The Pennsylvania State University
The Graduate School

BY DESIGN: EXCHANGE ALGORITHMS TO CONSTRUCT EXACT MODEL-ROBUST AND MULTIRESPONSE EXPERIMENTAL DESIGNS

A Dissertation in
Statistics and Operations Research
by
Byran J. Smucker

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2010

The dissertation of Byran J. Smucker was reviewed and approved* by the following:

Enrique del Castillo
Distinguished Professor of Industrial and Manufacturing Engineering
Dissertation Advisor and Co-Chair of Committee

James L. Rosenberger
Professor of Statistics
Co-Chair of Committee

Steven F. Arnold
Professor of Statistics

Bing Li
Professor of Statistics

Susan H. Xu
Professor of Management Science and Supply Chain Management

Bruce G. Lindsay
Willaman Professor of Statistics
Statistics Department Head

*Signatures are on file in the Graduate School.

Abstract

Optimal experimental design procedures, utilizing criteria such as $\mathcal{D}$-optimality, are often used under nonstandard experimental conditions such as constrained design spaces, and produce designs with desirable variance properties. However, to implement these methods the form of the regression function must be known *a priori*, an often unrealistic assumption. Model-robust designs are those which, from our perspective, are robust for a set of specified possible models. In this dissertation, we present new model-robust exchange algorithms for exact experimental designs which improve upon current, practical model-robust methodology. We also extend these ideas to experiments with multiple responses and split-plot structures, settings for which few or no flexible, practicable model-robust procedures exist.

We first develop a model-robust technique which, when the possible models are nested, is $\mathcal{D}$-optimal with respect to an associated multiresponse model. In addition to providing a justification for the procedure, this motivates a generalization of the modified Fedorov exchange algorithm which is used to construct exact model-robust designs. We give several examples and compare our designs with two model-robust procedures in the literature.

For a given set of models, the aforementioned algorithm tends to produce designs which have higher $\mathcal{D}$-efficiencies for some models and lower $\mathcal{D}$-efficiencies for others. To mitigate this unbalancedness, we develop a model-robust maximin exchange algorithm which maximizes the lowest efficiency over the set of models and consequently produces designs for which there is worst-case protection. Furthermore, we present a generalization of this technique which allows the user to express varying levels of interest in each model, often resulting in a design suggestive of these differences. Some asymptotic properties of this criterion are explored, including a condition which guarantees complete balance in terms of (generalized) efficiencies. We also show that even if this condition is not satisfied, this balance will be achieved in some subset of at least two models for nontrivial cases. We give

several examples illustrating the procedure.

Since many, if not most, experiments have multiple responses, we extend our methodology to such designs. In addition to the problem of unknown model forms, which in this case is exacerbated by the fact that there are multiple such forms to specify, the response covariance matrix is generally unknown at the design stage as well. We present an exchange algorithm for multiresponse $\mathcal{D}$-optimal designs, using generalizations of matrix-updating formulae to serve as its computational engine. However, this procedure requires knowledge of the model forms, so we develop an expanded multiresponse model which allows each response to accommodate a set of possible models. The optimal design with respect to this larger model constitutes a design robust to these sets. We find, as has been noted before, that the covariance matrix is generally of little import, and it is much less consequential than the unknown model forms. We use several examples to compare the model-robust designs to designs optimal for the largest assumed model (i.e. usual practice).

Finally, we consider model-robust split-plot designs using the maximin approach. Split-plot experiments are appropriate when some factors are difficult or expensive to change relative to other factors. They require two levels of randomization which induces an error structure that renders ordinary least squares analysis incorrect in general. The design of such experiments has garnered much attention over the last twenty years, and has spawned work in split-plot $\mathcal{D}$-optimal designs. However, as in the case of completely randomized experiments, these procedures rely on the assumption that the form of the model relating the factors to the response is correctly specified. We relax that assumption, again by allowing the experimenter to specify a set of model forms, and use the maximin criterion to produce designs that have high $\mathcal{D}$-efficiencies for each of the models in the set. Furthermore, a generalization allows the experimenter to exert some control over the efficiencies by specifying a level of interest in each model. We demonstrate the procedure with two examples.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\boldsymbol{\beta}$	The $p \times 1$ vector of regression parameters for basic regression model; or the $q \times 1$ vector of regression parameters for the multiresponse regression model.
κ	The prior precision parameter for the model-robust procedure of DuMouchel and Jones [37].
$\phi(\mathbf{M}(\xi))$	Optimal design criterion function.
$\boldsymbol{\rho}(\mathbf{x})$	Model forms, for the multiresponse regression model.
$\boldsymbol{\tau}$	The $p \times 1$ vector of regression parameters for split-plot regression model.
ξ	Asymptotic design.
ξ_n	Exact, n-point design for completely randomized and multiresponse experiments.
ξ_{nb}	Exact, n-point design with b whole plots for split-plot experiments.
Ξ	Set of all possible designs.
B	The set of whole plots in a split-plot design.
b	The number of whole plots in a split-plot design.
c	The number of candidate points in exchange algorithm candidate list.
C	The candidate list for the various exchange algorithms developed in this dissertation.

C_i	The set of candidate points with the same factor level settings as whole plot i.
D_f	$\mathcal{D}$-efficiency with respect to model f.
$f(\mathbf{x})$	Model form, for the univariate regression model.
$\mathcal{F}$	Set of models for which procedures are model-robust.
$g_i(\mathbf{M}_{\mathcal{F}}(\xi_n))$	Model-robust criterion function.
G_f	The generalized $\mathcal{D}$-efficiency with respect to model f.
H_i	The set of k_i design points in the i^{th} whole plot.
k	The number of factors in basic and/or multiresponse regression model.
k_i	The number of design points in the i^{th} whole plot.
$\mathbf{M}(\xi)$	Information matrix for design ξ, for univariate, completely randomized design.
$\mathbf{M}_m(\xi)$	Information matrix for design ξ, for multiresponse design.
$\mathbf{M}_{sp}(\xi)$	Information matrix for design ξ, for split-plot design.
$\mathbf{M}_{\mathcal{F}} = (\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_r)$	The set of information matrices with respect to the model forms in $\mathcal{F}$.
p	The number of parameters in basic regression model.
P	The set of possible whole plot factor level combinations.
q_i	The number of parameters for response i in the multiresponse regression model.
q	The total number of parameters in the multiresponse regression model.
r	The number of models in $\mathcal{F}$ and/or the number of responses in the multiresponse regression model.

$v = (v_1, v_2, \ldots, v_r)$	The model-interest vector for each of the r models in $\mathcal{F}$.
$\mathbf{V}(\mathbf{x}, \xi)$	Prediction variance for point $\mathbf{x}$ and design ξ, for univariate, completely randomized design.
$\mathbf{V}_m(\mathbf{x}, \xi)$	Prediction variance matrix for point $\mathbf{x}$ and design ξ, for multiresponse design.
$\mathbf{V}_{sp}(\mathbf{x}, \xi)$	Prediction variance for point $\mathbf{x}$ and design ξ, for split-plot design.
$\mathcal{X}$	Design space.
z_i	The i^{th} whole plot factor level combination.

Acknowledgments

I thank my Aunt Rosie for demonstrating that going to college was normal, even in a setting in which it was rare.

I thank my father and mother, Steve and Bonnie, for their implicit, explicit, and unwavering support of a son whose formal education, I'm sure, seemed like it was never to end.

I thank my classmates for their help. Grad school would have been significantly more difficult without them.

I thank James Rosenberger for his confidence in me, and for making it seem that he was never too busy to talk, even when he probably was.

I thank my advisor, Enrique del Castillo, for making our regular meetings uplifting. So often I left these exchanges feeling better about my progress.

I thank the U.S. Census Bureau for the Dissertation Fellowship that has allowed my academic energies to converge on this project.

I thank my son, Xavier, for infusing my life with a mixture of joy and more joy.

I thank my wife, Amy, for patiently waiting for the commencement of our "real" life, and for loving me in the meantime.

I thank God, the Great Designer Himself, for giving me all these things to enjoy.

Dedication

To Amy, who agreed to embark on a different sort of life by marrying me.

Chapter 1

Introduction and Setting

1.1 Introduction

Consider a manufacturing setting in which an engineer is examining the effect of five mixture factors on the hardness of a plastic product [96]. In addition to the mixture constraint (each factor, in each experimental run, comprises between 0% and 100% of the mixture, and the sum of these factors must be 100%), other constraints on the factors further restrict the design region. With only 25 experimental runs available, the experimenter would like to fit an appropriate polynomial-type regression model, the form of which is unknown. How might one choose the levels of the factors for these experimental runs? In other words, how should the experiment be designed?

Alternatively, perhaps food scientists are studying the effects of washing minced mullet flesh on several measures of quality, including texture, color, and how well the meat was preserved [101]. This is a multiresponse design problem and it suggests the question of whether a design can be constructed to take advantage of this multivariate structure? If yes, how? If not, does it reduce to a univariate design problem?

Or, suppose that one is producing vinyl to cover automobile seats and the measurement of interest is its thickness ([29], pp. 377-383). This material is comprised of many mixture components, including three plasticizers, which may affect its thickness. But two nonmixture factors, extrusion rate and drying temperature, are potentially important as well. Furthermore, it is inconvenient to independently

reset the levels of each factor for each run, as complete randomization would require. Both the design and analysis of this experiment is thus complicated by these considerations. How might one design an effective experiment in this case?

In this dissertation, we develop flexible and practical design methods to address these sorts of problems. We seek to construct efficient designs even when the experimenter lacks knowledge of the type of relationship between the response and factors. Our methods aim to accommodate not just univariate, completely randomized experiments, but multiresponse and split-plot experiments as well.

Standard designs, such as fractional factorial or central composite designs and their split-plot analogs, exist to address many experimental design situations. Sometimes, however, because of constraints on the design region, categorial factors, or nonstandard sample size requirements, these traditional designs are inadequate. Furthermore, they can be inefficient in their use of experimental resources.

A popular alternative to standard designs are optimal designs, which are chosen for their good variance properties. There are many optimality criteria that have been proposed, but they generally fall into one of two categories: 1) parameter variance minimizers; or 2) prediction variance minimizers. In contrast to standard designs which are model-independent (although this is not precisely true; they are generally chosen with a particular maximal model in mind), optimal designs depend strongly upon the assumed *a priori* knowledge of the form of the relationship between the response(s) and the factors.

This assumption is often unmet in practice, since experimenters generally do not know the true form of the model. Thus, the central and unifying theme of this thesis is model-robust experimental design. That is, we develop useful methods for practitioners which retain the optimal design paradigm, but allow robustness with respect to departures from the assumed model form(s).

This is a well-studied problem in the univariate case, but the new practical procedures developed herein are intuitive for users and constitute generalizations of current optimal design techniques for experiments with finite run sizes. Furthermore, little or no work has been done regarding model-robustness for multiresponse and split-plot experiments, so the univariate, completely randomized procedures are extended to these more complicated model-robust design problems.

1.2 Setting

To clarify some of the aforementioned ideas, we review and set notation for the basic linear regression model as well as completely randomized, univariate experimental design, concepts which are fundamental to the work in this dissertation. The linear regression model is written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\mathbf{y}$ is an n-vector of observations, $\mathbf{X}$ is an $n \times p$ matrix, $\boldsymbol{\beta}$ is a p-vector of model parameters, and $\boldsymbol{\epsilon}$ is an n-vector of errors with mean 0 and variance σ^2. A common way to estimate $\boldsymbol{\beta}$ is to use the least squares criterion, which minimizes the squared deviation between the data and the estimates. These are given by $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ with $Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$.

Let $\mathcal{X}$ be the design space and $\mathbf{x} \in \mathcal{X}$ be the set of all design points in this space. Further, let a design be a discrete probability measure ξ defined over $\mathcal{X}$. This implies that $\xi(\mathbf{x}) \geq 0 \ \forall \mathbf{x} \in \mathcal{X}$, $\xi(\chi) = 1$, and that there exist a countable number of design points upon which there is positive measure. Then, a design can be thought of as the proportion of the available experimental runs assigned to any particular design point in $\mathcal{X}$. This allows any design to be represented as its associated design measure: $\xi(\mathbf{x}) = \lambda(\mathbf{x})$. A more enlightening representation is

$$\xi = \begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_d \\ \lambda_1 & \dots & \lambda_d \end{pmatrix} \tag{1.2}$$

where $\mathbf{x}_i$, $i = 1, \dots, d$ are the d design points in $\mathcal{X}$ which have positive measure, and λ_i, $i = 1, \dots, d$ is the measure placed (or the fraction of the experiments performed) on the associated design points. In general, for a set of n experiments, such a design does not restrict the number of experiments performed at each of the d design points to be integer-valued (equivalent to assuming an infinite number of runs) and thus it is called an approximate (asymptotic) design. However, an exact experiment (one with a finite number of runs) can be represented as

$$\xi_n = \begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_d \\ n_1 & \dots & n_d \end{pmatrix} \tag{1.3}$$

where n is the total number of experiments, and n_i, $i = 1, \ldots, d$ is the number of experiments performed at design point, $\mathbf{x}_i$. Figure 1.1 gives an example of a continuous design with a discrete design space for which the measure at each design point is equal.

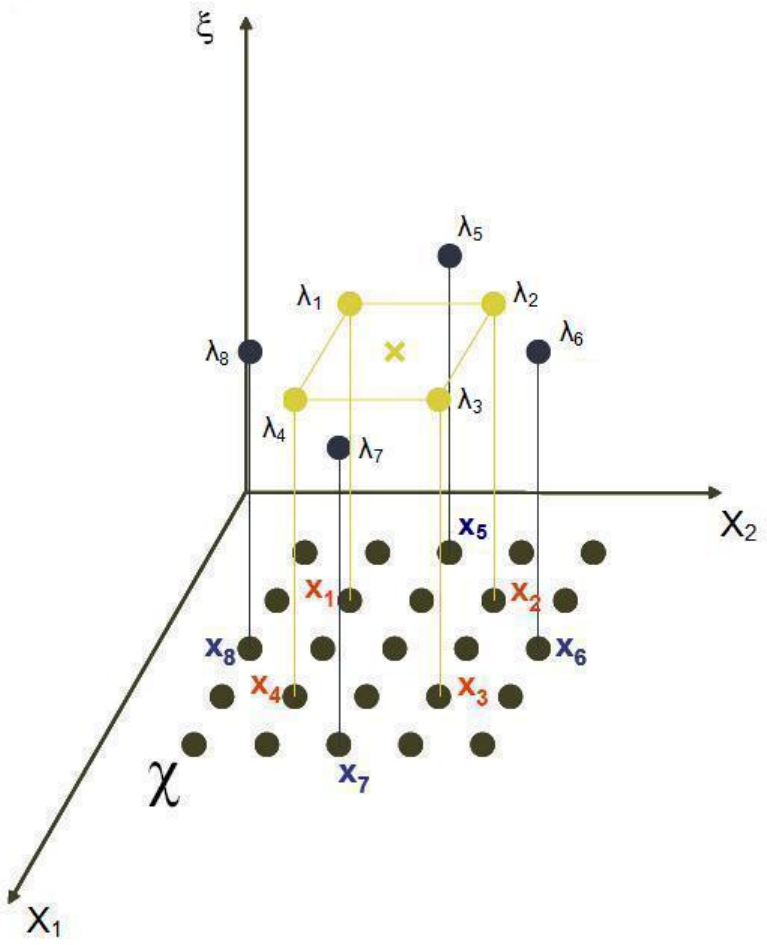

Figure 1.1. Example of continuous design with discrete design space in which there are two factors.

We further define the univariate information matrix as

$$\mathbf{M}(\xi) = \frac{1}{\sigma^2} \int_{\chi} f(\mathbf{x}) f^{'}(\mathbf{x})\, d\xi$$

where $f(\mathbf{x})$ is a p-vector with entries of the same form as the expanded design matrix $\mathbf{X}$ for design point $\mathbf{x}$. Further, for a particular point $\mathbf{x}$ and design ξ, the

prediction variance is

$$\mathbf{V}(\mathbf{x}, \xi) = f^{'}(\mathbf{x})\mathbf{M}^{-1}(\xi)f(\mathbf{x}) \tag{1.4}$$

For the exact design defined in (1.3), the information matrix can be simplified to

$$\mathbf{M}(\xi_n) = \frac{1}{\sigma^2}\sum_{i=1}^{d} n_i f(\mathbf{x}_i)f^{'}(\mathbf{x}_i) \tag{1.5}$$

$$= \frac{1}{\sigma^2}\mathbf{X}^{'}\mathbf{X} \tag{1.6}$$

$$= \left[Var(\hat{\boldsymbol{\beta}})\right]^{-1} \tag{1.7}$$

In traditional optimal design for completely randomized experiments, the goal is to optimally allocate experimental resources. A raft of criteria have been proposed (i.e. alphabetic optimality), the most common and popular of which is $\mathcal{D}$-optimality, largely because of its computational convenience. This criterion is defined as

$$\phi(\mathbf{M}(\xi)) = |\mathbf{M}(\xi)|\,, \tag{1.8}$$

and minimizes the generalized variance of the parameter estimates. Other parameter variance-based criteria include $\mathcal{A}$-optimality, which minimizes the trace of the information matrix and $\mathcal{E}$-optimality, which minimizes the maximum eigenvalue of the information matrix. Assuming normality, the $\mathcal{D}$-optimal design minimizes the volume of the confidence ellipsoid about the parameters, the $\mathcal{A}$-optimal design minimizes the volume of the enclosing box of this ellipsoid, and the $\mathcal{E}$-optimal design minimizes its major axis (1.2). Other criteria minimize a function of the prediction variance, (1.4). $\mathcal{G}$- and $\mathcal{IV}$-optimality are examples, which minimize the maximum prediction variance and the average prediction variance, respectively.

It is clear from (1.5) that the model form f is implicit in the information matrix which is a central part of the optimality criteria. Thus, any optimal design depends strongly on the model form chosen by the experimenter. In this thesis, we develop procedures which allow designs based upon more than just a single, assumed model.

Our model-robust methods are based upon $\mathcal{D}$-optimality. It is the most commonly used criterion, due to its computational amenability, but also attractive

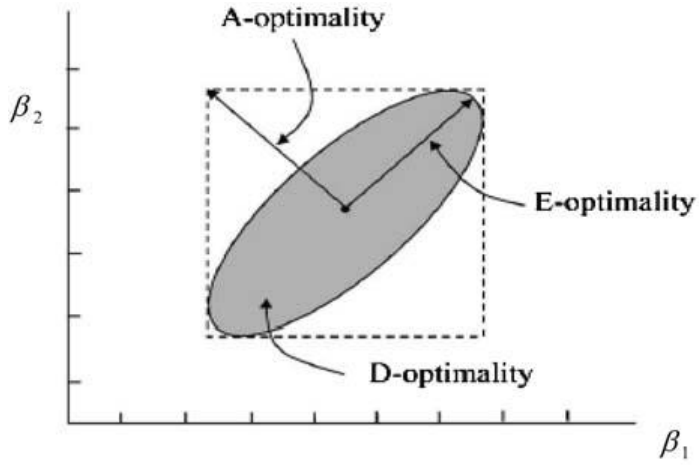

Figure 1.2. Diagram of several optimality criteria ([9] via [7]).

because it generally performs well with respect to other criteria (see, for example, [45]). With the exception of Chapter 5 on multiresponse model-robust design, the approach used to address this problem could be used in principle with other design criterion. However, many of the computational advantages would be lost.

1.3 Dissertation Topics

Classical optimal design assumes a single model form f. Based upon this model form, a design is expanded to $\mathbf{X}$, from which the information matrix, $\mathbf{M}$, can be calculated. Then, in the case of $\mathcal{D}$-optimality for an exact design, a design ξ_n is chosen to maximize $|\mathbf{M}|$.

In contrast, the model-robust approach taken in this dissertation assumes a set of models $\mathcal{F} = (f_1, f_2, \ldots, f_r)$. Based upon each of these user-specified models, a design can be expanded to $\mathbf{X}_i$, $i = 1, \ldots, r$, and a set of information matrices, $\mathbf{M}_\mathcal{F} = (\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_r)$, can be calculated. To construct a model-robust design, criteria are used which account for each of the information matrices. In this case, a design ξ_n is chosen to maximize a real-valued function $g(\mathbf{M}_\mathcal{F})$.

This device is used throughout, though the criteria (i.e., g) and/or experimental situation changes. If a design can be chosen which maximizes $g(\mathbf{M}_\mathcal{F})$, it should perform well with respect to each model form in $\mathcal{F}$ thus providing robustness for these models.

These model-robust criteria for univariate, completely randomized designs are not themselves new. However, their implementation via exchange algorithms for exact designs gives practitioners intuitive, useful design construction procedures and allows their extension to more complex experimental settings.

We focus first on two model-robust methods for univariate, completely randomized experiments, and then expand our methodology to include multiple response and split-plot experiments.

1.3.1 A Multiresponse Approach to Model-Robustness

The first technique, for a univariate, completely randomized experiment, chooses the design which maximizes the following model-robust criterion:

$$g_1(\mathbf{M}_{\mathcal{F}}(\xi_n)) = \prod_f |\mathbf{M}_f(\xi_n)|.$$

This criterion is motivated by a connection with multiresponse optimal design. When the model-robust design problem is framed using $\mathcal{F}$, the goal of multiresponse optimal design and model-robust design are very similar. In both cases, a design is desired that is efficient for a variety of models (because in the multiresponse case, the different responses can have different model forms). In fact, when the models chosen for $\mathcal{F}$ are nested (that is, if they are ordered from smallest to largest in terms of the number of parameters, each succeeding model includes the previous), the model-robust design is $\mathcal{D}$-optimal for the associated multiresponse model.

Thus, we can use multiresponse optimal design algorithms to construct model-robust designs for the single response case. This observation results in a multiresponse generalization of the modified Fedorov exchange algorithm, a simplification of which we use to find model-robust designs.

1.3.2 A Maximin Approach to Model-Robustness

Though the preceding procedure has a compelling interpretation, it affords the experimenter little flexibility in reflecting varying interest in each model and for model sets with relatively few elements tends to bestow much higher $\mathcal{D}$-efficiencies

for some models than others. To mitigate this lack of balance, another model-robust criterion is presented:

$$g_2(\mathbf{M}_{\mathcal{F}}(\xi_n)) = \min_{f\in\mathcal{F}}(|\mathbf{M}_f(\xi_n)|/|\mathbf{M}_f^*|)^{1/p_f} = \min_{f\in\mathcal{F}} E_f,$$

where $\mathbf{M}_f^*$ is the information matrix for the design optimal for model f, p_f is the number of parameters for model f, and E_f is the $\mathcal{D}$-efficiency with respect to model f. We seek the design which maximizes g_2.

We again develop a generalization of a standard exchange algorithm to implement this model-robust criterion. This maximin approach produces designs for which there is worst-case protection for all $f \in \mathcal{F}$, and the $\mathcal{D}$-efficiencies with respect to the models tend to be more balanced. We can generalize g_2 to allow varying emphases on particular models, if some are deemed more important or likely than others. This generalized criterion is

$$g_2(\mathbf{M}_{\mathcal{F}}(\xi_n)) = \min_{f\in\mathcal{F}}(E_f/v_f) = \min_{f\in\mathcal{F}}(G_f),$$

where $v_f \in (0, 1]$ is the level of interest in model f and G_f is the generalized $\mathcal{D}$-efficiency. Resulting designs are often suggestive of the differences in interest. We explore asymptotic properties of this criterion, including a condition which guarantees complete balance of the generalized $\mathcal{D}$-efficiencies. Even if this condition is not satisfied, we show that balance will be achieved for some subset of at least two models in nontrivial cases.

1.3.3 Multiresponse Model-Robustness

There are multiple barriers facing the practitioner whose experiment has multiple responses. The covariance matrix relating the responses to one another is usually unknown at the design stage, and the model forms relating each response to the factors are generally unknown as well. An approach reminiscent of §1.3.1 is adopted to provide a framework within which designs can be found that reflect the multiresponse nature of the problem as well as the lack of precise knowledge of the model form for each response.

To find multiresponse $\mathcal{D}$-optimal designs, we generalize a univariate exchange

algorithm, using matrix-updating formulae that reflect multiple responses. The basic procedure is not model-robust and assumes that the covariance matrix is known. However, using an expanded multiresponse model, we specify a set of models for each response and calculate the optimal design for this larger multiresponse model. In certain cases, the covariance matrix can be ignored, and even in those cases in which it affects the optimal design, its effect seems small particularly as compared to the effect of ignoring model misspecification.

1.3.4 Model-Robustness for Split-Plot Experiments

If the levels of some factors in an experiment are hard and/or expensive to change, while others are considerably easier and/or cheaper, a split-plot experiment is appropriate. Such an experiment involves two levels of randomization and consequently complicates the error structure of the model typically fit.

$\mathcal{D}$-optimal design for split-plot experiments has been used to increase the precision with which the parameters of the split-plot model can be estimated. However, such procedures require—just as in the completely randomized case—that the form of the model be specified. Failure to correctly specify this model form leads to a suboptimal design.

We develop model-robust split-plot procedures by using the maximin criterion discussed in §1.3.2 to give worst-case protection with respect to the models specified in $\mathcal{F}$. The same generalization of the maximin criterion is used to give the experimenter more control over these efficiencies.

1.4 Dissertation Research Objectives

The overarching objective of this dissertation is to provide usable, intuitive methodologies for experimenters whose approach of choice is $\mathcal{D}$-optimal design. The goal is to maintain the optimal design framework, while reducing the dependence upon a single, assumed model.

Specifically, for the univariate, completely randomized scenario, the objective is to develop exchange algorithms which are generalizations of those commonly implemented to find $\mathcal{D}$-optimal designs. Given a user-specified set of potential

models, and a model-robust criterion (either the product of the determinants of the information matrices, or maximizing the minimum $\mathcal{D}$-efficiency), these algorithms will produce exact designs which have desirable model-robust properties, including the ability to test for lack-of-fit.

For multiresponse experiments, the objective is to produce an exchange algorithm that will construct $\mathcal{D}$-optimal designs, given that the covariance matrix and model forms are known. Beyond that, the goal is that the algorithm has sufficient flexibility to handle a relaxation of the known model forms assumption.

The final objective is to extend the maximin criterion to the situation in which a split-plot experiment is to be designed. As in the simpler case, this will allow the user to relax the model form assumption, dictate to some extent the efficiencies with respect to each possible model, and be equipped to test for lack-of-fit.

1.5 Dissertation Outline

This dissertation is organized as follows. In Chapter 2, we review several streams of literature, the confluence of which have produced many of the ideas in this work: optimal and model-robust design, multiresponse regression models, multiresponse design, and split-plot design.

In Chapter 3 the first model-robust exchange algorithm is developed for univariate, completely randomized experiments, using as a criterion the product of the determinants of the information matrices with respect to all models in a specified set. In Chapter 4 another criterion is used—maximize the minimum $\mathcal{D}$-efficiency with respect to all models in a specified set—and another model-robust exchange algorithm is developed. Both of these methods are illustrated by examples.

Chapter 5 consists of a generalization to the multiresponse setting of the algorithm in Chapter 3, including the development of a multiresponse exchange algorithm and an empirical demonstration of its virtues. Chapter 6 extends the methodology of Chapter 4 to the case of split-plot designs, developing a maximin exchange algorithm, and demonstrating its use via examples.

Chapter 7 provides a discussion of our contributions, as well as some potential future work.

Chapter 2

Literature Review

The research in this dissertation pulls together ideas from a range of statistical topics—experimental design in particular. In this chapter, we review the literatures for optimal design, model-robust design, multiresponse design, multiresponse regression, and design for split-plot experiments.

2.1 Optimal Design

In §1.2, we reviewed the basic linear regression model, which serves as the setting for optimal design, and gave an overview of various optimality criteria including $\mathcal{D}$-optimality. The early leader of the optimal design movement was Jack Kiefer [64], and he set it upon the firm mathematical foundations which gave it legitimacy and allowed it to thrive. He used continuous design theory, which assumed asymptotically large run sizes and resulted in the famed General Equivalence Theorem [66] which showed that $\mathcal{D}$- and $\mathcal{G}$-optimality are equivalent. For more on optimal design continuous theory, see [87, 10].

In this thesis, we concentrate on exact designs; that is, designs for experiments with a specified, finite number of available runs. This is a more practical situation because an experimenter invariably wants a design conforming to an experimental budget.

Unfortunately, when the continuous approach is forsaken, the associated optimization problem becomes much harder, because the space of information matrices is no longer convex and neither are the criteria functions such as $|\cdot|$ and $tr(\cdot)$.

Consequently, heuristic exchange algorithms are generally employed to find optimal designs.

The basic exchange procedure was pioneered by Fedorov [41], and involved a simple idea supplemented by updating formulae which ameliorated the inherent computational difficulties to some extent. This exchange algorithm requires a candidate list of points which reasonably covers the design space, as well as an initial design. Then, it considers the effect of exchanges between each design point and each candidate point. When each has been evaluated, the exchange which most increases the determinant of the information matrix is executed and this completes one iteration of the algorithm. Iteration continues until convergence, which is guaranteed because of the nondecreasing sequence of determinants and the existence of an upper bound. This algorithm can accommodate irregularly-shaped design spaces as long as a suitable candidate list can be constructed. But even with the computational shortcut formulae, this method will sputter for large problems, because the required candidate list will necessarily be so large and each iteration searches it n times—once for each design point—but only makes a single exchange.

Consequently, many other approaches have been proposed to improve upon the original algorithm. Cook and Nachtsheim [26] increase computational efficiency by exchanging each design point (if the determinant can be improved) instead of just one every iteration. Johnson and Nachtsheim [58] only consider exchanging the k least important design points at each iteration. Though these adjustments are faster, they still require a candidate list which suffers from the curse of dimensionality as the number of factors grow large.

To address this issue, Atkinson et al. [10] limit not just the number of design points to consider exchanging, but also the size of the candidate list, at each iteration. Meyer and Nachtsheim [77] abolish the candidate list altogether by making exchanges coordinate-wise. The latter procedure is computationally superior and performs well compared to the more methodical algorithms. In the original paper, they indicated that this coordinate-exchange algorithm could not be implemented for irregular design spaces, but more recent work seems to indicate that it is possible and has been done [86, 59].

Another approach, developed independently of Fedorov, is the DETMAX al-

gorithm of Mitchell [79]. Instead of exchanging, his procedure adds or subtracts design points sequentially to increase the determinant, allowing excursions which result temporarily in designs several runs larger or smaller than n, before invariably returning to the original design size.

2.2 Model-Robust Design

A serious criticism of optimal design theory is that these designs have an undue reliance upon the assumed model form. They will not be optimal, and may not even be acceptable, if the true model form is different than what initially supposed. If the model is overspecified, efficiency is lost by devoting experimental resources to estimate unnecessary parameters; worse, if it is underspecified, the optimal design will not even be able to estimate the true model.

Box and Draper [18] first argued that instead of focusing on optimal designs, it is more important to design experiments that are robust to model misspecification. They proposed as a model-robust criterion the mean squared deviation from the true response, a quantity which can be decomposed into a variance term and a bias term. Thus, a model-robust design would be one that minimizes this quantity. However, since this criterion depends upon the parameters in the true model which are not in the assumed model, it is of limited practical usefulness.

There seems to be two streams of research in this area. One approach is theoretical, consisting largely of continuous design theory and/or special cases. The other is practical, focusing on flexible algorithms for exact designs. Our work falls into the latter category, developing methods that are intuitive and useful to the practitioner. However, as a matter of completeness and because ideas from more theoretical work are useful and important, we review both.

2.2.1 Theoretical Approaches

A sensible approach to the model-robust experimental design problem is one proposed by Läuter [71]. She assumes that the class of potential models, $\mathcal{F}$, is known though both a finite and infinite class of models is allowed for. In the finite case her criteria reduce to

1. $\max_{\xi \in \Xi} \sum_{f \in \mathcal{F}} Q^*(f) p(f) \, |\mathbf{M}_f(\xi)|$
2. $\max_{\xi \in \Xi} \sum_{f \in \mathcal{F}} Q^*(f) p(f) \ln\{|\mathbf{M}_f(\xi)|\}$
3. $\min_{\xi \in \Xi} \max_{\mathbf{x} \in \chi} \sum_{f \in \mathcal{F}} Q^*(f) p(f) \mathbf{V}_f(\mathbf{x}, \xi)$

where $Q^*(f)$ is a weight given to model f and $p(f)$ is a function which standardizes the criterion function (i.e. the determinant) so that they are of comparable order of magnitude. If $Q^*(f) = Q^* \; \forall f \in \mathcal{F}$ and $p(f) = p \; \forall f \in \mathcal{F}$, Criterion 2 is equivalent to $\prod_{f \in \mathcal{F}} |\mathbf{M}_f(\xi)|$. An equivalence theorem is proven, in the spirit of [65], which shows the correspondence between Criteria 2 and 3 and a computing procedure is given which guarantees convergence for Criterion 3. It is also shown that the design which maximizes a slightly stronger form of Criterion 1 can also be iteratively computed.

Though conceived for asymptotic designs, this approach is similar in some ways to the approach we will suggest, since the idea of allowing the experimenter to define a class of plausible models is compelling in its practicality. Several authors have adopted this approach as well. Cook and Nachtsheim [28] developed a parallel to Läuter for the case of linear optimality, an example of which is the integrated variance criterion. Later, Dette [31] used the theory of canonical moments to give more explicit solutions for this product criterion. These papers, however, are limited to continuous designs and unconstrained cuboidal design regions.

Another approach optimizes the determinant of the information matrix of one model subject to requiring the determinant (or efficiency) of other models to be above some value (e.g. Stigler [97] and Dette and Franke [32]). Imhof and Wong [57] give a graphical method to find maximin designs with respect to two criteria (instead of two model forms) and Dette and Franke [33] explicitly characterize continuous maximin designs in the specific case of polynomial regression on $[-1, 1]$, where they maximize the minimum efficiency with respect to possible polynomial models as well as a goodness-of-fit criterion.

An overview of some of the theoretical side was given by Pukelsheim and Rosenberger [88] who compared several design-construction methods and evaluated how well they met three goals for a single factor study: (1) discriminate between a second-order and third-order model; (2) make inferences about the second-order model; (3) make inferences about the third-order model. They used as a baseline

the $\mathcal{D}$-optimal designs for each of the goals separately. Then, they compared three methods: (1) Divide the design into two equal parts, half of which uses an equispaced design and the other half which is $\mathcal{D}$-optimal for the second-order model; (2) Designs where the geometric mean of the design criteria was optimized; and (3) $\mathcal{D}$-optimal constraint designs in which one of the goals was optimized while the others were constrained in some way. The methods of this paper produce continuous designs and apply to a one-factor polynomial regression model.

Since then, a significant amount of additional work has been done on various aspects of designing optimal experiments when the degree of polynomial regression is unknown, utilizing the constrained optimal design idea. Montepiedra and Fedorov [80] examine a linear model in which the fitted model is

$$y_i = \boldsymbol{\beta}' f_1(\mathbf{x}_i) + \epsilon$$

but the true response model includes $\boldsymbol{\delta}' f_2(\mathbf{x}_i)$, an additional contamination function which is not modeled.

