163	0.0169	0.0300
$\mathbf{u}_{12}$ :	-1.7290	-1.7300	-1.7280	0.0649	0.0647	0.0599	0.1056	0.1056	0.1056
$\mathbf{u}_{13}$ :	-1.7300	-0.0960	1.7300	0.0474	0.0478	0.0499	0.0354	0.0359	0.0460
$\mathbf{u}_{14}$ :	1.7290	1.7240	-1.7290	0.0377	0.0378	0.0384	0.0758	0.0759	0.0774
$\mathbf{u}_{15}$ :	-0.1540	1.7300	-1.7300	0.0822	0.0820	0.0669	0.0883	0.0882	0.0860
$\mathbf{u}_{16}$ :	-0.1010	-1.7300	1.7300	0.0694	0.0694	0.0709	0.0702	0.0703	0.0712
$\mathbf{u}_{17}$ :	1.7290	1.7290	1.7220	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\mathbf{u}_{18}$ :	-1.5168	-1.6182	0.6520	0.0146	0.0154	0.0350	0.0000	0.0000	0.0000
$\mathbf{u}_{19}$ :	0.1158	1.6289	1.5256	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4.3: A-optimal weights computed from SDP in (4.21) and Atashgah and Seifi (2009) (denoted by ATSE) with  $\text{trace}(\mathbf{A}^{-1}(\mathbf{w})) = 17.546$  and 18.012, respectively.

Design points	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	$\mathbf{u}_4$	$\mathbf{u}_5$	$\mathbf{u}_6$	$\mathbf{u}_7$	$\mathbf{u}_8$	$\mathbf{u}_9$	$\mathbf{u}_{10}$
SDP in (4.21)	0.0504	0.0124	0.3634	0.0000	0.0460	0.0544	0.0147	0.0323	0.0343	0.0575
ATSE	0.0536	0.0000	0.4080	0.0318	0.0456	0.0000	0.0000	0.0455	0.0243	0.0498
Design points	$\mathbf{u}_{11}$	$\mathbf{u}_{12}$	$\mathbf{u}_{13}$	$\mathbf{u}_{14}$	$\mathbf{u}_{15}$	$\mathbf{u}_{16}$	$\mathbf{u}_{17}$	$\mathbf{u}_{18}$	$\mathbf{u}_{19}$	
SDP in (4.21)	0.0174	0.0642	0.0374	0.0405	0.0769	0.0702	0.0000	0.0280	0.0000	
ATSE	0.0066	0.0796	0.0238	0.0000	0.0656	0.0687	0.0427	0.0544	0.0000	

Atashgah and Seifi (2009) also computed an A-optimal design when the covariance matrix is

$$\Sigma_0 = \begin{pmatrix} 2 & 0.4 \\ 0.4 & 1 \end{pmatrix}.$$

Their design is shown in Table 4.3, along with the A-optimal design found from solving the SDP problem in (4.21). The latter design has a smaller value for  $\text{trace}(\mathbf{A}^{-1}(\mathbf{w}))$ , which confirms the results in Theorems 4.5 and 4.6 and our correct SDP formulation in (4.21) for finding A-optimal designs.  $\square$

**Example 4.6.** Magnúsdóttir (2013) used a bivariate Emax model to investigate the efficacy and side-effects of a drug given by

$$y_1 = E_{\max} \frac{x}{x + ED_{50}} + \epsilon_1, \quad y_2 = S_{\max} \frac{x}{x + SD_{50}} + \epsilon_2,$$

respectively. Here  $E_{\max}$  is the maximal achievable effect from the drug,  $S_{\max}$  is the maximal realizable side-effect and  $x \geq 0$  is the dose level of the drug. The interesting parameters are  $ED_{50}$  and  $SD_{50}$  which are the dose levels that give half maximal effect and side-effect, respectively. For this bivariate nonlinear model, we let  $\boldsymbol{\theta}_1^\top = (E_{\max}, ED_{50})$ ,  $\boldsymbol{\theta}_2^\top = (S_{\max}, SD_{50})$ ,  $\sigma_1^2 = \text{Var}(\epsilon_1)$ ,  $\sigma_2^2 = \text{Var}(\epsilon_2)$  and  $\rho = \text{Cov}(\epsilon_1, \epsilon_2)$ . The range of dose of interest is  $[0, 500]$ .

We construct locally D-optimal designs for estimating the model parameters and a  $c$ -optimal design for estimating utility of a dose. Let the nominal values for the model parameters  $E_{\max}, S_{\max}, ED_{50}, SD_{50}$  be  $E_{\max}^*, S_{\max}^*, ED_{50}^*, SD_{50}^*$ , respectively. To apply our methods in Sections 4.2 and 4.4, we first substitute  $\mathbf{f}_1(x)$  and  $\mathbf{f}_2(x)$  by  $\mathbf{z}_1(x)$  and  $\mathbf{z}_2(x)$ , respectively, where

$$\mathbf{z}_1(x) = \frac{\partial}{\partial \boldsymbol{\theta}_1} \left( E_{\max} \frac{x}{x + ED_{50}} \right) \Big|_{\boldsymbol{\theta}_1 = (E_{\max}^*, ED_{50}^*)^\top},$$

$$\mathbf{z}_2(x) = \frac{\partial}{\partial \boldsymbol{\theta}_2} \left( S_{\max} \frac{x}{x + SD_{50}} \right) \Big|_{\boldsymbol{\theta}_2 = (S_{\max}^*, SD_{50}^*)^\top}$$

To implement our strategy, we next discretize the dose interval to form a discrete design space by  $S_N = \{500(j-1)/(N-1), j = 1, \dots, N\}$  with  $N = 10001$ . Table 4.4 displays locally  $D$ -optimal designs for various parameter values. We observe that (i) the support points of the  $D$ -optimal designs are clustered around two or three points, (ii) they all include the extreme dose 500 as a support point, and (iii) the correlation  $\rho$  between the two responses seems to have influence on the optimal designs. When  $\rho = 0$ , the optimal designs are not affected by the ratio  $\sigma_2^2/\sigma_1^2$ . We also observe that as  $SD_{50}^*/ED_{50}^*$  increases, signifying that the difference between the

two models increases, the number of support points tends to increase.

Table 4.4: Locally  $D$ -optimal designs for the bivariate Emax model (Example 4.6) when the nominal values are  $ED_{50}^* = 1$ ,  $E_{\max}^* = S_{\max}^* = 1$  and selected values of  $SD_{50}^*$ . The support points are  $x_1, x_2, x_3$  and 500 with corresponding weights  $w_1, w_2, w_3$  and  $1 - w_1 - w_2 - w_3$ . If a point has zero weight, then it is not specified in the table.

$SD_{50}^*$		$\rho$ ( $\frac{\sigma_2^2}{\sigma_1^2} = 1$ )			$\frac{\sigma_2^2}{\sigma_1^2}$ ( $\rho = 0$ )		
		0	$\pm 0.5$	$\pm 0.7$	0.5	1.5	5.0
2	$x_1$	1.4000	1.4000	1.4000	1.4000	1.4000	1.4000
	$w_1$	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
3	$x_1$	1.7000	1.7000	1.7000	1.7000	1.7000	1.7000
	$w_1$	0.3618	0.3390	0.3142	0.3618	0.3618	0.3618
	$x_2$	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500
	$w_2$	0.1382	0.1610	0.1858	0.1382	0.1382	0.1382
5	$x_1$	2.2000	1.3500	1.0500	2.2000	2.2000	2.2000
	$w_1$	0.3755	0.2757	0.2611	0.3755	0.3755	0.3755
	$x_2$	2.2500	4.3500	5.4500	2.2500	2.2500	2.2500
	$w_2$	0.1245	0.2465	0.2472	0.1245	0.1245	0.1245
	$x_3$			5.5000			
	$w_3$			0.0367			

Magnusdottir (2013) proposed with justification a composite utility index (CUI)  $CUI(x) = k_1 y_1 - k_2 y_2$  to measure the utility of using the dose  $x$ . As before  $y_1$  represents efficacy and  $y_2$  represents side effect and,  $k_1$  and  $k_2$  are two user-specified positive constants. The goal is to find the most desirable dose that maximizes  $E[CUI(x)]$ . Let  $\boldsymbol{\theta}^\top = (\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_2^\top)$  and let  $h_c(\boldsymbol{\theta}) = \max_x E[CUI(x)]$ . Magnusdottir (2013) provided selected locally  $c$ -optimal designs that minimize the variance of  $h_c(\hat{\boldsymbol{\theta}})$ . Table 4.5 shows selected  $c$ -optimal designs from our algorithm, where we observe that their support points are also clustered around two or three points. However, the  $c$ -optimal designs are more sensitive to the true parameter values than the  $D$ -optimal designs. In addition, the  $D$ -optimal designs depend only on  $|\rho|$ , but the  $c$ -optimal designs depend on  $\rho$ . The results are consistent with those in Magnusdottir (2013) with small differences due to the discretized design space.