They propose, essentially, constrained $\mathcal{D}$-optimal designs where the determinant of the information matrix is maximized subject to the bias being controlled. Liu and Wiens [74] and Fang and Wiens [40] study a similar setting except the contamination function contains an unknown but continuous and bounded function. They introduce *bounded bias* and *generalized bounded bias* designs which minimize the determinant of the information matrix while constraining a function of the bias to be less than a specified bound. For more research in this vein, see [106, 107, 108, 111, 109, 110, 53]. Though most of this sort of work is for continuous designs, we do note that Fang and Wiens [39] give a simulated annealing algorithm which allows the construction of exact designs.

Chang and Notz [24] provided an older review of this area of research and admit that to use methods like those employing contamination functions requires one to make unsubstantiated assumptions about the true model. They summarize the value of these models: "The practical value of [the model-robust] results ... is probably in alerting us to the dangers of ignoring the approximate nature of any assumed model and in providing some insight concerning what features a design should have in order to be robust against departures from an assumed model while

allowing good fit of the assumed model. This insight may be more valuable in practical settings than a slavish adoption of any particular mathematical model."

Indeed, a problem with many of these contamination function approaches is that they must make assumptions about the contamination function if they are able to do any analytical work with respect to the bias. For instance, some assume the contamination function is from a family of random functions with a specified variance, or that the family of functions are bounded by some known function. At the very least, they require the experimenter to specify a parameter quantifying the interest in bias versus variance. This may be difficult and unintuitive in practice.

2.2.2 Practical Approaches

In contrast to the theoretical stream, there has been some development of more flexible, exact methods as well. Heredia-Langner et al. [54] used genetic algorithms to generate model-robust optimal designs, using the set of models idea of Läuter. This was accomplished by introducing a desirability function which utilized $\mathbf{M}_{\mathcal{F}}$.

One approach which is in the spirit of the original optimal design critique [18] is by Welch [105], in which protection is sought from a maximum discrepancy between the model and the true response. This requires the specification of a "maximum discrepancy" parameter which is unlikely to be supplied by the experimenter. Welch provides an algorithm for both continuous and exact designs and suggests a compromise between all-bias and all-variance designs enables by a choice of the maximum discrepancy parameter that is robust to both.

A key development in the area of model-robustness is a Bayesian model-robust method proposed by DuMouchel and Jones [37] in which they assume all possible terms in the model can be categorized as either *potential* or *primary*. For instance, perhaps a screening experiment is to be designed so that certain main effects are of particular interest; these would be termed *primary terms*. However, the experimenter might want to hedge against higher-order terms; these would be categorized as *potential terms*. Suppose there are s_1 primary terms and s_2 potential terms. Let $\mathbf{X} = (\mathbf{X}_{pri}|\mathbf{X}_{pot})$ where $\mathbf{X}$ has $s_1 + s_2$ columns and let $\boldsymbol{\beta} = (\boldsymbol{\beta}_{pri}|\boldsymbol{\beta}_{pot})$ be a vector partitioned similarly.

For the primary terms, DuMouchel and Jones assume a noninformative prior

and for the potential terms, since they unlikely to have large effects, they use a $N(0, \kappa^2\mathbf{I})$ prior distribution, where κ is a prior precision parameter. Then, assuming $\sigma = 1$ and data normally distributed as $Y|\beta \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I})$, they calculate the posterior distribution of $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta}|Y \sim N(\mathbf{A}^{-1}\mathbf{X}'Y, \mathbf{A}^{-1}) \tag{2.1}$$

where $\mathbf{A} = [\mathbf{X}'\mathbf{X} + \mathbf{K}/\kappa^2]$ and $\mathbf{K}$ is a $(s_1 + s_2) \times (s_1 + s_2)$ diagonal matrix with 0 on the first s_1 diagonals and 1 on the last s_2. They choose the design that maximizes $|\mathbf{A}|$, which is model-robust in the sense that it accounts for both the "primary" model as well as the full "potential" model. This method can produce robust designs even in cases when the total number of parameters considered is greater than the number of observations.

Neff [83] used this idea in what she called a two-stage Bayesian $\mathcal{D}$-$\mathcal{D}$ optimal model-robust design. In the first stage the design is chosen to maximize $|\mathbf{A}|$. Then, using information from the first stage, the second stage design is constructed. Later Ruggoo and Vandebroek [91] improved this by putting a prior on κ and using its posterior distribution from the first stage to serve as its prior for the second stage.

A non-Bayesian two-stage robust-design procedure due to Montepiedra and Yeh [81] suggested that in the first stage an optimally discriminating design should be used and then incorporated into a second stage in which an optimal design should be chosen according to the most likely model from the first stage.

Li and Nachtsheim [73] focus on factorial designs in which in addition to main effects, up to b interactions may be present. They use *estimation capacity*, based on the work of Sun [98], the ratio of the number of estimable models to the number of possible models for a given design, and show that a new class of designs, model-robust factorial designs, outperform competitors with respect to a compound criteria which involves estimation capacity as well as $\mathcal{D}$-optimality. Later, Agboto and Nachtsheim [1] proposed a Bayesian alternative to the above method, framing the problem in a decision-theoretic context, defining a utility function whose expectation is maximized by choosing a particular design. They combined the approach of Li and Nachtsheim [73] with DuMouchel and Jones [37] to develop the so-called Bayesian Model Robust Optimal design criterion. A key component

to this method is the choice of model priors, which are based on the hierarchical model priors developed by Chipman et al. [25].

A more recent approach by Tsai and Gilmour [100] uses an approximation to $\mathcal{A}_s$ optimality to achieve model-robustness for all possible subsets of a specified largest model. Another interesting model-robust procedure, by Berger and Tan [11], for the case in which a mixed model is to be fit, utilizes a maximin approach to guard again potential models as well as possible parameter values.

2.3 Multiresponse Regression

An altogether different area, necessary in the development of Chapters 3 and 5, is multiresponse regression, which seeks to take advantage of relationships that may exist among the responses. We can extract more information from the data if we utilize those relationships and in fact multivariate regression allows for more precise parameter estimates than fitting each response separately [115]. The classical multivariate regression model [4] forces each response variable to have the same design matrix, while the more flexible multiresponse regression model allows the form of the relationship between each response and the factors to be different.

There are two multiresponse regression models, each with their own estimation methods, that were proposed at about the same time: The seemingly unrelated regression (SUR) model [115] and the Box-Draper method [19]. We will review the SUR model in the following section, give an overview of the Box-Draper model, and briefly describe the classical multivariate regression model.

2.3.1 Seemingly Unrelated Regression Model

Zellner [115] introduced the *seemingly unrelated regression* (SUR) model (see also [117], [116]), a multiresponse regression model which allows the correlation structure among the responses to be considered explicitly and also allows each response to have a unique functional relationship with the factors.

To define the model, suppose there are r responses, q_i parameters for response i, $q = \sum_i q_i$ total parameters in the multiresponse model, and n observations for

each response. Then let

$$\mathbf{y}_i = \mathbf{Z}_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i,\ i = 1, 2, \ldots, p$$

be the linear regression for a given response, with $\mathbf{y}_i$ an n-vector of observations for this response, $\mathbf{Z}_i$ the $n \times q_i$ expanded design matrix, $\boldsymbol{\beta}_i$ the $q_i \times 1$ vector of parameters, and $\boldsymbol{\epsilon}_i$ the n-vector of errors. This multivariate model can be completely specified as

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_r \end{pmatrix} = \begin{pmatrix} \mathbf{Z}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{Z}_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_r \end{pmatrix} + \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_r \end{pmatrix} \tag{2.2}$$

or more concisely as

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.3}$$

where $\mathbf{Y}$ and $\boldsymbol{\epsilon}$ are now $nr \times 1$ vectors, $\boldsymbol{\beta}$ is a $q \times 1$ vector, and $\mathbf{Z}$ is a $nr \times q$ matrix and all of these quantities are shown in (2.2).

We assume that the error vector is distributed as $N(\mathbf{0}, \boldsymbol{\Omega})$, where $\boldsymbol{\Omega} = \boldsymbol{\Sigma} \otimes \mathbf{I}_n$ and '$\otimes$' is the Kronecker product. This model assumes independence across observations within a particular response and correlation between the responses.

Once the model is in the form (2.3), an estimate for $\boldsymbol{\beta}$ is given by the generalized least squares estimator (also known as the Aitken estimator):

$$\hat{\boldsymbol{\beta}}^* = (\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z})^{-1}\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Y} \tag{2.4}$$

with

$$Var(\hat{\boldsymbol{\beta}}^*) = (\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z})^{-1} \tag{2.5}$$

Under the assumption of normality (i.e. that $\mathcal{G}$ is an np-variate normal distribution), $\hat{\boldsymbol{\beta}}^*$ is also the maximum likelihood estimator [115]. Since $\boldsymbol{\Sigma}$ is in most cases unknown, it must be estimated before (2.4) can be employed. Zellner [115] proposed

$$\hat{\boldsymbol{\Sigma}}_{ij} = \frac{1}{n-q}\{s_{ij}\} = \frac{1}{n-q}\{(\mathbf{y}_i - \mathbf{Z}_i\hat{\boldsymbol{\beta}}_i)'(\mathbf{y}_j - \mathbf{Z}_j\hat{\boldsymbol{\beta}}_j)\} \tag{2.6}$$

where $\hat{\boldsymbol{\beta}}_i = (\mathbf{Z}_i'\mathbf{Z}_i)^{-1}\mathbf{Z}_i'\mathbf{y}_i$ are the ordinary least squares estimates for each of the responses separately. Using (2.6) and letting $\hat{\boldsymbol{\Omega}} = \hat{\boldsymbol{\Sigma}} \otimes \mathbf{I_n}$, we can calculate an estimate of $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}'\hat{\boldsymbol{\Omega}}^{-1}\mathbf{Z})^{-1}\mathbf{Z}'\hat{\boldsymbol{\Omega}}^{-1}\mathbf{Y} \tag{2.7}$$

Zellner showed that (2.7) is asymptotically equivalent to (2.4), the "pure" Aitken estimator.

2.3.2 Box and Draper Estimation Method

Box and Draper [19] developed a more general Bayesian setting to handle multivariate regression, in which the primary goal is to calculate the posterior distribution of the parameters. Suppose we have n r-variate observations, and each response can be modeled as

$$y_{iu} = f_i(x_{iu}^1, x_{iu}^2, \ldots, x_{iu}^\ell; \theta_1, \theta_2, \ldots, \theta_m) + \epsilon_{iu}$$

where $i = 1, \ldots, r$, $u = 1, \ldots, n$, there are ℓ independent variables, and there are m common parameters (though it is not required that all of the independent variables or all of the parameters be explicitly related to each response). Let $\mathbf{y}_u' = (y_{1u}, y_{2u}, \ldots, y_{ru})$ be the response vector for the u^{th} observation, $\boldsymbol{\theta}' = (\theta_1, \ldots, \theta_m)$, and $\boldsymbol{\Sigma} = \{\sigma_{ij}\}$ be the covariance matrix among the responses. Further, define

$$v_{ij} = \sum_{u=1}^{n} \{y_{iu} - f_i(x_{iu}^q, \boldsymbol{\theta})\}\{y_{ju} - f_j(x_{ju}^q, \boldsymbol{\theta})\}$$

Then, assuming normal data and a noninformative, Jeffreys priors on $\boldsymbol{\theta}$ and $(\sigma_{ij})^{-1}$, they derive the posterior distribution for $\boldsymbol{\theta}$ to be

$$p(\boldsymbol{\theta}|\mathbf{y}) = C\,|v_{ij}|^{-\frac{n}{2}} \tag{2.8}$$

where $\mathbf{y}' = (\mathbf{y}_1', \ldots, \mathbf{y}_n')$ and C is a constant. To obtain an estimate of $\boldsymbol{\theta}$ we would choose the values of the parameters which would maximize (2.8). If there are linear dependencies between the responses, this method breaks down, but Khuri and Cornell [63] reserve a portion of their book to discuss this case and how to

deal with it.

2.3.3 Multivariate Regression Model

A special case of the SUR model is when the expanded design matrix is common to all responses. In that case, the model is

$$\underset{(N \times r)}{\mathbf{Y}} = \underset{(N \times q)}{\mathbf{Z}} \underset{(q \times r)}{\boldsymbol{\beta}} + \underset{(N \times r)}{\boldsymbol{\epsilon}}$$

and the least squares estimates are $\hat{\boldsymbol{\beta}} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}$, a $q \times r$ matrix whose i^{th} column is the ordinary least squares estimate for response i. These estimates are unbiased with

$$Var(\hat{\boldsymbol{\beta}}) = \boldsymbol{\Sigma} \otimes (\mathbf{Z}'\mathbf{Z})^{-1}$$

where $\boldsymbol{\Sigma}$ is the variance-covariance matrix of the responses.

2.4 Multiresponse Experimental Design

For experiments with multiple responses, one might consider designing it with this multivariate structure in mind. Thus, the design should account for the correlation between responses, if possible, and also for the differences in the form of the relationships between the factors and each of the responses.

In univariate optimal design, the form of the assumed model is of great importance in the determination of an optimal design. Thus, intuitively, if the form of the assumed model is the same for each response in a multiresponse situation, it would seem to follow that the univariate optimal design would be the same as the multivariate optimal design. This is, in fact, true in the case of $\mathcal{D}$-optimality (see, e.g., Chang [22]).

On the other hand, if the form of the assumed model differs across responses, the optimal design for the first response is probably different than the optimal design for the second. Developing designs that are optimal with respect to all responses taken together, while considering the covariance matrix as well, is the goal of multiresponse optimal design of experiments.

Perhaps the first to consider multiresponse experimental design was Daniel [30]. He extended standard fractional factorial designs to take advantage of the fact that in a two-response system both may not be related to all of the same factors. He did not consider the correlation between the responses, assumed that those factors influencing each response were known, and found it difficult to extend his approach to more than two responses. Another early consideration of multiresponse design was by Roy et al. [90], who also studied how the 2^k factorial design might be extended to account for situations in which different responses were related to different sets of factors, or the case in which different responses were related to the same set of factors but with different effects (i.e. different model-forms). Other work, such as [67] and [21], has also considered the special case of two responses.

Draper and Hunter [34] were the first to examine the problem of optimal multiresponse experimental design. Using a Bayesian formulation with a noninformative prior for the parameter vector $\boldsymbol{\beta}$ and the assumption of normal errors, they devised the following criterion, which if maximized would in turn maximize the posterior information available with respect to the parameters:

$$\left|\tilde{\mathbf{X}}'\left(\boldsymbol{\Sigma}^{-1}\otimes\mathbf{I}\right)\tilde{\mathbf{X}}\right| = \left|\sum_{i=1}^{r}\sum_{j=1}^{r}\sigma^{ij}\mathbf{X}_i'\mathbf{X}_j\right| \tag{2.9}$$

where r is the number of responses, σ^{ij} is a known variance or covariance, $\mathbf{X}_i$ is the Jacobian of the possibly nonlinear multivariate response function, and $\tilde{\mathbf{X}} = (\mathbf{X}_1, \ldots, \mathbf{X}_r)$. In the case of a linear model $\mathbf{X}_i$ is the customary expanded design matrix. Equation (2.9) is similar to the univariate D-optimal criteria, except it maximizes the determinant of a weighted sum of $\mathbf{X}_i'\mathbf{X}_i$ matrices. In order for the matrices in (2.9) to be conformable all $\mathbf{X}_i$ must have the same number of columns, which implies that each response depends on the same set of parameters. This is indeed the assumption, though some columns are allowed to be $\mathbf{0}$.

Later, the same authors [35] extended this work to the case of a multivariate normal prior on the parameters, proposing a criterion which incorporates the prior precision matrix:

$$\left|\tilde{\mathbf{X}}'\left(\boldsymbol{\Sigma}^{-1}\otimes\mathbf{I}\right)\tilde{\mathbf{X}} + \boldsymbol{\Omega}^{-1}\right|$$

Then, Box and Draper [20] extended the same basic idea to include provisions

for non-homogenous variances among blocks, some known and some unknown.

Another area of multiresponse design research has been to give conditions for which the optimal design can be obtained irrespective of $\mathbf{\Sigma}$. Krafft and Schaefer [69] first gave a condition, followed by Kurotschka and Schwabe [70] and Bischoff [15, 16]. Among the several conditions given, these authors showed that if the response models are nested, the $\mathcal{D}$-optimal design is invariant to $\mathbf{\Sigma}$. This, of course, assumes that the response model forms are known.

To calculate $\mathcal{D}$-optimal designs for multiresponse systems, there are four methods of interest which we will describe. Before that, we will extend the design notation of §1.2 to the multiresponse case.

Define the multiresponse information matrix

$$\mathbf{M}_m(\xi, \mathbf{\Sigma}) = \int_{\chi} \boldsymbol{\rho}(\mathbf{x}) \mathbf{\Sigma}^{-1} \boldsymbol{\rho}'(\mathbf{x}) \, d\xi \tag{2.10}$$

where ξ is a design measure defined in §1.2, $\boldsymbol{\rho}^{'}(\mathbf{x}) = diag(\mathbf{z}_1^{'}(\mathbf{x}), \mathbf{z}_2^{'}(\mathbf{x}), \ldots, \mathbf{z}_r^{'}(\mathbf{x}))$ is an $r \times q$ matrix, and $\mathbf{z}_i^{'}(\mathbf{x})$ is a vector with entries of the same form as the expanded design matrix $\mathbf{Z}_i$ in (2.2) for response i and design point $\mathbf{x}$. Further, for a particular point $\mathbf{x}$, the prediction variance matrix is

$$\mathbf{V}_m(\mathbf{x}, \xi, \mathbf{\Sigma}) = \boldsymbol{\rho}^{'}(\mathbf{x}) \mathbf{M}_m^{-1}(\xi, \mathbf{\Sigma}) \boldsymbol{\rho}(\mathbf{x}) \tag{2.11}$$

For the exact design defined in (1.3), we have

$$\mathbf{M}_m(\xi_n, \mathbf{\Sigma}) = \sum_{i=1}^{n} \boldsymbol{\rho}(\mathbf{x_i}) \mathbf{\Sigma}^{-1} \boldsymbol{\rho}'(\mathbf{x_i}) \tag{2.12}$$

$$= \mathbf{Z}' \mathbf{\Omega}^{-1} \mathbf{Z} \tag{2.13}$$

$$= \left[Var(\hat{\boldsymbol{\beta}}^*) \right]^{-1} \tag{2.14}$$

Fedorov [41] proved an equivalence theorem which facilitated an algorithm to generate continuous $\mathcal{D}$-optimal designs for the multiresponse regression model. Simply put, a design ξ^* is $\mathcal{D}$-optimal if and only if the maximum over all points in the design space of $tr\left[\mathbf{\Sigma}^{-1} \mathbf{V}(\mathbf{x}, \xi^*, \mathbf{\Sigma})\right]$ is equal to the total number of parameters being estimated. This allowed Fedorov to develop a design algorithm, which uses

this equivalence to move toward the optimal design.

Fedorov's algorithm assumes that $\mathbf{\Sigma}$ is known. Wijesinha [112] uses the above algorithm as a basis for a sequential procedure which begins with $\mathbf{\Sigma} = \mathbf{I}$ and calculates an ever-improving estimate of $\mathbf{\Sigma}$ at each iteration. Wijesinha shows that this procedure converges to the true multiresponse $\mathcal{D}$-optimal design.

Chang [23] concentrated on response surface designs (designs for polynomial models of order 1 and 2) and showed via simulation that using a design support consisting only of the union of the $\mathcal{D}$-optimal support for each of the individual responses, nearly $\mathcal{D}$-optimal continuous designs could be constructed with $\mathbf{\Sigma} = \mathbf{I}$.

All three of the algorithms above must solve an optimization subproblem during each iteration of the algorithm. Chang's algorithm would appear to be the most computationally inexpensive because it restricts the design space. However, Atashgah and Seifi [8] formulated the problem as a semi-definite program, which turns the problem into a single, large optimization problem. They construct both continuous and exact designs using their method, though they require a discretization of the design space reminiscent of the candidate list required by exchange algorithms. They also assume that $\mathbf{\Sigma}$ is known or estimated from previous data.

Almost no work has been done to extend the idea of model-robustness to the multivariate setting. Kim and Draper [67] examine the case of two responses with no common parameters. They assume that the fit will be linear with a small amount of quadratic bias and that the correlation between the responses can be approximately known. The general model-robust idea of Box and Draper [18] is used to define a metric, now a matrix, which represents the mean squared deviation from the true response, and can be decomposed into a variance and bias component. They seek to minimize the trace of this metric matrix, but the approach is severely limited by the fact that this method is not readily extendible to more than two responses. It also demands knowledge of the model being protected against.

Yue [114] used the idea of an approximately polynomial model with an unknown contamination function (as in [74, 40]) but with an extension to the multiresponse setting, using the Box-Draper model formulation of Section 2.3.2. They assume that the contamination function (i.e. bias) belongs to some class of random functions, and use an average expected loss criterion which is similar to that used by Box and Draper [18] which can be partitioned into a variance and bias compo-

nent. Later, Liu and Yue [75] used a different criteria, namely $\mathcal{P}$-optimality, which maximizes the coverage probability of the confidence ellipsoid on the regression coefficients, in the same general multiresponse context.

These multiresponse approaches, while general in the sense that they do not restrict the bias functions to be of a particular form, are not necessarily practical as they assume the bias originates from a class of random functions with 0 mean and specified variance. Designs can be produced under certain assumptions about the bias functions and responses, but it seems unlikely that in real situations there would be a plausible basis for making such assumptions.

2.5 Split-Plot Experimental Design

Though dating back to Yates [113], split-plot experiments have undergone a renaissance of sorts in the experimental design literature over the last twenty years. Research by Lucas and coauthors [76, 2] and Leitsinger et al. [72], as well as an abundance of later work, suggests a rising level of awareness of these experiments among researchers and practitioners. Recently, Jones and Nachtsheim [61] gave a thorough review of these developments.

Optimal design in this area, dominated by Goos and coauthors (e.g. [49, 50, 51, 59]), utilizes the $\mathcal{D}$-optimality criterion to produce exact designs that estimate model parameters as precisely as possible. Goos and Vandebroek [50] developed a $\mathcal{D}$-optimal split-plot exchange algorithm for which the experimenter specifies the number and size of each whole plot. The same authors have also presented algorithms which decide the number and size of the whole plot automatically [49, 51]. These procedures require estimates of the ratio between the whole plot and subplot variance, but empirical evidence suggests that the designs are not very sensitive to this quantity.

Various other papers by Goos and coauthors address several additional issues in the design of these experiments. Goos [46] compares the optimal design approach to equivalent estimation and orthogonal approaches in the context of split-plot experiments, and concludes that the optimal design approach provides adequate flexibility and superior efficiency. Jones and Goos [59] develop an alternative, computationally more efficient, split plot coordinate exchange algorithm, inspired

by Meyer and Nachtsheim [77] which does not require a candidate list. Goos and Donev [47] use optimal design algorithms for split plot designs to construct designs in which both mixture and process variables are present. Arnouts and Goos [5] give determinant- and inverse-updating formulae for the information matrix for several scenarios of interest in the split-plot situation. They give results for 1) changes in easy-to-change factor levels; 2) changes in hard-to-change factor levels; 3) points exchanged between two whole plots; and 4) changes in the number of runs in a whole plot. This is important because these updating formulae allow the exchange algorithms to be executed with computational advantage.

Besides optimal design, there are other approaches to the design of split-plot experiments. One early algorithmic approach was by Trinca and Gilmour [99], who give methodology for the more general setting in which there are any number of randomization strata (as opposed to the split-plot case in which there are only two). Instead of using traditional notions of design optimality, they adopt orthogonality as their chief criterion and develop a method which builds the design stratum by stratum, where each stratum is close to orthogonal to the others. This procedure does not require prior information about the variance components.

Lucas and coauthors [76, 2, 3] have studied the advantages of split-plot designs, and argue for superefficiency, the phenomenon that in some situations split-plot designs have higher efficiencies than CRDs. Goos and Vandebroek [51] also make this claim, in terms of both $\mathcal{D}$- and $\mathcal{G}$-efficiencies. In another line of research [43, 62, 44, 104], researchers examine the devastating effects of analyzing a split-plot experiment as a CRD, and note in particular that if the levels for all factors are not independently reset, a *de facto* split-plot design has been run.

Fractional Factorial Split Plot designs have also been extensively studied [55, 12, 17, 14], often using the minimum aberration criteria to differentiate competing designs. Other authors have generalized second-order response surface designs, such as central composite and Box-Behnken, to the split-plot case (e.g. [36, 103, 84]). A related area of research is the so-called equivalent estimation designs which allow estimation of split-plot models via ordinary least squares [72, 103, 84, 102].

Chapter 3

Multiresponse Exchange Algorithms for Constructing Single Response Model-Robust Experimental Designs

3.1 Introduction

Since Kiefer [64] debuted the idea of optimal design of experiments, a vast literature has grown up around the notion of choosing a design based upon some numerical criterion. The most common is $\mathcal{D}$-optimality, which chooses the design minimizing the generalized variance of the regression parameter estimates. Though standard designs can be used in most design situations, optimal procedures are useful when, for instance, there are constraints on the design space, some factors are categorical, or nonstandard sample sizes are required.

An example in which optimal designs are a natural choice is in the case of mixture experiments because of the constrained nature of the design region. Heinsman and Montgomery [52] describe an experiment involving a household product with four surfactant mixture factors. Beyond the mixture constraint, the factors were restricted as well which made an optimal design natural. However, such a design would require the complete specification of the form of the mixture regression model. For instance a special cubic Scheffé polynomial model might be chosen, though it is unknown before the experiment whether this is the correct model

form. We provide a procedure which allows the experimenter to obtain a design which does not assume a single model form, but rather accounts for a class of user-specified models. We revisit this example later.

We propose a new, practical method which produces designs robust for a set of user-defined possible models by maximizing the product of the determinants of the information matrices. These ideas are motivated by the a connection between multiresponse regression [115], multiresponse optimal design [41], and a continuous model-robust optimal design approach due to Läuter [71]. To implement these ideas, we develop an exchange algorithm which generalizes existing univariate methods.

This chapter is organized as follows. In the next section we review the technical background and describe the basic approach taken to find model-robust designs. We then review some basic univariate exchange algorithms and give a generalization of the univariate determinant-updating formula which is used to drive the model-robust exchange algorithm. We next give several examples illustrating our method and compare our designs to those of DuMouchel and Jones [37] and Heredia-Langner et al. [54]. We conclude with a discussion of the procedure and its results.

3.2 Setting and Proposed Approach

Though given in §1.2, the regression and design setting are reviewed here. Suppose one is interested in performing an experiment with a single quantitative response variable, y, and a factors (quantitative or categorical), $\mathbf{x} = (x_1, \ldots, x_a)$. We assume that the classical univariate linear regression model will be fit, where $y_i = f'(\mathbf{x}_i)\boldsymbol{\beta} + \epsilon_i, i = 1, \ldots, n$ with $\boldsymbol{\beta}$ a p-vector of parameters and $f(\mathbf{x})$ the p-vector valued model function, though p and the precise form of $f(\mathbf{x})$ are unknown. In matrix notation, we have $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where $\mathbf{y}$ is an n-vector (independent observations), $\mathbf{X}$ is an $n \times p$ expanded design matrix, and $\boldsymbol{\epsilon}$ is also an n-vector with $E(\boldsymbol{\epsilon}) = \mathbf{0}$ and $Var(\boldsymbol{\epsilon}) = \sigma^2\mathbf{I}_n$. We assume also that the least squares criterion is used to estimate $\boldsymbol{\beta}$, in which case the estimator is $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ with $Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$.

To fit such a model, the design must be chosen and y_i observed at each of the designs points, $\mathbf{x}_i$. Let $\mathcal{X}$ be the design space, Ξ be the set of all possible designs

and $\xi_n(\mathbf{x}) \in \Xi$ be a discrete, n-point design:

$$\xi_n = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_d \\ n_1 & n_2 & \dots & n_d \end{pmatrix}$$

where n is the total number of experimental runs, and n_i, $i, \dots, d$ is the number of runs performed at design point, $\mathbf{x}_i$. We define the information matrix in this case as $\mathbf{M}(\xi) = \sigma^{-2} \sum_{i=1}^{n} f(\mathbf{x}) f'(\mathbf{x})$ and in the specific instance of the linear regression model, $\mathbf{M}(\xi) = (\mathbf{X}'\mathbf{X})/\sigma^2 = [Var(\hat{\boldsymbol{\beta}})]^{-1}$.

An optimal design approach would attempt to find the n points, $\mathbf{x}_i \in \chi, i = 1, \dots, n$, such that some criterion, $\phi(\mathbf{M}(\xi))$, is optimized. Many criteria have been proposed, but probably the most popular and mathematically tractable is the $\mathcal{D}$-optimality, for which $\phi(\mathbf{M}(\xi)) = |\mathbf{M}(\xi)|$. Assuming normality, such an optimal design minimizes the volume of the confidence ellipsoid of the parameters.

Since the precise form of $f(\mathbf{x})$ is generally unknown, we might make the weaker assumption that there exists a set of r possible models $\mathcal{F}$ that might be fit. Läuter [71] presented this idea for continuous designs ξ (designs with asymptotic run sizes), and introduced a model-robust criterion similar to $\phi\left(\mathbf{M}_{\mathcal{F}}(\xi)\right) = \prod_{f \in \mathcal{F}} |\mathbf{M}_f(\xi)|$, where $\mathbf{M}_{\mathcal{F}} = (\mathbf{M}_1, \dots, \mathbf{M}_r)$ and $\mathbf{M}_f$ is the information matrix for model f. Thus, the design which maximizes $\phi(\mathbf{M}_{\mathcal{F}}(\xi))$ over all possible designs might be considered robust to the models in $\mathcal{F}$. Cook and Nachtsheim [28] utilized this idea to develop linear-optimal designs focusing on prediction, and Dette [31] used the theory of canonical moments to give more explicit solutions for this product criterion. These papers, however, are limited to continuous designs and unconstrained cuboidal design regions.

Our discrete approach springs from Läuter's idea, since allowing the experimenter to define a class of possible models is practically compelling. When model-robustness is viewed in this way, it is closely related to multiresponse optimal design, which has a literature in its own right; see [41], [63], [23], and [9]. These methods are based upon a multiresponse regression model due to Zellner [115] which allows the functional form of the factors to be different for each response and can produce more precise estimates of the regression parameters by considering the covariance structure of the responses.

Zellner's seemingly unrelated regression (SUR) model was given in §2.3.1, but

is reviewed here as well. This model, with r responses, can be written as

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_r \end{pmatrix} = \begin{pmatrix} \mathbf{X}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{X}_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_r \end{pmatrix} + \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_r \end{pmatrix} \tag{3.1}$$

where each $\mathbf{y}_i$ and $\boldsymbol{\epsilon}_i$ are n-vectors, $\boldsymbol{\beta}_i$ is a q_i-vector, and $\mathbf{X}_i$ is a $n \times q_i$ expanded design matrix for response i and the total number of parameters is $\sum_{i=1}^{r} q_i = q$. It is assumed that the n observations are independent, but the r responses for the i^{th} observation are correlated as specified by the $r \times r$ covariance matrix $\boldsymbol{\Sigma}$. This leads to an error covariance matrix which is $\boldsymbol{\Omega} = \boldsymbol{\Sigma} \otimes \mathbf{I}_n$ where '$\otimes$' is the Kronecker product. Consequently, the generalized least squares estimator is $\hat{\boldsymbol{\beta}}^* = (\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z})^{-1}\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Y}$ with $Var(\hat{\boldsymbol{\beta}}^*) = (\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z})^{-1}$ where

$$\mathbf{Z} = \begin{pmatrix} \mathbf{X}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{X}_r \end{pmatrix}$$

as seen in (3.1). Then the $q \times q$ multiresponse information matrix is $\mathbf{M}_m = \mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z}$. Thus, for a given $\boldsymbol{\Sigma}$, to find a multiresponse $\mathcal{D}$-optimal design, one must find that which maximizes $|\mathbf{M}_m|$, which, as in the univariate case, will be the design which minimizes the volume of the confidence ellipsoid for the parameters when normality is assumed.

Notice, however, that finding the multiresponse optimal design for r responses with different regression functions should give a design that is simultaneously "good" for all the response models, though not optimal for any particular one. Consequently, when a univariate model-robust design is viewed as one which performs well for a set of specified models, finding such a design is similar to a parallel multiresponse situation in which there are r response models and we calculate the corresponding multiresponse $\mathcal{D}$-optimal design. Since this work was begun, we discovered a technical report [38] which makes the same connection, though the basis of our work is independent of theirs.

Results by Bischoff [15, 16] and Kurotschka and Schwabe [70] prove that when the response models are nested (i.e., when ordered from smallest to largest in terms of the number of parameters, each succeeding model contains the previous), multiresponse optimal designs are invariant to $\mathbf{\Sigma}$. Moreover, since our primary concern is model-robustness, it seems reasonable to assume the identity matrix as the covariance between the r "responses" or models, which when the models in $\mathcal{F}$ are nested gives an attractive multiresponse $\mathcal{D}$-optimal interpretation for the model-robust design. To implement these ideas, we will develop a generalization of the determinant-updating formula used in univariate exchange algorithms, and implement a model-robust exchange algorithm based upon it.

3.3 Multiresponse and Model-Robust Exchange Algorithms

In this section we first review the basic univariate exchange algorithms upon which our methods are based. Then we present a generalization to the matrix-updating formulas used in the univariate procedures, which is equivalent to a multiresponse generalization when when $\mathbf{\Sigma} = \mathbf{I}$. Finally, we introduce our model-robust exchange algorithm, which utilizes this generalization to avoid calculating determinants when evaluating potential exchanges.