The designs in Magnusdottir (2013) were derived by informative guessing, which can be difficult for practitioners. Our SDP-based algorithms can provide an effective and systematic way for finding optimal designs, which remain very efficient even for  $N$  as large as 10001 in this example.  $\square$

Table 4.5: Locally  $c$ -optimal designs for the bivariate Emax model (Example 4.6) when the nominal values for the model parameters are  $ED_{50}^* = 1$ ,  $E_{\max}^* = S_{\max}^* = 1$ ,  $k_1 = k_2 = 1$  and selected values of  $SD_{50}^*$ . The support points of the optimal designs are  $x_1, x_2, x_3$  and 500 with corresponding weights  $w_1, w_2, w_3$  and  $1 - w_1 - w_2 - w_3$ .

$SD_{50}^*$		$\rho \left( \frac{\sigma_2^2}{\sigma_1^2} = 1 \right)$				$\frac{\sigma_2^2}{\sigma_1^2} \left( \rho = 0 \right)$		
		-0.5	0	0.5	0.7	0.5	1.5	5
2	$x_1$	1.0500	1.1000	1.1000	1.0500	0.9500	1.2000	1.7500
	$w_1$	0.0863	0.3948	0.5116	0.2534	0.3801	0.0700	0.0933
	$x_2$	1.1000			1.1000		1.2500	1.8000
	$w_2$	0.1935			0.3176		0.3343	0.3494
3	$x_1$	1.2500	1.2500	0.8500	0.6500	1.0500	1.4000	1.8500
	$w_1$	0.2325	0.1331	0.4744	0.4784	0.1431	0.1617	0.4908
	$x_2$	1.3000	1.3000	6.0000	6.4000	1.1000	1.4500	4.8500
	$w_2$	0.1400	0.3570	0.2194	0.3408	0.3269	0.3425	0.0791
5	$x_1$	1.3500	0.9500	0.7000	0.6000	0.9500	0.9500	0.9500
	$w_1$	0.3951	0.4363	0.4745	0.4948	0.4620	0.4198	0.3679
	$x_2$	7.2500	8.7500	8.7000	8.0500	8.7000	8.8500	9.0500
	$w_2$	0.0857	0.0524	0.3837	0.4186	0.1561	0.3144	0.2976
	$x_3$		8.8000					9.1000
	$w_3$		0.1995					0.2196

## 4.6 Multi-response generalized linear models

SDP can also be used to generate optimal designs for GLMs. We apply our method for an application discussion in Kpamegan (1998), where he found D-optimal designs for making accurate inference on the probabilities of patients experiencing adverse events affecting the liver or the heart. Each of the cancer patients receiving chemotherapy received two drugs, so there were four possible binary outcomes.

Maximum likelihood estimator (MLE) is often used to estimate unknown parameters in GLMs. Let  $l(\boldsymbol{\theta}|\mathbf{y}, \xi)$  denote the log-likelihood function for a GLM, where  $\boldsymbol{\theta} \in R^q$  is the unknown parameter vector,  $\mathbf{y}$  is the vector of multi-response variables. If we have resources to take  $n$  observations and  $\xi$  is the distribution of design points  $\mathbf{x}_1, \dots, \mathbf{x}_n \in R^p$ , the  $(i, k)^{th}$  element of the Fisher information matrix of the MLE is  $A_{i,k} = -E \left( \frac{\partial^2 l(\boldsymbol{\theta}|\mathbf{y}, \xi)}{\partial \theta_i \partial \theta_k} \right)$ ,  $i, k = 1, \dots, q$ , and the expectation is taken over the distribution of  $\mathbf{y}$ . Using the discrete design space  $S_N$  for  $\mathbf{x}$  and the discrete distribution  $\xi$  in (4.7), we can write all the elements  $A_{i,k}$  in  $\mathbf{A}(\xi, \boldsymbol{\theta})$  as linear functions of  $\mathbf{w}$ . Locally optimal designs minimize some scalar functions of  $\mathbf{A}(\xi, \boldsymbol{\theta})$  over  $\mathbf{w}$ . We now use CVX and SeDuMi to find D-optimal, A-optimal and other optimal designs for GLMs and discuss one such application in some detail.

Kpamegan (1998) found locally D-optimal designs for a bivariate probit response model to study the relationship between two toxic responses  $y_1$  and  $y_2$  and the levels of two drugs,  $x_1$  and  $x_2$ . The two binary response variables are  $y_1$  and  $y_2$ , which take on values 1 for toxicity and 0 for non-toxicity responses. These bivariate responses  $F_{ij}(x_1, x_2) = P(y_1 = i, y_2 = j | (x_1, x_2))$ ,  $i, j = 0, 1$ , are modeled using the bivariate normal probability density function (pdf)  $\phi(u, v)$  with parameter vector  $\boldsymbol{\theta} = (\mu_1, \sigma_1, \mu_2, \sigma_2, \rho)^\top$ , where  $\mu_1$  and  $\mu_2$  are the location parameters,  $\sigma_1$  and  $\sigma_2$  are the scale parameters and  $\rho$  is the correlation parameter for the two response variables. For example, we have  $F_{11}(x_1, x_2) = \int_{-\infty}^{x_2} \int_{-\infty}^{x_1} \phi(u, v) du dv$ .

Kpamegan (1998) found D-optimal designs for the following three cases:

Case (i): The two response variables are independent and  $\rho = 0$ . The parameter vector is reduced to  $\boldsymbol{\theta} = (\mu_1, \sigma_1, \mu_2, \sigma_2)^\top$ .

Case (ii): The two response variables are independent and  $\sigma_1 = \sigma_2 = \sigma$ . The

parameter vector is reduced to  $\boldsymbol{\theta} = (\mu_1, \mu_2, \sigma)^\top$ .

Case (iii): The two response variables are correlated and the parameter vector is  $\boldsymbol{\theta} = (\mu_1, \sigma_1, \mu_2, \sigma_2, \rho)^\top$ .

We construct D-optimal designs for the three cases as follows. Let  $S_N = \{(x_{1j}, x_{2j}), j = 1, \dots, N\}$  contain  $N$  possible combinations of the levels of the two drugs. Define transformed variables  $z_{ij} = (x_{ij} - \mu_i)/\sigma_i$ ,  $j = 1, \dots, N$ ,  $i = 1, 2$ . In Case (i), the Fisher information matrix is given by

$$\mathbf{A}_1(\xi, \boldsymbol{\theta}) = N \sum_{j=1}^N w_j \begin{pmatrix} \frac{1}{\sigma_1^2} m(z_{1j}) & \frac{1}{\sigma_1^2} z_{1j} m(z_{1j}) & 0 & 0 \\ \frac{1}{\sigma_1^2} z_{1j} m(z_{1j}) & \frac{1}{\sigma_1^2} z_{1j}^2 m(z_{1j}) & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_2^2} m(z_{2j}) & \frac{1}{\sigma_2^2} z_{2j} m(z_{2j}) \\ 0 & 0 & \frac{1}{\sigma_2^2} z_{2j} m(z_{2j}) & \frac{1}{\sigma_2^2} z_{2j}^2 m(z_{2j}) \end{pmatrix},$$

where  $m(\cdot) = \phi^2(\cdot)/[\Phi(\cdot)(1 - \Phi(\cdot))]$ , and  $\phi(\cdot)$  and  $\Phi(\cdot)$  are the pdf and cumulative distribution function of the univariate standard normal random variable. In Case (ii), the Fisher information matrix is given by

$$\mathbf{A}_2(\xi, \boldsymbol{\theta}) = \frac{N}{\sigma^2} \sum_{j=1}^N w_j \begin{pmatrix} m(z_{1j}) & 0 & z_{1j} m(z_{1j}) \\ 0 & m(z_{2j}) & z_{2j} m(z_{2j}) \\ z_{1j} m(z_{1j}) & z_{2j} m(z_{2j}) & z_{1j}^2 m(z_{1j}) + z_{2j}^2 m(z_{2j}) \end{pmatrix}.$$

Both  $\mathbf{A}_1(\xi, \boldsymbol{\theta})$  and  $\mathbf{A}_2(\xi, \boldsymbol{\theta})$  are linear in  $\mathbf{w}$ . As in Kpamegan (1998), we find optimal designs in terms of standardized variables  $z_1$  and  $z_2$ , where  $z_i = (x_i - \mu_i)/\sigma_i$ ,  $i = 1, 2$ . Let  $z_{il} = -3 + 6(l - 1)/(N_0 - 1)$ ,  $l = 1, \dots, N_0$ ,  $i = 1, 2$ . The design space  $S_N \subset [-3, 3] \times [-3, 3]$  contains  $N = N_0^2$  grid points formed by  $z_{1l}$  and  $z_{2l}$ . Using CVX program we compute the D-optimal designs for Case (i) with  $N = 101^2$  and Case (ii) with  $N = 201^2$  and obtain the results in Table 4.6. It took less than 2 seconds for Case (i) and 2 minutes for Case (ii). Our results are similar to those in Kpamegan (1998), which are also listed in Table 4.6. The search is reduced to one dimensional in Kpamegan (1998) after analyzing the information matrix and using the symmetry of the D-optimal designs, which worked well for Cases (i) and (ii). However, if the information matrix is complicated, the method discussed in Kpamegan (1998) may not work and we will show this for Case (iii).

Table 4.6: D-optimal designs for the bivariate probit response model

Case/Method	Support points [weights]	function value
Case (i): $\sigma_1 = \sigma_2 = 1$ CVX ( $N = 101^2$ )	$(-1.140, -1.140), (-1.140, 1.140), (1.140, -1.140), (1.140, 1.140)$ [0.25], [0.25], [0.25], [0.25]	$\det(\mathbf{A}_1/N) = 0.0394748$
Kpamegan (1998)	$(-1.138, -1.138), (-1.138, 1.138), (1.138, -1.138), (1.138, 1.138)$ [0.25], [0.25], [0.25], [0.25]	$\det(\mathbf{A}_1/N) = 0.0394752$
Case (ii): $\sigma = 1$ CVX ( $N = 201^2$ )	$(-0.940, -0.940), (-0.940, 0.940), (0.940, -0.940), (0.940, 0.940)$ [0.25], [0.25], [0.25], [0.25]	$\det(\mathbf{A}_2/N) = 0.1703124$
Kpamegan (1998)	$(-0.937, -0.937), (-0.937, 0.937), (0.937, -0.937), (0.937, 0.937)$ [0.25], [0.25], [0.25], [0.25]	$\det(\mathbf{A}_2/N) = 0.1703150$

For nonlinear model, the optimal designs depend on the true values of unknown parameters. If the initial guess for the true value is wrong, we would like to do the sensitivity analysis by defining the D-efficiency. Let

$$l_1(\xi^*; \boldsymbol{\theta}^*) = \det(\mathbf{A}_1(\xi^*, \boldsymbol{\theta}^*)), \quad l_2(\xi^{**}; \boldsymbol{\theta}^*) = \det(\mathbf{A}_1(\xi^{**}, \boldsymbol{\theta}^*)).$$

where  $\boldsymbol{\theta}^*$  are the true values for parameters and  $\xi^*$  is the calculated optimal design under the true values  $\boldsymbol{\theta}^*$ .  $\xi^{**}$  is the optimal design calculated under the wrong guess of parameters.

And the D-efficiency measures as

$$Eff = \frac{l_2(\xi^{**}; \boldsymbol{\theta}^*)}{l_1(\xi^*; \boldsymbol{\theta}^*)}$$

Table 4.7 shows some representative results of D-efficiency for Case (i) when the true values of parameters are  $\boldsymbol{\theta}^* = (0, 1.2, 0, 0.95)$ . In practice, It is efficient for small disturbances.

Table 4.7: D-efficiency for Case (i)

wrong guess of unknown parameters	D-efficiency
(0,1,0,1)	0.9357
(0,1,0,0.8)	0.8846
(0.1,1,0.1,0.8)	0.8743
(0.1,1.3,0.1,0.75)	0.8846
(0.1,1.4,0.1,0.7)	0.8036

In Case (iii), the Fisher information matrix  $\mathbf{A}_3(\xi, \boldsymbol{\theta})$  ( $5 \times 5$  matrix) is also linear in  $\mathbf{w}$ . We have

$$\begin{aligned} F_{11}(z_{1j}, z_{2j}) &= \Phi\left(\frac{x_{1j} - \mu_1}{\sigma_1}, \frac{x_{2j} - \mu_2}{\sigma_2}\right), \\ F_{10}(z_{1j}, z_{2j}) &= \Phi\left(\frac{x_{1j} - \mu_1}{\sigma_1}\right) - \Phi\left(\frac{x_{1j} - \mu_1}{\sigma_1}, \frac{x_{2j} - \mu_2}{\sigma_2}\right), \\ F_{01}(z_{1j}, z_{2j}) &= \Phi\left(\frac{x_{2j} - \mu_2}{\sigma_2}\right) - \Phi\left(\frac{x_{1j} - \mu_1}{\sigma_1}, \frac{x_{2j} - \mu_2}{\sigma_2}\right), \\ F_{00}(z_{1j}, z_{2j}) &= 1 - \Phi\left(\frac{x_{1j} - \mu_1}{\sigma_1}\right) - \Phi\left(\frac{x_{1j} - \mu_1}{\sigma_1}\right) + \Phi\left(\frac{x_{1j} - \mu_1}{\sigma_1}, \frac{x_{2j} - \mu_2}{\sigma_2}\right), \end{aligned}$$

where  $\Phi(z_{1j}, z_{2j})$  is the standard bivariate normal cumulative distribution function.

In Case (iii), the Fisher's information matrix is given by

$$\mathbf{A}_3(\xi, \boldsymbol{\theta}) = \frac{N}{\sigma^2} \sum_{j=1}^N w_j \begin{pmatrix} \mathbf{I}_{11}(z_{1j}, z_{2j}) & \mathbf{I}_{12}(z_{1j}, z_{2j}) & \mathbf{I}_{13}(z_{1j}, z_{2j}) \\ \mathbf{I}_{12}^\top(z_{1j}, z_{2j}) & \mathbf{I}_{22}(z_{1j}, z_{2j}) & \mathbf{I}_{23}(z_{1j}, z_{2j}) \\ \mathbf{I}_{13}^\top(z_{1j}, z_{2j}) & \mathbf{I}_{23}^\top(z_{1j}, z_{2j}) & \mathbf{I}_{33}(z_{1j}, z_{2j}) \end{pmatrix},$$

where

$$\mathbf{I}_{11}(z_{1j}, z_{2j}) = \begin{pmatrix} \frac{1}{\sigma_1^2} K(z_{1j}, z_{2j}) & \frac{z_{1j}}{\sigma_1^2} K(z_{1j}, z_{2j}) \\ \frac{z_{1j}}{\sigma_1^2} K(z_{1j}, z_{2j}) & \frac{z_{1j}^2}{\sigma_1^2} K(z_{1j}, z_{2j}) \end{pmatrix}, \quad \mathbf{I}_{13}(z_{1j}, z_{2j}) = \begin{pmatrix} \frac{1}{\sigma_1} R(z_{1j}, z_{2j}) \\ \frac{z_{1j}}{\sigma_1} R(z_{1j}, z_{2j}) \end{pmatrix},$$

$$\mathbf{I}_{22}(z_{1j}, z_{2j}) = \begin{pmatrix} \frac{1}{\sigma_2^2} S(z_{1j}, z_{2j}) & \frac{z_{2j}}{\sigma_2^2} S(z_{1j}, z_{2j}) \\ \frac{z_{2j}}{\sigma_2^2} S(z_{1j}, z_{2j}) & \frac{z_{2j}^2}{\sigma_2^2} S(z_{1j}, z_{2j}) \end{pmatrix}, \quad \mathbf{I}_{23}(z_{1j}, z_{2j}) = \begin{pmatrix} \frac{1}{\sigma_2} U(z_{1j}, z_{2j}) \\ \frac{z_{2j}}{\sigma_2} U(z_{1j}, z_{2j}) \end{pmatrix},$$

$$\mathbf{I}_{12}(z_{1j}, z_{2j}) = \begin{pmatrix} \frac{1}{\sigma_1 \sigma_2} P(z_{1j}, z_{2j}) & \frac{z_{2j}}{\sigma_1 \sigma_2} P(z_{1j}, z_{2j}) \\ \frac{z_{1j}}{\sigma_1 \sigma_2} P(z_{1j}, z_{2j}) & \frac{z_{1j} z_{2j}}{\sigma_1 \sigma_2} P(z_{1j}, z_{2j}) \end{pmatrix}, \quad \mathbf{I}_{33}(z_{1j}, z_{2j}) = V(z_{1j}, z_{2j}),$$

and the functions  $K$ ,  $P$ ,  $S$ ,  $R$ ,  $U$  and  $V$  are

$$K(z_{1j}, z_{2j}) = \left( \frac{\partial F_{11}}{\partial z_{1j}} \right)^2 \left( \frac{1}{F_{11}} + \frac{1}{F_{01}} \right) + \left[ \phi(z_{1j}) - \frac{\partial F_{11}}{\partial z_{1j}} \right]^2 \left( \frac{1}{F_{10}} + \frac{1}{F_{00}} \right),$$