3.3.1 Univariate Exchange Algorithms

The idea which buttresses the original exchange algorithm [41] is simple. Starting with a nonsingular design, consider exchanges between each design point and each member of a candidate list of points covering the design space, $\mathcal{X}$. Choose the exchange which most increases the determinant of the information matrix, and repeat until convergence. This brute force method has been made computationally feasible by a quick determinant-updating formula. Fedorov [41] showed that given design ξ_n and model form f, if $\mathbf{x}_j \in \xi_n$ is exchanged for $\mathbf{x} \in \mathcal{X}$ resulting in the new design $\tilde{\xi}_n$, then

$$|\mathbf{M}_f(\tilde{\xi}_n)| = |\mathbf{M}_f(\xi_n)| \, (1 + \Delta_f(\mathbf{x}_j, \mathbf{x}, \xi_n)) \tag{3.2}$$

where

$$\Delta_f(\mathbf{x}_j, \mathbf{x}, \xi_n) = \mathbf{V}_f(\mathbf{x}, \xi_n) - \mathbf{V}_f(\mathbf{x}, \xi_n)\mathbf{V}_f(\mathbf{x}_j, \xi_n) + \mathbf{V}_f^2(\mathbf{x}, \mathbf{x}_j, \xi_n) - \mathbf{V}_f(\mathbf{x}_j, \xi_n) \quad (3.3)$$

under the assumption that $\sigma^2 = 1$, with

$$\mathbf{V}_f(\mathbf{x}, \xi_n) = f^{'}(\mathbf{x})\mathbf{M}_f^{-1} f(\mathbf{x})$$

and

$$\mathbf{V}_f(\mathbf{x}, \mathbf{x}_j, \xi_n) = f^{'}(\mathbf{x})\mathbf{M}_f^{-1} f(\mathbf{x}_j).$$

We can also update the inverse of the information matrix using Lemma 3.3.1 in Fedorov [41] (using notation from Meyer and Nachtsheim [77]):

$$\mathbf{M}_f^{-1}(\tilde{\xi}_n) = \mathbf{M}_f^{-1}(\xi_n) - \mathbf{M}_f^{-1}(\xi_n)\mathbf{F}_1(\mathbf{I}_2 + \mathbf{F}_2^{'}\mathbf{M}_f^{-1}(\xi_n)\mathbf{F}_1)^{-1}\mathbf{F}_2^{'}\mathbf{M}_f^{-1}(\xi_n) \quad (3.4)$$

with $\mathbf{F}_1 = [f(\mathbf{x}), -f(\mathbf{x}_j)]$ and $\mathbf{F}_2 = [f(\mathbf{x}), f(\mathbf{x}_j)]$. The Fedorov algorithm is as follows:

1. Initialize algorithm: Begin with a nonsingular design; construct grid, $C \subset \mathcal{X}$
2. Let $j = 1$.
3. For design point $\mathbf{x}_j$, calculate $\Delta(\mathbf{x}_j, \mathbf{x}, \xi_n)$ as in (3.3) for all $\mathbf{x} \in C$. Choose $\mathbf{x}_j^* = \arg\max_{\mathbf{x}\in\chi} \Delta(\mathbf{x}_j, \mathbf{x}, \xi_n)$.
4. Increment j and if $j < n$ return to Step 3. Else choose

$$j^* = \underset{j\in\{1,\cdots,n\}}{\arg\max}\, \Delta(\mathbf{x}_j, \mathbf{x}_j^*, \xi_n),$$

   exchange $\mathbf{x}_{j^*}$ and $\mathbf{x}_{j^*}^*$, and update the determinant.
5. Update the inverse of the information matrix according to (3.4).
6. If $\Delta(\mathbf{x}_{j^*}, \mathbf{x}_{j^*}^*, \xi_n) < \epsilon$, STOP. Else return to Step 2.

This algorithm generates a convergent nondecreasing sequence of determinants, but will not in general converge to the global optimum. Therefore, it is necessary

to run many instances of the algorithm each with a randomly generated initial design. Despite the cheap updates, Fedorov's eponymous procedure will sputter for large problems, since each iteration searches the candidate list n times—once for each design point—but only makes a single exchange. Consequently, many improvements and alternatives have been proposed over the years; see §2.1.

Cook and Nachtsheim [26] proposed a modified Fedorov exchange algorithm, which mimics Fedorov's original procedure but exchanges each $\mathbf{x}_j$ and $\mathbf{x}_j^*$ in Step 3. This capitalizes on each of the n optimizations that are performed during each iteration, and seems to be as effective as its archetype. It is actually a special case of the k-exchange algorithm [58], which considers only the k least critical design points (those with the smallest prediction variance) for exchange.

In the remainder of this chapter, we develop a multiresponse generalization of the modified Fedorov exchange algorithm and use it to construct single response model-robust designs. We focus on this algorithm since we found it to be faster than the original Fedorov algorithm while producing better designs than the k-exchange. Similar extensions to other existing univariate algorithms, such as DETMAX [79], BLKL [10], and coordinate-exchange [77], could be developed.

3.3.2 Model-Robust Exchange Algorithm

Model-robust exchange algorithms arise from a confluence of motivating factors. First, there is a need to develop practical and intuitive tools which allow experimenters to design experiments for nonstandard situations. Since the form of the model is rarely known in advance, traditional optimal design methods fall short in providing the necessary technical machinery.

Secondly, by noting the similarity between multiresponse optimal design and the single response model-robust design problem we might consider the use of existing multiresponse optimal design methods to construct model-robust designs. However, there exists almost no exact design methods for multiresponse optimal design. This has led us to the development of multiresponse optimal design exchange algorithms based on the multiresponse determinant updating formula (see Chapter 5). In the present context, we use a simplification of the multiresponse procedure to produce model-robust designs.

3.3.2.1 Model-Robust Updating Formula

Recall that q is the total number of parameters in the multiresponse regression model given in (3.1) and r is the number of responses. A multiresponse generalization of the determinant updating formula (5.6) can be developed (again, see Chapter 5) which allows the determinant of the $q \times q$ multiresponse information matrix to be updated by evaluating the determinant of a $2r \times 2r$ matrix when a single point is exchanged.

However, if we assume that $\mathbf{\Sigma} = \mathbf{I}_r$ we can simplify this updating formula so that the update involves only scalar quantities. It is well known that the determinant of a block diagonal matrix is the product of the determinants of the blocks. Thus,

$$\begin{aligned} \left|\mathbf{M}_m(\tilde{\xi})\right| = \left|\mathbf{Z}'_{new}\mathbf{Z}_{new}\right| &= \prod_{i=1}^{r} \left|\mathbf{X}'_{i,new}\mathbf{X}_{i,new}\right| \\ &= \prod_{i=1}^{r} \left|\mathbf{X}'_i\mathbf{X}_i\right| \cdot (1 + \Delta_i(\mathbf{x}_j, \mathbf{x})) \end{aligned} \tag{3.5}$$

where $\mathbf{Z}_{new}$ is the multiresponse expanded design matrix for post-exchange design $\tilde{\xi}$, $\mathbf{X}_{i,new}$ is the univariate expanded design matrix for model i and $\tilde{\xi}$, $\mathbf{X}_i$ is the univariate expanded design matrix for model i and the pre-exchange design ξ, and Δ_i is as in (3.3). The last equality follows from the univariate identity (3.2). This allows us to update the information matrix via (3.5), a scalar. We are now prepared to describe the proposed model-robust modified Fedorov exchange algorithm.

3.3.2.2 Model-Robust Modified Fedorov Exchange Algorithm

As in Läuter [71] we consider, instead of a single model, a finite set of possible models, $\mathcal{F}$. More specifically, let ξ_n be an n-point design and $\mathbf{M}_i(\xi_n)$ be the information matrix for model i where $f_i \in \mathcal{F}$, $i = 1, \cdots, r$. Suppose that we exchange a design point $\mathbf{x}_j$ for an arbitrary point $\mathbf{x}$ in the design region, resulting in a new design $\tilde{\xi}_n$. Then the model-robust optimization criteria can be written as:

$$\boldsymbol{\phi}(\mathbf{M}_{\mathcal{F}}(\tilde{\xi}_n)) = \prod_{i=1}^{r} \left|\mathbf{M}_i(\tilde{\xi}_n)\right| \tag{3.6}$$

$$= \prod_{i=1}^{r} |\mathbf{M}_i(\xi_n)| \, (1 + \Delta_i(\mathbf{x}_j, \mathbf{x}))$$
$$= \boldsymbol{\phi}(\mathbf{M}_f(\xi_n)) \prod_{i=1}^{r} (1 + \Delta_i(\mathbf{x}_j, \mathbf{x}))$$

so that for each iteration of the algorithm, we need to just calculate and maximize $\prod_{i=1}^{r}(1+\Delta_i(\mathbf{x}_j, \mathbf{x}))$ where Δ_i is calculated as in (3.3) for model i. We make a slight adjustment to this criterion so our algorithm will not choose to exchange a point that is so bad that $(1 + \Delta_i(\mathbf{x}_j, \mathbf{x})) < 0$ for an even number of models, which would result in a positive value of our criterion even though the exchange is undesirable. Thus, we choose the exchange which maximizes

$$\prod_{i=1}^{r} (1 + \Delta_i(\mathbf{x}_j, \mathbf{x})) \mathbb{I}(1 + \Delta_i(\mathbf{x}_j, \mathbf{x}) > 0) \tag{3.7}$$

where $\mathbb{I}$ is the indicator function. By (3.5) this is equivalent to updating the multiresponse information matrix under the assumption that $\boldsymbol{\Sigma} = \mathbf{I}$.

Based on the above development, the algorithm is as follows:

1. Initialize algorithm: Begin with a nonsingular design ξ_n; construct grid, $C \subset \mathcal{X}$.

2. Let $j = 1$.

3. For design point $\mathbf{x}_j$, calculate (3.7) for all $\mathbf{x} \in C$. Choose

$$\mathbf{x}_j^* = \arg\max_{\mathbf{x} \in \chi} \prod_{i=1}^{r} (1 + \Delta_i(\mathbf{x}_j, \mathbf{x})) \mathbb{I}(1 + \Delta_i(\mathbf{x}_j, \mathbf{x}) > 0).$$

4. Perform exchange $\mathbf{x}_j^*$ for $\mathbf{x}_j$, updating ξ_n. Update the determinant and also $\left(\mathbf{X}_i' \mathbf{X}_i\right)^{-1}$ for each model using (3.4).

5. Increment j and if $j < n$ return to Step 3. Else, if

$$\max_j \prod_{i=1}^{r} (1 + \Delta_i(\mathbf{x}_j, \mathbf{x}_j^*)) < 1 + \epsilon,$$

STOP. Else return to Step 2.

As in the univariate algorithm, to find a global optimum for larger problems it is necessary to perform many runs of the algorithm using different initial designs. We use $\epsilon = 0.01$ for the convergence parameter.

3.4 Examples

In this section we present several examples illustrating the proposed *model-robust modified Fedorov* (MRMF) exchange algorithm, and compare it with two other exact model-robust design methods in the literature. Before giving the examples, we briefly describe these methods and discuss how the designs will be evaluated.

DuMouchel and Jones [37] use a Bayesian approach to provide protection against higher-order terms. They set s_1 terms as primary and s_2 terms as potential and after scaling the two groups to make them nearly orthogonal, they assume an informative prior for the potential terms and calculate a posterior distribution for the parameters with variance $\mathbf{A} = [\mathbf{X}'\mathbf{X} + \mathbf{K}/\kappa^2]^{-1}$, where $\mathbf{X} = (\mathbf{X}_{pri}|\mathbf{X}_{pot})$ and $\mathbf{K}$ is a $(s_1 + s_2) \times (s_1 + s_2)$ diagonal matrix with 0 on the first s_1 diagonals and 1 on the last s_2. The prior variance parameter, κ, is to be chosen by the user. Then, they simply choose the design that minimizes $|\mathbf{A}|$ using straightforward alterations to existing exchange algorithms.

A distinct advantage of this method is that it can provide protection against models with more parameters than observations. On the other hand, it is not designed to produce model-robust designs with respect to more than two models. Since it is a prominent model-robust technique for exact designs, we compare its results to ours. Difficulties associated with this method are the choice of the prior precision value, $\frac{1}{\kappa}$, and how to designate the primary and potential terms. We use $\frac{1}{\kappa} = 1$, as recommended by DuMouchel and Jones, but also include designs based upon $\frac{1}{\kappa} = 16$. Because of the structure of $\mathbf{A}$, larger prior precision values will result in less consideration of the potential terms as manifested by lower efficiencies for models involving those terms. We also generally assume more primary terms as opposed to less. The results are based upon the implementation of this method in the $SAS^{®}$ software's PROC OPTEX [92].

Heredia-Langner et al. [54] use a genetic algorithm to calculate exact model-robust designs. They consider r possible models and optimize a desirability func-

tion which incorporates the determinants of the information matrices of each of the models. Their procedure does not require a candidate list, though implementation of a tuned genetic algorithm is not trivial. Examples in §3.4.1 and §3.4.3 are taken from their paper, which allows comparisons to be made.

We compare designs on the basis of $\mathcal{D}$-efficiencies with respect to each model $f \in \mathcal{F}$. The $\mathcal{D}$-efficiency for model f is $D_f = \left(\frac{|\mathbf{M}_f|}{|\mathbf{M}_f^*|}\right)^{1/p}$ where $\mathbf{M}_f^*$ is the information matrix for the design optimal for f alone, and p is the number of parameters for model f. Since determinants can roughly be viewed as measures of volume, this quantity takes the ratio of the volumes and scales the comparison to a per-parameter basis.

For the individual model optimal designs in all examples save the last, Fedorov's algorithm via PROC OPTEX was run 50 times from randomly chosen initial designs and the best final design was chosen. For the final example, the MRMF algorithm was used to find the best designs for the models individually. Furthermore, all model-robust designs produced by the methods in this chapter, as well as those based upon the Bayesian procedure [37], were also generated based on 50 separate algorithm instances.

All designs referred to in the following examples are given in Appendix A. Matlab® (Version 7.8) code can be found at `http://www.stat.psu.edu/~jlr/pub/Smucker/`.

3.4.1 Constrained Response Surface Experiment

A constrained two-factor example, taken from Heredia-Langner et al. [54], will serve as an initial example illustrating our method. The design region, shown in Figure 3.1, is $\mathcal{X} = \{\mathbf{x} = (x_1, x_2) : -1 \leq x_1, x_2 \leq 1, x_1 + x_2 \leq 1, -0.5 \leq x_1 + x_2\}$ with $n = 6$. The experimenter would like a design robust for a first-order, a first-order with interaction, and full quadratic polynomial; i.e. $\mathcal{F} = \{f_i'(\mathbf{x})\boldsymbol{\beta}_i, 1 \leq i \leq 3, \mathbf{x} \in \mathcal{X}\}$ where

$$f_1'(\mathbf{x}) = (1, x_1, x_2) \tag{3.8}$$

$$f_2'(\mathbf{x}) = (1, x_1, x_2, x_1x_2) \tag{3.9}$$

$$f_3'(\mathbf{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2) \tag{3.10}$$

The candidate list for this example consisted of 266 points constituting a grid of resolution 0.1 placed over the design space. For the Bayesian method, we adopt $\frac{1}{\kappa} = 1$ and assign $f'_{pri} = (1, x_1, x_2, x_1x_2)$ and $f'_{pot} = (x_1^2, x_2^2)$ in accordance with recommendations in DuMouchel and Jones. We also include in our comparison the model-robust design of Heredia-Langner et al. [54] as well as the optimal design for the largest model.

The model-robust designs are given in Tables A.1-A.3 in Appendix A and are shown graphically in Figure 3.1. Three design points are common to all four designs, $\{(0,1),(1,0),(1,-1)\}$, and the MRMF and Bayes methods produced the same design. Table 3.1 also compares the designs in terms of the determinant and $\mathcal{D}$-efficiency for each of the considered models, and the last column gives the product of the determinants and efficiencies. The last row gives the determinant of the information matrix for the $\mathcal{D}$-optimal design for each of the models individually (see Tables A.3-A.5 in Appendix A), and the efficiencies are calculated using these values.

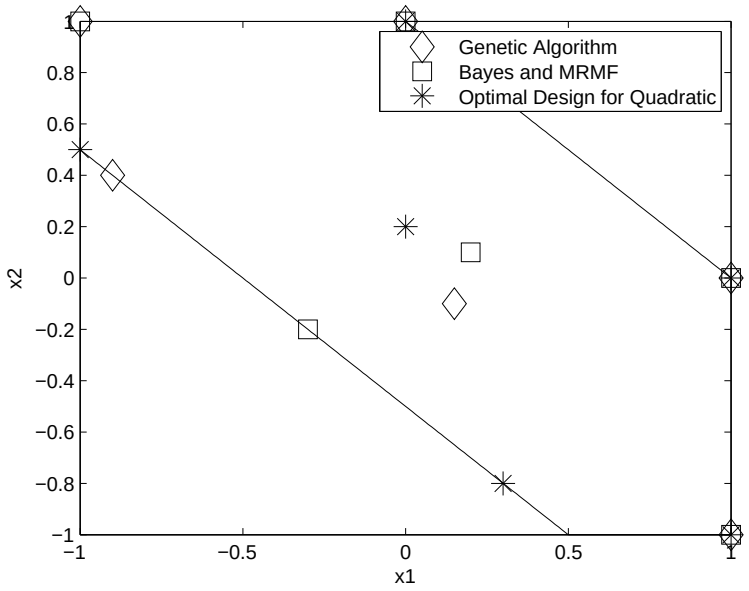

Figure 3.1. Model-robust designs for example in §3.4.1

Even though the Bayesian and MRMF designs seem close to the optimal design for the quadratic model (since their $\mathcal{D}$-efficiency for the quadratic model is nearly 1), the individually optimal design produces a poor efficiency with respect

to the interaction model. It is also somewhat surprising that the Bayesian method produced the same design as the MRMF method, given that three models were to be guarded against. However, in this simple example the MRMF design for the three models is the same as that obtained when considering only models (3.9) and (3.10) and ignoring (3.8). Therefore, it appears that the first-order model has no effect upon the MRMF algorithm, so that there are essentially two models under consideration, a situation for which the Bayesian procedure is natural.

		Model			
Design	Measure	(3.8)	(3.9)	(3.10)	Product
MRMF	$\lvert\mathbf{M}_f\rvert$	27.04	33	3.01	2685.88
	D_f	.810	.907	.995	.731
Genetic Algorithm	$\lvert\mathbf{M}_f\rvert$	31.14	26.91	2.21	1851.93
	D_f	.849	.862	.945	.692
Bayes ($\frac{1}{\kappa} = 1$)	$\lvert\mathbf{M}_f\rvert$	27.04	33	3.01	2685.88
	D_f	.810	.907	.995	.731
Optimal Design for (4.10)	$\lvert\mathbf{M}_f\rvert$	31.63	14.35	3.11	1411.60
	D_f	.853	.737	1	.629
Optimal (for each model)	$\lvert\mathbf{M}_f\rvert$	50.88	48.77	3.11	

Table 3.1. Determinants, with $\mathcal{D}$-efficiencies, for example in §3.4.1 with $n = 6$, protecting against three models.

3.4.2 Hypothetical Constrained 3-factor Experiment

To further explore our method and how it compares to the Bayesian method in particular, consider a three-factor example with design region $\chi = \{\mathbf{x} = (x_1, x_2, x_3) : -1 \leq x_1, x_2, x_3 \leq 1, -1 \leq x_1 + x_2 + x_3 \leq 1, -1 \leq x_1 + x_2 \leq 1, -1 \leq x_1 + x_3 \leq 1, -1 \leq x_2 + x_3 \leq 1\}$ and five models of interest:

$$f_1^{'}(\mathbf{x}) = (1, x_1, x_2, x_3) \tag{3.11}$$

$$f_2^{'}(\mathbf{x}) = (f_1^{'}, x_1x_2, x_1x_3, x_2x_3) \tag{3.12}$$

$$f_3^{'}(\mathbf{x}) = (f_2^{'}, x_1^2, x_2^2, x_3^2) \tag{3.13}$$

$$f_4^{'}(\mathbf{x}) = (f_3^{'}, x_1^2x_2, x_1^2x_3, x_1x_2^2, x_2^2x_3, x_1x_3^2, x_2x_3^2, x_1x_2x_3) \tag{3.14}$$

$$f_5^{'}(\mathbf{x}) = (f_4^{'}, x_1^3, x_2^3, x_3^3) \tag{3.15}$$

so that $\mathcal{F} = \{f_i^{'}(\mathbf{x})\beta_i, 1 \leq i \leq 5, \mathbf{x} \in \chi\}$.

In particular, assume that the experimenter would like to use $n = 20$ runs and would like a design that can fit each of these models well. To specify the Bayesian procedure, we take as primary all terms in (3.13) and designate the rest as potential. We give the MRMF design in Table 3.2, as well as Bayesian designs with $\frac{1}{\kappa} = 1$ and $\frac{1}{\kappa} = 16$ and the optimal design for the largest model, all using a candidate list consisting of a grid of points with resolution 0.1 placed over the design space.

The Bayesian designs are competitive for most of the models, but the designs lack efficiency for model (3.14) when compared to the MRMF design, which might be expected since it is in between the primary and full model and as such not explicitly considered. None of the designs perform very well for model (3.12), though the MRMF design is marginally better. As we expect, when a larger prior precision value is used in the Bayesian procedure, the efficiency of models containing primary terms is reduced, and in this case significantly degrades the design in terms of the product criterion. The optimal design for the largest model is competitive with the Bayesian designs in terms of model-robustness, though the MRMF design would likely be preferred because of its higher efficiencies in models (3.12), (3.13), and (3.14).

		Model					
Design	Measure	(3.11)	(3.12)	(3.13)	(3.14)	(3.15)	Product
MRMF	$\lvert\mathbf{M}_f\rvert$	6.58e3	5.57e4	1.10e5	3.21e0	5.24e-3	6.78e11
	D_f	.864	.756	.870	.955	.979	.531
Bayes ($\frac{1}{\kappa} = 1$)	$\lvert\mathbf{M}_f\rvert$	6.63e3	5.21e4	9.74e4	9.92e-1	7.94e-3	2.65e11
	D_f	.867	.749	.860	.892	.999	.498
Bayes ($\frac{1}{\kappa} = 16$)	$\lvert\mathbf{M}_f\rvert$	5.93e3	4.39e4	1.12e5	4.61e-1	4.41e-3	5.93e10
	D_f	.843	.731	.872	.852	.970	.444
Optimal for (3.15)	$\lvert\mathbf{M}_f\rvert$	6.44e3	4.94e4	9.62e4	7.63e-1	8.07e-3	1.88e11
	D_f	.860	.744	.859	.878	1	.4826
Optimal (for each model)	$\lvert\mathbf{M}_f\rvert$	1.18e4	3.93e5	4.42e5	6.97e0	8.07e-3	

Table 3.2. Determinants, with $\mathcal{D}$-efficiencies, for example in §3.4.2 with $n = 20$, protecting against five models.

3.4.3 Constrained Mixture Experiment

We now revisit the example briefly described at the outset. This is a four-factor constrained mixture experiment regarding the formulation of a household product in which 20 runs are available. The design region can be defined by:

$$\mathcal{X} = \left\{ \mathbf{x} = (x_1, x_2, x_3, x_4) : \sum_{i=1}^{4} x_i = 1, .5 \leq x_1 \leq 1, 0 \leq x_2, x_3 \leq .5, 0 \leq x_4 \leq .05 \right\}$$

where x_1 is a nonionic surfactant, x_2 is an anionic surfactant, x_3 is a second nonionic surfactant, and x_4 is a zwitterionic surfactant. Because of the dependency induced by the mixture constraint, standard mixture design models are considered which do not include an intercept:

$$f_1^{'}(\mathbf{x}) = (x_1, x_2, x_3, x_4) \tag{3.16}$$

$$f_2^{'}(\mathbf{x}) = (f_1^{'}, \{x_i x_j, i < j \leq 4\}) \tag{3.17}$$

$$f_3^{'}(\mathbf{x}) = (f_2^{'}, \{x_i x_j x_k, i < j < k \leq 4\}) \tag{3.18}$$

$$f_4^{'}(\mathbf{x}) = (f_3^{'}, \{x_i x_j (x_i - x_j), i < j \leq 4\}) \tag{3.19}$$

so that $\mathcal{F} = \{f_i^{'}(\mathbf{x})\beta_i, 1 \leq i \leq 4, \mathbf{x} \in \chi\}$. Heredia-Langner et al. [54] also used this example, and so we compare our method to their Genetic Algorithm as well as to the Bayesian method of DuMouchel and Jones [37]. For the latter, we use both a standard value for the prior precision, $\frac{1}{\kappa} = 1$, and a larger value, $\frac{1}{\kappa} = 16$, with all terms primary except those unique to f_4, which are regarded as potential.

Since this is a large mixture design, we supplemented a regular grid (resolution 0.01) with extreme vertices and approximate centroids of the design region using code as described by Piepel [85].

		Model				
Design	Measure	(3.16)	(3.17)	(3.18)	(3.19)	Product
MRMF	$\lvert\mathbf{M}_f\rvert$	5.31e-2	7.22e-22	2.65e-43	8.36e-78	8.49e-143
	D_f	.728	.897	.931	.996	.606
Genetic Algorithm	$\lvert\mathbf{M}_f\rvert$	5.23e-2	7.46e-22	2.90e-43	7.80e-78	8.83e-143
	D_f	.725	.900	.937	.992	.607
Bayes ($\frac{1}{\kappa} = 1$)	$\lvert\mathbf{M}_f\rvert$	5.46e-2	6.74e-22	2.24e-43	9.08e-78	7.48e-143
	D_f	.733	.890	.919	1	.600
Bayes ($\frac{1}{\kappa} = 16$)	$\lvert\mathbf{M}_f\rvert$	5.64e-2	6.12e-22	3.01e-43	3.08e-78	3.20e-143
	D_f	.739	.882	.939	.947	.580
Optimal Design for (3.19)	$\lvert\mathbf{M}_f\rvert$	5.46e-2	6.74e-22	2.24e-43	9.08e-78	7.48e-143
	D_f	.733	.890	.919	1	.600
Optimal (for each model)	$\lvert\mathbf{M}_f\rvert$	1.89e-1	2.15e-21	7.26e-43	9.08e-78	

Table 3.3. Determinants, with $\mathcal{D}$-efficiencies, for example in §3.4.3 with $n = 20$, protecting against four models.

In Table 3.3, the MRMF method can be seen to be competitive with the Genetic Algorithm, though their design is slightly superior by our product optimality criterion. This is likely a function of the discretization in our candidate list. Note that the optimal design for model (3.19) alone has a significantly higher objective function value (9.08e-78) than that given in Heredia-Langner et al. [54] (7.83e-78), though theirs was asserted to have been obtained from PROC OPTEX in SAS as well.

It is the case again in this example that the best design found by the MRMF method is relatively close to that of the optimal design for the largest model. The Bayesian design with precision of 1 actually chooses the optimal design for the largest model, and shows that this design is competitive with those that look to maximize the product of the determinants. When the precision is increased, we see the same behavior as was noted before: The Bayesian design becomes less efficient for the model that involves potential terms. The resulting Bayesian design gives slightly more balance, but suffers against the product optimality criterion.

3.4.4 Mixture Experiment with Disparate Models

For our final example we use an unconstrained mixture experiment by Frisbee and McGinity [42] with $n = 11$. The response is the glass transition temperature of a certain film with three nonionic surfactant factors. The goal was to minimize this transition temperature, and Frisbee and McGinity fit a traditional polynomial model. However, another class of models, the so-called Becker models [29, Sec. 6.5], were shown by Rajagopal and Castillo [89] to also fit the data well and lead to a significantly different optimal solution. These models, originally considered to address certain shortcomings in the Sheffé polynomial models, use $\min(\cdot)$ instead of $\text{prod}(\cdot)$ to model second order mixture blending.

In this case,

$$\mathcal{X} = \left\{ \mathbf{x} = (x_1, x_2, x_3) : \sum_{i=1}^{3} x_i = 1, 0 \leq x_i \leq 1, i = 1, 2, 3 \right\}$$

and we take five possible models:

$$f_1^{'}(\mathbf{x}) = (\{x_i, i = 1, 2, 3\}) \tag{3.20}$$
$$f_2^{'}(\mathbf{x}) = (f_1^{'}, \{x_i x_j, i < j \leq 3\}) \tag{3.21}$$
$$f_3^{'}(\mathbf{x}) = (f_2^{'}, \{x_1 x_2 x_3\}) \tag{3.22}$$
$$f_4^{'}(\mathbf{x}) = (f_1^{'}, \{min(x_i, x_j), i < j \leq 3\}) \tag{3.23}$$
$$f_5^{'}(\mathbf{x}) = (f_4^{'}, \{min(x_1, x_2, x_3)\}) \tag{3.24}$$

so that $\mathcal{F} = \{f_i^{'}(\mathbf{x})\beta_i, 1 \leq i \leq 5, \mathbf{x} \in \mathcal{X}\}$.

In addition to the five models we are guarding against, we also examine effectiveness of our design with respect to the model fit by Frisbee and McGinity, as well as the most probable model found *a posteriori* by Rajagopal and Castillo, respectively:

$$f_{fm}^{'}(\mathbf{x}) = (x_1, x_2, x_3, x_1 x_3, x_2 x_3) \tag{3.25}$$
$$f_{rc}^{'}(\mathbf{x}) = (x_1, x_2, x_3, min(x_1, x_3), min(x_2, x_3)) \tag{3.26}$$

For a candidate list, we used a regular grid with resolution 1/12, which because of the regular design region, contained the vertices and centroids of the region.

With the disparate model types, the Bayes procedure, with its primary and potential factors, cannot be easily applied. Instead, we examine the results of the MRMF design and compare it in Table 3.4 to the design that was actually used. In terms of efficiency, the design used by Frisbee and McGinity is much inferior for all models considered because it includes, in addition to two centroid points, three other points on the interior of the simplex design region.

As seen in Table 3.4, the MRMF design is optimal for models (3.22), (3.23), and (3.24). This is because the optimal designs for these models individually are interchangeable; i.e. the optimal design for one is also optimal for another. Note that since the models are not nested we do not have the multiresponse $\mathcal{D}$-optimality interpretation.

		Model						
Design	Measure	(3.20)	(3.21)	(3.22)	(3.23)	(3.24)	(4.20)	(4.21)
MRMF	$\lvert\mathbf{M}_f\rvert$	19.81	5.91e-3	5.36e-6	0.569	2.78e-2	6.61e-2	1.46
	D_f	.745	.954	1	1	1	.812	.866
Frisbee and McGinity	$\lvert\mathbf{M}_f\rvert$	8.25	1.22e-3	1.51e-6	.146	8.82e-3	2.30e-2	.588
	D_f	.556	.733	.835	.797	.849	.658	.722
Optimal (for each model)	$\lvert\mathbf{M}_f\rvert$	48	7.8e-3	5.36e-6	.569	2.78e-2	.188	3

Table 3.4. Determinants, with $\mathcal{D}$-efficiencies, for example in §3.4.4, with $n = 11$, protecting against five models.

3.5 Discussion

The Model-Robust Modified Fedorov (MRMF) exchange algorithm presented in this chapter provides a natural tool with which to find designs when an optimal design is desired but the model-form is unknown. The mechanism to achieve this is intuitive and simple: The experimenter chooses r models for which he/she would like to design. Then, a design is found which maximizes the product of the determinant of the information matrices of each of the models. In the case that the models under consideration are nested, this is the $\mathcal{D}$-optimal design for the associated multiresponse model with r responses and thus, under normality, minimizes the volume of the confidence ellipsoid of the parameters.

Furthermore, the MRMF method produces designs that are competitive, with simpler algorithmic machinery, than the Genetic Algorithm (GA) approach of Heredia-Langner et al. [54]. The strength of the MRMF method with respect to the GA technique is that it is automatic and a straightforward extension of commonly used exchange algorithms. The GA requires tuning of several parameters and is nontrivial to implement effectively.

We also compared our procedure to the Bayesian method of DuMouchel and Jones [37], a widely available model-robust technique. We initially hypothesized that the Bayesian method would suffer when confronted with multiple possible models, since it categorizes terms into just two groups. This is supported by the second example, though the procedure performed well in the first and third examples. The choice of $\frac{1}{\kappa}$ certainly affects the model-robustness of the design; indeed for certain values of $\frac{1}{\kappa}$ (i.e. $\frac{1}{\kappa} = 1$ in the third example) the method seems to produce a design optimal for the highest-order model, while for large enough values of $\frac{1}{\kappa}$ the full model is not even estimable. The choice of terms as primary or potential also makes an impact. Our procedure does not suffer from these uncertainties, has a multiresponse $\mathcal{D}$-optimal interpretation (for $\mathcal{F}$ nested) and explicitly considers a larger class of models; it can also handle situations as in Example 4 in which the possible models are disparate and impossible to nest.

One strategy, if faced with a situation necessitating a $\mathcal{D}$-optimal design, might be to design for the highest-order model possible. If, as assumed in this dissertation, there are a sufficient number of runs to estimate the largest model, one might

question whether the efficiency gained using model-robust methods is worth the additional methodology. In certain cases, as in the third example, the gains appear to be limited. But as demonstrated by the first and second examples, significant gains can be made by utilizing the model-robust approach. Therefore, a dedicated procedure based upon accepted univariate exchange algorithms will be useful to produce model-robust designs.

In terms of $\mathcal{D}$-efficiency, the MRMF designs seem to favor larger models. In other words, the efficiency of the smaller models suffer as compared to the larger ones. To mitigate this, one might consider the following optimization criterion [10, 38], instead of (3.6):

$$\boldsymbol{\phi}(\mathbf{M}_{\mathcal{F}}(\tilde{\xi}_n)) = \prod_{i=1}^{r} \left|\mathbf{M}_i(\tilde{\xi}_n)\right|^{1/q_i} \tag{3.27}$$

where q_i is the number of parameters in the i^{th} model. It is straightforward to derive an exchange algorithm using this criterion–call it the scaled MRMF–which has the effect of shrinking values of dissimilar orders of magnitude toward each other, in essence weighting more heavily those models with fewer parameters. We implemented this procedure using several examples, and the results were surprisingly similar to the unscaled MRMF. For instance, for the constrained mixture experiment in the third example, the scaled MRMF design resulted in a design very close to the MRMF in Table 3.3. For the second example in §3.4.2, the scaled MRMF design produced more of a difference, with $\mathcal{D}$-efficiencies increasing from 86.4% to about 89% for model (3.11) and from 75.6% to about 78% for model (3.12), while decreasing the efficiencies of model (3.13) from 87% to about 85.5% and model (3.15) from 97.9% to about 96.5%. In any case, the scaled MRMF does not result in dramatic changes in efficiency.

The model-robust criterion used in this chapter could easily be extended to include prior information in terms of model weights, if certain models are preferred over the others. However, since this work was motivated in part by multiresponse optimal design theory, the minimal volume of the parameter confidence ellipsoid interpretation of $\mathcal{D}$-optimality is used and thus we only consider equally weighted models. Furthermore, the relative ineffectiveness of the scaled MRMF to provide designs with balanced $\mathcal{D}$-efficiencies underscores the difficulty in balancing the

designs using weights.

Finally, assume that T_e is the time it takes to run the univariate exchange algorithm. The runtime for these model-robust algorithms should be rT_e where r is the number of models considered. Commercial software programs have fast implementations of exchange algorithms, so the computational burden imposed by a similarly implemented model-robust exchange algorithm should not be heavy.

Chapter 4

A Maximin Model-Robust Exchange Algorithm and its Generalization to Include Model Preferences

In the previous chapter, we considered a model-robust procedure for an experimental scenario in which we allow the specification of multiple models and a maximization of the product of the determinant of the information matrices with respect to these models. In this chapter, we stay within the same model-set framework, but maximize the *minimum* $\mathcal{D}$-efficiency.

4.1 Introduction

Again, assume that an experiment is to be designed for which there is a single response and k factors of interest. As before, the linear regression model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}$ will be fit, where $\mathbf{y}$ is an n-vector of independent observations, $\mathbf{X}$ is the $n \times p$ expanded design matrix, $\boldsymbol{\beta}$ is the $p \times 1$ vector of unknown parameters, and $\boldsymbol{\epsilon}$ is also an n-vector with $E(\boldsymbol{\epsilon}) = 0$ and $Var(\boldsymbol{\epsilon}) = \sigma^2$. Assume also that the classical least squares criterion will be used to estimate $\boldsymbol{\beta}$, in which case the estimator is $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ with $Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. Recall that the inverse of $Var(\hat{\boldsymbol{\beta}})$ is called the information matrix, denoted as $\mathbf{M} = (\mathbf{X}'\mathbf{X})/\sigma^2 = [Var(\hat{\boldsymbol{\beta}})]^{-1}$.