$$\begin{aligned} P(z_{1j}, z_{2j}) &= \frac{1}{F_{11}} \left( \frac{\partial F_{11}}{\partial z_{1j}} \right) \left( \frac{\partial F_{11}}{\partial z_{2j}} \right) - \frac{1}{F_{10}} \left( \frac{\partial F_{11}}{\partial z_{2j}} \right) \left[ \phi(z_{1j}) - \frac{\partial F_{11}}{\partial z_{1j}} \right] \\ &\quad - \frac{1}{F_{01}} \left( \frac{\partial F_{11}}{\partial z_{1j}} \right) \left[ \phi(z_{2j}) - \frac{\partial F_{11}}{\partial z_{2j}} \right] + \frac{1}{F_{00}} \left[ \phi(z_{1j}) - \frac{\partial F_{11}}{\partial z_{1j}} \right] \\ &\quad \times \left[ \phi(z_{2j}) - \frac{\partial F_{11}}{\partial z_{2j}} \right], \end{aligned}$$

$$S(z_{1j}, z_{2j}) = \left( \frac{\partial F_{11}}{\partial z_{2j}} \right)^2 \left( \frac{1}{F_{11}} + \frac{1}{F_{10}} \right) + \left[ \phi(z_{2j}) - \frac{\partial F_{11}}{\partial z_{2j}} \right]^2 \left( \frac{1}{F_{01}} + \frac{1}{F_{00}} \right),$$

$$R(z_{1j}, z_{2j}) = \frac{\partial F_{11}}{\partial \rho} \left[ - \left( \frac{\partial F_{11}}{\partial z_{1j}} \right) \left( \frac{1}{F_{11}} + \frac{1}{F_{10}} + \frac{1}{F_{01}} + \frac{1}{F_{00}} \right) + \phi(z_{1j}) \left( \frac{1}{F_{10}} + \frac{1}{F_{00}} \right) \right],$$

$$U(z_{1j}, z_{2j}) = \frac{\partial F_{11}}{\partial \rho} \left[ - \left( \frac{\partial F_{11}}{\partial z_{2j}} \right) \left( \frac{1}{F_{11}} + \frac{1}{F_{10}} + \frac{1}{F_{01}} + \frac{1}{F_{00}} \right) + \phi(z_{2j}) \left( \frac{1}{F_{01}} + \frac{1}{F_{00}} \right) \right],$$

$$V(z_{1j}, z_{2j}) = \left( \frac{\partial F_{11}}{\partial \rho} \right)^2 \left( \frac{1}{F_{11}} + \frac{1}{F_{10}} + \frac{1}{F_{01}} + \frac{1}{F_{00}} \right).$$

Notice that there are typos in expressions  $R$ ,  $U$  and  $V$  in Kpamegan (1998, page 70), and each expression is missing a factor. Expressions  $R$ ,  $U$  and  $V$  should include factors  $\frac{\partial F_{11}}{\partial \rho}$ ,  $\frac{\partial F_{11}}{\partial \rho}$  and  $\left(\frac{\partial F_{11}}{\partial \rho}\right)^2$ , respectively.  $\mathbf{A}_3(\xi, \boldsymbol{\theta})$  is more complicated than  $\mathbf{A}_1(\xi, \boldsymbol{\theta})$  or  $\mathbf{A}_2(\xi, \boldsymbol{\theta})$ , and it is hard to find the D-optimal design for Case (iii). Because of computational difficulties, Kpamegan (1998) only considered symmetric designs, which have equal weights on four support points  $(-a, -a)$ ,  $(-a, a)$ ,  $(a, -a)$ ,  $(a, a)$  for some positive number  $a$ , to find the D-optimal design. Using CVX, we can find the D-optimal design without any assumption on the design. Some representative results are given in Table 4.8, which shows that the optimal designs found by CVX are better than those in Kpamegan (1998). For  $\rho > 0$ , the D-optimal designs do not have equal weights on support points  $(-a, -a)$ ,  $(-a, a)$ ,  $(a, -a)$ ,  $(a, a)$ . In fact, they have equal weights on points  $(-a, -a)$  and  $(a, a)$ .

Table 4.8: D-optimal designs for different values of  $\rho$  for the GLM

D-optimal designs by CVX					D-optimal designs in Kpamegan (1998)	
$\rho$	$N$	support points	weights	$\det(\mathbf{A}_3/N)$	support points (weights)	$\det(\mathbf{A}_3/N)$
0	$61^2$	$(-1.0000, -1.0000)$ $(-1.0000, 1.0000)$ $(-0.9000, -0.9000)$ $(-0.9000, 0.9000)$ $(0.9000, -0.9000)$ $(0.9000, 0.9000)$ $(1.0000, -1.0000)$ $(1.0000, 1.0000)$	0.0838 0.0838 0.1662 0.1662 0.1662 0.1662 0.0838 0.0838	0.0072	$(1.138, 1.138)$ (0.2500) $(-1.138, 1.138)$ (0.2500) $(1.138, -1.138)$ (0.2500) $(-1.138, -1.138)$ (0.2500)	0.0061
0.1	$101^2$	$(-0.9600, -0.9600)$ $(-0.9000, -0.9000)$ $(0.9000, 0.9000)$ $(0.9600, 0.9600)$	0.4390 0.0610 0.0610 0.4390	0.0077	$(0.94, 0.94)$ (0.2500) $(-0.94, 0.94)$ (0.2500) $(0.94, -0.94)$ (0.2500) $(-0.94, -0.94)$ (0.2500)	0.0074
0.5	$11^2$	$(-1.2000, -1.2000)$ $(-0.6000, -0.6000)$ $(0.6000, 0.6000)$ $(1.2000, 1.2000)$	0.3264 0.1736 0.1736 0.3264	0.0134	$(0.89, 0.89)$ (0.2500) $(-0.89, 0.89)$ (0.2500) $(0.89, -0.89)$ (0.2500) $(-0.89, -0.89)$ (0.2500)	0.0119

$T$ -invariant A- and D-optimal designs can also be investigated for GLMs. This can be seen by first writing the information matrix as  $\mathbf{A}(\xi, \boldsymbol{\theta}) = \sum_{j=1}^N w_j \mathbf{B}(\mathbf{u}_j; \boldsymbol{\theta})$ , for some symmetric and PSD matrices  $\mathbf{B}(\mathbf{u}_j; \boldsymbol{\theta})$ .

**Theorem 4.7.** *Let  $T$  be an one-to-one function defined on a given design space  $S_N$  and  $T^2$  is an identity map. If there exists a diagonal matrix  $\mathbf{Q}$  with diagonal elements all equal to  $-1$  or  $1$  such that  $\mathbf{B}(T(\mathbf{u}_j); \boldsymbol{\theta}) = \mathbf{Q}\mathbf{B}(\mathbf{u}_j; \boldsymbol{\theta})\mathbf{Q}$ , for all  $\mathbf{u}_j \in S_N$ ,  $j = 1, \dots, N$ , then there exists  $T$ -invariant A- and D-optimal designs.*

The proof is similar to that of Theorem 4.4 and is omitted. For  $\mathbf{A}_1(\xi, \boldsymbol{\theta})$  and  $\mathbf{A}_2(\xi, \boldsymbol{\theta})$ , one can show that there exists  $T$ -invariant A- and D-optimal designs under transformations  $T_1$  and  $T_3$  in Example 4.2. It follows that the D-optimal designs have equal weights at points  $(-a, -a)$ ,  $(-a, a)$ ,  $(a, -a)$ ,  $(a, a)$ , which is consistent with the results in Table 4.6 and in Kpamegan (1998). However, for  $\mathbf{A}_3(\xi, \boldsymbol{\theta})$  with  $\rho > 0$ , one can only verify that there exists  $T$ -invariant A- and D-optimal designs under transformation  $T_1(\mathbf{u}) = -\mathbf{u}$ . This confirms the results in Table 4.8.

## 4.7 Conclusion

We have investigated A-,  $A_s$ - and D-optimal designs for multi-response regression models using the invariance property and dependence on the error covariance matrix  $\Sigma_0$ . Symmetry of optimal designs can be examined through the  $T$ -invariance property obtained, and it can be applied for various models and design spaces. In general, optimal designs may depend on  $\Sigma_0$ , but we have showed there are situations that optimal designs are independent of  $\Sigma_0$ . When optimal designs depend on  $\Sigma_0$ , our numerical results indicate that some optimal designs may not be sensitive to small changes in  $\Sigma_0$ .

We have also developed numerical algorithms for finding A-,  $A_s$ - and c-optimal designs based on methods for SDP problems. They are effective and efficient, and they can be used for finding optimal designs systematically for any discrete design spaces. They can be also extended to other optimality criteria such as I-optimality and L-optimality. D-optimal designs can be computed using CVX program. All the results can be applied for linear and nonlinear models and generalized linear models.

# Chapter 5

## Discussions

In this dissertation, we have derived several properties for various optimal design criteria and developed several algorithms for computing optimal regression designs. The big picture is shown as a flow chart in Figure 5.1.

Chapter 2 proposed an algorithm based on SeDuMi to solve various optimal design problems. A general transformation for A-,  $A_s$ -, c-, I- and L-optimal design problems was derived to transform the design problems into corresponding SDP problems. We showed step by step how we can use SeDuMi in MATLAB to solve these SDP problems, and we derived the Kiefer-Wolfowitz equivalence theorem for this algorithm to verify the optimality of the results. We presented several examples, including polynomial and nonlinear regression models, cubic regression model with unequal error variances, and models with linear constraints on weights. Our algorithm works efficiently for all these examples.