Furthermore, we again let $\mathcal{X}$ be the design space, Ξ be the set of all possible

designs and $\xi_n \in \Xi$ be a discrete, n-point design which can be represented as

$$\xi_n = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_d \\ n_1 & n_2 & \dots & n_d \end{pmatrix} \tag{4.1}$$

where $n = \sum_{i=1}^{d} n_i$ is the total number of experiments, and n_i is the number of experiments performed at design point $\mathbf{x}_i$, $i = 1, \dots, d$.

An optimal design strategy chooses a design by optimizing a function of the information matrix. The $\mathcal{D}$-criterion maximizes $\phi(\mathbf{M}) = |\mathbf{M}|$, which is equivalent to minimizing the volume of the confidence ellipsoid about the parameter estimates if normality is assumed.

Implicit in the structure of $\mathbf{X}$ is the design, ξ_n, as well as a model form, $f(\mathbf{x})$, which is a $p \times 1$ expanded design vector that is usually assumed given though in reality is rarely known. Consequently, we represent the information matrix as $\mathbf{M}_f(\xi_n)$, though we may express it as $\mathbf{M}_f$ in the interest of simplicity. Thus, $\phi(\mathbf{M}_f(\xi_n))$ depends upon the model form f and an optimal design for one model is not in general optimal for another. To mitigate against an undue reliance upon a single model form f, we again allow a user-chosen set of models, $\mathcal{F}$.

In our experience the model-robust procedures which use a form of $\mathcal{D}$-optimality (such as our procedure in Chapter 3, the Genetic Algorithm approach of Heredia-Langner et al. [54], and the Bayesian approach of DuMouchel and Jones [37]) often produce designs which have high $\mathcal{D}$-efficiencies for some models (generally those with the most parameters), but considerably lower $\mathcal{D}$-efficiencies for others. Furthermore, these algorithms give experimenters little *a priori* control over the efficiencies of the design with respect to the given possible models and indeed exhibit a substantial lack of efficiency balance across them. *Efficiency balance*, which we formally define in §4.3, occurs when a design has the same efficiency with respect to each model in a specified set.

We propose a maximin model-robust exchange algorithm which gives exact designs with substantially more balance than those from the aforementioned procedures, while optimizing the worst-case efficiency with respect to a user-specified set of models. We develop a generalization of this algorithm which allows the specification of model interest levels (weights), providing additional control over the

$\mathcal{D}$-efficiencies with respect to each model. Though the procedure is explicitly for finite run sizes, we explore some asymptotic properties of this generalized maximin criterion, showing that even if there is not perfect balance among the efficiencies of all models, there will be among a subset of at least two models. We also specify a condition which guarantees complete asymptotic efficiency balance among all models in $\mathcal{F}$.

Several authors have studied a similar criterion. For instance, Imhof and Wong [57] give a graphical method to find asymptotic maximin designs when two different optimality criteria are used. Dette and Franke [33] explicitly characterize continuous maximin designs in the specific case of polynomial regression on $[-1, 1]$, where they maximize the minimum efficiency with respect to possible polynomial models as well as a goodness-of-fit criterion. Berger and Tan [11] use the maximin criterion for mixed effects models. In contrast, the motivation for our work in this chapter is to provide a useful, flexible tool for experimenters to construct exact model-robust designs, for completely randomized experiments, for general design spaces and small run sizes, allowing the user to specify a class of possible models and exert as much control as possible over the design performance with respect to each. The continuous design results in this chapter are an almost incidental consequence of our observations of the discrete procedure's performance, and we will show empirically that even small run sizes exhibit similar behavior.

The chapter is arranged as follows. In the next section, we describe classic exchange algorithms for $\mathcal{D}$-optimality, and then present our generalized maximin procedure. Following that, we give some results concerning the balance properties of this criterion, though here we resort to continuous design theory. We then present several examples illustrating the exact procedure, and conclude with a discussion.

4.2 A Generalized Maximin Model-Robust Exchange Algorithm

The maximin procedure we describe is built upon a classic exchange algorithm for $\mathcal{D}$-optimality. This and other such algorithms were reviewed in Chapter 3.3.1. We

follow our previous work and use the modified Fedorov algorithm [26] as the basis for our model-robust procedure. The determinant and inverse of the information matrix for a particular model f can be updated via (3.2) and (3.4).

We assume that a set of r models, $\mathcal{F}$, will be chosen and explicitly formulated. For instance, for single-factor polynomial regression, the experimenter might desire protection for possible polynomial models of degree 1 to 4.

Furthermore, a model interest (weight) vector, $v = (v_1, \ldots, v_r) \in (0, 1]$, must be supplied which quantifies the interest the experimenter has in each model. By default, the model interest is unity for all models in which case the algorithm maximizes the minimum efficiency. If certain models are less important, however, they can be assigned lower interest levels which will often decrease their efficiencies compared to those with larger values. We require that $\max_{f \in \mathcal{F}} v_f = 1$. Note that these model interest levels can also be framed as prior probabilities. For instance, if there are two possible models, f_1 and f_2, and the model interest vector is $v = (1, 0.8)$, this is equivalent to specifying a prior probability of $1/(1+0.8) = 0.56$ for the first model and $0.8/(1 + 0.8) = 0.44$ for the second.

The algorithm also requires as input the determinant of the information matrix of the optimal design for each model individually so that efficiencies can be calculated. This is not a stringent requirement in light of the capabilities of readily available software; for instance, the $SAS^{®}$ software's PROC OPTEX [92]. Finally, the algorithm requires a user-constructed candidate list, C, of design points.

The algorithm works on efficiencies, and the most prevalent in this context is $\mathcal{D}$-efficiency. Assuming ξ_n^* is the optimal design for model f, the $\mathcal{D}$-efficiency for a design ξ_n with respect to model f is defined as:

$$D_f(\xi_n) = \left(\frac{|\mathbf{M}_f(\xi_n)|}{|\mathbf{M}_f(\xi_n^*)|} \right)^{1/p} \tag{4.2}$$

where $\mathbf{M}_f(\xi_n)$ is the information matrix with respect to model f and design ξ_n, $\mathbf{M}_f(\xi_n^*)$ is the information matrix for the design optimal for f alone, and p is the number of parameters for model f. The $\mathcal{D}$-efficiency roughly scales the efficiency of a design to a per-parameter basis and the reciprocal is a factor indicating the additional sample size required to estimate the parameters to the same level of precision as the optimal design (note: $\mathcal{D}_f(\xi_n) \in [0, 1]$). In the generalized version

of our maximin algorithm, we define *generalized $\mathcal{D}$-efficiency*, for model f, as

$$G_f(\xi_n) = \frac{D_f(\xi_n)}{v_f} \tag{4.3}$$

where v_f is the interest weight for model f. Note that unlike $\mathcal{D}$-efficiency, $G_f(\xi_n) \in [0, \infty)$.

To illustrate the utility of generalized $\mathcal{D}$-efficiency, consider an experimental situation in which there are two models of interest, f_1 and f_2, with a model interest vector $v = (1, .8)$. In this case, finding a design maximizing the minimum $\mathcal{D}$-efficiency will not reflect this prior belief. Instead, we find the maximin design using $G_f(\xi_n)$. Thus, if the current design gives $D_{f_1}(\xi_n) = 0.85$ and $D_{f_2}(\xi_n) = 0.75$, we find that $G_{f_1}(\xi_n) = 0.85$ and $G_{f_2}(\xi_n) = 0.75/v_2 = 0.75/.8 = .9375$ so that f_1 has the lower *generalized* $\mathcal{D}$-efficiency even though it has the larger $\mathcal{D}$-efficiency. By working on $G_f(\xi_n)$ instead of $D_f(\xi_n)$, the model interest levels are reflected in the $\mathcal{D}$-efficiencies of the generalized maximin design.

In the algorithm, we extend the notation of (4.3) to $G_{f,jk}(\xi_n)$ to represent the generalized $\mathcal{D}$-efficiency when the j^{th} design point is exchanged with the k^{th} candidate point. The algorithm is as follows.

1. Initialize algorithm: Input optimal determinants, $|\mathbf{M}_f^*|$, for each $f \in \mathcal{F}$ as well as model interest vector, v. Construct candidate list, $C \subset \mathcal{X}$, with c candidate points. Specify nonsingular initial design, ξ_n. For each model, calculate $G_f(\xi_n)$ via (4.3) and determine $f_{min} = \arg\min_{f \in \mathcal{F}} G_f(\xi_n)$.

2. Let $j = 1$ (index for design points).

3. Let $k = 1$ (index for candidate points).

4. For design point $\mathbf{x}_j$ and candidate point $\mathbf{x}_k$, calculate $\Delta_{f_{min}}(\mathbf{x}_j, \mathbf{x}_k, \xi_n)$ as in (3.3).

   (a) If $\Delta_{f_{min}}(\mathbf{x}_j, \mathbf{x}_k, \xi_n) < 0$, do not consider $\mathbf{x}_k$ for exchange because the minimum efficiency would not be increased.

      i. If $k = c$, go to step 5.

      ii. If $k < c$ set $k = k + 1$ and return to step 4.

(b) If $\Delta_{f_{min}}(\mathbf{x}_j, \mathbf{x}_k, \xi_n) \geq 0$, calculate $\Delta_f(\mathbf{x}_j, \mathbf{x}_k, \xi_n)$ for all $f \neq f_{min}$ and evaluate the determinant of the information matrices for each $f \in \mathcal{F}$ according to (3.2). Calculate the generalized efficiencies via (4.2) and (4.3) and determine $G_{f_{min},jk}(\xi_n) = \min_{f \in \mathcal{F}} G_{f,jk}(\xi_n)$.

i. If $k = c$, go to step 5.

ii. If $k < c$ set $k = k + 1$ and return to step 4.

5. Exchange $\mathbf{x}_j$ with $\mathbf{x}_{k_j^*}$, where $k_j^* = \arg\max_k G_{f_{min},jk}(\xi_n)$.

6. Update $\mathbf{M}_f^{-1}$ for all $f \in \mathcal{F}$ using (3.4). Calculate $|\mathbf{M}_f|$ for all $f \in \mathcal{F}$.

7. Set $j = j+1$ and if $j < n$ return to step 3. If $j = n$ and termination criterion is met, STOP. If $j = n$ and termination criterion not met, return to Step 2.

The termination criterion stops the algorithm if $\max_{j,f \in \mathcal{F}} \Delta_f(\mathbf{x}_j, \mathbf{x}_{k_j}^*, \xi_n) < \epsilon$, where ϵ is a small, positive constant (typically $\epsilon = 0.01$).

If the experimenter is interested in each model equally, this algorithm specializes to a maximin procedure which finds a design maximizing the smallest $\mathcal{D}$-efficiency. The model interest vector, however, affords the user a great deal of flexibility because designs focusing primarily on those models of most interest can be generated, while protection against and estimability for less likely models can be maintained.

Though we explore some asymptotic properties of this model-robust criterion in the following section, we have not found these properties severely compromised in small-sample situations. In what follows, the "maximin" design or exchange algorithm refers to the specific case in which each element of the model interest vector is 1. "Generalized maximin" refers to possibly unequal model interest elements. Similarly, "efficiency" refers to D_f, while "generalized efficiency" refers to the efficiencies adjusted by the model interest vector as in (4.3).

4.3 Asymptotic Properties of Generalized Maximin Criterion

Although we believe that exact designs are most useful and relevant to experimenters, we can obtain insight into these maximin designs by studying their con-

tinuous counterparts. In what follows, we give a condition which ensures that the generalized maximin criterion will produce designs balanced in terms of generalized efficiencies. Technical details, and proofs of the results in this section, are given in Appendix B.

For the purposes of this section we define an asymptotic, or continuous, design as a discrete probability measure ξ over $\mathcal{X}$, which implies that there exists a countable number of design points upon which there is positive measure. A design defined in this way can be thought of as the proportion of the available experimental budget assigned to any particular design point $\mathbf{x}$ in $\mathcal{X}$ and represented as

$$\xi(\mathbf{x}) = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_d \\ \lambda_1 & \lambda_2 & \dots & \lambda_d \end{pmatrix} \tag{4.4}$$

where λ_i, $i = 1, \dots, d$ is the measure on the designs points with positive measure. The information matrix is $p \times p$:

$$\mathbf{M}(\xi) = \sigma^{-2} \sum_{i=1}^{d} \lambda_i f(\mathbf{x}_i) f^{'}(\mathbf{x}_i) \tag{4.5}$$

Before proceeding further, we formally define *generalized efficiency balance* and *limiting models*.

Definition 1. *Given a design, ξ, along with a set of models, $\mathcal{F}$,* generalized efficiency balance *is achieved for a subset of models $\mathcal{F}' \subseteq \mathcal{F}$ if*

$$G_{f_{max}}(\xi) - G_{f_{min}}(\xi) = 0 \tag{4.6}$$

where $f_{min} = \arg\min_{f \in \mathcal{F}'} G_f(\xi)$, $f_{max} = \arg\max_{f \in \mathcal{F}'} G_f(\xi)$, and $f_{min} \neq f_{max}$. Alternatively, we refer to the elements of $\mathcal{F}'$ as limiting models.

Notice that a design could have generalized efficiency balance with respect to $\mathcal{F}$, the complete set of models, in which case we say that the design has *complete generalized efficiency balance* (CGEB).

The following result guarantees that the maximin design will have generalized efficiency balance for some subset of $\mathcal{F}$. In other words, a set of at least two limiting models is sure to exist.

Theorem 1. *Assume $\mathcal{F}$ has $r \geq 2$ elements and let $\xi^* = \arg\max_{\xi\in\Xi} \min_{f\in\mathcal{F}} G_f(\xi)$ and $f_{min} = \arg\min_{f\in\mathcal{F}} G_f(\xi^*)$. Also, assume that ξ^* is not optimal for any $f \in \mathcal{F}$ individually. Then, $G_f(\xi^*) - G_{f_{min}}(\xi^*) = 0$ for at least one $f \in \mathcal{F}\backslash f_{min}$.*

We next give a condition which, if satisfied, ensures that CGEB is attained for the continuous maximin design. Assume $\mathcal{F}$ is a set of models with an arbitrary proper subset $\mathcal{F}' \subset \mathcal{F}$, and let $\xi^* = \arg\max_{\xi\in\Xi} \min_{f\in\mathcal{F}} G_f(\xi)$, and $\xi' = \arg\max_{\xi\in\Xi} \min_{f\in\mathcal{F}'} G_f(\xi)$. Then, the condition is

$$\min_{f\in\mathcal{F}'} G_f(\xi') > \min_{f\in\mathcal{F}'} G_f(\xi^*) \tag{4.7}$$

The condition specifies, essentially, that we can do better than ξ^* when the set of models is a strict subset of $\mathcal{F}$. While Theorem 2 shows that (4.7) is a sufficient condition for CGEB, we know of no *a priori* way to check it for a given $\mathcal{F}$. Consequently, the value of the following balance theorem rests on the insight it provides when a particular set of models does not provide balance.

Theorem 2. *Assume $\mathcal{F}$ has r elements and satisfies* (4.7)*. Let*

$$\xi^* = \arg\max_{\xi\in\Xi} \min_{f\in\mathcal{F}} G_f(\xi)$$

and $f_{min} = \arg\min_{f\in\mathcal{F}} G_f(\xi^*)$. Then, $G_f(\xi^*) - G_{f_{min}}(\xi^*) = 0 \ \forall f \in \mathcal{F}$.

This result is more descriptive than prescriptive, because it produces understanding in the case that balance does not happen instead of showing what classes of models achieve generalized efficiency balance. The contrapositive to Theorem 2 is that if the models do not produce balance, condition (4.7) does not hold; i.e. for some subset or group of subsets of $\mathcal{F}$, the maximin design is no better than for $\mathcal{F}$ itself. This suggests a subclass of limiting models that constrain the maximin, which is guaranteed by Theorem 1.

The final result is a special case of Theorem 2 when $r = 2$, and is somewhat like a result in Imhof and Wong [57], though in a different context.

Corollary 1. *Let $\mathcal{F}$ have just 2 elements, f_1 and f_2, and*

$$\xi^* = \arg\max_{\xi\in\Xi} \min_{f\in\mathcal{F}} G_f(\xi),$$

and $\xi^*_{f_i}$ *be the optimal design for model* f_i *alone. If neither* $\xi^* \neq \xi^*_{f_1}$ *nor* $\xi^* \neq \xi^*_{f_2}$, *then* $G_f(\xi^*) - G_{f_{min}}(\xi^*) = 0$ *for* $f \neq f_{min}$.

This corollary says that when the maximin design is not the same as one of the individually optimal designs, efficiency balance (when model interest elements are equal) is achieved when only two models are considered. This holds, too, for generalized maximin designs, though there are nontrivial examples in which the maximin design is the same as the optimal design for one of the models individually.

For instance, consider the situation in which we have two nested possible models, f_1 and f_2, with $v = (0.5, 1)$. It is likely that the design optimal for f_2 will have a $\mathcal{D}$-efficiency with respect to f_1 of more than 50%, in which case $G_{f_1} > 1$. This implies that the maximin design, ξ^*, is the same as the optimal design for f_2, $\xi^*_{f_2}$. There is obviously not CGEB here, but Corollary 1 is not invalidated because $\xi^* = \xi^*_{f_2}$.

Because we cannot check the truth of condition (4.7) for a particular experimental situation, we cannot guarantee balance of the generalized efficiencies, though Theorem 1 guarantees balance of some subset of $\mathcal{F}$ in most interesting situations. In that sense, then, we achieve balance and the higher efficiencies for the other models are a surplus.

However, after running the generalized maximin procedure, limiting models often suggest themselves. Consequently, it is possible that "taking" a small amount of efficiency from one or more of these limiting models will result in a substantial amount of "giving" to the efficiencies of the others. Therefore, a sensitivity analysis can be performed in which the model interest entries for the limiting models are reduced by some user-specified amount and the algorithm is rerun. The results will likely be lower efficiencies for the original limiting models but higher efficiencies for the others. We will explore this in the subsequent section.

4.4 Examples

In this section we present several examples illustrating the generalized maximin model-robust design procedure. Unless otherwise noted, all designs for a given example were constructed using the same candidate list and 50 randomly started algorithm tries. Optimal designs for individual models were generated using $SAS^{®}$

software's PROC OPTEX [92]. All designs referred to in the following examples are given in Appendix C (or, for those that appear in both this chapter and the previous, in Appendix A). Matlab® (Version 7.8) code can be found at `http://www.stat.psu.edu/~jlr/pub/Smucker/`.

4.4.1 Constrained Two-Factor Experiment

We give first a simple six run, two-factor example used in Chapter 3 as well as Heredia-Langner et al. [54] with a constrained design region $\mathcal{X} = \{\mathbf{x} = (x_1, x_2) : -1 \leq x_1, x_2 \leq 1, -0.5 \leq x_1 + x_2 \leq 1\}$. Suppose the experimenter would like to guard against three models:

$$f_1^{'}(\mathbf{x}) = (1, x_1, x_2) \tag{4.8}$$

$$f_2^{'}(\mathbf{x}) = (1, x_1, x_2, x_1x_2) \tag{4.9}$$

$$f_3^{'}(\mathbf{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2) \tag{4.10}$$

so that $\mathcal{F} = \{f_i^{'}(\mathbf{x})\boldsymbol{\beta}_i, 1 \leq i \leq 3, \mathbf{x} \in \mathcal{X}\}$.

We use as a candidate list the points in a resolution 0.1 grid placed over $\mathcal{X}$. In Table 4.1 we give results for a $\mathcal{D}$-maximin design and a generalized $\mathcal{D}$-maximin design with model interest vector $v = (1, 1, .6)$, as well as two other model-robust designs for comparison, the one produced by the MRMF algorithm of Chapter 3 and the one via the Genetic Algorithm (GA) approach of Heredia-Langner et al. [54]. We also include the optimal design for the largest considered model, representing what might be considered the default optimal design strategy. The final row in the table gives $|\mathbf{M}|$ for the optimal design for each individual model.

Notice that the non-maximin designs favor the largest model in terms of $\mathcal{D}$-efficiencies. On the other hand, the $\mathcal{D}$-maximin design is close to balanced among the three models ($\mathcal{D}$-efficiences of 88.9%, 89.4%, and 88.8%, respectively), and provides a substantial increase in the worst-case efficiency (88.8% versus 81% for the MRMF design and 84.9% for the GA design). Complete generalized efficiency balance, in the sense of the previous section, is not achieved because of the finite sample size. The $(1, 1, .6)$-$\mathcal{D}$-maximin design provides some protection for the quadratic model while allowing very high efficiencies for models (4.8) and (4.9).

Notice there is not even the suggestion of CGEB here, since $G_{f_3} = \frac{1}{0.6} \cdot 0.721 = 1.2$ while $G_{f_1} = .951$ and $G_{f_2} = .959$. It seems likely that, asymptotically, the first two models are the limiting ones in this case.

Our algorithm can mimic other designs, which shows its flexibility. For instance, setting $v = (.8, .9, 1)$ gives a design equivalent in $\mathcal{D}$-efficiency to the MRMF design, and setting $v = (.9, .9, 1)$ produces a design with almost the same efficiencies as that from the Genetic algorithm.

		Model		
Design	Measure	(4.8)	(4.9)	(4.10)
$\mathcal{D}$-Maximin	$\lvert\mathbf{M}_f\rvert$	35.70	31.14	1.52
	D_f	.889	.894	.888
$(1, 1, .6)$-$\mathcal{D}$-Maximin	$\lvert\mathbf{M}_f\rvert$	43.79	41.34	0.44
	D_f	.951	.959	.721
MRMF	$\lvert\mathbf{M}_f\rvert$	27.04	33	3.01
	D_f	.810	.907	.995
Genetic Algorithm	$\lvert\mathbf{M}_f\rvert$	31.14	26.91	2.21
	D_f	.849	.862	.945
Design Optimal for (4.10)	$\lvert\mathbf{M}_f\rvert$	31.63	14.35	3.11
	D_f	.853	.737	1
Optimal (for each model)	$\lvert\mathbf{M}_f\rvert$	50.88	48.77	3.11

Table 4.1. Determinants, with $\mathcal{D}$-efficiencies, for example in §4.4.1 with $n = 6$, protecting against three models.

In Figure 4.1, we give a schematic illustrating the model-robust designs in Table 4.1. The MRMF and GA designs, which are close to optimal for the quadratic model, each place a design point near the center of the design space with the rest on the periphery. In contrast, the $\mathcal{D}$-maximin design, with a lower efficiency for the largest model, has an interior point but places it close to the boundary, while the $(1, 1, .6)$-$\mathcal{D}$-maximin has none strictly inside the space. Of course, by using the model interest vector the generalized maximin algorithm can certainly accommodate more emphasis on the quadratic model and less on the others if desired, which would result in an interior point.

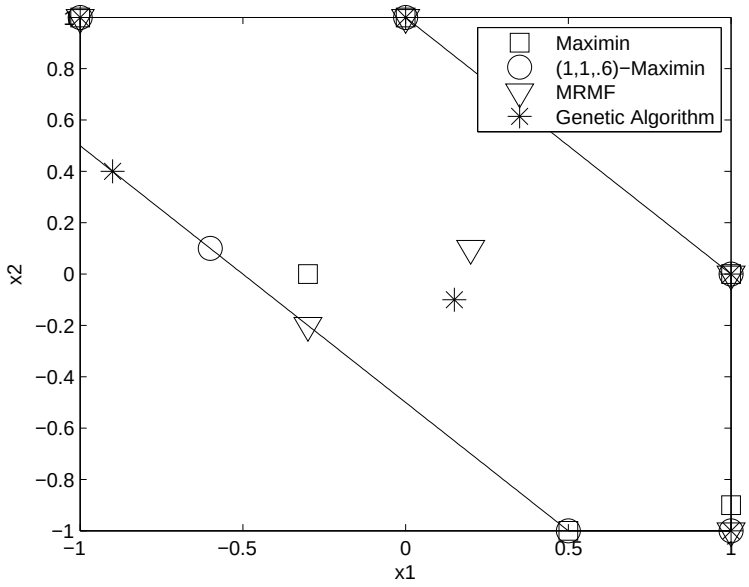

Figure 4.1. Model-robust designs for example in §4.4.1

4.4.2 Constrained Mixture Experiment

A constrained mixture example, originally from Snee [96], involves five mixture factors and their effect on the formulation of a certain plastic. The five factors were a binder (x_1), cobinder (x_2), plasticizer (x_3), and two monomers (x_4 and x_5). In addition to the mixture constraint, each component is constrained, and there are some multicomponent constraints as well:

$$\begin{aligned}\mathcal{X} = \{\mathbf{x} = (x_1, \ldots, x_5) : \ & \sum_{i=1}^{5} x_i = 1 \\ & 0.50 \le x_1 \le 0.70 \\ & 0.05 \le x_2 \le 0.15 \\ & 0.05 \le x_3 \le 0.15 \\ & 0.10 \le x_4 \le 0.25 \\ & 0.00 \le x_5 \le 0.15 \\ & 0.18 \le x_4 + x_5 \le 0.26 \\ & 0.00 \le x_3 + x_4 + x_5 \le 0.35\}\end{aligned}$$

We consider three possible models:

$$f_1^{'}(\mathbf{x}) = (\{x_i, i = 1, \ldots, 5\}) \tag{4.11}$$

$$f_2^{'}(\mathbf{x}) = (f_1^{'}, \{x_i x_j, i < j \leq 5\}) \tag{4.12}$$

$$f_3^{'}(\mathbf{x}) = (f_2^{'}, \{x_i x_j x_k, i < j < k \leq 5\}) \tag{4.13}$$

so that $\mathcal{F} = \{f_i^{'}(\mathbf{x})\beta_i, 1 \leq i \leq 3, \mathbf{x} \in \mathcal{X}\}$. The experimental budget was $n = 25$ which means model f_3 is saturated. For a candidate list of design points, we utilized the extreme vertices and approximate centroids, using the code by Piepel [85].

In Table 4.2, we include several generalized maximin designs, the MRMF design, as well as the optimal design for f_3. These are compared to the $|\mathbf{M}|$ for the optimal design for each individual model (note that in addition to the extreme vertices and approximate centroid, the candidate list for these individually optimal designs included a grid of resolution 0.01).

For the $\mathcal{D}$-maximin design it appears that, asymptotically, models f_1 and f_3 would be limiting, both with $\mathcal{D}$-efficiencies of about 87% in this finite-sample case. This design's minimum $\mathcal{D}$-efficiency is larger than that of the other designs, though it has lower $\mathcal{D}$-efficiency with respect to models f_2 and f_3 when compared to other model-robust strategies (i.e. the MRMF design or the design optimal for model f_3).

If the experimenter is willing to give up some efficiency for the first model the algorithm could be rerun, for instance, with $v = (.9, 1, 1)$. The results are in Table 4.2 and is nearly CGEB with a $\mathcal{D}$-efficiency for f_1 about 90% that of the others. By giving up a relatively small amount for model f_1, we gain substantially on the others. Notice that this design is closer to the MRMF design.

Alternatively, suppose that the experimenter believes *a priori* that f_2 is most likely, but varying degrees of protection are desired for f_1 and f_3. Then, $v = (.9, 1, .5)$ might be chosen. The results in Table 4.2 suggest that the first two models would have GEB asymptotically ($G_{f_1} = 0.882/0.9 = 0.98$ and $G_{f_2} = 0.979$, while $G_{f_3} = 0.507/0.5 = 1.014$). Overall, it is clear that the generalized maximin algorithm gives great flexibility for designing with respect to the specified models.

		Model		
Design	Measure	(4.11)	(4.12)	(4.13)
$\mathcal{D}$-Maximin	$\lvert\mathbf{M}_f\rvert$	8.76e-5	1.83e-49	1.13e-124
	D_f	.869	.903	.868
$(.9, 1, 1)$-$\mathcal{D}$-Maximin	$\lvert\mathbf{M}_f\rvert$	7.56e-5	3.01e-49	6.87e-124
	D_f	.844	.934	.933
$(.9, 1, .5)$-$\mathcal{D}$-Maximin	$\lvert\mathbf{M}_f\rvert$	9.43e-5	6.09e-49	1.61e-130
	D_f	.882	.979	.507
MRMF	$\lvert\mathbf{M}_f\rvert$	6.66e-5	4.21e-49	1.08e-123
	D_f	.822	.955	.950
Design Optimal for (4.13)	$\lvert\mathbf{M}_f\rvert$	5.78e-5	2.56e-49	3.86e-123
	D_f	.800	.924	1
Optimal (for each model)	$\lvert\mathbf{M}_f\rvert$	1.77e-4	8.42e-49	3.86e-123

Table 4.2. Determinants, with $\mathcal{D}$-efficiencies, for example in §4.4.2 with $n = 25$, protecting against three models.

4.4.3 Mixture Experiment With Disparate Models

We present an additional example based on a pharmaceutical experiment reported by Frisbee and McGinity [42]. This 11-run, mixture experiment investigated the effect of three nonionic surfactant factors on the glass transition temperature of films obtained from poly(DL-lactide) pseudolatex. Though Frisbee and McGinity fit a traditional mixture model, Rajagopal and Castillo [89] showed that a model from the class of Becker mixture models had a better fit. Consequently, we will choose $\mathcal{F} = \{f_i^{'}(\mathbf{x})\beta_i, 1 \leq i \leq 5, \mathbf{x} \in \mathcal{X}\}$ with

$$f_1^{'}(\mathbf{x}) = (\{x_i, i = 1, 2, 3\}) \tag{4.14}$$

$$f_2^{'}(\mathbf{x}) = (f_1^{'}, \{x_i x_j, i < j \leq 3\}) \tag{4.15}$$

$$f_3^{'}(\mathbf{x}) = (f_2^{'}, x_1 x_2 x_3) \tag{4.16}$$

$$f_4^{'}(\mathbf{x}) = (f_1^{'}, \{min(x_i, x_j), i < j \leq 3\}) \tag{4.17}$$

$$f_5^{'}(\mathbf{x}) = (f_4^{'}, min(x_1, x_2, x_3)) \tag{4.18}$$

with

$$\mathcal{X} = \left\{ \mathbf{x} = (x_1, x_2, x_3) : \sum_{i=1}^{3} x_i = 1, 0 \leq x_i \leq 1, i = 1, 2, 3 \right\} \tag{4.19}$$

We will also evaluate the designs in terms of two models not explicitly consid-

ered in our model-robust design. The first is the model fit by Frisbee and McGinity, and the second is the model found most probable *a posteriori* by Rajagopal and Castillo:

$$f'_{fm}(\mathbf{x}) = (x_1, x_2, x_3, x_1x_3, x_2x_3) \tag{4.20}$$

$$f'_{rc}(\mathbf{x}) = (x_1, x_2, x_3, min(x_1, x_3), min(x_2, x_3)) \tag{4.21}$$

For a candidate list, we used a regular grid with resolution 1/12, which because of the regular design region, contained its vertices and centroids.

The efficiencies for the $\mathcal{D}$-maximin design are given in Table 4.3 with probable limiting models f_1 and f_5 (this was confirmed by additional work in which we set $n = 50$ with a denser grid). Based on this result, the experimenter might want to generate another design in which $v = (.9, 1, 1, 1, .9)$, which can be seen in Table 4.3 to have f_1 and f_4 as probable limiting models, with $G_{f_1} = 0.883$ and $G_{f_4} = 0.888$. In this case, allowing such a sensitivity analysis produces gains for the final four models, while degrading f_1 only slightly.

Also in Table 4.3 is the MRMF design from [95] as well as the original design of Frisbee and McGinity. For comparison purposes, we include also the optimal designs for the two "maximal" models, f_3 and f_5. All designs excepting these latter two are included in Figure 4.2.

		Model						
Design	Measure	(4.14)	(4.15)	(4.16)	(4.17)	(4.18)	(4.20)	(4.21)
$\mathcal{D}$-Maximin	$\lvert\mathbf{M}_f\rvert$	24.73	3.71e-3	3.00e-6	0.200	6.43e-3	7.97e-2	1.44
	D_f	.802	.883	.921	.840	.811	.843	.863
$(.9, 1, 1, 1, .9)$-$\mathcal{D}$-Maximin	$\lvert\mathbf{M}_f\rvert$	24.12	4.48e-3	3.34e-6	.280	8.14e-3	7.31e-2	1.07
	D_f	.795	.911	.935	.888	.839	.828	.814
MRMF	$\lvert\mathbf{M}_f\rvert$	19.81	5.91e-3	5.36e-6	0.569	2.78e-2	6.61e-2	1.46
	D_f	.745	.955	1	1	1	.812	.866
Design Optimal for (4.16)	$\lvert\mathbf{M}_f\rvert$	19.81	5.9e-3	5.36e-6	.569	2.78e-2	.1025	2.15
	D_f	.745	.955	1	1	1	.886	.936
Design Optimal for (4.18)	$\lvert\mathbf{M}_f\rvert$	11.58	5.2e-3	5.36e-6	.486	2.78e-2	7.6e-3	1.36
	D_f	.623	.935	1	.9741	1	.834	.854
Frisbee and McGinity	$\lvert\mathbf{M}_f\rvert$	8.25	1.22e-3	1.51e-6	.146	8.82e-3	2.30e-2	.588
	D_f	.556	.733	.835	.797	.849	.658	.722
Optimal (for each model)	$\lvert\mathbf{M}_f\rvert$	48	7.8e-3	5.36e-6	.569	2.78e-2	.188	3

Table 4.3. Determinants, with $\mathcal{D}$-efficiencies, for example in §4.4.3 with $n = 11$, protecting against five models.

The results of the maximin designs are as advertised: Their minimum $\mathcal{D}$-efficiency is at least 5% higher than the others. They do, however, give up efficiency for models f_2-f_5. We note that choosing $v = (.75, 1, 1, 1, 1)$ results in a generalized maximin design equivalent to the MRMF or the design optimal for f_3 alone. Care must be taken when interpreting the efficiencies with respect to the fitted models, f_{fm} and f_{rc}. For instance, the MRMF design and the design optimal for f_3 alone appear to be equivalent, though their efficiencies for f_{fm} and f_{rc} differ. This is because, as seen in Figure 4.2, the MRMF design includes two design points at $(.5, .5, 0)$ and one at $(0, .5, .5)$, while the design optimal for f_3 alone (not shown) has only one design point at $(.5, .5, 0)$ and two at $(0, .5, .5)$. Since the fitted models include a blending term with both x_2 and x_3 but not one with x_1 and x_2, the latter design is preferred over the former. Unless an *a priori* inclination for the correct blending term exists, there is no reason to prefer one over the other at the design stage.

4.5 Discussion

The maximin exchange algorithm presented in this chapter is a practical model-robust procedure which produces designs robust for a set of experimenter-specified model forms by maximizing the minimum $\mathcal{D}$-efficiency with respect to each model. It also admits specification of varying levels of interest in each model so that a design reflective of those interests will be generated and as such affords additional control to the user. This method provides a worst-case efficiency guarantee and in our experience usually has a significantly higher minimum efficiency than those of the procedures against which they were compared.

We noticed that the designs produced by this procedure tend to have less variable efficiencies (with respect to the models designed for) than those from other methods. This led to the asymptotic analysis in §4.3 which gives a condition which guarantees complete generalized efficiency balance. This condition is often unsatisfied; however, we also proved that even when complete balance is unattainable, some subset of the specified models is limiting and therefore balanced. We emphasize that the utility of this procedure is not dependent upon these asymptotic results, but that they can provide insight into the algorithm and its designs.