Chapter 3 investigated the properties and numerical algorithms for optimal design problems under the SLSE. We derived the expressions for A- and D-optimality criteria and developed algorithms based on SeDuMi and CVX programs for solving optimal design problems. The transformation invariance property for A-optimal designs, the equivalence results and a property of D-optimal designs were derived. We compared multiplicative algorithm with CVX and SeDuMi algorithms by applying them for various applications. CVX and SeDuMi programs are faster than the multiplicative algorithm, especially for the nonlinear models. Several models including second-order models with two variables, compartmental model, Michaelis-Menten model and Emax model were discussed for both A- and D-optimality criteria. When the error distri-

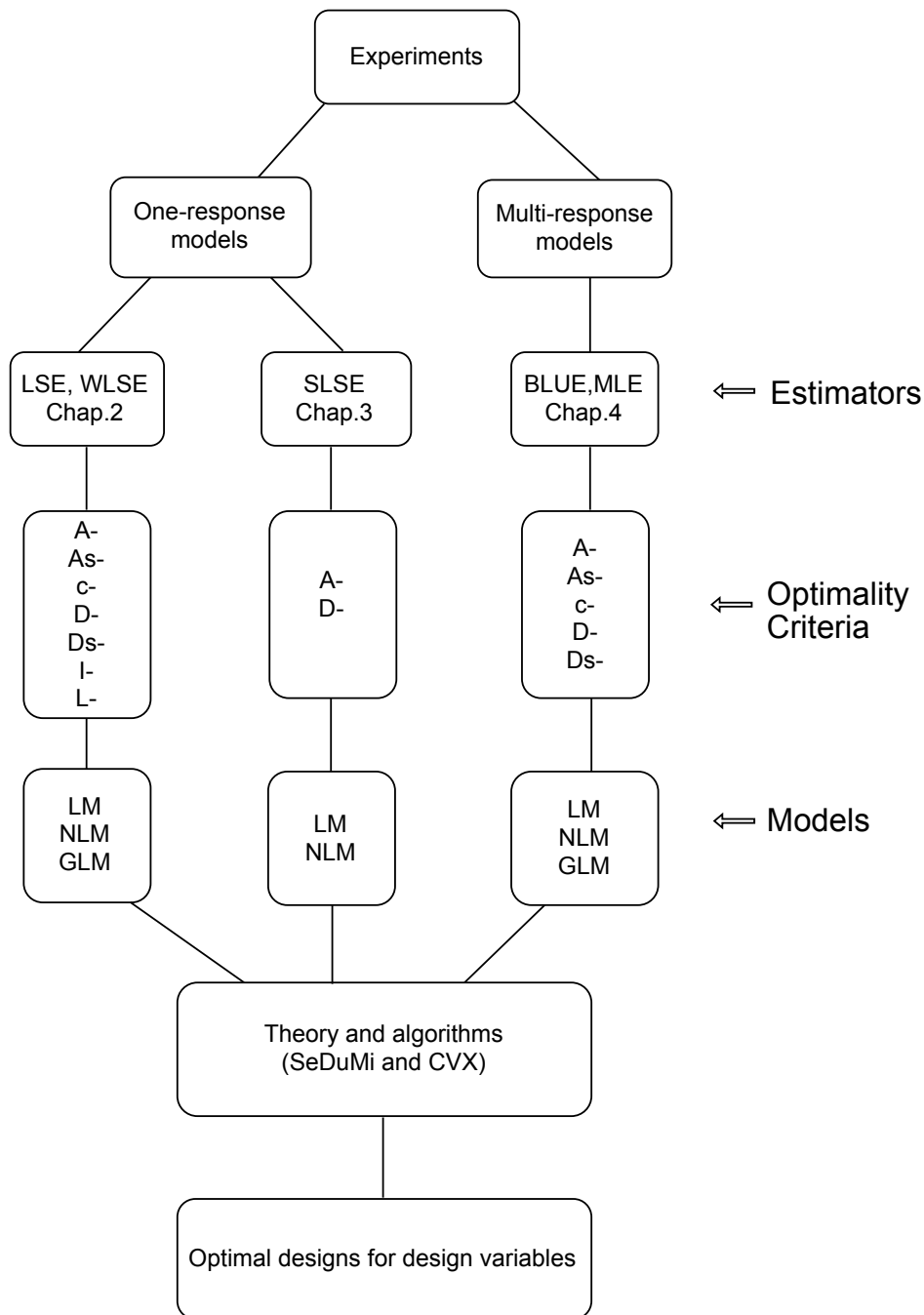


Figure 5.1: Various optimal designs studied in this dissertation

bution is asymmetric, our results showed that the optimal designs under the SLSE are more efficient than the optimal designs under the LSE.

Chapter 4 studied various optimal designs for multi-response models. The transformation invariance property and dependence on the covariance matrix of the correlated errors were derived for A- and D-optimal designs. We constructed a transformation to connect A-optimal design problems with SDP problems, and proposed a numerical algorithm based on SeDuMi program to compute A-optimal designs. We also corrected the error in Atashgah and Seifi (2009). Several examples including two-response models with three variables, bivariate Emax model and GLMs are computed for A-, c- and D-optimal designs.

CVX and SeDuMi programs are powerful for computing optimal designs. Currently our algorithms are effective and efficient for  $N$  as large as 10000.

There are still open issues for further research in the field of optimal regression designs. We will mention two issues directly related to this dissertation.

One issue is about the minimum number of support points in optimal designs. The number of support points for some regression models and design criteria, has been found to be either  $q$  or  $q + 1$ , where  $q$  is the number of unknown parameters. In the Michaelis-Menten model, from the theory in Yang and Stufken (2012), we know that the number of support points is two for LSE and one of the support points is the upper bound of the design space for the D-optimal designs. Under SLSE, from the numerical results in Yin and Zhou (2016), the number of support points is three when  $t$  (a measure of skewness of error distribution) is large. For small  $t$ , the number of support points is still two. One further research issue is to study the theoretical number of support points for regression models under the SLSE. It seems that the numbers depend on the values of  $t$ ,  $q$ , the model and the design space.

Another issue is about the robustness and extension of optimal designs in Chapter 4. For the multi-response model with correlated errors, we assume that the covariance matrix  $\Sigma_0$  is known. We studied the multi-response regression models under the assumption that all the response variables are correlated in the same run and independent for different runs. If we relax this assumption and consider the case

in which  $\Sigma_0$  is unknown, one possible future research issue would be to study the robust regression design if there is a misspecification in the covariance matrix  $\Sigma_0$ . In the Chapter 4, we only studied the A-,  $A_s$ -, c- and D-optimal design problems for multi-response regression models. How to generalize an optimal algorithm for other design criteria, such as I- or L-optimality criteria, is worth further attention too.

# Appendix: MATLAB Codes

- A1. MATLAB program using SeDuMi for Example 2.2. (c-optimality)
- A2. MATLAB program using SeDuMi for Example 2.5. (I-optimality)
- A3. MATLAB program using CVX for Example 3.5. (D-optimality)
- A4. MATLAB program using multiplicative algorithm for Example 3.5. (D-optimality)
- A5. MATLAB program using CVX for Example 4.5. (A-optimality)

## A1. MATLAB program using SeDuMi for Example 2.2

---

```

% Model  $y = \theta_0 + \theta_1x + \theta_2x^2$ , design space  $S = [-1, 1]$ 
% c-optimal design with vector  $c^\top = (1, 2, 4)$ 
% N is the number of design points, design space  $S=[a,b]$ ,
% q is the number of parameters in the model,
% vector u stores all the design points in  $S_N$ ,
% h1, h2 and h3 store all the values for  $f(u), f(u) * f(u)^\top$ , and they are used to
% compute matrices  $\mathbf{H}'_i$ s.
clear
format long
N=501; a=-1; b=1; q=3;
u=zeros(1,N);
h1=zeros(q,N); h2=zeros(q,q,N); h3=h2;

D=[1 2 4;0 1 0;0 0 1]; % Matrix  $\mathbf{D} : 3 \times 3$ 
D1=inv(D'); D2=inv(D);

% Compute h1, h2 and h3 at each design point
for i=1:N

```

```

    u(1,i)=a+(i-1)*(b-a)/(N-1);
    for j=1:q
        h1(j,i)=u(i)^(j-1);
    end
    h2(:,i)=h1(:,i)*h1(:,i)';
    h3(:,i)=D1*h2(:,i)*D2;
end

% Construct matrices  $\mathbf{H}'_i$ s,  $i = 0, \dots, N$ .
H=zeros(q+1+N,q+1+N,N+1);
% In MATLAB, H(...,1) is for  $\mathbf{H}_0$ .
H(1:q,1:q,1)=h3(:,i,N);
H(q+1+N,q+1+N,1)=1;
H(1,q+1,1)=1;
H(q+1,1,1)=1;

% H(...,2),...,H(...,N) are for  $\mathbf{H}_1, \dots, \mathbf{H}_{(N-1)}$ .
for i=2:N
    H(1:q,1:q,i)=h3(:,i,i-1)-h3(:,i,N);
    H(q+1+N,q+1+N,i)=-1;
    H(q+i,q+i,i)=1;
end

% H(...,N+1) is for  $\mathbf{H}_N$ .
H(q+1,q+1,N+1)=1;

% Objective function is  $0 * w_1 + \dots + 0 * w_{(N-1)} + 1 * v_N$ .
c=zeros(N,1);
c(N)=1;
bt=-c;
ct=vec(H(:,i,1));
for i=1:N
    At(:,i)=-vec(H(:,i+1));
end
K.s=size(H(:,i,1),1);
pars.eps=0;

```

```

[x,v,info]=sedumi(At,bt,ct,K,pars);
answer=zeros(N,1);
answer(1:N-1,1)=v(1:N-1,1)';
answer(N)=1-sum(v(1:N-1));
design=[u([find(answer>0.0009)]);answer(find(answer>0.0009))]' % a solution

% Check for optimality
z=zeros(q,q,N);
A=zeros(q,q);
for i=1:N
    z(:, :, i)=answer(i)*h2(:, :, i);
    A=A+z(:, :, i);
end
T=[1 2 4];
phi=trace(T*inv(A)*T');
for i=1:N
    phi_A(i)=h1(:, i)'*inv(A)*T'*T*inv(A)*h1(:, i);
    check(i)=phi_A(i)-phi;
end
delta=max(check)

```

□

---

## A2. MATLAB program using SeDuMi for Example 2.5

---

```

% Model  $y = \frac{\theta_1}{\theta_1 - \theta_2}(e^{-\theta_2 x} - e^{-\theta_1 x})$ , design space  $S = [0, 20]$ 
% I-optimal design with  $\theta_1^* = 0.7$  and  $\theta_2^* = 0.2$ 
% N is the number of design points on  $[a, b]$ 
% vector u stores all the design points in  $S_N$ ,
% h1 and h2 store all the values for  $f(u)$ ,  $f(u) * f(u)^\top$ , respectively, and they are
% used to compute matrices  $\mathbf{M}$ .
% h3 store the values of matrices of  $\mathbf{C}'_i$ 's.
clear
format long
N=501; a=0; b=20;
theta1=0.7; theta2=0.2;

u=zeros(1,N);

```

```

h1=zeros(2,N); h2=zeros(2,2,N); h3=h2;

% get h1 and h2 at each design point
for i=1:N
    u(1,i)=a+(i-1)*(b-a)/(N-1);
    h1(:,i)=[-(theta2/(theta1-theta2)^2)*exp(-theta2*u(i))
             +theta2/(theta1-theta2)^2*exp(-theta1*u(i))
             +theta1/(theta1-theta2)*u(i)*exp(-theta1*u(i))

             theta1/(theta1-theta2)^2*exp(-theta2*u(i))
             -theta1/(theta1-theta2)^2*exp(-theta1*u(i))
             -theta1/(theta1-theta2)*u(i)*exp(-theta2*u(i))];
    h2(:,i)=h1(:,i)*h1(:,i)';
end

% find M matrix
M=zeros(2,2);
for i=1:N
    M=M+h2(:,i);
end M=1/N*M;
% singular value decomposition to find T
[U,S,V] = svd(M); T=U*sqrt(S)*V;

% find C's
for i=1:N
    h3(:,i)=T'\h2(:,i)/T;
end

H=zeros(6+N,6+N,N+2);
H(1:2,1:2,1)=h3(:,N);
H(4:5,4:5,1)=h3(:,N);
H(6+N,6+N,1)=1;
H(1,3,1)=1;
H(3,1,1)=1;
H(5,6,1)=1;
H(6,5,1)=1;

```

```

for i=2:N
    H(1:2,1:2,i)=h3(:,:,i-1)-h3(:,:,N);
    H(4:5,4:5,i)=h3(:,:,i-1)-h3(:,:,N);
    H(6+N,6+N,i)=-1;
    H(5+i,5+i,i)=1;
end

H(3,3,N+1)=1; H(6,6,N+2)=1;

% target function is 0*w1+...+0*w(N-1)+1*r1+1*r2
c=zeros(N+1,1);
c(N)=1;
c(N+1)=1;
bt=-c;
ct=vec(H(:,:,1));
for i=1:N+1
    At(:,i)=-vec(H(:,:,i+1));
end K.s=size(H(:,:,1),1);
pars.eps=0;
[x,y,info]=sedumi(At,bt,ct,K,pars);
info;
answer=zeros(N,1);
answer(1:N-1,1)=y(1:N-1,1)';
answer(N)=1-sum(y(1:N-1));
design=[u([find(answer<0.0009)]);answer(find(answer<0.0009))']

% check for optimality
z=zeros(2,2,N);
A_w=zeros(2,2);
for i=1:N
    z(:,:,i)=answer(i)*h2(:,:,i);
    A_w=A_w+z(:,:,i);
end

phi=trace(A_w\M);

```

```

for i=1:N
    phi_A(i)=h1(:,i)'/A_w*M/A_w*h1(:,i);
    check(i)=phi_A(i)-phi;
end
delta=max(check)

```

□

---

### A3. MATLAB program using CVX for Example 3.5

---

```

% minimize  $\det((G2(w) - t * g1(w) * g1(w)^T)^{-1})$ ,  $\sum(w)=1$ ,  $w \geq 0$ 
% where G2 is a 4 by 4 matrix, g1 is 4 by 1 vector, t is from 0 to 1.
% Input
% N-number of support points
%  $\alpha_1, \alpha_2, \beta_1, \beta_2$ 
% t-ratio parameter t belongs to [0,1]
% S-design space, lower bound and upper bound
%
% Output
% nonzero weights with corresponding points
% running time
% tolerance error value to check the design
clear
runningtime=cputime; %record the starting time
N=501; % # of design points
startvalue=0; %lower bound for interval
endvalue=50; %upper bound for interval

%model parameters
a=1; % $\alpha_1$ 
b=0.1; % $\beta_1$ 
c=1; % $\alpha_2$ 
d=1; % $\beta_2$ 
t=0.9; %ratio parameter

cvx_begin
    cvx_precision high

```

```

variable w(N);
expression g1(4,1);
expression G2(4,4);
expression A(5,5); %target function

u=zeros(1,N); %initial value for design points
h1=zeros(4,N);
h2=zeros(4,4,N);
g1=zeros(4,1);
G2=zeros(4,4);

for i=1:N
    u(i)=startvalue+(i-1)*(endvalue-startvalue)/(N-1);
    h1(:,i)=[exp(-b*u(i)), -a*u(i)*exp(-b*u(i)), exp(-d*u(i)), -c*u(i)*exp(-d*u(i))];
    h2(:,i)=h1(:,i)*h1(:,i);
end

%get g1
for i=1:N
    g1=g1+h1(:,i)*w(i);
end

%get G2
for i=1:N
    G2=G2+h2(:,i)*w(i);
end

%get A
A(1,1)=1;
A(1,2:5)=sqrt(t)*g1';
A(2:5,1)=sqrt(t)*g1;
A(2:5,2:5)=G2;

minimize (-det_rootn(A))
subject to

```

```

        sum(w)==1;
        w>=0;

cvx_end
resulttime=cputime-runningtime;
w=w';
design=[u([find(w>0.001)]);w(find(w>0.001))]

G=zeros(4,4);
g=zeros(4,1);
for i=1:N
    G=G+w(i)*h2(:,i);
    g=g+w(i)*h1(:,i);
end
%Judge if all the weights satisfy the criterion
for i=1:N
    a(i) = MMPHi_D(G,g,t,h1,i)-4;
end
error=max(a);
designpoints=N;
ratio=t;
method='cvx_D';
X = sprintf('%s, N=%d, t=%d time= %d,
tolerance= %d.',method,designpoints,ratio,resulttime,error);
disp(X)

%MMPHi_D function
function phi=MMPHi_D(G,g,t,h1,j)
H=G-t*(g*g');
x=h1(:,j);
phi=(1-t)*x'*inv(H)*x+t*(x-g)'+inv(H)*(x-g);

```

□

---

#### **A4. MATLAB program using multiplicative algorithm for Example 3.5**

```

clear
runningtime = cputime; %record the starting time