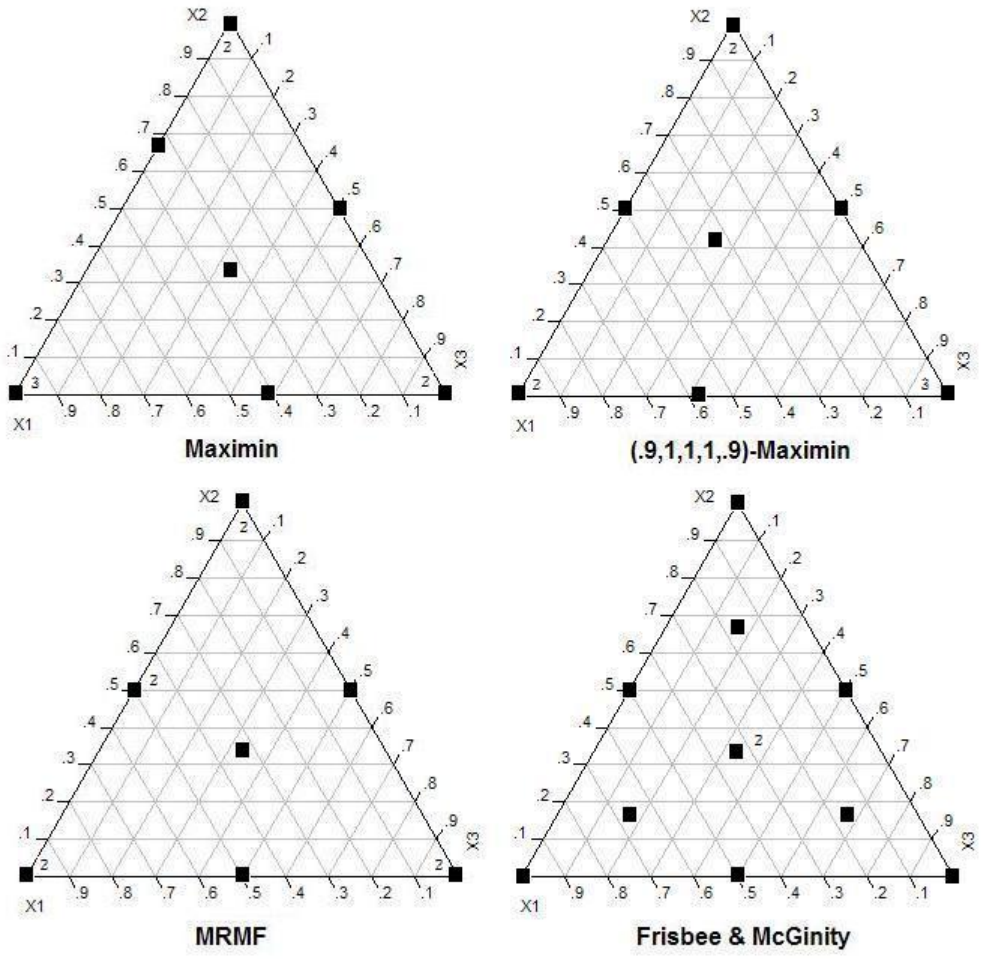

Figure 4.2. Some 11-run designs for example in §4.4.3, with repeated points noted

Included in our comparisons are optimal designs for the largest model under consideration. This represents a reasonable model-robust design strategy that might be employed using commercially available software. This approach gives no worst-case protection, however, which may result in unnecessarily small efficiencies for smaller models. It also gives the experimenter no control over the efficiencies with respect to any model besides the one that is assumed. Our maximin procedure addresses both of these issues.

Our method can also be used similarly to the Bayesian approach of DuMouchel and Jones [37], in which an assumed model exists but protection is desired for some larger model. The maximin procedure allows a quantification of the required degree of protection by assigning an appropriate model interest element. For instance,

if a main effects model is likely, but at least some estimation of the two-factor interaction model as well as the full quadratic model is needed, a model interest vector $v = (1, .5, .5)$ could be specified and would result in a design with a high efficiency for the main effects model, but much less for the larger models (though still with the capacity to estimate them). If possible, the design would be CGEB, but even if it is not, the differences in efficiency will be suggestive of v. We do note that our procedure requires $n \geq p_l$, where p_l is the number of parameters in the largest model, while the Bayesian procedure eludes this restriction by using prior information.

A near upper bound on the runtime of this algorithm is mT_{Ex}, where m is the number of models considered, and T_{Ex} is the time taken by the classical exchange algorithm (in the case of this chapter, the modified Fedorov algorithm). We say "near upper bound" because there is a small amount of computational energy expended in calculating efficiencies which is unnecessary in the classical case. In practice, the runtime should be less than the near upper bound because exchanges are only evaluated for all models if the exchange increases the efficiency for the model currently with the lowest efficiency.

Chapter 5

Multiresponse, Model-Robust Experimental Design

5.1 Introduction

Many times, when an experiment is to be designed, there will be more than one response of interest. For instance, a food scientist may want to explore the effect of different combinations of ingredients on taste, texture, nutritional value, and cost. Or, an engineer may examine how various levels of constituent components of a tire tread compound affect properties such as rate of wear and hardness.

The preponderance of research in the design of experiments, however, has been directed toward the univariate case. In the multiresponse realm, there have been primarily theoretical advances with little guidance for practical situations. Is it worth the extra effort inherent in methodology that explicitly accounts for multiple responses? If so, can we develop tools that leverage multiresponse characteristics to produce an advantage over usual practice?

There are good reasons for the relative dearth of research in optimal multiresponse experimental design. In general, to design such an optimal experiment the relationship between the responses as quantified in the covariance matrix $\mathbf{\Sigma}$ must be known *a priori*. Furthermore, the form of the relationship—the model form—between each response and the factors must also be specified in advance. In the absence of these elements, the experimenter is forced to ignore both and design for

an assumed but unknown single model.

We approach this problem from a model-robust perspective, and argue that in many cases a good design is a model-robust one for which $\mathbf{\Sigma}$ is irrelevant or unimportant anyway. When both the model form and covariance matrix are unknown, the multiresponse $\mathcal{D}$-optimal design problem can be cast effectively as a univariate model-robust design problem.

One contribution of this chapter is to develop and present an exchange algorithm to construct multiresponse $\mathcal{D}$-optimal designs when the covariance matrix and response model forms are known. But since this information is rarely known in practice, we extend the multiresponse model and the algorithm to allow for the construction of a design which is robust for an experimenter-specified set of possible models for each response.

We first review the multiresponse model, and then give an overview of our approach. Subsequently, we generalize matrix- and determinant-updating formulae given for the univariate case by Fedorov [41], and use them in a multiresponse exchange algorithm which assumes knowledge of both the covariance matrix and model forms. This corresponds to the most studied and least likely practical scenario for multiresponse design, so we then explore the relaxation of these assumptions, illustrating the methodology with several examples.

5.2 Background

The classical multivariate regression model [4] assumes the model form is common to all responses, and it has been shown [22, 70] that in this case the multiresponse optimal design solution is identical to the univariate optimal design. We use the more flexible seemingly unrelated regression (SUR) model [115] which allows different model forms for each response.

We now review this multiresponse model. Suppose there are r responses, k factors of interest, and n observations for each response. Let

$$\mathbf{y}_i = \mathbf{Z}_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i$$

be the linear regression for response i, with $\mathbf{y}_i$ the n-vector of observations, $\mathbf{Z}_i$

the $n \times q_i$ expanded design matrix, $\boldsymbol{\beta}_i$ the $q_i \times 1$ vector of parameters, and $\boldsymbol{\epsilon}_i$ the n-vector of errors. Further, let $q = \sum_i q_i$. Then, this multivariate model can be specified as

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_r \end{pmatrix} = \begin{pmatrix} \mathbf{Z}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{Z}_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_r \end{pmatrix} + \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_r \end{pmatrix} \tag{5.1}$$

or more concisely as

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{5.2}$$

where $\mathbf{Y}$ and $\boldsymbol{\epsilon}$ are now $nr \times 1$ vectors, $\boldsymbol{\beta}$ is a $q \times 1$ vector, and $\mathbf{Z}$ is a $nr \times q$ matrix and all of these quantities are shown in (5.1).

We make the standard normality assumption that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Omega})$ where $\boldsymbol{\Omega} = \boldsymbol{\Sigma} \otimes \mathbf{I}_n$ and '$\otimes$' is the Kronecker product. Here $\boldsymbol{\Sigma}$ models the covariance between the responses and $\mathbf{I}_n$ models the assumed independence across observations within a particular response.

An estimate for $\boldsymbol{\beta}$ in (5.2) is given by the generalized least squares estimator (also known as the Aitken estimator):

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z})^{-1}\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Y} \tag{5.3}$$

with

$$Var(\hat{\boldsymbol{\beta}}) = (\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z})^{-1}$$

Under the assumption of normality, $\hat{\boldsymbol{\beta}}$ is also the maximum likelihood estimator [115]. Since $\boldsymbol{\Sigma}$ is usually unknown, the estimator in (5.3) cannot be employed directly, though estimators for $\boldsymbol{\Sigma}$ have been proposed (see [115, 63]). We omit this discussion because of the present focus on design.

The multiresponse design problem is to specify n design points with respect to the aforementioned multiresponse regression model. Let $\mathcal{X}$ be the design space, Ξ be the set of all possible designs and $\xi_n(\mathbf{x}) \in \Xi$ be a discrete, n-point design:

$$\xi_n = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_d \\ n_1 & n_2 & \dots & n_d \end{pmatrix}$$

where d is the number of distinct design points, and n_i, $i = 1, \ldots, d$ is the number of runs performed at design point $\mathbf{x}_i$.

For this exact design we define the multiresponse information matrix:

$$\begin{aligned}\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma}) &= \sum_{i=1}^{n} \boldsymbol{\rho}(\mathbf{x}_i)\boldsymbol{\Sigma}^{-1}\boldsymbol{\rho}'(\mathbf{x}_i) \\ &= \mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z} \\ &= \left[Var(\hat{\boldsymbol{\beta}})\right]^{-1}\end{aligned}$$

where

$$\boldsymbol{\rho}'(\mathbf{x}) = \begin{pmatrix} \mathbf{z}_1'(\mathbf{x}) & 0 & \ldots & 0 \\ 0 & \mathbf{z}_2'(\mathbf{x}) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbf{z}_r'(\mathbf{x}) \end{pmatrix} \tag{5.4}$$

is an $r \times q$ matrix, and $\mathbf{z}_i'(\mathbf{x})$ is a vector with entries of the same form as the expanded design matrix $\mathbf{Z}_i$, for response i and design point $\mathbf{x}$. The multiresponse $\mathcal{D}$-optimal design maximizes the determinant of the information matrix:

$$\xi_n^* = \arg\max_{\xi_n \in \Xi} |\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma})|$$

Implicit in the information matrix is a dependence upon $\boldsymbol{\rho}(\mathbf{x})$, which includes the model form for each response, as well as the response covariance matrix $\boldsymbol{\Sigma}$. The work in this chapter examines these assumptions and design options in the face of them.

5.3 Model-Robust Design for the SUR Model

Similar to what has been proposed in the single response case (see, for instance, Chapters 3 and 4, as well as [71, 26, 54]), we allow the experimenter to choose a set of potential model forms, $\mathcal{F}_i$, for each response i, where each set has p_i models. What follows is an artificial, admittedly byzantine exercise meant only to establish a structure within which model-robust designs for multiresponse experiments can be constructed.

For the purposes of model-robust design, imagine a multiresponse model which has, instead of r responses as described above, $\tilde{p} = \sum_{i=1}^{r} p_i$ responses, one for each possible model form assigned to each response. To specify such a framework, we let $\mathbf{y}_{\ell j}$ be an n-vector of observations for response ℓ and potential model j. Then, all of the "data" for response ℓ is an np_ℓ-vector, $\mathbf{Y}_\ell = (\mathbf{y}'_{\ell 1}, \ldots, \mathbf{y}'_{\ell p_\ell})'$ and $\tilde{\mathbf{Y}} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_r)'$ is an $n\tilde{p}$-vector. Further, let $\mathbf{Z}_{\ell j}$ be the $n \times q_{\ell j}$ expanded design matrix for response ℓ and model j, where $q_{\ell j}$ is the number of parameters and $\sum_{\ell=1}^{r} \sum_{j=1}^{p_\ell} q_{\ell j} = \tilde{q}$. Then, we have the SUR model

$$\begin{pmatrix} \mathbf{y}_{11} \\ \vdots \\ \mathbf{y}_{1p_1} \\ \mathbf{y}_{21} \\ \vdots \\ \mathbf{y}_{2p_2} \\ \vdots \\ \mathbf{y}_{r1} \\ \vdots \\ \mathbf{y}_{rp_r} \end{pmatrix} = \begin{pmatrix} \mathbf{Z}_{11} & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{Z}_{1p_1} & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{Z}_{21} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{Z}_{2p_2} & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{Z}_{r1} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{Z}_{rp_r} \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_{11} \\ \vdots \\ \boldsymbol{\beta}_{1p_1} \\ \boldsymbol{\beta}_{21} \\ \vdots \\ \boldsymbol{\beta}_{2p_1} \\ \vdots \\ \boldsymbol{\beta}_{r1} \\ \vdots \\ \boldsymbol{\beta}_{rp_r} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\epsilon}_{11} \\ \vdots \\ \boldsymbol{\epsilon}_{1p_1} \\ \boldsymbol{\epsilon}_{21} \\ \vdots \\ \boldsymbol{\epsilon}_{2p_2} \\ \vdots \\ \boldsymbol{\epsilon}_{r1} \\ \vdots \\ \boldsymbol{\epsilon}_{rp_r} \end{pmatrix}$$

where $\boldsymbol{\beta}_{\ell j}$ is a $q_{\ell j} \times 1$ vector, and $\mathbf{y}_{\ell j}$ and $\boldsymbol{\epsilon}_{\ell j}$ are $n \times 1$ vectors. This model can be written more succinctly as

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{Z}}\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\epsilon}} \tag{5.5}$$

where $\tilde{\mathbf{Y}}$ and $\tilde{\boldsymbol{\epsilon}}$ are $n\tilde{p}$-vectors, $\tilde{\mathbf{Z}}$ is an $n\tilde{p} \times \tilde{q}$ matrix, and $\tilde{\boldsymbol{\beta}}$ is a $\tilde{q}$-vector. Suppose we further make the distributional assumption that $\tilde{\boldsymbol{\epsilon}}$ has mean $\mathbf{0}$ and $n\tilde{p} \times n\tilde{p}$ covariance matrix $\tilde{\boldsymbol{\Omega}}$. Then, the information matrix is $\mathbf{M}_m(\xi_n, \tilde{\boldsymbol{\Sigma}}) =$

$\sum_{i=1}^{n} \tilde{\boldsymbol{\rho}}(\mathbf{x})\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\rho}}'(\mathbf{x}) = \tilde{\mathbf{Z}}'\tilde{\boldsymbol{\Omega}}^{-1}\tilde{\mathbf{Z}}$ where

$$\tilde{\boldsymbol{\rho}}'(\mathbf{x}) = \begin{pmatrix} \mathbf{z}'_{11}(\mathbf{x}) & 0 & \dots & 0 & \dots & 0 \\ 0 & \mathbf{z}'_{12}(\mathbf{x}) & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{z}'_{1p_1}(\mathbf{x}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & \mathbf{z}_{rp_r}(\mathbf{x}) \end{pmatrix}$$

and $\tilde{\boldsymbol{\Sigma}}$ is given (see below) with $\tilde{\boldsymbol{\Omega}} = \tilde{\boldsymbol{\Sigma}} \otimes \mathbf{I}_n$.

Note that if $\tilde{\boldsymbol{\Omega}}$ and $\tilde{\boldsymbol{\rho}}(\mathbf{x})$ are specified, the multiresponse $\mathcal{D}$-optimal design for (5.5) will account for all model forms implicit in the SUR model. Thus, we can relax the model-form assumption for each response by framing the model-robust problem as a large multiresponse optimal design problem in which each response is assigned a set of possible model forms.

Again, this is an artificial device because the design problem at hand only involve r responses, not $\tilde{p}$. However, if the determinant of this expanded information matrix is maximized, the resulting design should be robust to the models represented in $\mathcal{F}_u = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_r$ in the same way that a multiresponse $\mathcal{D}$-optimal design can give a model-robust single response design (see Chapter 3).

To utilize this idea we must specify $\tilde{\boldsymbol{\Sigma}}$. In the expanded model, there are $\tilde{p}$ "responses" (really response/model combinations) which means $\tilde{\boldsymbol{\Omega}} = \tilde{\boldsymbol{\Sigma}} \otimes \mathbf{I}_n$ where $\tilde{\boldsymbol{\Sigma}}$ is $\tilde{p} \times \tilde{p}$. Recall that the true response covariance matrix is $\boldsymbol{\Sigma} = \{\sigma_{ij}^2\}$ for $i, j = 1, \ldots, r$, and in the case that this quantity is known in advance, we must incorporate it into $\tilde{\boldsymbol{\Sigma}}$. Then, a reasonable specification is $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} \otimes \mathbf{I}_p$, where p is the number of possible models for *each* response. This is somewhat restrictive, because the number of models must be the same for all responses; however, we found that a more complex specification allowing the number of models to vary over responses—that different possible models within a given response are uncorrelated, but for any two response/model combinations ij and $i'j'$ so that $i \neq i'$ the correlation is $\sigma_{ii'}^2$—produced $\tilde{\boldsymbol{\Omega}}$ not always invertible. Thus, we take $\tilde{\boldsymbol{\Omega}} = \boldsymbol{\Sigma} \otimes \mathbf{I}_p \otimes \mathbf{I}_n$.

Recall that when models are nested, multiresponse optimal designs can be constructed in the absence of knowledge of $\boldsymbol{\Sigma}$ [69, 15, 16, 70]. The same results

suggest that for the expanded, model-robust version of the multiresponse model, if the union of all possible models are nested, the design optimal for the expanded model is invariant to $\mathbf{\Sigma}$, which leads to a simplified criterion in which the present problem is essentially the same as the one for univariate model-robustness (see §5.4.1.2).

In summary, an artificially large SUR model can be constructed by assigning a set of possible models to each response, and the optimal design for this larger model is, in a sense, robust to the set of all possible models specified. In the case that the specified possible models are nested, the best design with respect to this large model can be found irrespective of $\mathbf{\Sigma}$. If they are nonnested, there may be a (usually small) loss of efficiency due to the necessity of using $\mathbf{\Sigma} = \mathbf{I}$.

5.3.1 Choosing a Set of Models

Optimally, when there is no preference for certain models, one would like to construct a design that was robust for the set of all possible models, or the set of all possible hierarchical models. However, the former grows exponentially in the number of model parameters and the latter quickly grows unmanageable as well, even for problems with as few as four or five factors.

In this section, we explore two different types of model sets for the general case in which it is desired to provide robustness for first- and second-order terms. We reiterate that the set of models is user-defined, and suggest these sets only as defaults in case no preference is given to any particular first- or second-order polynomial models. The first, which we call the SSPS (small, symmetric, pseudo-spanning) set, uses just three models, regardless of the number of factors: main effects; main effects and all two-factor interactions; and a full quadratic model. The second, which we call SPS (symmetric, pseudo-spanning), scales linearly with the number of factors (3k+3 models), and consists of the three models in SSPS as well as

1. k models, each of which includes only a single main effect.

2. k models (for $k > 2$), each of which includes all main effects plus a single two-factor interaction. The k interactions should be chosen so that each factor shows up exactly twice.

3. k models, each of which includes all main effects and all two-factor interactions, as well as a single quadratic term.

The term "symmetric" is used to illustrate the lack of preference for any particular factor. Furthermore, "pseudo-spanning" should not be taken in a technical sense since it simply represents our attempt to "cover" the space of possible models. We note that there has been recent work assigning priors to the space of possible models [13] based on certain axiomatic assumptions, an approach discussed in the final section of this chapter.

5.4 Design Algorithms

As stated above, there are two significant hurdles inherent in the design of optimal multiresponse experimental designs. First, the response covariance matrix, $\mathbf{\Sigma}$, is often unknown at the design stage. At times, there may be historical estimates of $\mathbf{\Sigma}$ but many experiments do not have this luxury. Secondly, the form of the model with respect to each response is likely unavailable. A typical design gambit for the multiresponse experimenter is to simply design for a large, assumed, univariate model (i.e. full quadratic). This ignores, however, both of the aforementioned issues.

In what follows, we develop an exchange algorithm to address the basic case in which both $\mathbf{\Sigma}$ and $\boldsymbol{\rho}(\mathbf{x})$ are known. This, in itself, is a contribution because of the scant literature on constructing exact designs for multiresponse experiments. We facilitate this procedure by generalizing matrix- and determinant-updating formulas given in the univariate case by Fedorov [41], and then discuss the model-robust extension.

5.4.1 Basic Multiresponse Exchange Algorithm

We again base our algorithm on the modified Fedorov algorithm [26]. As in the univariate case, we exploit updating formulae to enable the computational feasibility of the multiresponse exchange algorithm In what follows, we first develop the updating formulae, then we give the algorithm and discuss how it can be used to find multiresponse model-robust designs.

5.4.1.1 Updating Formulae

The univariate modified Fedorov exchange algorithm seems, at first glance, to be a brute force optimization heuristic. Although it is, the computational requirements are moderated by a determinant-updating formula which allows the effect of an exchange to be calculated quickly. Here, we generalize standard univariate updating formulae and use them as the computational engine for our multiresponse exchange algorithm. But first, we give the updating formula for the standard univariate regression model: Given design ξ_n, design point $\mathbf{x}_j \in \xi_n$, and candidate list point $\mathbf{x} \in \chi$, a new design $\tilde{\xi}_n$ is generated when the two points are swapped and the new determinant of the univariate information matrix (denoted by $\mathbf{M}(\xi_n)$) is

$$|\mathbf{M}(\tilde{\xi}_n)| = |\mathbf{M}(\xi_n)| \, (1 + \Delta(\mathbf{x}_j, \mathbf{x}, \xi_n)) \tag{5.6}$$

where

$$\Delta(\mathbf{x}_j, \mathbf{x}, \xi_n) = \mathbf{V}(\mathbf{x}, \xi_n) - \mathbf{V}(\mathbf{x}, \xi_n)\mathbf{V}(\mathbf{x_i}, \xi_n) + \mathbf{V}^2(\mathbf{x}, \mathbf{x}_j, \xi_n) - \mathbf{V}(\mathbf{x}_j, \xi_n)$$

with $\mathbf{V}(\mathbf{x}, \xi_n) = f'(\mathbf{x})\mathbf{M}^{-1}(\xi_n)f(\mathbf{x})$ and $\mathbf{V}(\mathbf{x}, \mathbf{x}_j, \xi_n) = f'(\mathbf{x})\mathbf{M}^{-1}(\xi_n)f(\mathbf{x}_j)$. This is under standard assumptions about the univariate model, including, without loss of generality, that the variance of the error term is $\sigma^2 = 1$.

We prove a multivariate generalization of a result from Fedorov (Lemma 3.2.1, [41]) from which the univariate determinant update (5.6) is derived, using the same sorts of arguments. This is essentially identical to that given in Huizenga et al. [56], but we present it here with an explicit proof and give a corollary which further reduces computational burden in certain cases. We also follow Fedorov and give a more general result in which ℓ points are swapped, instead of just 1. Some supporting lemmas are given in Appendix D.

Theorem 3. *Let ξ_n be an exact design consisting of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ and $\tilde{\xi}_n$ be the design produced when $\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \ldots, \mathbf{x}_{j_\ell}$, $\mathbf{x}_{j_i} \in \xi_n$, are exchanged for $\tilde{\mathbf{x}}_k \in \chi$, $k = 1, \ldots, \ell$. Further, let $\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma})$ be the $q \times q$ multivariate information matrix of the design ξ_n and $\boldsymbol{\rho}'(\mathbf{x}_{j_k})$ be the $r \times q$ multiresponse basis matrix, where $q = \sum_{i=1}^{r} q_i$*

and q_i is the number of parameters for the i^{th} response; then

$$|\mathbf{M}_m(\tilde{\xi}_n, \boldsymbol{\Sigma})| = |\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma})||\mathbf{I}_{2\ell r} + \mathbf{A}_2^{'}\mathbf{M}_m^{-1}(\xi_n, \boldsymbol{\Sigma})\mathbf{A}_1| \tag{5.7}$$

where

$$\mathbf{A}_1 = \left(-\boldsymbol{\rho}(\mathbf{x}_{j_1})\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}(\tilde{\mathbf{x}}_1)\boldsymbol{\Sigma}^{-1/2}, \ldots, -\boldsymbol{\rho}(\mathbf{x}_{j_\ell})\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}(\tilde{\mathbf{x}}_\ell)\boldsymbol{\Sigma}^{-1/2}\right)$$

and

$$\mathbf{A}_2 = \left(\boldsymbol{\rho}(\mathbf{x}_{j_1})\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}(\tilde{\mathbf{x}}_1)\boldsymbol{\Sigma}^{-1/2}, \ldots, \boldsymbol{\rho}(\mathbf{x}_{j_\ell})\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}(\tilde{\mathbf{x}}_\ell)\boldsymbol{\Sigma}^{-1/2}\right)$$

and both matrices are $q \times 2\ell r$.

Proof. For brevity, we denote $\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma})$ as $\mathbf{M}_m$ and $\mathbf{M}_m(\tilde{\xi}_n, \boldsymbol{\Sigma})$ as $\tilde{\mathbf{M}}_m$.

By definition,

$$\tilde{\mathbf{M}}_m = \mathbf{M}_m - \sum_{k=1}^{\ell} \boldsymbol{\rho}(\mathbf{x}_{j_k})\boldsymbol{\Sigma}^{-1}\boldsymbol{\rho}^{'}(\mathbf{x}_{j_k}) + \sum_{k=1}^{\ell} \boldsymbol{\rho}(\tilde{\mathbf{x}}_k)\boldsymbol{\Sigma}^{-1}\boldsymbol{\rho}^{'}(\tilde{\mathbf{x}}_k)$$

Now,

$$\mathbf{A}_1\mathbf{A}_2^{'} = \left(-\boldsymbol{\rho}(\mathbf{x}_{j_1})\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}(\tilde{\mathbf{x}}_1)\boldsymbol{\Sigma}^{-1/2}, \ldots, -\boldsymbol{\rho}(\mathbf{x}_{j_\ell})\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}(\tilde{\mathbf{x}}_\ell)\boldsymbol{\Sigma}^{-1/2}\right) \begin{pmatrix} \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}^{'}(\mathbf{x}_{j_1}) \\ \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}^{'}(\tilde{\mathbf{x}}_1) \\ \vdots \\ \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}^{'}(\mathbf{x}_{j_\ell}) \\ \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}^{'}(\tilde{\mathbf{x}}_\ell) \end{pmatrix}$$

$$= \sum_{k=1}^{\ell} -\boldsymbol{\rho}(\mathbf{x}_{j_k})\boldsymbol{\Sigma}^{-1}\boldsymbol{\rho}^{'}(\mathbf{x}_{j_k}) + \boldsymbol{\rho}(\tilde{\mathbf{x}}_k)\boldsymbol{\Sigma}^{-1}\boldsymbol{\rho}^{'}(\tilde{\mathbf{x}}_k)$$

This implies that $\tilde{\mathbf{M}}_m = \mathbf{M}_m + \mathbf{A}_1\mathbf{A}_2^{'}$ and by Lemma 6 (Appendix D),

$$|\mathbf{M}_m + \mathbf{A}_1\mathbf{A}_2^{'}| = |\mathbf{M}_m||\mathbf{I}_{2\ell r} + \mathbf{A}_2^{'}\mathbf{M}_m^{-1}\mathbf{A}_1| \tag{5.8}$$

which implies what we wanted to prove. □

We can generalize this further, in the case where $\ell = 1$ and a particular matrix is invertible. Let $\mathbf{d}(\mathbf{x}, \xi_n) = \boldsymbol{\rho}'(\mathbf{x})\mathbf{M}_m^{-1}(\xi_n, \boldsymbol{\Sigma})\boldsymbol{\rho}(\mathbf{x})$ and $\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n) = \boldsymbol{\rho}'(\mathbf{x}_j)\mathbf{M}_m^{-1}(\xi_n, \boldsymbol{\Sigma})\boldsymbol{\rho}(\mathbf{x})$ and $\mathbf{x} \in \mathcal{X}$.

Corollary 2. *Let the setup be as in Theorem 3. Additionally, assume that* $\ell = 1$ *and* $\mathbf{B} = \mathbf{I}_r - \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \xi_n)\boldsymbol{\Sigma}^{-1/2}$ *is invertible. Then*

$$|\mathbf{M}_m(\tilde{\xi}_n, \boldsymbol{\Sigma})| = |\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma})| \times |\mathbf{B}| \times |\mathbf{I}_r + \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} + \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2}\mathbf{B}^{-1}\boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2}| \quad (5.9)$$

Proof. For convenience, let $\mathbf{M}_m^{-1} = \mathbf{M}_m^{-1}(\xi_n, \boldsymbol{\Sigma})$ and $\tilde{\mathbf{M}}_m{}^{-1} = \mathbf{M}_m^{-1}(\tilde{\xi}_n, \boldsymbol{\Sigma})$. Since $\ell = 1$,

$$\mathbf{A}_1 = \left(-\boldsymbol{\rho}'(\mathbf{x}_j)\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}'(\mathbf{x})\boldsymbol{\Sigma}^{-1/2}\right)$$

and

$$\mathbf{A}_2 = \left(\boldsymbol{\rho}'(\mathbf{x}_j)\boldsymbol{\Sigma}^{-1/2}, \boldsymbol{\rho}'(\mathbf{x})\boldsymbol{\Sigma}^{-1/2}\right).$$

By Theorem 3, $|\tilde{\mathbf{M}}_m| = |\mathbf{M}_m||\mathbf{I}_{2r} + \mathbf{A}_2^{'}\mathbf{M}_m^{-1}\mathbf{A}_1|$. Now,

$$\begin{aligned}
\mathbf{A}_2^{'}\mathbf{M}_m^{-1}\mathbf{A}_1 &= \begin{pmatrix} \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x}_j) \\ \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x}) \end{pmatrix} \mathbf{M}_m^{-1} \left(-\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x}_j), \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x})\right) \\
&= \begin{pmatrix} -\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x}_j)\mathbf{M}_m^{-1}\boldsymbol{\rho}(\mathbf{x}_j)\boldsymbol{\Sigma}^{-1/2} & \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x}_j)\mathbf{M}_m^{-1}\boldsymbol{\rho}(\mathbf{x})\boldsymbol{\Sigma}^{-1/2}. \\ -\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x}_j)\mathbf{M}_m^{-1}\boldsymbol{\rho}(\mathbf{x})\boldsymbol{\Sigma}^{-1/2} & \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\rho}'(\mathbf{x})\mathbf{M}_m^{-1}\boldsymbol{\rho}(\mathbf{x})\boldsymbol{\Sigma}^{-1/2} \end{pmatrix} \\
&= \begin{pmatrix} -\boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \xi_n)\boldsymbol{\Sigma}^{-1/2} & \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} \\ -\boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} & \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} \end{pmatrix}
\end{aligned}$$

Then,

$$\begin{aligned}
|\mathbf{I}_{2r} + \mathbf{A}_2^{'}\mathbf{M}_m^{-1}\mathbf{A}_1| &= \begin{vmatrix} -\boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \xi_n)\boldsymbol{\Sigma}^{-1/2} & \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} \\ -\boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} & \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} \end{vmatrix} \\
&= |\mathbf{I}_r - \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \xi_n)\boldsymbol{\Sigma}^{-1/2}| \\
&\quad |\mathbf{I}_r + \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} + \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} \\
&\quad (\mathbf{I}_r - \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \xi_n)\boldsymbol{\Sigma}^{-1/2})^{-1}\boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2}|
\end{aligned}$$

where the first equality follows from Lemma 5 and the second from Lemma 4. This implies that which we wanted to prove.

□

Theorem 3 (assuming a single exchange, i.e., $\ell = 1$) provides computational relief when $2r < q$, and Corollary 2 reduces the burden even further, by computing determinants and an inverse of smaller, $r \times r$ matrices.

We also would like updating formulae for the information matrix as well as its inverse. The justification for the former is given in the proof for Theorem 3. The proof of the latter is essentially from Fedorov [41], though it is explicitly for a multiresponse situation.

Corollary 3. *Using the assumptions of Theorem 3,*

$$\mathbf{M}_m(\tilde{\xi}_n, \mathbf{\Sigma}) = \mathbf{M}_m(\xi_n, \mathbf{\Sigma}) + \mathbf{A}_1\mathbf{A}_2' \tag{5.10}$$

Theorem 4. *Using the assumptions of Theorem 3,*

$$\mathbf{M}_m^{-1}(\tilde{\xi}_n, \mathbf{\Sigma}) = \left(\mathbf{I}_q - \mathbf{M}_m^{-1}(\xi_n, \mathbf{\Sigma})\mathbf{A}_1\left(\mathbf{I}_{2\ell} + \mathbf{A}_2'\mathbf{M}_m^{-1}(\xi_n, \mathbf{\Sigma})\mathbf{A}_1\right)^{-1}\mathbf{A}_2'\right)\mathbf{M}_m^{-1}(\xi_n, \mathbf{\Sigma}) \tag{5.11}$$

Proof. First, take the inverse of both sides of (5.10) to get

$$\tilde{\mathbf{M}}_m^{-1} = (\mathbf{M}_m + \mathbf{A}_1\mathbf{A}_2')^{-1} = (\mathbf{I}_q + \mathbf{M}_m^{-1}\mathbf{A}_1\mathbf{A}_2')^{-1}\mathbf{M}_m^{-1}$$

The last equality can be verified by right multiplying both sides by $\mathbf{M}_m + \mathbf{A}_1\mathbf{A}_2'$:

$$(\mathbf{M}_m + \mathbf{A}_1\mathbf{A}_2')^{-1}(\mathbf{M}_m + \mathbf{A}_1\mathbf{A}_2') = \mathbf{I}_q$$

and

$$\begin{aligned}(\mathbf{I}_q + \mathbf{M}_m^{-1}\mathbf{A}_1\mathbf{A}_2')^{-1}&\mathbf{M}_m^{-1}(\mathbf{M}_m + \mathbf{A}_1\mathbf{A}_2') \\ &= (\mathbf{I}_q + \mathbf{M}_m^{-1}\mathbf{A}_1\mathbf{A}_2')^{-1}(\mathbf{I}_q + \mathbf{M}_m^{-1}\mathbf{A}_1\mathbf{A}_2') = \mathbf{I}_q\end{aligned}$$

Then, using Lemma 7 from Appendix D, it follows that

$$\begin{aligned}\tilde{\mathbf{M}}_m^{-1} &= \left(\mathbf{I}_q + \mathbf{M}_m^{-1}\mathbf{A}_1\mathbf{A}_2'\right)^{-1}\mathbf{M}_m^{-1} \\ &= \left(\mathbf{I}_q - \mathbf{M}_m^{-1}\mathbf{A}_1\left(\mathbf{I}_{2\ell} + \mathbf{A}_2'\mathbf{M}_m^{-1}\mathbf{A}_1\right)^{-1}\mathbf{A}_2'\right)\mathbf{M}_m^{-1}\end{aligned}$$

□

5.4.1.2 Multiresponse Exchange Algorithm for $\mathcal{D}$-Optimal Designs

The $\mathcal{D}$-optimality criterion for the n-point multiresponse model is to maximize

$$\phi(\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma})) = |\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma})| = |\mathbf{Z}'\boldsymbol{\Omega}^{-1}\mathbf{Z}| = |\mathbf{Z}'(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_n)\mathbf{Z}|.$$

However, because of the above development, when $\mathbf{x}_j \in \xi_n$ and $\mathbf{x} \in \mathcal{X}$ are swapped, we have that

$$\phi(\mathbf{M}_m(\tilde{\xi}_n, \boldsymbol{\Sigma})) = \phi(\mathbf{M}_m(\xi_n, \boldsymbol{\Sigma}))\Delta_m(\mathbf{x}_j, \mathbf{x}, \boldsymbol{\Sigma})$$

where, assuming $\mathbf{B} = \mathbf{I}_r - \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \xi_n)\boldsymbol{\Sigma}^{-1/2}$ is invertible,

$$\begin{aligned}\Delta_m(\mathbf{x}_j, \mathbf{x}, \boldsymbol{\Sigma}) = |\mathbf{B}| \times |\mathbf{I}_r &+ \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2} \\ &+ \boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2}\mathbf{B}^{-1}\boldsymbol{\Sigma}^{-1/2}\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)\boldsymbol{\Sigma}^{-1/2}|\end{aligned} \tag{5.12}$$

with $\mathbf{d}(\mathbf{x}, \xi_n)$ and $\mathbf{d}(\mathbf{x}_j, \mathbf{x}, \xi_n)$ defined as in the prelude to Corollary 2. Otherwise, if $\mathbf{B}^{-1}$ does not exist,

$$\Delta_m(\mathbf{x}_j, \mathbf{x}, \boldsymbol{\Sigma}) = |\mathbf{I}_{2r} + \mathbf{A}_2^{'}\mathbf{M}_m^{-1}(\xi_n, \boldsymbol{\Sigma})\mathbf{A}_1| \tag{5.13}$$

with $\mathbf{A}_1$ and $\mathbf{A}_2$ defined as in the proof of Corollary 2. Consequently, as the exchange algorithm considers swaps, it can choose the one which maximizes Δ_m, not ϕ.

If the known model forms are nested, or if $\boldsymbol{\Sigma} = \mathbf{I}_r$, it is possible to simplify ϕ even further, as done in Chapter 3:

$$\begin{aligned}\phi(\mathbf{M}_m(\tilde{\xi}_n)) &= \prod_{i=1}^{r} \left|\mathbf{M}_i(\tilde{\xi}_n)\right| \\ &= \prod_{i=1}^{r} |\mathbf{M}_i(\xi_n)| \left(1 + \Delta_i(\mathbf{x}_j, \mathbf{x})\right) \\ &= \prod_{i=1}^{r} |\mathbf{M}_i(\xi_n)| \prod_{i=1}^{r} (1 + \Delta_i(\mathbf{x}_j, \mathbf{x}))\end{aligned} \tag{5.14}$$

where $\mathbf{M}_i$ and Δ_i are the univariate information matrices and update functions for response model i. Then, to evaluate an exchange, we need only to calculate $\prod_{i=1}^{r}(1+\Delta_i(\mathbf{x}_j,\mathbf{x}))$ where Δ_i is calculated, for response i, as in (5.6).

We can make yet another simplification if some of the model forms are the same for the different responses. Assume there are a total of $R \leq r$ distinct model forms, and let t_d be the number of times model d appears among all the model forms. Then we have that

$$\prod_{i=1}^{r}(1+\Delta_i(\mathbf{x}_j,\mathbf{x})) = \prod_{d=1}^{R}(1+\Delta_d(\mathbf{x}_j,\mathbf{x}))^{t_d} \tag{5.15}$$

We make a slight adjustment to this criterion so our algorithm will not choose an exchange so bad that $(1+\Delta_d(\mathbf{x}_j,\mathbf{x})) < 0$ for an even t_d and/or an even number of distinct models, which would result in a positive value of our criterion even though the exchange is undesirable. Thus, we choose the exchange which maximizes

$$\prod_{d=1}^{R}(1+\Delta_d(\mathbf{x}_j,\mathbf{x}))^{t_d}\mathbb{I}(1+\Delta_d(\mathbf{x}_j,\mathbf{x})>0) \tag{5.16}$$

where $\mathbb{I}$ is the indicator function.

Based on the above development, the Multiresponse Exchange (MX) algorithm is as follows:

1. Initialize algorithm constructing a nonsingular initial design ξ_n and a grid, $C \subset \mathcal{X}$.

2. Let $j = 1$.

3. For design point $\mathbf{x}_j$:

   (a) If models are nested or $\mathbf{\Sigma} = \mathbf{I}$, calculate (5.16) for all $\mathbf{x} \in C$. Let $\mathbf{x}_j^* = \arg\max_{\mathbf{x}\in C}\prod_{i=1}^{r}(1+\Delta_i(\mathbf{x}_j,\mathbf{x}))\mathbb{I}(1+\Delta_i(\mathbf{x}_j,\mathbf{x})>0)$.

   (b) Otherwise, if models are not nested and $\mathbf{\Sigma} \neq \mathbf{I}$, calculate $\Delta_m(\mathbf{x}_j,\mathbf{x},\mathbf{\Sigma})$, according to (5.12) or (5.13) as appropriate, for all $\mathbf{x} \in C$. Let $\mathbf{x}_j^* = \arg\max_{\mathbf{x}\in C}\Delta_m(\mathbf{x}_j,\mathbf{x},\mathbf{\Sigma})$.

4. Perform exchange $\mathbf{x}_j^*$ for $\mathbf{x}_j$, updating ξ_n. Update the determinant and also $\mathbf{M}^{-1}(\xi_n, \boldsymbol{\Sigma})$ via Theorem 4.

5. Increment j and if $j < N$ return to Step 3. Else, if algorithm has converged, STOP. Else return to Step 2.

The convergence criterion used here is similar to that for the algorithm in Chapter **??** ($\epsilon = 0.01$). This algorithm is guaranteed to converge because it produces a bounded, nondecreasing sequence of determinants. It will not necessarily converge to the global solution, so multiple algorithm tries using different initial designs should be executed. Also, for large problems in particular, the determinants may need to be calculated on the log scale.

5.4.2 Model-robust, Multiresponse Exchange Algorithm

In §5.4.1, we developed an exchange algorithm for the multiresponse model when the response covariance matrix, $\boldsymbol{\Sigma}$, and the model form of each response, $\boldsymbol{\rho}$, are known *a priori*. To extend this to the model-robust situation, for which it is assumed that the form of the response models is not known precisely, we refer back to §5.3 and choose a set of possible models, $\mathcal{F}_i$, for each response and use them to construct the artificial SUR model in (5.5) with covariance matrix $\tilde{\boldsymbol{\Omega}} = \boldsymbol{\Sigma} \otimes \mathbf{I}_p \otimes \mathbf{I}_n$. Then, the MX algorithm can be used to calculate the model-robust design.

At this point, there still is the problem of unknown $\boldsymbol{\Sigma}$. However, if the choice of possible models are such that the union of them are nested, the model-robust design is invariant to $\boldsymbol{\Sigma}$. Even for nonnested cases, our experience is that there is little or no efficiency lost by using $\boldsymbol{\Sigma} = \mathbf{I}$ (see the examples).

5.5 Examples

In this section, we demonstrate the model-robust procedures using two simple theoretical examples, as well as a real example. All designs referred to are in Appendix E. Matlab® (Version 7.8) code can be found at `http://www.stat.psu.edu/~jlr/pub/Smucker/`.

5.5.1 3-factor, 2-response Experiment

The first example, used in the multiresponse design literature [112, 9], is one in which $\boldsymbol{\Sigma}$ and $\boldsymbol{\rho}(\mathbf{x})$ are given and there is an experimental budget of $n = 20$ runs. To remain consistent with previous literature, we take the design space to be $\mathcal{X} = \{\mathbf{x} = (x_1, x_2, x_3) : -1.73 \leq x_1, x_2, x_3 \leq 1.73\}$ with known model forms, for each response,

$$z_1^{'}(\mathbf{x}) = (1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_1^2, x_3^2)$$
$$z_2^{'}(\mathbf{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$$

so that $\boldsymbol{\rho}^{'}(\mathbf{x}) = \begin{pmatrix} z_1^{'} & 0 \\ 0 & z_2^{'} \end{pmatrix}$. We also have that the variance-covariance matrix of the responses are

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

We will use the multiresponse exchange algorithm developed in §5.4.1.2 to produce an exact solution to the multiresponse $\mathcal{D}$-optimal design problem using the given $\boldsymbol{\Sigma}$ and $\boldsymbol{\rho}$. The same problem is addressed in Atashgah and Seifi [9] using semidefinite programming, and we will compare our solution with theirs. We then will demonstrate how the problem and solution changes when $\boldsymbol{\Sigma}$ and/or $\boldsymbol{\rho}(\mathbf{x})$ are unknown. Before going further, we define the $\mathcal{D}$-efficiency of a design.

Definition 2. *The $\mathcal{D}$-efficiency of design ξ_n for the multiresponse regression model with covariance matrix $\boldsymbol{\Sigma}$ is defined as*

$$D(\xi_n, \boldsymbol{\Sigma}) = \left(\frac{\left| \mathbf{Z}_{\xi_n}^{'} \left(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_n \right) \mathbf{Z}_{\xi_n} \right|}{\left| \mathbf{Z}_{\xi_n^*}^{'} \left(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_n \right) \mathbf{Z}_{\xi_n^*} \right|} \right)^{1/q}$$

where $\xi_n^* = \arg\max_{\xi_n \in \Xi} \left| \mathbf{Z}_{\xi_n}^{'} \left(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_n \right) \mathbf{Z}_{\xi_n}^{'} \right|$.

5.5.1.1 When $\boldsymbol{\rho}(\mathbf{x})$ is Known

When both $\boldsymbol{\rho}(\mathbf{x})$ and $\boldsymbol{\Sigma}$ are known, the MX algorithm in §5.4.1.2 can be used to find the optimal design. We use a 27-element candidate list with just 3 levels (-1.73,

0, 1.73) for each factor, since the models have no terms higher than quadratic (a denser candidate list with seven levels was tried also, but effected no change in the solution). The design, in Table E.1, is the best out of 50 algorithm tries. Atashgah and Seifi [9], on the other hand, used a semi-definite programming algorithm and 50 so-called "test vectors" to generate their design. It is unclear why they did not limit themselves to the 3^3 set of test vectors, or, if those 27 were included, why the procedure chose others, as shown in Table E.2.

However, if the experimenter is confident in the form of the regression function for each response, but does not know the covariance matrix which relates them, let $\mathbf{\Sigma} = \mathbf{I}$ and use the MX algorithm to construct a design (Table E.3), using the same 27-point candidate list and the best of 50 algorithm tries. Table 5.1 compares the two designs and shows the superiority of that constructed using our exchange algorithm. Note also that very little efficiency is lost when $\mathbf{\Sigma}$ is unknown. The $\mathcal{D}$-efficiencies are with respect to the best design we found, in this case using the MX algorithm.

Design	Covariance	$\lvert\mathbf{M}_m\rvert$	$\mathcal{D}$-Eff.
MX	$\mathbf{I}$	8.30e20	0.999
MX	$\mathbf{\Sigma}$	8.38e20	1
SDP	$\mathbf{\Sigma}$	5.48e20	0.970

Table 5.1. Determinants, with $\mathcal{D}$-efficiencies, for example in §5.5.1 with $\boldsymbol{\rho}(\mathbf{x})$ known.

5.5.1.2 When $\boldsymbol{\rho}(\mathbf{x})$ is Unknown

Now we consider the case in which the model form for each response is unknown. Whether the covariance matrix is known or unknown, the experimenter specifies $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_r)$, where each $\mathcal{F}_i$ is a set of possible model forms for response i. If $\mathcal{F}_u = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_r$ is nested, we can utilize theory by Bischoff [15] and others to justify the use of the simpler determinant updates via (5.15) for the MX algorithm. If the model forms are not nested and $\mathbf{\Sigma} \neq \mathbf{I}$, then we must use the determinant-updating formula (5.12) which accounts for $\mathbf{\Sigma}$.

Two model sets, as described in §5.3.1, will be considered. For three factors, the SSPS set is

$$f_{11}^{'}(\mathbf{x}) = (1, x_1, x_2, x_3) \tag{5.17}$$

$$f_{12}^{'}(\mathbf{x}) = (f_{11}, x_1x_2, x_1x_3, x_2x_3) \tag{5.18}$$

$$f_{13}^{'}(\mathbf{x}) = (f_{12}, x_1^2, x_2^2, x_3^2) \tag{5.19}$$

so that $\mathcal{F}_{1,SSPS} = \{f_i^{'}(\mathbf{x})\boldsymbol{\beta}_{1i}, 1 \leq i \leq 3, \mathbf{x} \in \chi\}$. Further, assume that $\mathcal{F}_{2,SSPS} = \mathcal{F}_{1,SSPS}$ so that both responses have the same set of possible models. Then, $\mathcal{F}_{u,SSPS} = \mathcal{F}_{1,SSPS} \cup \mathcal{F}_{2,SSPS}$ is nested and consequently we can use the simpler algorithm.

The SPS set includes twelve models

$$f_{1+}^{'}(\mathbf{x}) = (1, x_1) \tag{5.20}$$

$$f_{2+}^{'}(\mathbf{x}) = (1, x_2) \tag{5.21}$$

$$f_{3+}^{'}(\mathbf{x}) = (1, x_3) \tag{5.22}$$

$$f_{4+}^{'}(\mathbf{x}) = (1, x_1, x_2, x_3) \tag{5.23}$$

$$f_{5+}^{'}(\mathbf{x}) = (f_{4+}, x_1x_2) \tag{5.24}$$

$$f_{6+}^{'}(\mathbf{x}) = (f_{4+}, x_1x_3) \tag{5.25}$$

$$f_{7+}^{'}(\mathbf{x}) = (f_{4+}, x_2x_3) \tag{5.26}$$

$$f_{8+}^{'}(\mathbf{x}) = (f_{4+}, x_1x_2, x_1x_3, x_2x_3) \tag{5.27}$$

$$f_{9+}^{'}(\mathbf{x}) = (f_{8+}, x_1^2) \tag{5.28}$$

$$f_{10+}^{'}(\mathbf{x}) = (f_{8+}, x_2^2) \tag{5.29}$$

$$f_{11+}^{'}(\mathbf{x}) = (f_{8+}, x_3^2) \tag{5.30}$$

$$f_{12+}^{'}(\mathbf{x}) = (f_{8+}, x_1^2, x_2^2, x_3^2) \tag{5.31}$$

so that $\mathcal{F}_{1,SPS} = \{f_{i+}^{'}(\mathbf{x})\boldsymbol{\beta}_{1i+}, 1 \leq i \leq 12, \mathbf{x} \in \mathcal{X}\}$ and $\mathcal{F}_{2,SPS} = \mathcal{F}_{1,SPS}$. This set of models is not nested, so the SPS-model-robust design will depend upon $\boldsymbol{\Sigma}$. Using the MX algorithm of §5.4.1.2, we can calculate three model-robust designs: SSPSMR (SSPS-model-robust, given in Table E.4); SPSMR-$\boldsymbol{\Sigma}$ (in the case that the covariance matrix is known); and SPSMR-$\mathbf{I}$ (when the covariance is unknown), and compare with the design optimal for the full quadratic alone (i.e. usual practice, given in Table E.6). In this case SPSMR-$\boldsymbol{\Sigma}$ and SPSMR-$\mathbf{I}$ are the same (Table E.5).

For a three-factor experiment, there are in total 63 possible models which obey

the oft-observed hierarchical principal (see [82, 13]), which says that higher-order terms are included in the model only if appropriate lower-order terms appears. Since there are two responses, this results in $63^2 = 3969$ total possible hierarchical models. To evaluate the model-robust designs described above, we have found the optimal designs for each of these possible models individually (based upon only 10 algorithm tries due to computational constraints, and using a multiresponse exchange algorithm in which only one exchange is made per iteration), allowing us to calculate the $\mathcal{D}$-efficiency of the model-robust designs with respect to each of the 3969 models. We give relevant, comparative statistics in Table 5.2. For these hierarchical models, the model-robust designs have higher $\mathcal{D}$-efficiencies on average, with less variability, than the design optimal for the full quadratic model. Interestingly, while the design for the larger SPS set has less variability in its efficiencies, its average is inferior to the design with respect to the smaller SSPS set.

Design	Mean	StDev	Min
SSPSMR	0.940	0.037	0.829
SPSMR-$\boldsymbol{\Sigma}$/SPSMR-$\mathbf{I}$	0.920	0.024	0.838
Quadratic	0.915	0.056	0.740

Table 5.2. Comparison of model-robust designs for example in §5.5.1, in terms of $\mathcal{D}$-efficiencies with respect to all 3969 possible true models.

5.5.2 Two-factor, Two-response Experiment

We also consider a theoretical example in which there are two responses, two factors and 12 runs, with both $\boldsymbol{\Sigma}$ and $\boldsymbol{\rho}(\mathbf{x})$ unknown. Since the covariance matrix is unknown we assume $\boldsymbol{\Sigma} = \mathbf{I}$ when constructing designs.

We can again use the algorithm of §5.4.1.2 to construct these model-robust designs, one each with respect to the SSPS and SPS sets, using 50 algorithm tries and a small 3^2 candidate list. In this case, the two designs turn out to be equivalent (Table E.7). To evaluate the model-robustness of this design, we again consider the set of all possible hierarchical models up through second order polynomials. Since there are seven such models for each response, there are $7^2 = 49$ models in total.

We found the optimal designs for each of the 49 models (note that for convenience we did not use the simplification of Equation (5.14) even when appropriate; also, we again used an exchange algorithm similar to the one described in §5.4.1.2 except with only one swap made per iteration), for each of three assumed "true" $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{5.32}$$

$$\boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & .8 \\ .8 & 1 \end{pmatrix}, \tag{5.33}$$

$$\boldsymbol{\Sigma}_3 = \begin{pmatrix} 9 & 1.5 \\ 1.5 & 1 \end{pmatrix}. \tag{5.34}$$

We also constructed the design optimal for the quadratic model alone (Table E.8) and compared it with the model-robust design. Table 5.3 shows that the model-robust design has a higher average and minimum $\mathcal{D}$-efficiency, with smaller standard deviation. For $\boldsymbol{\Sigma}_1$, we give histograms of these efficiencies for both designs, as well.

	SSPS/SPS Model-Robust			Optimal for Quadratic		
	Mean	StDev	Min	Mean	StDev	Min
$\boldsymbol{\Sigma}_1$	0.931	0.0468	0.825	0.893	0.0726	0.740
$\boldsymbol{\Sigma}_2$	0.932	0.0482	0.825	0.894	0.0737	0.740
$\boldsymbol{\Sigma}_3$	0.932	0.0481	0.825	0.894	0.0735	0.740

Table 5.3. Comparison of model-robust design and design optimal for full quadratic model, for each of three assumed true covariance matrices, as measured by the mean $\mathcal{D}$-efficiency, standard deviation of the $\mathcal{D}$-efficiency, and the minimum $\mathcal{D}$-efficiency.

5.5.3 Mullet Meat Experiment

The final example, taken from Shah et al. [94] via Tseo et al. [101], was an experiment in which washing minced mullet flesh was investigated. The controllable factors were washing temperature (x_1), washing time (x_2), and washing ratio (x_3) and there were four responses: springiness (y_1), thiobarbituric acid number (y_2), percent cooking loss (y_3), and whiteness index (y_4). The original experiment was a central composite design with eight corner points, six axial points, and three

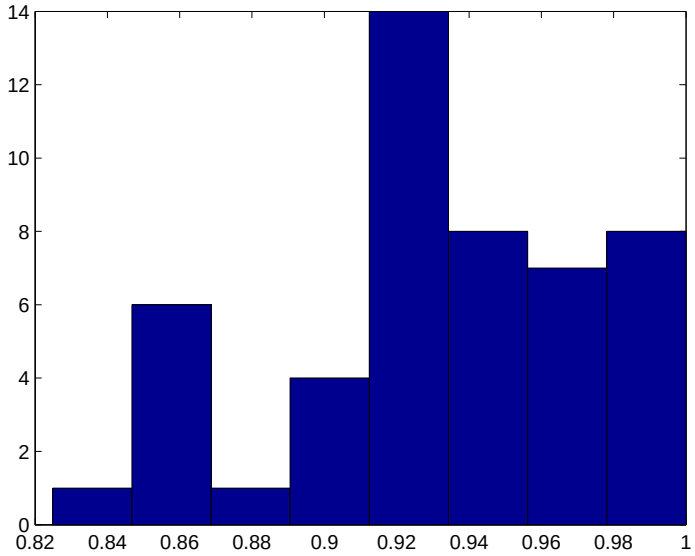

Figure 5.1. Histogram of $\mathcal{D}$-efficiencies with respect to the 49 models and $\boldsymbol{\Sigma}_1$, for the model-robust design.

center points ($N = 17$). Shah et al. [94] used an SUR model to fit the following models:

$$y_1 = 1.884634 - 0.097383x_1 - 0.103901x_1^2 \tag{5.35}$$

$$y_2 = 22.648811 + 5.614804x_1 - 0.341069x_2 + 7.830449x_1^2 + 2.681968x_1x_2 \tag{5.36}$$

$$y_3 = 18.956432 + 0.744422x_1 - 0.207512x_2 - 1.331086x_3 + 3.222674x_1^2 + 1.392487x_3^2 + 1.587398x_1x_2 + 1.804884x_1x_3 \tag{5.37}$$

$$y_4 = 51.910033 + 2.436441x_1 - 3.428739x_1^2 \tag{5.38}$$

with estimated covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} .0016525 & -.0086836 & -.0358445 & -.0804776 \\ -.0086836 & 7.5417141 & -.5390002 & 2.4998582 \\ -.0358445 & -.5390002 & 4.5640517 & 4.8343599 \\ -.0804776 & 2.4998582 & 4.8343599 & 14.2181839 \end{pmatrix}. \tag{5.39}$$

We first use the MX algorithm to produce, as a standard, the optimal design using (5.35)-(5.38) and (5.39) as the truth (Table E.9). Then, we also construct

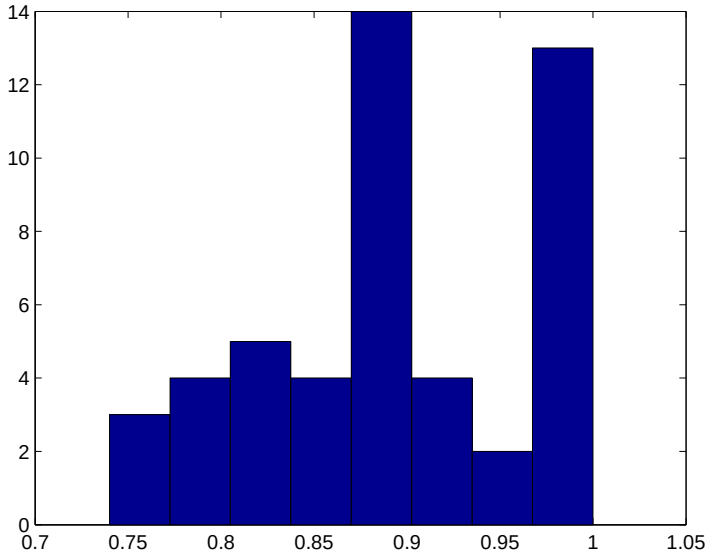

Figure 5.2. Histogram of $\mathcal{D}$-efficiencies with respect to the 49 models and $\boldsymbol{\Sigma}_1$, for the design optimal for the quadratic model alone

two model-robust designs, one using the SSPS set (Table E.10) and the other using the SPS set (Table E.11). We also include the design optimal for the quadratic model alone in Table E.12. With respect to the particular "true" model forms and covariance matrix, the design optimal for the quadratic model is 91.9% $\mathcal{D}$-efficient, while the model robust designs are only 89.2% and 88.2%, respectively. This is not surprising because the fitted model forms include quadratic terms. If the "true" models are modified such that three of the five quadratic terms are removed from the model forms in (5.35)-(5.38) (x_1^2 from y_1 and y_2 and x_3^2 from y_3), the SSPS-MR and SPS-MR are 92.5% and 95.3% efficient, while the full quadratic is 91.1%.

5.6 Discussion

In this chapter we have generalized the matrix updating formulae of Fedorov [41] to facilitate the development of an exchange algorithm to construct exact multiresponse designs. However, since neither the covariance matrix of the responses or the model forms for each response is likely to be known at the design stage, we extend the methodology and demonstrate that a model-robust approach can

be effective in constructing designs in the multiresponse context, compared to the usual practice of ignoring the multiple responses and designing for the largest assumed univariate model. Furthermore, our work confirms the general conclusion of Chang [23] which found the effect of $\mathbf{\Sigma}$ on the optimal multiresponse design to be minimal. Compared to the model response forms, the covariance matrix is inconsequential.

When large problems are coupled with a large potential model set, computational requirements could become crippling, particularly if a higher granularity is desired in the candidate list. For the small-to-moderate sized problems in this chapter, model-robust designs can be computed in a reasonable amount of time. However, to counteract debilitating computational demands for larger problems, further work might include a coordinate exchange algorithm [77] for multiresponse design, which would eliminate the need for a candidate list. For regular design regions in particular, the extension should be straightforward.

Work has been done by Bingham and Chipman [13] which takes a more sophisticated view of selecting sets of *a priori* model forms. Instead of choosing more or less *ad hoc* sets as we have in this work, these authors develop a prior distribution of possible models based upon three axiomatic principles: Effect sparsity, effect hierarchy, and effect heredity. Their approach does not eliminate the problem of exploding model size (in the end, they only handle a small subset of the models with the largest prior probabilities), but it does represent a principled approach that might be adapted to the current setting. However, we note that 1) this approach would further exaggerate the superiority of the model-robust designs, because of the effect hierarchy assumption which favors lower-order effects to higher-order ones; and 2) it would not be a straightforward application of their methodology because we consider quadratic effects and require designs for which these effects are estimable.

Chapter 6

Maximin Model-Robust Designs for Split Plot Experiments

6.1 Introduction

Split-plot experiments arise when certain factors are significantly harder and/or more expensive to change (whole plot factors) than others (subplot factors). To alleviate these difficulties, a two-step randomization scheme is employed. First, the designer specifies the number of whole plots, and randomly assigns some combination of the whole plot factors to each. Then, within each whole plot, combinations of subplot factor levels are randomly assigned to the subplots within each whole plot. Thus, a split-plot design reduces the number of required independent resets of the levels of the whole plot factors, compared to the completely randomized design (CRD), and saves the experimenter time and/or money.

For instance, suppose an experiment is to be conducted to learn about factors affecting the strength of a type of ceramic pipe. Two of the factors are concerned with the temperature of different parts of a furnace, and are difficult to change from run to run. The other two factors are the amount of binder and the grinding speed, and are easy to change. A desirable design minimizes the number of changes to the temperature factors–thus a split-plot design is appropriate. We return to this example in §6.4.1.

For most of their history, split-plot experiments have been viewed as categorical

designs; that is, designs with qualitative factors. However, the recent split-plot resurgence has viewed these as designed experiments with continuous factors and this approach has spawned relatively recent work in the optimal design of split-plot experiments. This helps to explain why there is not an asymptotic design theory for split-plot experiments, though CRDs have a thorough asymptotic development.

Optimal design in this area, led by Goos and coauthors (e.g. [49, 50, 51, 59]), utilizes the $\mathcal{D}$-optimality criterion to produce exact designs that estimate model parameters as precisely as possible. As in the case of CRDs, these optimal designs require the specification of a particular model form–an often unrealistic assumption.

There is a paucity of extant research in model-robustness for split-plot experiments. In this chapter, we provide a framework in which the model form assumption can be relaxed and an exchange algorithm used to construct model-robust split-plot designs. Instead of a single model, we allow the user to specify a set of models; instead of directly using the $\mathcal{D}$-criterion to construct the design, we use a criterion which maximizes the minimum $\mathcal{D}$-efficiency with respect to this set. A generalization allows the experimenter to specify a level of interest in each model, giving the experimenter recourse if certain models are preferred over others, and flexibility to inject prior belief into the design process. We find also, corroborating previous work, that the ratio of variance components has little impact on the optimal design.

6.2 The Split-Plot Model and Design

Here, we follow the notation of Goos and Vandebroek [49]. A split-plot experiment consists of two disparate groups of factors: the n_w difficult to change whole plot variables, denoted $\mathbf{z} = (z_1, z_2, \ldots, z_{n_w})$, and the remaining n_s easy to change subplot factors, denoted $\mathbf{x} = (x_1, x_2, \ldots, x_{n_s})$. Whole plot factor level combinations are randomly assigned to the whole plots, and then within each whole plot factor level combinations of $\mathbf{x}$ are randomly assigned. We can represent an exact

split-plot design as

$$\xi_{nb} = \begin{pmatrix} (\mathbf{z}_1, \mathbf{x}_{11}) & (\mathbf{z}_1, \mathbf{x}_{12}) & \dots & (\mathbf{z}_1, \mathbf{x}_{1u_1}) & (\mathbf{z}_2, \mathbf{x}_{21}) & \dots & (\mathbf{z}_b, \mathbf{x}_{bu_b}) \\ n_{11} & n_{12} & \dots & n_{1u_1} & n_{21} & \dots & n_{bu_b} \end{pmatrix} \tag{6.1}$$

where n is the total number of runs, b is the number of whole plots, u_i is the number of distinct points in the i^{th} whole plot, and n_{ik} is the number of runs for the k^{th} distinct point in the i^{th} whole plot.

The regression model for the j^{th} observation within the i^{th} whole plot is

$$y_{ij} = \mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{ij})\boldsymbol{\tau} + \delta_i + \epsilon_{ij} \tag{6.2}$$

where $\mathbf{f}$ is the model expansion in terms of the whole plot and subplot factors, $\boldsymbol{\tau}$ is a p-vector of model parameters, δ_i is the whole plot error term, and ϵ_{ij} is the subplot error term. The parameter vector includes p_w parameters for whole plot factors only, p_s parameters for subplot factors only, and p_{ws} parameters for whole plot-by-subplot interactions. We can write the model in matrix notation as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\tau} + \boldsymbol{\delta} + \boldsymbol{\epsilon} \tag{6.3}$$

Both δ_i and ϵ_{ij} are assumed to be normally distributed with mean 0 and variances σ_δ^2 and σ_ϵ^2, respectively. Each observation y_{ij} has two variance components, one contributed by the whole plot, σ_δ^2, and one contributed by the subplot, σ_ϵ^2. Thus, $Var(y_{ij}) = \sigma_\delta^2 + \sigma_\epsilon^2$. However, within whole plot i, observations are correlated because the whole plot is fixed. This is represented by $Cov(y_{ij}, y_{ij'}) = \sigma_\delta^2$, where $j \neq j'$. Thus, the variance-covariance matrix for whole plot i is

$$\begin{aligned} \mathbf{W}_i &= \sigma_\epsilon^2 \mathbf{I}_{s_i} + \sigma_\delta^2 \mathbf{1}_{s_i} \mathbf{1}_{s_i}' \\ &= \sigma_\epsilon^2 (\mathbf{I}_{s_i} + \eta \mathbf{1}_{s_i} \mathbf{1}_{s_i}') \end{aligned}$$

where $\eta = \sigma_\delta^2 / \sigma_\epsilon^2$, s_i is the number of runs in whole plot i, $\mathbf{I}_{s_i}$ is the identity matrix, and $\mathbf{1}_{s_i}$ is an s_i-vector of ones. Since the whole plots are assumed to be

independent of one another, the covariance matrix for (6.3) is block diagonal:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{W}_b \end{pmatrix}$$

To estimate the parameters of this model, we use Generalized Least Squares (GLS), which give $\hat{\boldsymbol{\tau}} = \left(\mathbf{X}'\mathbf{W}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{R}^{-1}\mathbf{Y}$ so that $Var(\hat{\boldsymbol{\tau}}) = \left(\mathbf{X}'\mathbf{W}^{-1}\mathbf{X}\right)^{-1}$. The information matrix, then, is

$$\mathbf{M}_{sp} = \mathbf{X}'\mathbf{W}^{-1}\mathbf{X}.$$

This expression for the information matrix can be simplified [49]; see Appendix F.1 for details. First, $\mathbf{W}_i^{-1}$ can be written as

$$\mathbf{W}_i^{-1} = \frac{1}{\sigma_\epsilon^2}\left(\mathbf{I}_{s_i} - \frac{d}{1+s_i d}\mathbf{1}_{s_i}\mathbf{1}_{s_i}'\right)$$

which leads to a convenient, updatable form of the information matrix:

$$\begin{aligned}\mathbf{M}_{sp} &= \sum_{i=1}^{w}\mathbf{X}_i'\mathbf{W}_i^{-1}\mathbf{X}_i \\ &= \frac{1}{\sigma_\epsilon^2}\left(\sum_{i=1}^{w}\sum_{j=1}^{s_i}\mathbf{f}(\mathbf{z}_i,\mathbf{x}_{ij})\mathbf{f}'(\mathbf{z}_i,\mathbf{x}_{ij}) - \sum_{i=1}^{w}\frac{d}{1+s_i d}(\mathbf{X}_i'\mathbf{1}_{s_i})(\mathbf{X}_i'\mathbf{1}_{s_i})'\right).\end{aligned}$$

In this chapter, we are primarily interested in $\mathcal{D}$-optimality, which is given by

$$|\mathbf{M}_{sp}| = \left|\mathbf{X}'\mathbf{W}^{-1}\mathbf{X}\right|. \tag{6.4}$$

We also give the prediction variance for design ξ and point $(\mathbf{z}_i, \mathbf{x}_{ij})$, because our algorithm uses it to construct an initial design:

$$V_{sp}(\mathbf{z}_i,\mathbf{x}_{ij},\xi) = f'(\mathbf{z}_i,\mathbf{x}_{ij})\mathbf{M}_{sp}^{-1}(\xi)f(\mathbf{z}_i,\mathbf{x}_{ij}). \tag{6.5}$$

6.3 Model-Robust, Maximin Split-Plot Design Algorithm

In this section, we first review the basic idea of the exchange algorithm for $\mathcal{D}$-optimal split-plot designs, given by Goos and Vandebroek [50]. We then give a generalization of the algorithm which is robust to a set of user-specified models, achieving this robustness by seeking the design which maximizes the minimum $\mathcal{D}$-efficiency with respect to the model set.

6.3.1 $\mathcal{D}$-Optimal Split-Plot Exchange Algorithm

The algorithm upon which ours is based [50] requires the specification of the number of whole plots as well as the number of subplots per whole plot. The same authors [51] also give an algorithm for which these requirements are relaxed. However, when split-plot experiments are called for, it is most often because certain factors are difficult or expensive to change. Thus, at the very least an algorithm is desired for which the number of whole plots can be constrained, and for our purposes we will consider only the case in which the number and size of the whole plots are explicitly specified.