```

```

N=101; % # of design points
startvalue=0; %lower bound for interval
endvalue=50; %upper bound for interval

%model parameters
a=1; % $\alpha_1$ 
b=0.1; % $\beta_1$ 
c=1; % $\alpha_2$ 
d=1; % $\beta_2$ 
t=0.9; %ratio parameter

u=zeros(1,N); %initial value for design points
h1=zeros(4,N); %initial value for g1
h2=zeros(4,4,N); %initial value for G2

%calculate the design points
%get g1 and G2 at each design points
for i=1:N
    u(i)=startvalue+(i-1)*(endvalue-startvalue)/(N-1);
    h1(:,i)=[exp(-b*u(i)), -a*u(i)*exp(-b*u(i)), exp(-d*u(i)), -c*u(i)*exp(-d*u(i))];
    h2(:,:,i)=h1(:,i)*h1(:,i)';
end

%initial value of weights
w=zeros(1,N);
for i=1:N
    w(i)=1/N;
end

%new weights for iteration
w1=zeros(1,N);
b=w1;

%set up an indicator
valid = false;

```

```

while valid == false
    G=zeros(4,4);
    g=zeros(4,1);
    for i=1:N
        G=G+w(i)*h2(:,i);
        g=g+w(i)*h1(:,i);
    end

    H=G-t*(g*g');
    v=inv(H);
    for i=1:N
        x=h1(:,i);
        b(i)= (1-t)*x'*v*x+t*(x-g)'*v*(x-g);
    end
    for i=1:N
        valid = b(i)>4+1e-6;
        if(valid == false)
            break;
        end
    end
    end
    %get new iteration of weights
    if(valid==false)
        w1=w.*b/4;
        sumvalue=sum(w1)-w1(N);
        w1(N)=1-sumvalue;
        w=w1;
    end
end
end
design=[u([find(w>0.0001)]);w(find(w>0.0001))]
resulttime = cputime-runningtime;
G=zeros(4,4);
g=zeros(4,1);
for i=1:N
    G=G+w(i)*h2(:,i);
    g=g+w(i)*h1(:,i);

```

```

end
for i=1:N
    a(i) = MMPHi4_D(G,g,t,h1,i)-4;
end
error=max(a);
designpoints=N;
ratio=t;
method='multiplicative_D';
X = sprintf('%s, N=%d, t=%d time= %d, tolerance= %d.',
method,designpoints,ratio,resulttime,error);
disp(X)

```

□

---

### A5. MATLAB program using CVX for Example 4.5

---

```

% A-optimal design for multi-response (Atashgah and Seifi, 2007, 2009)
%  $\Sigma_0 = [2 \ 0.4; 0.4 \ 1]$ .
% 19 user-selected design points
% two response variables and three independent variables
clear
runningtime=cputime; %record the starting time
% design space 19X3 matrix, x1 x2 x3, 19 test vectors
u=[1.68 0 0; 0 1.68 0; 0 0 0; 1.729 1.727 -1.703; 1.728 -1.729 -1.72;
    1.729 1.729 1.729; -1.725 -1.723 1.715; -1.73 1.721 1.729;
    1.73 -1.729 1.729; -1.73 1.73 0.026; 1.73 -1.73 -0.045;
    -1.729 -1.73 -1.728; -1.73 -0.096 1.73; 1.729 1.724 -1.729;
    -0.154 1.73 -1.73; -0.101 -1.73 1.73; 1.729 1.729 1.722;
    -1.5168 -1.6182 0.652; 0.1158 1.6289 1.5256];
N=19;
sigma=[2 0.4;0.4 1];
invs=inv(sigma);

cvx_begin
    cvx_precision high
    variable w(N);
    expression H(14,14);

```

```

U=zeros(2,14,19);
G=zeros(14,14,19);
% construct Uj j=1,...,19.
for i=1:19
    U(1,1:8,i)=[1 u(i,1) u(i,2) u(i,3) u(i,1)*u(i,2) u(i,1)*u(i,3) u(i,1)^2 u(i,3)^2];
    U(2,9:14,i)=[1 u(i,1) u(i,2) u(i,1)*u(i,2) u(i,1)^2 u(i,2)^2];
    G(:,:,i)=U(:,:,i)'invs*U(:,:,i);
end
%get H
for i=1:N
    H=H+G(:,:,i)*w(i);
end
minimize (trace_inv(H))
subject to
    sum(w)==1;
    w>=0;
cvx_end

result=[u w]

```

---

□

# My Publications

In the publications, we use the alphabetical order for the authors.

- [1] Wong, W.K., Yin, Y. and Zhou, J. (2016). Optimal designs for multi-response nonlinear regression models with several factors via semi-definite programming, submitted.
- [2] Wong, W.K., Yin, Y. and Zhou, J. (2017). Using SeDuMi to find various optimal designs for regression models, to appear in *Statistical Papers*.
- [3] Yin, Y. and Zhou, J. (2015). Minimax design criterion for fractional factorial designs, *Annals of the Institute of Statistical Mathematics*, 67, 673-685.
- [4] Yin, Y. and Zhou, J. (2016). Optimal designs for regression models using the second-order least squares estimator, to appear in *Statistica Sinica*.
- [5] Yin, Y. and Zhou, J. (2016). Optimal designs for regression models using the second-order least squares estimator, Poster, The 28th International Biometric Conference, Victoria, Canada, July 10-15, 2016.

# Bibliography

- [1] Atashgah, A.B. and Seifi, A. (2007). Application of semi-definite programming to the design of multi-response experiments. *IIE Transactions*, 39, 763-769.
- [2] Atashgah, A.B. and Seifi, A. (2009). Optimal design of multi-response experiments using semi-definite programming. *Optimization and Engineering*, 10, 75-90.
- [3] Berger, M.P.F. and Wong, W.K. (2009). *An Introduction to Optimal Designs for Social and Biomedical Research*. Wiley, Chichester, UK.
- [4] Bose, M. and Mukerjee, R. (2015). Optimal design measures under asymmetric errors, with application to binary design points. *Journal of Statistical Planning and Inference*. 159, 28-36.
- [5] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York.
- [6] Broudiscou, A., Leardi, R. and Phan-Tan-Luu, R. (1996). Genetic algorithm as a tool for selection of D-optimal design. *Chemometrics and Intelligent Laboratory Systems*, 35, 105-116.
- [7] Chang, S. (1997). An algorithm to generate near D-optimal designs for multiple response surface models. *IIE Transactions*, 29, 1073-1081.
- [8] Chang, F.C., Huang, M.N.L., Lin, D.K.J. and Yang, H.C. (2001). Optimal designs for dual response polynomial regression models. *Journal of Statistical Planning and Inference*, 93, 309-322.
- [9] Chen, R.B., Chang, S.P., Wang, W., Tung, H.C. and Wong, W.K. (2015). Minimax optimal designs via particle swarm optimization methods. *Statistics and Computing*, 25, 975-988.

- [10] Cuervo, D.P., Goos, P. and Sorensen, K. (2016). Optimal design of large-scale screening experiments: a critical look at the coordinate-exchange algorithm. *Statistics and Computing*, 26,15-28.
- [11] Dean, A., Morris, M., Stufken, J., and Bingham, D. (2015). *Handbook of Design and Analysis of Experiments*. CRC Press, Boca Raton, Florida.
- [12] Dette, H., Haines, L.M., and Imhof, L. (1999). Optimal designs for rational models and weighted polynomial regression. *The Annals of Statistics*, 27, 1272-1293.
- [13] Dette, H., Hoyden, L., Kuhnt, S. and Schorning, K. (2013). Optimal designs for multi-response generalized linear models with applications in thermal spraying. <http://arxiv.org/pdf/1312.4472.pdf>
- [14] Dette, H., Melas, V.B., and Pepelyshev, A. (2004). Optimal designs for a class of nonlinear regression models. *The Annals of Statistics*, 2142-2167.
- [15] Dette, H., Melas, V.B., and Wong, W.K. (2005). Optimal design for goodness of fit of the Michaelis-Menten enzyme kinetic function. *Journal of the American Statistical Association*. 100, 1370-1381.
- [16] Dette, H., Melas, V.B., and Wong, W.K. (2006). Locally D-optimal designs for exponential regression models. *Statistica Sinica*. 16, 789-803.
- [17] Dette, H., and O'Brien, T.E. (1999). Optimality criteria for regression models based on predicted variance. *Biometrika*, 86, 93-106.
- [18] Dette, H. and Studden, W.J. (1997). *The Theory of Canonical Moments with Applications in Statistics, Probability, and Analysis*. Wiley, New York.
- [19] Drain, D., Carlyle, W.M., Montgomery, D.C., Borrow, C. and Anderson-Cook, C. (2004). A genetic algorithm hybrid for constructing optimal response surface designs *Quality of Reliability Engineering International*, 637-650.
- [20] Duarte, B.P and Wong, W.K. (2014). A semi-infinite programming based algorithm for finding minimax optimal designs for nonlinear models. *Statistics and Computing*, 24, 1063-1080.