Thus, the $\mathcal{D}$-optimal split-plot algorithm requires the construction of a candidate list, the number and size of the whole plots, the ratio of variance components, $\eta = \sigma_\delta^2/\sigma_\epsilon^2$, and a given, assumed model form. The algorithm first randomly specifies an initial design, and then iteratively improves upon the design using three exchange mechanisms: 1) Consider swaps of current design points with candidate points having the same whole plot factor settings; 2) Consider swaps of current design points with other design points having the same whole plot factor settings; and 3) Consider swaps of current whole plot factor settings with other possible whole plot factor settings. For each mechanism in turn, all possible exchanges are considered for each appropriate design point (or whole plot). The algorithm iterates until convergence. Since this is not a concave optimization problem, the algorithm is not guaranteed to converge to the globally optimal solution. Consequently, numerous algorithm tries, each with a randomly generated initial design, are used to overcome possibly local optima (we use 50 tries, and Goos and Vande-

broek [50] give evidence that this is adequate to produce an optimal design with high probability). The model-robust algorithm developed below is similar to this procedure in structure and approach.

Many updates of the determinant of the split-plot information matrix are required as exchanges are considered. Arnouts and Goos [5] developed updating formulae that speed the computation of these updates. We reproduce them in Appendix F.2, and refer to them in our algorithm as they are used.

6.3.2 Maximin Split-Plot Exchange Algorithm

In this section we describe a model-robust procedure for the split-plot optimal design problem which parallels that in Chapter 4 for the completely randomized setting. This is based upon $\mathcal{F}$, a user-specified set of r models. Our goal is to find a design that is maximin optimal with respect to a generalized measure of $\mathcal{D}$-efficiency, $G_f(\xi_{nb})$, defined as

$$G_f(\xi_{nb}) = \frac{D_f(\xi_{nb})}{v_f} \tag{6.6}$$

where v_f represents the model interest (weight) element for model f and $D_f(\xi_{nb})$ is the $\mathcal{D}$-efficiency for a design ξ_{nb} with respect to model f, calculated as

$$D_f(\xi_{nb}) = \left(\frac{|\mathbf{M}_{sp,f}(\xi_{nb})|}{|\mathbf{M}_{sp,f}(\xi^*_{nb})|} \right)^{1/p} \tag{6.7}$$

where ξ^*_{nb} is the optimal design for model f. The model interest vector is $v = (v_1, \ldots, v_r) \in (0, 1]$ and represents the experimenter's level of interest in each of the r specified models. By convention, we scale this vector so that the *a priori* most likely model(s) has (have) a model interest level of 1. (Note that the user could renormalize these model interest weights as prior probabilities.) The algorithm gives a lower priority to models with smaller interest levels and in fact gives precedence to each model commensurate with its model interest level. This hierarchy is sometimes reflected in the efficiencies of the design with respect to the models in $\mathcal{F}$ (i.e. if $\mathcal{F} = (f_1, f_2)$ and $v = (1, .8)$, the resulting design may have a $\mathcal{D}$-efficiency with respect to the second model form that is about 80% that of the

first). Due to the discrete nature of the exact split-plot design problem, as well as complications inherent in model sets with more than two elements (see Chapter 4), the efficiency precedence may be obscured or nonexistent, as demonstrated in our examples.

This model-robust procedure is computationally more demanding than the original split-plot exchange algorithm. First, there are r models to consider instead of just one. Secondly, it requires that the optimal design for each of the models individually be found. There are fewer commercially available optimal design algorithms for split-plot experiments than for CRDs, though they do exist (e.g. SAS JMP). However, if the model-robust algorithm is implemented the optimal design split-plot algorithm is a special case when $\mathcal{F}$ consists of just a single model, and thus could be used to furnish the individually optimal designs.

To describe our algorithm, we adopt the same basic structure (though different notation) as Goos and Vandebroek [50]: There is assumed to be c candidate points, the set of which is denoted by C. Similarly, there are b whole plots and the set of them is B, and the k_i design points in the i^{th} whole plot for a design ξ_{nb} are denoted by H_i. Thus, the total number of design points is $n = \sum_{i=1}^{b} k_i$. The whole plot factor level combinations for the i^{th} whole plot is given by $\mathbf{z}_i$, and the set of candidate points with the same whole plot factor combinations as the i^{th} whole plot is given by C_i. The generalized $\mathcal{D}$-efficiency for model f is G_f, as in (6.6). The best minimum generalized efficiency found at any given point in the algorithm is represented as $G^*_{f_{min}}$. Also, we let the set of possible whole plot factor levels be P. The number of algorithm tries is given by t and the current try is t_c. See §6.4.1 for an illustration of this notation.

For the algorithm, we must specify 1) a candidate list, C; 2) the number of whole plots, b; 3) the size of each whole plot, $k_1, k_2, \ldots, k_b$; 4) the ratio of variance components, η; 5) the set of possible model forms, $\mathcal{F}$; 6) a model interest vector, $\mathbf{v}$, specifying the relative preference for each model (if no preferences, enter $v_f = 1$ for all $f \in \mathcal{F}$; and 7) for each model $f \in \mathcal{F}$, $\left|\mathbf{M}_{sp,f}(\xi^*_f)\right|$, where $\xi^*_{nb,f}$ is the optimal design for model f alone. One consequence of our set of models approach is that the number of whole plots, b, must be greater than $p_{w,f_{max}}$, the number of parameters with only whole plot terms, for the largest model. The algorithm is as follows:

1. Set $t_c = 1$.

2. Determine $p_{w,f}$, the number of coefficients for whole plot factors only, for all $f \in \mathcal{F}$.

3. Determine $p_{s,f}$, the number of coefficients for subplot factors only, for all $f \in \mathcal{F}$.

4. Set $\mathbf{M}_{sp,f} = \omega\mathbf{I}$ (where ω is a small constant set at 0.01) for all $f \in \mathcal{F}$; set $H_i = \emptyset$.

5. Construct initial design

   (a) Randomly assign $p_{w,f_{max}}$ unique whole-plot factor settings to $p_{w,f_{max}}$ whole plots, where f_{max} is the model with the most parameters (to ensure estimability of whole plot coefficients for largest model).

   (b) Randomly assign $b - p_{w,f_{max}}$ levels of the whole-plot factors to the rest of the whole plots.

   (c) Randomly choose u ($1 \leq u \leq p_{f_{max}}$), where $p_{f_{max}}$ is the number of parameters for model f_{max}.

   (d) Do u times (this step gives initial design a measure of randomness):

      i. Randomly select $i \in B$ (select a whole plot at random)

      ii. Randomly select $j \in C_i$ (select a candidate point with the i^{th} setting of whole plot, at random)

      iii. If $\#H_i < k_i$, then $H_i = H_i \cup j$; otherwise, go back to step i.

      iv. Update $\mathbf{M}_{sp,f}^{-1}$ for all $f \in \mathcal{F}$ via (F.1).

   (e) Do $n - u$ times (this step attempts to give initial design a measure of quality with respect to all models $f \in \mathcal{F}$):

      i. Randomly select $f \in \mathcal{F}$.

      ii. Set $l = 1$.

      iii. For model f, determine $j \in C$ with the l^{th} biggest prediction variance via (6.5).

      iv. Find i, where $i \in B$, $j \in C_i$, and $\#H_i < k_i$ (find a nonfull whole plot into which j can be inserted). If no such i exists, set $l = l + 1$ and return to step iii.

v. $H_i = H_i \cup j$.

vi. Update $\mathbf{M}_{sp,f}^{-1}$ for all $f \in \mathcal{F}$ via (F.1).

6. Compute $\mathbf{M}_{sp,f}^{-1}$ for all $f \in \mathcal{F}$ and $G_{f_{min}}$ (the smallest generalized efficiency). If $G_{f_{min}} = 0$, go back to step 4. Otherwise, continue.

7. Set $\nu = 0$.

8. Evaluate design point exchanges (swapping design points with candidate points, where whole plot factors settings are the same):

   (a) Set $\gamma = G_{f_{min}} = \min_{f \in \mathcal{F}} G_f$.

   (b) $\forall i \in B$, $\forall j \in H_i$, $\forall k \in C_i$, $j \neq k$:

      i. Determine the effect $\delta_{jk,f_{min}}^{i} = |\mathbf{M}_{jk,sp,f_{min}}^{i}|/|\mathbf{M}_{sp,f_{min}}|$ of exchanging, in the i^{th} whole plot, points j and k, using (F.4).

      ii. If $\delta_{jk,f_{min}}^{i} > 1$, calculate $\delta_{jk,f}^{i} = |\mathbf{M}_{jk,sp,f_{min}}^{i}|/|\mathbf{M}_{sp,f_{min}}|$ for all $f \neq f_{min}$ and update $|\mathbf{M}_{sp,f}|$ for all $f \in \mathcal{F}$ via (F.4). Then, compute $G_{jk,f}^{i}$ for all $f \in \mathcal{F}$.

      iii. If $\min_{f \in \mathcal{F}} G_{jk,f}^{i} > \gamma$, then $\gamma = \min_{f \in \mathcal{F}} G_{jk,f}^{i}$ and store i, j, and k.

9. If $\gamma > G_{f_{min}} + \epsilon$, then go to step 10; otherwise, go to step 11.

10. Perform the best exchange:

    (a) $H_i = H_i \setminus j \cup k$.

    (b) Update $\mathbf{M}_{sp,f}^{-1}$ and $|\mathbf{M}_{sp,f}|$ for all $f \in \mathcal{F}$ via (F.5) and (F.4), respectively.

    (c) Set $\nu = 1$.

11. Evaluate interchanges of points within whole plots with the same factor levels:

    (a) Set $\gamma = G_{f_{min}} = \min_{f \in \mathcal{F}} G_f$.

    (b) $\forall i, j \in B$, $i < j$, $\mathbf{z}_i = \mathbf{z}_j$, $\forall k \in H_i$, $\forall l \in H_j$, $k \neq l$:

       i. Determine the effect $\delta_{ik,f_{min}}^{jl} = |\mathbf{M}_{ik,sp,f_{min}}^{jl}|/|\mathbf{M}_{sp,f_{min}}|$ of moving k to whole plot j (from whole plot i) and l to whole plot i (from whole plot j), via (F.7).

ii. If $\delta^{jl}_{ij,f_{min}} > 1$, calculate $\delta^{jl}_{ik,f} = |\mathbf{M}^{jl}_{ik,sp,f_{min}}|/|\mathbf{M}_{sp,f_{min}}|$ for all $f \neq f_{min}$ and update $|\mathbf{M}_{sp,f}|$ for all $f \in \mathcal{F}$ via (F.7). Then, compute $G^{jl}_{ik,f}$ for all $f \in \mathcal{F}$.

iii. If $\min_{f\in\mathcal{F}} G^{jl}_{ik,f} > \gamma$, then $\gamma = \min_{f\in\mathcal{F}} G^{jl}_{ik,f}$ and store i, j, k, and l.

12. If $\gamma > G_{f_{min}} + \epsilon$, go to step 13; otherwise, go to step 14.

13. Perform the best interchange:

    (a) $H_i = H_i \setminus k \cup l$.

    (b) $H_j = H_j \setminus l \cup k$.

    (c) Recalculate $\mathbf{M}^{-1}_{sp,f}$ and update $|\mathbf{M}_{sp,f}|$ via (F.7).

    (d) Set $\eta = 1$.

14. Evaluate exchanges of whole-plot factor settings:

    (a) Set $\gamma = G_{f_{min}} = \min_{f\in\mathcal{F}} G_f$.

    (b) $\forall i \in B$, $\forall j \in P$, $\mathbf{z}_i \neq \mathbf{z}_j$:

        i. Determine the effect $\delta_{ij,f_{min}} = |\mathbf{M}_{ij,sp,f_{min}}|/|\mathbf{M}_{sp,f_{min}}|$ of exchanging $\mathbf{z}_i$ by $\mathbf{z}_j$ in the i^{th} whole plot, via (F.10).

        ii. If $\delta_{ij,f_{min}} > 1$, calculate $\delta_{ij,f} = |\mathbf{M}_{ij,sp,f_{min}}|/|\mathbf{M}_{sp,f_{min}}|$ for all $f \neq f_{min}$ and update $|\mathbf{M}_{sp,f}|$ for all $f \in \mathcal{F}$ via (F.10). Then, compute $G_{ij,f}$ for all $f \in \mathcal{F}$.

        iii. If $\min_{f\in\mathcal{F}} G_{ij,f} > \gamma$, then $\gamma = \min_{f\in\mathcal{F}} G_{ij,f}$ and store i and j.

15. If $\gamma > G_{f_{min}} + \epsilon$, go to step 16; otherwise, go to step 17.

16. Perform best exchange:

    (a) Update H_i and C_i.

    (b) Update $\mathbf{M}^{-1}_{sp,f}$ and $|\mathbf{M}_{sp,f}|$ for all $f \in \mathcal{F}$ via (F.11) and (F.10), respectively.

    (c) Set $\eta = 1$.

17. If $\eta = 1$, go to step 7.

18. If $G_{f_{min}} = \min_{f \in \mathcal{F}} G_f > G^*_{f_{min}}$, then $G^*_{f_{min}} = G_{f_{min}}$; also, update the design ξ_{nb}.

19. If $t_c < t$, then $t_c = t_c + 1$, and go back to step 4; otherwise, STOP.

Notes: 1) This algorithm, including Step 5 which generates the initial design, generalizes Goos and Vandebroek [50] but follows its basic outline. 2) In 13(c), we recalculate $\mathbf{M}^{-1}$ by directly taking its inverse, because of numerical issues associated with the updated formula in (F.8). This should have a minimal effect on the speed of the algorithm because it occurs only once per iteration. 3) In steps 9, 12, and 15, we have changed the original algorithm of Goos and Vandebroek [50] slightly by requiring that, for any considered exchange/interchange, the increase in the minimum efficiency is greater than $\epsilon = 0.0001$ (instead of 0, as it was originally), to encourage algorithmic stability. 4) Note that the ϵ here is not a convergence parameter as in the previous chapters. For a particular algorithm try, t_c, the procedure terminates when an iteration results in no exchanges.

6.4 Examples

In this section we illustrate our model-robust procedure using two examples from the literature. The first involves two whole plot factors and two subplot factors. The second includes two process (whole plot) factors and three mixture (subplot) factors. In what follows, we reference and discuss a large number of designs. In the interest of space, they can be found in the supplementary material accompanying this article. All designs referred to in the following examples are given in Appendix H. Matlab® (Version 7.8) code can be found at `http://www.stat.psu.edu/~jlr/pub/Smucker/`.

6.4.1 Strength of Ceramic Pipe Experiment

In a reprise of the example at the outset, the first example is taken from Vining et al. [103] and concerns an experiment on the strength of ceramic pipe. To illustrate and clarify the notation for the algorithm in the previous section, we will include that notation in the description of this example. There were four factors,

two hard-to-change (zone 1 temperature of furnace, z_1; zone 2 temperature of furnace, z_2) and two easy-to-change (amount of binder, x_1; grinding speed, x_2). The experiment used $b = 12$ whole plots each of size 4 ($k_i = 4$ for $i = 1, \ldots, 12$ so that $n = \sum_{i=1}^{12} k_i = 48$). A given design for this experiment is denoted by ξ_{nb} and the part of the design associated with the i^{th} whole plot is H_i.

After running the experiment, it was found that $\hat{\eta} = \hat{\sigma}_\delta^2/\hat{\sigma}_\epsilon^2 = .52828/.09348 = 5.65$. The model to be fit was full quadratic in all factors:

$$f^{'}(\mathbf{z}, \mathbf{x}) = (1, z_1, z_2, z_1z_2, z_1^2, z_2^2, x_1, x_2, x_1x_2, z_1x_1, z_1x_2, z_2x_1, z_2x_2, x_1^2, x_2^2) \quad (6.8)$$

with $\mathcal{X} = \{\mathbf{z} = (z_1, z_2), \mathbf{x} = (x_1, x_2) : -1 \leq \mathbf{z}, \mathbf{x} \leq 1\}$.

Vining et al. [103] used a face-centered central composite design (CCD) modified to accommodate the split-plot structure. Another design approach would be to use the $\mathcal{D}$-optimal exchange algorithm of Goos and Vandebroek [50] (or, alternatively, the candidate list-free exchange algorithm of, [59]) assuming (6.8) to be the true model. In retrospect, this model seemed to be a good approximation of the true model [103], but if the experimenter were not confident about the form of the model *a priori*, it would be advantageous to choose a set of models and construct a design robust for all of them. For instance, we might choose $\mathcal{F}_1 = \{f_i^{'}(\mathbf{z}, \mathbf{x})\boldsymbol{\tau}_i, 1 \leq i \leq 3; \mathbf{z}, \mathbf{x} \in \mathcal{X}\}$ with

$$f_1^{'}(\mathbf{z}, \mathbf{x}) = (1, z_1, z_2, x_1, x_2) \quad (6.9)$$

$$f_2^{'}(\mathbf{z}, \mathbf{x}) = (f_1^{'}, z_1z_2, x_1x_2, z_1x_1, z_1x_2, z_2x_1, z_2x_2) \quad (6.10)$$

$$f_3^{'}(\mathbf{z}, \mathbf{x}) = (f_2^{'}, z_1^2, z_2^2, x_1^2, x_2^2). \quad (6.11)$$

A weakness of optimal design—addressed for exact designs in an ad-hoc way in the split-plot case by Goos and Donev [47] and in a more systematic way in the CRD case by, for instance, DuMouchel and Jones [37] and Goos et al. [48]—is that it devotes all of the available experimental resources to increase the efficiency of the design and none to allow lack-of-fit testing. To overcome this shortcoming, while remaining in the optimal design paradigm and thus enjoying a great deal of efficiency, a strategy would be to construct a design somewhat less efficient with respect to models considered likely, while affording the experimenter estimability

for a larger model. This approach can serve as a hedge against a larger-than-expected model, though it does not necessarily allow traditional lack-of-fit testing. To illustrate, we take $\mathcal{F}_2 = \{\mathcal{F}_1, f_4'\}$ with

$$\begin{aligned} f_4 = (f_3, & x_1x_2x_3, x_1x_2x_4, x_1x_3x_4, x_2x_3x_4, \\ & x_1^2x_2, x_1^2x_3, x_1^2x_4, x_2^2x_1, x_2^2x_3, x_2^2x_4, x_3^2x_1, x_3^2x_2, x_3^2x_4, x_4^2x_1, x_4^2x_2, x_4^2x_3, \\ & x_1^3, x_2^3, x_3^3, x_4^3), \end{aligned} \tag{6.12}$$

the full cubic model.

In Table 6.1, we consider seven designs in all, evaluating them by calculating their $\mathcal{D}$-efficiencies with respect to each of the possible models we have discussed. The first four are based on the methodology developed in this work, and the last three are included for comparative purposes. The maximin designs, as well as the optimal designs for models (6.11) and (6.12), have been calculated under the assumption that the variance components are the same (i.e. no prior knowledge of η) and the best of $t = 50$ algorithm tries taken, each starting with a randomly chosen design, using a 5^4 candidate list–all combinations of $(-1, -0.5, 0, 0.5, 1)$ for each of the four factors. Thus, $c = 5^4 = 625$ and C is the set of these points, while C_i is the portion of these points whose whole plot factor combinations match that of the i^{th} whole plot. This also implies that P, the possible factor level combinations of the two whole plot factors, consists of 5^2 points (all combinations of $(-1, -0.5, 0, 0.5, 1)$).

	$\eta = 1$				$\eta = 5.65$			
Design	(6.9)	(6.10)	(6.11)	(6.12)	(6.9)	(6.10)	(6.11)	(6.12)
$\mathcal{F}_1$-Maximin	.933	.879	.879	0	.933	.878	.879	0
(.9,.9,1)-$\mathcal{F}_1$-Maximin	.905	.838	.931	0	.904	.837	.931	0
$\mathcal{F}_2$-Maximin	.876	.778	.874	.777	.876	.777	.872	.769
(.8,.8,1,.5)-$\mathcal{F}_2$-Maximin	.852	.754	.942	.708	.850	.752	.942	.707
Optimal for (6.11)	.850	.752	1	0	.847	.750	1.000	0
Optimal for (6.12)	.759	.600	.882	1	.758	.599	.882	.999
Vining et al. [103]	.512	.405	.582	0	.512	.405	.527	0

Table 6.1. For example in §6.4.1, $\mathcal{D}$-efficiencies for various designs with respect to models (6.9)-(6.12). We give efficiencies assuming $\eta = 1$ as well as $\eta = 5.65$.

In Table 6.1, we calculate the efficiencies both under the assumption that $\eta = 1$ and $\eta = 5.65$ (estimated by Vining et al. [103]). It can be seen that η has almost no effect on the efficiencies of the designs. The maximin design with respect to $\mathcal{F}_1$ is relatively efficient for the three models, but cannot estimate the larger cubic model, (6.12). The (.9,.9,1)-$\mathcal{F}_1$-maximin design is appropriate if more emphasis is desired for the quadratic model, (6.11), than the smaller ones. In comparison to the design optimal for the quadratic model alone, this design gives up some precision for the quadratic model but is more efficient for the two smaller models.

However, if some protection is desired for a lurking, higher-order model we expand to the set $\mathcal{F}_2$. The $\mathcal{F}_2$-maximin design shows, unsurprisingly, that adding the full cubic model dampens the ability to estimate the other models. However, if we take $v = (.8, .8, 1, .5)$ and compare the resulting maximin design to the optimal design for (6.11), this design estimates the smaller models as well, gives up just 6% in efficiency for the quadratic model and simultaneously allows for 71%-efficient estimation of the full cubic model.

In terms of $\mathcal{D}$-efficiency, the design proposed by Vining et al. [103] (a CCD in the whole plot factors with various subplot designs at each whole plot) is not competitive, but this does not necessarily preclude it from the consideration of the experimenter. It has other qualities—for instance, it allows for a model-independent estimate of the variance components and a replication-based lack-of-fit test—that recommend it to an experimenter that wishes to remain within the structure of classic designs (i.e. CCDs).

6.4.2 Vinyl-Thickness Experiment

We next consider an example given in Cornell [29] and used later by Kowalski et al. [68] and Goos and Donev [47]. This experiment measured the thickness of manufactured vinyl for automobile seat covers and included two process factors (extrusion rate, z_1, and drying temperature, z_2) and three mixture factors (plasticizers, x_1, x_2, and x_3). Complete randomization was unwieldy, so the process variables were treated as whole plot factors, and the mixture variables as subplot factors. Though the original design was run in eight whole plots each of size six, Kowalski et al. [68] and Goos and Donev [47] used only 28 total runs, with 7 whole

plots each of size 4. We will adopt this approach as well.

The model used by Kowalski et al. [68] and Goos and Donev [47] was as in (6.2) with $\mathbf{z}, \mathbf{x} \in \mathcal{X}$ where

$$f^{'}(\mathbf{z}, \mathbf{x}) = (x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, z_1x_1, z_2x_1, z_1x_2, z_2x_2, z_1x_3, z_2x_3, z_1z_2) \tag{6.13}$$

and

$$\mathcal{X} = \left\{ \mathbf{z} = (z_1, z_2), \mathbf{x} = (x_1, x_2, x_3) : -1 \leq \mathbf{z}, \mathbf{x} \leq 1; \sum_{i=1}^{3} x_i = 1 \right\}.$$

Since this experiment involved both mixture and process variables, the model is a combination of an interaction model for the process factors and a second-order Sheffé polynomial model for the mixture factors (see [68]). The designs of [68] and [47] assumed the model in (6.13) but in the analysis done by Kowalski et al. [68] it was concluded that few of the terms were significant and thus the assumed model was overspecified. This was not, of course, known before the experiment was conducted, and illustrates why it might be desirable to use a design that accounts for more than just a single, assumed model.

To demonstrate the flexibility of our algorithm, we will consider two experimental scenarios. The first assumes that there is a high degree of confidence that the true model includes some combination of terms in the original fitted model. In this case, it would make sense to choose $\mathcal{F}_1 = \{f_{1i}^{'}(\mathbf{z}, \mathbf{x})\boldsymbol{\tau}_{1i}, 1 \leq i \leq 6; \mathbf{z}, \mathbf{x} \in \mathcal{X}\}$ with

$$f_{11}^{'}(\mathbf{z}, \mathbf{x}) = (x_1, x_2, x_3) \tag{6.14}$$

$$f_{12}^{'}(\mathbf{z}, \mathbf{x}) = (f_{11}^{'}, x_1x_2, x_1x_3, x_2x_3) \tag{6.15}$$

$$f_{13}^{'}(\mathbf{z}, \mathbf{x}) = (f_{11}^{'}, z_1x_1, z_2x_1, z_1x_2, z_2x_2, z_1x_3, z_2x_3) \tag{6.16}$$

$$f_{14}^{'}(\mathbf{z}, \mathbf{x}) = (f_{13}^{'}, z_1z_2) \tag{6.17}$$

$$f_{15}^{'}(\mathbf{z}, \mathbf{x}) = (f_{12}^{'}, z_1x_1, z_2x_1, z_1x_2, z_2x_2, z_1x_3, z_2x_3) \tag{6.18}$$

$$f_{16}^{'}(\mathbf{z}, \mathbf{x}) = (f_{15}^{'}, z_1z_2) \tag{6.19}$$

as a set of possible models, the largest of which is equivalent to (6.13). These designs have no estimation abilities for models with higher order terms, so if the experimenter wishes to guard against this possibility, consider augmenting $\mathcal{F}_1$ with

several such models: $\mathcal{F}_2 = (\mathcal{F}_1, \{f_{2i}^{'}(\mathbf{z},\mathbf{x})\boldsymbol{\tau}_{2i}, 1 \leq i \leq 6; \mathbf{z},\mathbf{x} \in \mathcal{X}\}$ where

$$f_{21}^{'}(\mathbf{z},\mathbf{x}) = (f_{12}^{'}, x_1x_2x_3) \tag{6.20}$$
$$f_{22}^{'}(\mathbf{z},\mathbf{x}) = (f_{14}^{'}, z_1^2, z_2^2) \tag{6.21}$$
$$f_{23}^{'}(\mathbf{z},\mathbf{x}) = (f_{15}^{'}, x_1x_2x_3) \tag{6.22}$$
$$f_{24}^{'}(\mathbf{z},\mathbf{x}) = (f_{15}^{'}, z_1z_2, x_1x_2x_3) \tag{6.23}$$
$$f_{25}^{'}(\mathbf{z},\mathbf{x}) = (f_{16}^{'}, z_1^2, z_2^2) \tag{6.24}$$
$$f_{26}^{'}(\mathbf{z},\mathbf{x}) = (f_{24}^{'}, z_1^2, z_2^2) \tag{6.25}$$

In all the maximin (and individually optimal) designs for this example, we use 50 algorithm tries with a 90-element candidate list (a 3^2 list for the whole plot variables crossed with a 10-element set of mixture points: $(1,0,0)$, $(0,1,0)$, $(0,0,1)$, $(1/2,1/2,0)$, $(1/2,0,1/2)$, $(0,1/2,1/2)$, $(1/3,1/3,1/3)$, $(2/3,1/6,1/6)$, $(1/6,2/3,1/6)$, and $(1/6,1/6,2/3)$). We have assumed $\eta = 1$ throughout.

In Table 6.2 we show the efficiencies of the $\mathcal{F}_1$-maximin and $(.7,.9,1,1,1,1)$-$\mathcal{F}_1$-maximin designs with respect to models (6.14)-(6.25). We also give a design which explicitly accounts for all models in $\mathcal{F}_2$, the $(1,1,1,1,1,1,.5,.5,.5,.5,.5,.5)$-$\mathcal{F}_2$-maximin design. Compared to the design optimal for the original assumed model,(6.13), the $\mathcal{F}_1$-Maximin design successfully raises the lowest of the efficiencies for the models in $\mathcal{F}_1$. The second design represents the situation in which the experimenter cares most about estimating models (6.16)-(6.19).

The third design gives up estimability for models (6.14)-(6.19) but allows the estimability of the models (6.20)-(6.25). These latter models have model interest vector elements of 0.5, so we might expect the efficiencies of these models to be lower, say about 50% of the efficiencies for those models with model interest vector elements of 1. This is not the case, however, and we conjecture that in the asymptotic case (i.e. when both the number and size of the whole plots tends to infinity), and with a larger candidate list, the efficiencies would be more suggestive of the differences in the model interest vector. Even with model interest vector elements for the last six models of 0.3 or 0.15 instead of 0.5, designs with similar efficiencies resulted. See the last section for more discussion of this point.

When the experimenter believes that a model more complex than (6.13) is some-

what likely, designs giving more weight to the higher order models in $\mathcal{F}_2$ are appropriate. In Table 6.3 we include the $\mathcal{F}_2$-maximin design, as well as one in which the larger models are downweighted slightly, with $\mathbf{v} = (1, 1, 1, 1, 1, 1, .9, .9, .9, .9, .9, .9)$. To demonstrate that quality designs can be obtained using a well-chosen subset of $\mathcal{F}_2$, we include two designs based upon

$$\mathcal{F}_3 = \{f_{11}^{'}(\mathbf{z}, \mathbf{x})\boldsymbol{\tau}_{11}, f_{16}^{'}(\mathbf{z}, \mathbf{x})\boldsymbol{\tau}_{16}, f_{26}^{'}(\mathbf{z}, \mathbf{x})\boldsymbol{\tau}_{26}\}, \tag{6.26}$$

as well. As a benchmark, we compare these designs with the design optimal for (6.25).

The design optimal for (6.25) does not estimate models (6.22) or (6.23) precisely. All of the maximin designs do much better for these models, and it underscores the general point that the maximin designs allow control of the worst-case much better than using a design optimal for a single model. Furthermore, the $(.7, 1, .9)$-$\mathcal{F}_3$-Maximin design dominates the individually optimal design for 10 of 12 models (the $\mathcal{F}_2$-Maximin design is better for 9 of 12, whereas the other two designs only 6 of 12).

Design	(6.14)	(6.15)	(6.16)	(6.17)	(6.18)	(6.19)	(6.20)	(6.21)	(6.22)	(6.23)	(6.24)	(6.25)
$\mathcal{F}_1$-Maximin	.871	.896	.872	.885	.973	.975	0	0	0	0	0	0
$(.7, .9, 1, 1, 1, 1)$-$\mathcal{F}_1$-Maximin	.907	.814	.915	.923	.967	.969	0	0	0	0	.012	0
$(1, 1, 1, 1, 1, 1, .5, .5, .5, .5, .5, .5)$-$\mathcal{F}_2$-Maximin	.888	.823	.823	.812	.904	.888	.626	.923	.806	.799	.956	.854
Optimal for (6.13)	.831	.958	.831	.847	1	1	0	0	0	0	0	0
Kowalski et al. [68]	.430	.555	.354	.375	.475	.481	.593	0	.520	.511	0	0

Table 6.2. For example in §6.4.2, designs focused on smaller models, with $\mathcal{D}$-efficiencies for models (6.14)-(6.25), assuming $\eta = 1$

Design	(6.14)	(6.15)	(6.16)	(6.17)	(6.18)	(6.19)	(6.20)	(6.21)	(6.22)	(6.23)	(6.24)	(6.25)
$\mathcal{F}_2$-Maximin	.866	.874	.801	.792	.919	.901	.798	.905	.908	.894	.969	.942
$(1, 1, 1, 1, 1, 1, .9, .9, .9, .9, .9, .9)$-$\mathcal{F}_2$-Maximin	.882	.833	.817	.807	.899	.882	.763	.918	.887	.873	.952	.926
$\mathcal{F}_3$-Maximin	.887	.834	.815	.805	.905	.889	.771	.916	.895	.880	.956	.932
$(.7, 1, .9)$-$\mathcal{F}_3$-Maximin	.822	.948	.767	.762	.937	.918	.859	.875	.928	.910	.986	.958
Optimal for (6.25)	.763	.904	.662	.658	.838	.819	.954	.854	.906	.882	.960	1

Table 6.3. For example §6.4.2, designs focusing on both small and large models, with $\mathcal{D}$-efficiencies for models (6.14)-(6.25), assuming $\eta = 1$

6.5 Discussion

In this chapter we have presented a generalization of the split-plot exchange algorithm of Goos and Vandebroek [50] which relaxes the model form assumption inherent in optimal design by designing with respect to a set of specified possible models. Our algorithm tries to choose the design which maximizes the smallest $\mathcal{D}$-efficiency with respect to the specified models, robustifying the experiment to the models in the set. This approach requires that the optimal designs for each of the individual models in the set be found, but once these are in hand the experimenter is afforded much more flexibility because they can specify a model interest vector that quantifies the relative interest in each model.

In Chapter 4, we have used this criterion to construct model-robust designs in the CRD case. There, the efficiencies more faithfully reflect the model interest vector than do the maximin designs in this chapter. In fact, several results were proven in the CRD case regarding the balance of these designs in terms of their $\mathcal{D}$-efficiencies for a given set of models (e.g. for a given set of models, the maximin design will include at least two whose generalized $\mathcal{D}$-efficiencies are the same). However, these were results assuming infinite run sizes, drawing on classical asymptotic design theory. The same basic asymptotic development does not exist, to our knowledge, for restricted randomization designs; further research is necessary. We conjecture that these "balance" type results have split-plot analogs.

However, even if our conjecture is correct, for finite run sizes such balance may not be evident, due to the typically small number and size of the whole plots in these experiments and the lack of a dense enough candidate list. This latter difficulty could be mitigated somewhat by generalizing the algorithm of Jones and Goos [59], which requires no candidate list. The original algorithm was limited in that it could not work for constrained design spaces, but recent work reports that it is possible [86, 59].

Within the "set of models" framework, other criteria could have been used. For instance, several authors [71, 27, 54, 95] use something akin to $\prod_{i=1}^{m} |\mathbf{M}_i(\xi_n)|$ as a model-robust criterion. This obviates the need for first obtaining the optimal designs for each of the individual models. The downside is the lack of flexibility for the experimenter in terms of the efficiencies with respect to each model: If

the design favors one model over another (say, by endowing the former with an efficiency very close to 1 and the latter with a much lower efficiency), the only recourse might be to try weighting the criterion: $\prod_{i=1}^{m} |\mathbf{M}_i(\xi_n)|^{a_i}$. However, in the CRD case, this proved ineffective. Overall, with the ever-increasing computational power at hand, we feel that the extra work is worth the additional flexibility afforded by the maximin criterion.

Extensive simulations have been done (e.g. [49]) demonstrating that the ratio of variance components, η, has a minimal effect on the optimal designs. Our work here confirms that, and is shown in particular in Table 6.1. The second example, although we do not show the efficiencies under the estimated η, gives similar results. In short, very little information is lost by a lack of *a priori* knowledge of η, particularly when compared to the precision lost when an incorrect model is assumed. Though we do not understand precisely why this is the case, it comports with other research (Chapter 5, [23]) in which the correlated, multiple responses seem to have little effect on experimental design.

Chapter 7

Contributions and Future Work

In this dissertation, a practical model-robust framework for exact experimental design is presented and implemented via generalizations of an existing exchange algorithm. We consider several experimental situations, including two model-robust procedures for the univariate and completely randomized design scenario, as well as model-robust procedures to account for multiple responses and split-plot experiments. In this chapter, we review the contributions of this work, and give some possible future research directions.