- [21] Duarte, B.P.M. and Wong, W.K. (2015). Finding Bayesian optimal designs for nonlinear models: a semidefinite programming-based approach. *International Statistical Review*, 83, 239-262.
- [22] Duarte, B.P.M., Wong, W.K. and Atkinson, A.C. (2015). A semi-infinite programming based algorithm for determining T-optimum designs for model discrimination. *Journal of Multivariate Analysis*, 135, 11-24.
- [23] Duarte, B.P.M., Wong, W.K. and Oliveira, N.M.C. (2016). Model-based optimal design of experiments - semidefinite and nonlinear programming formulations. *Chemometrics and Intelligent Laboratory Systems*, 151, 153-163.
- [24] Fang, Z., and Wiens, D. P. (2000). Integer-valued, minimax robust designs for estimation and extrapolation in heteroscedastic, approximately linear models. *Journal of the American Statistical Association*, 95, 807-818.
- [25] Faddy, M.J. (1993). A structured compartmental model for drug kinetics. *Biometrics*, 49, 243-248.
- [26] Fedorov, V.V. (1972). *Theory Of Optimal Experiments*. Academic Press, New York.
- [27] Fedorov, V., Gagnon, R. and Leonov, S. (2001). Optimal design for multiple responses with variance depending on unknown parameters. GlaxoSmithKline Pharmaceuticals, GSK BDS Technical Report, August 2001.
- [28] Gao, L.L. and Zhou, J. (2014). New optimal design criteria for regression models with asymmetric errors. *Journal of Statistical Planning and Inference*. 149, 140-151.
- [29] Gao, L.L. and Zhou, J. (2015). D-optimal designs based on the second-order least squares estimator. *Statistical Papers*, 58, 77-94.
- [30] Gianchandani, Y.B., and Crary, S.B. (1998). Parametric modeling of a microaccelerometer: comparing I-and D-optimal design of experiments for finite-element analysis. *Journal of Microelectromechanical Systems*, 7, 274-282.
- [31] Givens G.H. and Hoeting J.A. (2005). *Computational Statistics*. Wiley, New York.

- [32] Grant, M.C. and Boyd, S.P. (2013). *The CVX Users Guide*. Release 2.0 (beta), CVX Research, Inc. (<http://cvxr.com/cvx/doc/CVX.pdf>, October 14, 2013.)
- [33] Hamada, M., Martz, H.F., Reese, C.S. and Wilson, A.G. (2001) Finding near-optimal Bayesian experimental designs via genetic algorithms. *The American Statistician*, 155, 175–181,
- [34] Han, C., and Chaloner, K. (2003). D- and c-optimal designs for exponential regression models used in viral dynamics and other applications. *Journal of Statistical Planning and Inference*, 115, 585-601.
- [35] Hardin, R.H., and Sloane, N.J.A. (1993). A new approach to the construction of optimal designs. *Journal of Statistical Planning and Inference*, 37, 339-369.
- [36] He, Z., Studden, W.J., and Sun, D. (1996). Optimal designs for rational models. *The Annals of Statistics*, 24, 2128-2147.
- [37] Herrmann, J. W. (1999). A genetic algorithm for minimax optimization problems. *IEEE Evolutionary Algorithms*, 1099-1103.
- [38] Horn, R.A. and Johnson, C.R. (2009). *Matrix Analysis*. Cambridge University Press, New York.
- [39] Imhof, L.A., and Studden, W.J. (2001). E-optimal designs for rational models. *The Annals of Statistics*, 763-783.
- [40] Karami, J.H., and Wiens, D.P. (2014). Robust static designs for approximately specified nonlinear regression models. *Journal of Statistical Planning and Inference*, 144, 55-62.
- [41] Kirby, S., Brain, P., and Jones, B. (2011). Fitting Emax models to clinical trial dose-response data. *Pharmaceutical statistics*, 10, 143-149.
- [42] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.
- [43] Kpamegan, E.P. (1998). D-optimal designs given a bivariate probit response function, in *New Developments and Applications in Experimental Design*, eds, Flournoy, N., Rosenberger, W.F., and Wong, W.K., 62-72, Institute of Mathematical Statistics, Hayward, California.

- [44] Krafft, O. and Schaefer, M. (1992). D-optimal designs for a multivariate regression model. *Journal of Multivariate Analysis*, 42, 130-140.
- [45] Liu, X., Yue, R.X. and Hickernell, F.J. (2011). Optimality criteria for multiresponse linear models based on predictive ellipsoids. *Statistica Sinica*, 21, 421-432.
- [46] Lu, Z. and Pong, T.K. (2013). Computing optimal experimental designs via interior point method. *SIAM Journal on Matrix Analysis and Application*, 34, 1556-1580.
- [47] Magnúsdóttir, B.T. (2013). c-Optimal designs for the bivariate Emax model. In *mODa 10-Advances in Model-Oriented Design and Analysis*, edited by Uciński, D., Atkinson, A.C., and Patan, M., page 153-161. Springer, Switzerland.
- [48] Meyer, R.K., and Nachtsheim, C.J. (1988). Constructing exact D-optimal experimental designs by simulated annealing. *American Journal of Mathematical and Management Sciences*, 8, 329-359.
- [49] Meyer, R.K., and Nachtsheim, C.J. (1995). The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, 37, 60-69.
- [50] Michaelis, L. and Menten, M.L. (1913). Die Kinetik der invertinwirkung. *Biochemische Zeitschrift*. 49, 333-369.
- [51] Montepiedra, G. and Myers, D. and Yeh, A. (1998). Application of genetic algorithms to the construction of exact D-optimal designs. *Journal of Applied Statistics*, 817-826.
- [52] Montgomery, D.C. (2013). *Design and Analysis of Experiments*, Eighth Edition. Wiley, New York.
- [53] Papp, D. (2012). Optimal designs for rational function regression. *Journal of the American Statistical Association*, 107, 400-411.
- [54] Pukelsheim, F. (1993). *Optimal Design of Experiments*. Wiley, New York.
- [55] Pukelsheim, F., and Studden, W.J. (1993). E-optimal designs for polynomial regression. *The Annals of Statistics*, 402-415.

- [56] Qiu, J., Chen, R.B., Wang, W. and Wong, W.K. (2014). Using animal instincts to design efficient biomedical studies via particle swarm optimization. *Swarm and Evolutionary Computation*, 18, 1-10.
- [57] Rempel, M.F., and Zhou, J. (2014). On exact K-optimal designs minimizing the condition number. *Communications in Statistics - Theory and Methods*, 43, 1114-1131.
- [58] Seber, G.A.F. and Wild, C.J. (1989). *Nonlinear Regression*. Wiley, New York.
- [59] Silvey, S.D., Titterington, D.H., and Torsney, B. (1978). An algorithm for optimal designs on a design space. *Communications in Statistics-Theory and Methods*, 7, 1379-1389.
- [60] Sturm, J.F. (1999). Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*. 11, 625-653.
- [61] Wang, L. and Leblanc, A. (2008). Second-order nonlinear least squares estimation. *Annals of the Institute of Statistical Mathematics*. 60, 883-900.
- [62] Weiss, M. (1983). Use of gamma distributed residence times in pharmacokinetics. *European Journal of Clinical Pharmacology*, 25, 695-702.
- [63] Wise, M.E. (1985). Negative power functions of time in pharmacokinetics and their implications. *Journal of Pharmacokinetics and Biopharmaceutics*, 13, 309-346.
- [64] Wilmut, M., and Zhou, J. (2011). D-optimal minimax design criterion for two-level fractional factorial designs. *Journal of Statistical Planning and Inference*, 141, 576-587.
- [65] Wong, W.K. and Chen, R.B. and Huang, C.C. and Wang, W. (2015). A modified particle swarm optimization technique for finding optimal designs for mixture models. *PLoS One*, 10, 1-23.
- [66] Yang, M., Biedermann, S., and Tang, E. (2013). On optimal designs for nonlinear models: a general and efficient algorithm. *Journal of the American Statistical Association*. 108, 1411-1420.

- [67] Yang, M. and Stufken, J. (2009). Support points of locally optimal designs for nonlinear models with two parameters. *The Annals of Statistics*. 37, 518-541.
- [68] Yang, M. and Stufken, J. (2012). Identifying locally optimal designs for nonlinear models: a simple extension with profound consequences. *The Annals of Statistics*. 40, 1665-1681.
- [69] Ye, J.J., Zhou, J., and Zhou, W. (2015). Computing A-optimal and E-optimal designs for regression models via semidefinite programming. *Communications in Statistics - Simulation and Computation*, 46, 2011-2024.
- [70] Yin, Y. and J. Zhou (2015). Minimax design criterion for fractional factorial designs. *Annals of the Institute of Statistical Mathematics*, 67, 673-685.
- [71] Yu, Y. (2010). Monotonic convergence of a general algorithm for computing optimal designs. *The Annals of Statistics*, 38, 1593-1606.
- [72] Zhang, R. and Mukerjee, R. (2013). Highly efficient factorial designs for cDNA microarray experiments: use of approximate theory together with a step-up step-down procedure. *Statistical Applications in Genetics and Molecular Biology*. 12, 489-503.