7.1 Contributions

7.1.1 Single Response Model-Robust Experimental Design

In Chapter 3 a generalization of the modified Fedorov exchange algorithm has been developed to construct exact designs which are robust to a set of user-specified potential model forms. The product of the determinants of the information matrices for each of the possible models is used as a criterion and a connection is made between these model-robust designs and the $\mathcal{D}$-optimal multiresponse design whose responses have the same model forms as the potential models. Specifically, if the model forms are nested the model-robust design is the same as the optimal design for this associated multiresponse model.

The designs produced by this model-robust procedure compare favorably to the designs produced by competing procedures in the literature. The MRMF

designs are competitive with and simpler to implement than the Genetic Algorithm approach of Heredia-Langner et al. [54]. Compared to the Bayesian approach of DuMouchel and Jones [37], the MRMF procedure is more flexible in the sense that we can explicitly account for more than two models and we do not have the uncertainty of choosing a prior variance parameter.

The second method for exact model-robust designs for single response, completely randomized experiments gives even more flexibility to the user. Using the same "set of potential models" framework, a maximin criterion is used and another generalization of the modified Fedorov exchange algorithm implemented, this time to produce designs which maximize the smallest $\mathcal{D}$-efficiency. This provides experimenters with worst case protection against the specified models and generally seems to give designs whose $\mathcal{D}$-efficiencies are less variable with respect to the possible models than the MRMF, Genetic Algorithm, or Bayesian methods.

A generalization of this maximin procedure allows the specification of a level of interest in each model, resulting in designs that are suggestive of the varying interest levels. This allows the experimenter to exert considerable control over how the design performs with respect to each model for which the design accounts. It is shown that, once the $\mathcal{D}$-efficiencies with respect to each model are standardized by the level of interest, the generalized maximin criterion will produce a design for which the generalized efficiencies are the same, for at least two of the models in the set. Furthermore, a condition is given which guarantees that there is balance for each model in the set, though experience has shown that this complete balance is not usually achieved.

7.1.2 Multiresponse Model-Robust Experimental Design

A new exchange algorithm for multiresponse $\mathcal{D}$-optimal designs is developed in Chapter 5 by deriving multiresponse generalizations of matrix- and determinant-updating formulae originally given in Fedorov [41]. This algorithm requires the specification of the response covariance matrix as well as the form of the relationship between each response and the factors. This procedure seems to be the most effective and practical approach for constructing exact, multiresponse experimental designs that exists in the literature. Others either produce asymptotic designs

[41, 112, 23] or produce sub-optimal designs [9].

However, the required model and covariance matrix specifications are shortcomings of all of these methods. We relax the model form assumption by allowing the user to specify a set of possible models for each response and constructing an artificially large multiresponse regression model for which the $\mathcal{D}$-optimal design is found via the multiresponse exchange algorithm. This is a generalization of the methodology of Chapter 3.

The response covariance matrix is found to have little impact on the $\mathcal{D}$-optimal design, so experimenters will lose little by the almost inevitable pre-experiment lack of knowledge of the covariance matrix. Fortunately, if the experimenter is forced, out of ignorance, to use $\mathbf{\Sigma} = \mathbf{I}$ a simplification of the exchange algorithm is possible which significantly reduces the computational requirements of the multiresponse exchange algorithm. Similarly, if the union of the potential models for all responses are nested, the simplification can be adopted.

The multiresponse methodology developed in this thesis builds upon commonly used exact $\mathcal{D}$-optimal design techniques and allows the construction of exact, multiresponse $\mathcal{D}$-optimal designs, and is extended to account for the model uncertainty that is almost sure to exist before the experiment is conducted.

7.1.3 Split-Plot Model-Robust Experimental Design

In the final part of this dissertation, the methodology developed in Chapter 4 is extended to the more complex case of split-plot experiments. In particular, a maximin algorithm is used to construct $\mathcal{D}$-optimal-like designs that are robust for a set of user-specified models, just as before. The setting, and required algorithm, is complicated by the randomization and covariance structure of a split-plot experiment.

The procedure works to maximize the smallest $\mathcal{D}$-efficiency with respect to the set of user-specified models, and user flexibility is enhanced by the allowance of the specification of varying levels of interest in each possible model. This methodology allows the experimenter to guard against models that are seen as unlikely but possible, while focusing primarily on *a priori* likely models.

7.2 Future Work

The most natural extensions to the work in this thesis are to other experimental settings. For instance, split-plot experiments are considered herein, but split-split-plot experiments (where there are three levels of randomization and three categories of factors: very-hard-to-change, hard-to-change, and easy-to-change; see [60]) and strip-plot experiments (in the two-factor case, both the row factor and column factor are randomized separately, resulting in a matrix of factor level combinations; see [6]) are not considered. The maximin model-robust approach in particular could prove effective if the form of the model being fit is uncertain.

Another class of experiments to which these approaches could be applied are those that fit generalized linear models. For instance, we might consider discrete choice designs modeled via a multinomial distribution, or experiments involving count data using Poisson or Negative Binomial distributions. These experiments have the additional complication of depending not only on the form of the model but also on the particular values of the parameters. This makes for a much more difficult problem and suggests a Bayesian approach, to allow a prior distribution to be put on the parameter values.

Though the "set of models" idea gives experimenters the ability to ensure estimability for particular models other than those deemed most likely, it does not guarantee the ability to formally check for lack-of-fit. Thus, as Dette and Franke [33] did in the one-factor, asymptotic design case, a lack-of-fit criterion could be added to the set of models. Then, the minimum efficiency with respect to each model and the lack-of-fit criterion could be maximized using the exchange algorithm.

The modified Fedorov exchange algorithm [26], upon which our work is based, requires a candidate list which quickly becomes large as its resolution and number of factors increases. An alternative is the candidate-set-free coordinate exchange algorithm [77] which in its original form could not handle irregular design spaces. Recently, we became aware that this algorithm could be adapted to such cases and as such this presents an attractive alternative upon which to base our model-robust procedures.

Another area of additional research would be to study how to better choose

the set of models. In the simplest case, a default of just three different model forms is used: main effects only; main effects plus two-way interactions; and full quadratic. In Chapter 5 we explore larger sets, but in an *ad hoc* way. Bingham and Chipman [13] develop prior distributions on possible models using several assumptions about how effects distribute themselves. It is possible that their approach could be integrated with ours to produce additional, more compelling, sets of possible models.

A striking deficiency in the literature is continuous design theory for restricted randomization experiments. In Chapter 4, we were able to lean upon well developed continuous design theory for completely randomized designs to prove some asymptotic results about the designs produced by the maximin criterion. However, the same does not exist, to our knowledge, for split-plot designs. Thus, an area of future research would be to develop such theory which would allow analogous maximin results for the split-plot case. This study is complicated by the complexity of the two-stage design and its information matrix, and it is unclear whether significant progress can be made. Still, it warrants additional study.

Appendix A

Designs for Chapter 3

A.1 Designs for Example in §3.4.1

Run	x_1	x_2
1	-1	1
2	-0.3	-0.2
3	0	1
4	0.2	0.1
5	1	-1
6	1	0

Table A.1. Model-robust design using MRMF algorithm, for the example in §3.4.1. This is also the Bayesian model-robust design.

Run	x_1	x_2
1	-1	1
2	0	1
3	1	0
4	1	-1
5	-0.9	0.4
6	0.15	-0.1

Table A.2. Model-robust design using Genetic Algorithm [54], for the example in §3.4.1.

Run	x_1	x_2
1	-1	0.5
2	0	0.2
3	0	1
4	0.3	-0.8
5	1	-1
6	1	0

Table A.3. Optimal design for model (3.10), for the example in §3.4.1.

Run	x_1	x_2
1	-1	0.5
2	-1	0.5
3	0	1
4	0.5	-1
5	1	-1
6	1	0

Table A.4. Optimal design for model (3.8), for the example in §3.4.1.

Run	x_1	x_2
1	-1	1
2	-0.3	-0.2
3	-0.2	-0.3
4	0	1
5	1	-1
6	1	0

Table A.5. Optimal design for model (3.9), for the example in §3.4.1.

A.2 Designs for Example in §3.4.2

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1	0	-1	11	0	1	-1
2	0	0.6	0.4	12	0.1	0.3	-0.4
3	1	0	0	13	0.5	-1	0.5
4	0	-1	1	14	0	-1	0
5	0	0	-1	15	-1	0	1
6	0	1	-0.2	16	-1	0	0
7	-0.5	0	1	17	0.4	-0.3	0.6
8	1	-1	0	18	-0.5	-0.5	0.4
9	0.7	-0.6	-0.4	19	0	0	1
10	-1	1	0	20	-1	0.5	0

Table A.6. Model-robust design constructed using MRMF algorithm, for the example in §3.4.2.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	0	0	11	0	0	1
2	-1	0	1	12	0	1	-1
3	-1	1	0	13	0	1	0
4	-0.5	-0.5	0.3	14	0.4	-0.6	-0.4
5	-0.5	0.5	-0.5	15	0.4	0.2	0.4
6	-0.3	-0.1	0.7	16	0.5	0.5	-1
7	-0.3	1	-0.3	17	0.9	-0.3	-0.3
8	0	-1	0	18	1	-1	0
9	0	-1	1	19	1	0	-1
10	0	0	-1	20	1	0	0

Table A.7. Model-robust design constructed using the Bayesian method of [37] with $\frac{1}{\kappa} = 1$, for the example in §3.4.2.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	0	0	11	0	0	1
2	-1	0	1	12	0	1	-1
3	-1	1	0	13	0	1	0
4	-0.7	0.3	0.2	14	0.2	0.2	0.2
5	-0.5	0	-0.5	15	0.2	0.3	-0.7
6	-0.5	1	0	16	0.5	-0.7	0.5
7	-0.4	-0.6	0.5	17	0.5	-0.5	-0.5
8	0	-1	0	18	1	-1	0
9	0	-1	1	19	1	0	-1
10	0	0	-1	20	1	0	0

Table A.8. Model-robust design constructed using the Bayesian method of [37] with $\frac{1}{\kappa} = 16$, for the example in §3.4.2.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	0	0	11	0	0	1
2	-1	0	1	12	0	1	-1
3	-1	1	0	13	0	1	0
4	-0.5	0.6	0.4	14	0.3	-1	0.3
5	-0.5	0.7	-0.5	15	0.4	0.6	-0.5
6	-0.3	-0.3	0.9	16	0.5	0	0.5
7	-0.2	-0.3	-0.2	17	0.9	-0.3	-0.3
8	0	-1	0	18	1	-1	0
9	0	-1	1	19	1	0	-1
10	0	0	-1	20	1	0	0

Table A.9. Design optimal for model (3.15), for the example in §3.4.2.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	0	0	11	0	0	1
2	-1	0	0	12	0	1	-1
3	-1	0	1	13	0	1	0
4	-1	0	1	14	0	1	0
5	-1	1	0	15	0	1	0
6	0	-1	0	16	1	-1	0
7	0	-1	1	17	1	-1	0
8	0	-1	1	18	1	0	-1
9	0	0	-1	19	1	0	-1
10	0	0	-1	20	1	0	0

Table A.10. Design optimal for model (3.11), for the example in §3.4.2.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	0	1	11	0	0.5	0.5
2	-1	0	1	12	0	1	-1
3	-1	1	0	13	0	1	-1
4	-1	1	0	14	0.4	0.2	0.4
5	-0.4	-0.2	-0.4	15	0.5	0.5	0
6	-0.3	-0.4	-0.3	16	1	-1	0
7	-0.3	-0.4	-0.3	17	1	-1	0
8	0	-1	1	18	1	-1	0
9	0	-1	1	19	1	0	-1
10	0	-1	1	20	1	0	-1

Table A.11. Design optimal for model (3.12), for the example in §3.4.2.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	0	0	11	0	0	-1
2	-1	0	0	12	0	0	0
3	-1	0	1	13	0	0	0
4	-1	0	1	14	0	0	1
5	-1	1	0	15	0	0	1
6	-1	1	0	16	0	1	-1
7	0	-1	0	17	0	1	0
8	0	-1	0	18	1	-1	0
9	0	-1	1	19	1	0	-1
10	0	-1	1	20	1	0	0

Table A.12. Design optimal for model (3.13), for the example in §3.4.2.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	0	0	11	0	0.1	-0.1
2	-1	0	1	12	0	0.5	0.5
3	-1	0.4	0	13	0	1	-1
4	-1	0.5	0.5	14	0	1	-0.2
5	-1	1	0	15	0.3	-1	0
6	-0.4	-0.6	0	16	0.6	0	0.4
7	-0.4	0	1	17	1	-1	0
8	0	-1	1	18	1	-0.5	-0.5
9	0	-0.4	1	19	1	0	-1
10	0	0	-1	20	1	0	-0.3

Table A.13. Design optimal for model (3.14), for the example in §3.4.2.

A.3 Designs for Example in §3.4.3

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.66	0.17	0.16	0.01	11	0.65	0	0.35	0
2	0.5	0	0.475	0.025	12	0.725	0.225	0	0.05
3	1	0	0	0	13	0.975	0	0	0.025
4	0.5	0.225	0.225	0.05	14	0.85	0.15	0	0
5	0.5	0	0.5	0	15	0.95	0	0	0.05
6	0.5	0.45	0	0.05	16	0.5	0.5	0	0
7	0.725	0	0.225	0.05	17	0.5	0.35	0.15	0
8	0.5	0.15	0.35	0	18	0.85	0	0.15	0
9	0.5	0.475	0	0.025	19	0.65	0.15	0.16	0.04
10	0.65	0.35	0	0	20	0.5	0	0.45	0.05

Table A.14. Model-robust design using the MRMF algorithm, for the example in §3.4.3.

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.5	0.5	0	0	11	0.5	0.1534	0.3466	0
2	0.5	0.3457	0.1543	0	12	0.5	0.4759	0	0.0241
3	0.8471	0.1529	0	0	13	0.8444	0	0.1556	0
4	0.95	0	0	0.05	14	0.5	0.45	0	0.05
5	0.725	0	0.225	0.05	15	0.6543	0.1544	0.1531	0.0382
6	0.6550	0.3450	0	0	16	1	0	0	0
7	0.9759	0	0	0.0241	17	0.5	0	0.4755	0.0245
8	0.725	0.225	0	0.05	18	0.5	0.225	0.225	0.05
9	0.6576	0	0.3424	0	19	0.6635	0.1637	0.1640	0.0088
10	0.5	0	0.5	0	20	0.5	0	0.45	0.05

Table A.15. Model-robust design using the Genetic Algorithm of [54], for the example in §3.4.3. Note: numbers have been rounded to four decimal places if necessary.

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.5	0	0.45	0.05	11	0.64	0.36	0	0
2	0.5	0	0.475	0.025	12	0.66	0.15	0.15	0.04
3	0.5	0	0.5	0	13	0.66	0.16	0.17	0.01
4	0.5	0.14	0.36	0	14	0.725	0	0.225	0.05
5	0.5	0.225	0.225	0.05	15	0.725	0.225	0	0.05
6	0.5	0.36	0.14	0	16	0.86	0	0.14	0
7	0.5	0.45	0	0.05	17	0.86	0.14	0	0
8	0.5	0.475	0	0.025	18	0.95	0	0	0.05
9	0.5	0.5	0	0	19	0.975	0	0	0.025
10	0.64	0	0.36	0	20	1	0	0	0

Table A.16. Model-robust design using Bayesian method of [37] with $\frac{1}{\kappa} = 1$, for the example in §3.4.3.

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.5	0	0.45	0.05	11	0.64	0.36	0	0
2	0.5	0	0.475	0.025	12	0.66	0.15	0.15	0.04
3	0.5	0	0.5	0	13	0.6667	0.1667	0.1667	0
4	0.5	0.14	0.36	0	14	0.725	0	0.225	0.05
5	0.5	0.225	0.225	0.05	15	0.725	0.225	0	0.05
6	0.5	0.36	0.14	0	16	0.86	0	0.14	0
7	0.5	0.45	0	0.05	17	0.86	0.14	0	0
8	0.5	0.475	0	0.025	18	0.95	0	0	0.05
9	0.5	0.5	0	0	19	0.975	0	0	0.025
10	0.64	0	0.36	0	20	1	0	0	0

Table A.17. Model-robust design using Bayesian method of [37] with $\frac{1}{\kappa} = 16$, for the example in §3.4.3.

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.5	0	0.45	0.05	11	0.64	0.36	0	0
2	0.5	0	0.475	0.025	12	0.65	0.16	0.15	0.04
3	0.5	0	0.5	0	13	0.67	0.16	0.16	0.01
4	0.5	0.14	0.36	0	14	0.725	0	0.225	0.05
5	0.5	0.225	0.225	0.05	15	0.725	0.225	0	0.05
6	0.5	0.36	0.14	0	16	0.86	0	0.14	0
7	0.5	0.45	0	0.05	17	0.86	0.14	0	0
8	0.5	0.475	0	0.025	18	0.95	0	0	0.05
9	0.5	0.5	0	0	19	0.975	0	0	0.025
10	0.64	0	0.36	0	20	1	0	0	0

Table A.18. Optimal design for model (3.19), for the example in §3.4.3.

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.5	0	0.45	0.05	11	0.5	0.5	0	0
2	0.5	0	0.45	0.05	12	0.5	0.5	0	0
3	0.5	0	0.45	0.05	13	0.5	0.5	0	0
4	0.5	0	0.5	0	14	0.5	0.5	0	0
5	0.5	0	0.5	0	15	0.95	0	0	0.05
6	0.5	0	0.5	0	16	0.95	0	0	0.05
7	0.5	0	0.5	0	17	0.95	0	0	0.05
8	0.5	0.45	0	0.05	18	1	0	0	0
9	0.5	0.45	0	0.05	19	1	0	0	0
10	0.5	0.45	0	0.05	20	1	0	0	0

Table A.19. Optimal design for model (3.16), for the example in §3.4.3.

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.5	0	0.45	0.05	11	0.7375	0	0.2375	0.025
2	0.5	0	0.45	0.05	12	0.7375	0.2375	0	0.025
3	0.5	0	0.5	0	13	0.75	0	0.2	0.05
4	0.5	0	0.5	0	14	0.75	0	0.25	0
5	0.5	0.2375	0.2375	0.025	15	0.75	0.2	0	0.05
6	0.5	0.25	0.25	0	16	0.75	0.25	0	0
7	0.5	0.45	0	0.05	17	0.95	0	0	0.05
8	0.5	0.45	0	0.05	18	0.97	0	0	0.03
9	0.5	0.5	0	0	19	1	0	0	0
10	0.5	0.5	0	0	20	1	0	0	0

Table A.20. Optimal design for model (3.17), for the example in §3.4.3.

Run	x_1	x_2	x_3	x_4	Run	x_1	x_2	x_3	x_4
1	0.5	0	0.45	0.05	11	0.67	0.16	0.17	0
2	0.5	0	0.475	0.025	12	0.725	0	0.225	0.05
3	0.5	0	0.5	0	13	0.725	0.225	0	0.05
4	0.5	0.225	0.225	0.05	14	0.7375	0	0.2375	0.025
5	0.5	0.25	0.25	0	15	0.75	0	0.25	0
6	0.5	0.45	0	0.05	16	0.75	0.25	0	0
7	0.5	0.45	0	0.05	17	0.75	0.25	0	0
8	0.5	0.5	0	0	18	0.95	0	0	0.05
9	0.5	0.5	0	0	19	0.975	0	0	0.025
10	0.65	0.17	0.15	0.03	20	1	0	0	0

Table A.21. Optimal design for model (3.18), for the example in §3.4.3.

A.4 Designs for Example in §3.4.4

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1	0	0	7	0.5	0.5	0
2	0.5	0.5	0	8	0	0.5	0.5
3	0	0	1	9	1/3	1/3	1/3
4	0	1	0	10	1	0	0
5	0	0	1	11	0	1	0
6	0.5	0	0.5				

Table A.22. Design constructed using the MRMF algorithm, for the example in §3.4.4.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1	0	0	7	1/3	1/3	1/3
2	0	1	0	8	2/3	1/6	1/6
3	0	0	1	9	1/6	2/3	1/6
4	0.5	0.5	0	10	1/6	1/6	2/3
5	0.5	0	0.5	11	1/3	1/3	1/3
6	0	0.5	0.5				

Table A.23. Design used by Frisbee and McGinity [42], for the example in §3.4.4.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	0	1	0	7	0	0	1
2	0	1	0	8	0	0	1
3	1	0	0	9	0	1	0
4	1	0	0	10	0	1	0
5	1	0	0	11	0	0	1
6	0	0	1				

Table A.24. Design optimal for model (3.20), for the example in §3.4.4.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1	0	0	7	0	1	0
2	0.5	0	0.5	8	0	0.5	0.5
3	1	0	0	9	0.5	0.5	0
4	0	1	0	10	0.5	0	0.5
5	0	0	1	11	0.5	0.5	0
6	0	0	1				

Table A.25. Design optimal for model (3.21), for the example in §3.4.4.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	0	0	1	7	0.5	0	0.5
2	0.5	0.5	0	8	1	0	0
3	1	0	0	9	0.5	0	0.5
4	0	0.5	0.5	10	1/3	1/3	1/3
5	0	0.5	0.5	11	0	1	0
6	1/3	1/3	1/3				

Table A.26. Design optimal for model (3.22), for the example in §3.4.4.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	0.5	0.5	0	7	0	0.5	0.5
2	0	1	0	8	1/3	1/3	1/3
3	0	0	1	9	1	0	0
4	0	1	0	10	0.5	0	0.5
5	0	0	1	11	0.5	0	0.5
6	1	0	0				

Table A.27. Design optimal for model (3.23), for the example in §3.4.4.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	0	0.5	0.5	7	0	1	0
2	1/3	1/3	1/3	8	0.5	0	0.5
3	1	0	0	9	0	0.5	0.5
4	0	1	0	10	0	0	1
5	1	0	0	11	0.5	0.5	0
6	0.5	0.5	0				

Table A.28. Design optimal for model (3.24), for the example in §3.4.4.

Appendix B

Proof of Results in Chapter 4

B.1 Preliminaries

Fedorov [41, Lemma 2.2.2] shows that $\log(|\mathbf{M}_f(\xi)|)$ is strictly concave. Consequently, we can show that $\log(D_f(\xi)) = \frac{1}{p}\left(\log(|\mathbf{M}_f(\xi)|) - \log(|\mathbf{M}_f(\xi^*)|)\right)$ as well as $\min_{f\in\mathcal{F}} \log(D_f(\xi))$ are also strictly concave.

Lemma 1. *$h_1(\mathbf{M}_f(\xi)) = \frac{1}{p}\left(\log(|\mathbf{M}_f(\xi)|) - \log(|\mathbf{M}_f(\xi^*)|)\right)$ is a strictly concave function.*

Proof. Let ξ^* be the optimal design for model f. We must show that for any two arbitrary designs ξ_1 and ξ_2,

$$\frac{1}{p}\left(\log|(1-\alpha)\mathbf{M}_f(\xi_1) + \alpha\mathbf{M}_f(\xi_2)| - \log|\mathbf{M}_f(\xi^*)|\right) > \frac{1-\alpha}{p}\left(\log|\mathbf{M}_f(\xi_1)| - \log|\mathbf{M}_f(\xi^*)|\right) + \frac{\alpha}{p}\left(\log|\mathbf{M}_f(\xi_2)| - \log|\mathbf{M}_f(\xi^*)|\right).$$

Since $\log(|\mathbf{M}(\xi)|)$ is strictly concave, we have that

$$\begin{aligned}
&\frac{1}{p}\left(\log|(1-\alpha)\mathbf{M}_f(\xi_1) + \alpha\mathbf{M}_f(\xi_2)| - \log|\mathbf{M}_f(\xi^*)|\right) \\
&\quad> \frac{1}{p}\left((1-\alpha)\log|\mathbf{M}_f(\xi_1)| + \alpha\log|\mathbf{M}_f(\xi_2)| - \log|\mathbf{M}_f(\xi^*)|\right) \\
&\quad= \frac{1-\alpha}{p}\log|\mathbf{M}_f(\xi_1)| + \frac{\alpha}{p}\log|\mathbf{M}_f(\xi_2)| - \frac{1}{p}\log|\mathbf{M}_f(\xi^*)| \\
&\quad= \frac{1-\alpha}{p}\left(\log|\mathbf{M}_f(\xi_1)| - \log|\mathbf{M}_f(\xi^*)|\right) + \frac{\alpha}{p}\left(\log|\mathbf{M}_f(\xi_2)| - \log|\mathbf{M}_f(\xi^*)|\right)
\end{aligned}$$

Therefore, $h_1(\mathbf{M}_f(\xi))$ is strictly concave. □

Lemma 2. *Let $a_1, \ldots, a_r, a_{r+1}$ and $b_1, \ldots, b_r, b_{r+1}$ be real numbers. Then,*

$$\min(a_1, \ldots, a_r, a_{r+1}) + \min(b_1, \ldots, b_r, b_{r+1}) \leq \min(a_1 + b_1, \ldots, a_r + b_r, a_{r+1} + b_{r+1}) \tag{B.1}$$

Proof. We prove by induction. When $r = 1$, the result is trivially true. Then, assume

$$\min(a_1, \ldots, a_r) + \min(b_1, \ldots, b_r) \leq \min(a_1 + b_1, \ldots, a_r + b_r), \tag{B.2}$$

and show it is true for $r + 1$. We break it into four cases. Case 1:

$$\min(a_1, \ldots, a_r) \leq a_{r+1}, \quad \min(b_1, \ldots, b_r) \leq b_{r+1} \tag{B.3}$$

Thus,

$$\begin{aligned}
&\min(a_1, \ldots, a_r, a_{r+1}) + \min(b_1, \ldots, b_r, b_{r+1}) \\
&\qquad = \min(\min(a_1, \ldots, a_r), a_{r+1}) + \min(\min(b_1, \ldots, b_r), b_{r+1}) \\
&\qquad = \min(a_1, \ldots, a_r) + \min(b_1, \ldots, b_r) && \text{(B.4)} \\
&\qquad = \min(\min(a_1, \ldots, a_r) + \min(b_1, \ldots, b_r), a_{r+1} + b_{r+1}) && \text{(B.5)} \\
&\qquad \leq \min(\min(a_1 + b_1, \ldots, a_r + b_r), a_{r+1} + b_{r+1}) && \text{(B.6)} \\
&\qquad = \min(a_1 + b_1, \ldots, a_r + b_r, a_{r+1} + b_{r+1}) && \text{(B.7)}
\end{aligned}$$

where (B.4) and (B.5) are true because of (B.3), and the inequality in (B.6) is due to (B.2). Case 2:

$$\min(a_1, \ldots, a_r) \leq a_{r+1}, \quad \min(b_1, \ldots, b_r) > b_{r+1} \tag{B.8}$$

so that

$$\begin{aligned}
&\min(a_1, \ldots, a_r, a_{r+1}) + \min(b_1, \ldots, b_r, b_{r+1}) \\
&\qquad = \min(\min(a_1, \ldots, a_r), a_{r+1}) + \min(\min(b_1, \ldots, b_r), b_{r+1}) \\
&\qquad = \min(a_1, \ldots, a_r) + b_{r+1} && \text{(B.9)}
\end{aligned}$$

$$\leq \min(\min(a_1, \ldots, a_r) + \min(b_1, \ldots, b_r), a_{r+1} + b_{r+1}) \tag{B.10}$$

$$\leq \min(\min(a_1 + b_1, \ldots, a_r + b_r), a_{r+1} + b_{r+1}) \tag{B.11}$$

$$= \min(a_1 + b_1, \ldots, a_r + b_r, a_{r+1} + b_{r+1}) \tag{B.12}$$

by the same arguments as above. Case 3

$$\min(a_1, \ldots, a_r) > a_{r+1}, \quad \min(b_1, \ldots, b_r) \leq b_{r+1}, \tag{B.13}$$

and Case 4

$$\min(a_1, \ldots, a_r) > a_{r+1}, \quad \min(b_1, \ldots, b_r) > b_{r+1}, \tag{B.14}$$

can be shown using essentially the same arguments. Thus, the Lemma is proven. □

Lemma 3. *$h_2(\xi) = \min_{f \in \mathcal{F}} (\log (D_f(\xi)))$ is a strictly concave function.*

Proof. We will show this directly by the definition of concavity.

$$\begin{aligned}
&\min (\log D_1((1-\alpha)\xi_1 + \alpha\xi_2), \ldots, \log D_r((1-\alpha)\xi_1 + \alpha\xi_2)) \\
&\quad > \min ((1-\alpha) \log D_1(\xi_1) + \alpha \log D_1(\xi_2), \ldots, (1-\alpha) \log D_r(\xi_1) + \alpha \log D_r(\xi_2)) \\
&\quad \geq \min ((1-\alpha) (\log D_1(\xi_1), \ldots, \log D_r(\xi_1))) + \min (\alpha (\log D_1(\xi_2), \ldots, \log D_r(\xi_2))) \\
&\quad = (1-\alpha) \min (\log D_1(\xi_1), \ldots, \log D_r(\xi_1)) + \alpha \min (\log D_1(\xi_2), \ldots, \log D_r(\xi_2))
\end{aligned}$$

where the first inequality is using the fact that the $\log D_f(\xi)$ are strictly concave $\forall f \in \mathcal{F}$, the second is based upon Lemma 2, and the final equality based upon $\min (c(a_1, \ldots, a_r)) = c \min (a_1, \ldots, a_r)$ where c and $a_1, \ldots, a_r$ are real. Thus we have shown the strict concavity of this function. □

It is straightforward to see that these results hold also for $G_f(\xi) = D_{\mathcal{D},f}(\xi)/v_f$, the generalized efficiency.

The subsequent proofs rely in part on the strict concavity of $\log(G_f(\xi))$, which can then be used to gain insight about $G_f(\xi)$ itself because $\log(\cdot)$ is a strictly increasing function, so that for any two designs ξ_1 and ξ_2,

$$G_f(\xi_1) > G_f(\xi_2) \Leftrightarrow \log(G_f(\xi_1)) > \log(G_f(\xi_2)) \tag{B.15}$$

A direct, and useful, consequence of (B.15) is:

$$\min_{f\in\mathcal{F}} G_f(\xi_1) > \min_{f\in\mathcal{F}} G_f(\xi_2) \Leftrightarrow \min_{f\in\mathcal{F}} \log(G_f(\xi_1)) > \min_{f\in\mathcal{F}} \log(G_f(\xi_2)) \tag{B.16}$$

B.2 Proof of Theorem 1

Proof. We prove by contradiction. Assume that for design ξ^*, $G_f(\xi^*) - G_{f_{min}}(\xi^*) > 0$ for all $f \in \mathcal{F}\backslash f_{min}$. Let $f_g = \arg\min_{f\in\mathcal{F}\backslash f_{min}} G_f(\xi^*)$ and $G_{f_g}(\xi^*) - G_{f_{min}}(\xi^*) = c > 0$.

By the strict concavity of $\log(G_f(\cdot))$, and since we assume that ξ^* is not optimal for f_{min}, there exists a direction $d = (\lambda_1, \ldots, \lambda_d)$ *s.t.* for an infinitesimal step we have a new design $\xi^{'}$ for which $\log(G_{f_{min}}(\xi^{'})) > \log(G_{f_{min}}(\xi^*))$. By the equivalence in (B.15), we have that $G_{f_{min}}(\xi^{'}) > G_{f_{min}}(\xi^*)$. Note that without loss of generality, this new design could exclude design points in or include design points not in ξ^*. Then by the maximin optimality of ξ^*, $G_f(\xi^{'})$ for some $f \in \mathcal{F}\backslash f_{min}$ must be reduced by at least c (otherwise, there would exist a larger minimum so that ξ^* would not be maximin optimal).

Now, with the information matrix as in (4.5), an arbitrarily small step in direction d results in arbitrarily small changes in ξ^* in terms of design measures λ_i for all i so that the elements of $\mathbf{M}$ are negligibly changed, and by the continuity of $|\cdot|$ with respect to the elements of the matrix, $|\mathbf{M}|$ only changes negligibly, say $\delta < c$. But if $\delta < c$, then ξ^* is not maximin optimal because $\min_{f\in\mathcal{F}} G_f(\xi') > \min_{f\in\mathcal{F}} G_f(\xi^*)$. Since we assumed ξ^* is maximin optimal, this is a contradiction. Thus the assertion holds. □

B.3 Proof of Theorem 2

Proof. Proof is by contradiction, similar to that for Theorem 1. Assume that for the generalized maximin optimal design ξ^*, $G_f(\xi^*) - G_{f_{min}}(\xi^*) > 0$ for some $f \in \mathcal{F}$, a subset of models which we denote $\mathcal{F}_g$. Let $\mathcal{F}_l = \mathcal{F}\backslash\mathcal{F}_g$. Then, among all $f \in \mathcal{F}_g$, let $f_g = \arg\min_{f\in\mathcal{F}_g} G_f(\xi^*)$ and $G_{f_g}(\xi^*) - G_{f_{min}}(\xi^*) = c > 0$.

Since $\mathcal{F}_l \subset \mathcal{F}$, condition (4.7) gives that there exists a design $\xi^{'}$ *s.t.*

$$\min_{f \in \mathcal{F}_l} G_f(\xi^{'}) > \min_{f \in \mathcal{F}_l} G_f(\xi^*).$$

By the equivalence in (B.16),

$$\min_{f \in \mathcal{F}_l} log(G_f(\xi^{'})) > \min_{f \in \mathcal{F}_l} log(G_f(\xi^*)).$$

Since $\min_{f \in \mathcal{F}_l} \log(G_f(\cdot))$ is strictly concave and, evaluated at ξ^*, is not at its maximum, there exists a direction $d = (\lambda_1, \ldots, \lambda_d)$ *s.t.* for an infinitesimal step we have a new design, $\tilde{\xi}$, for which $\min_{f \in \mathcal{F}_l} \log(G_f(\tilde{\xi})) > \min_{f \in \mathcal{F}_l} \log(G_f(\xi^*))$. By (B.16) again, we have that $\min_{f \in \mathcal{F}_l} G_f(\tilde{\xi}) > \min_{f \in \mathcal{F}_l} G_f(\xi^*)$. The remainder follows by the same arguments as the previous proof. □

B.4 Proof of Corollary 1

Proof. We have that $\min_{f \in \mathcal{F}} G_f(\xi^*) < 1$ because we assume that $\xi^* \neq \xi^*_{f_1}$ and $\xi^* \neq \xi^*_{f_2}$ and that $v_1 = 1$ or $v_2 = 1$. Then, (4.7) is satisfied because we can always find a direction such that the minimum generalized efficiency is improved for the model associated with the smaller generalized efficiency. Then, we can use the same arguments as in the previous results. □

Appendix C

Designs for Chapter 4

C.1 Designs for Example in §4.4.1

In this section, we give only the maximin and $(1, 1, .6)$-maximin designs, since the others referred to in Table 4.1 are given in Appendix A, Tables A.1-A.5.

Run	x_1	x_2
1	-0.3	0
2	1	-0.9
3	0	1
4	1	0
5	0.5	-1
6	-1	1

Table C.1. Maximin model-robust design, for the example in §4.4.1.

Run	x_1	x_2
1	0.5	-1
2	1	0
3	-0.6	0.1
4	-1	1
5	0	1
6	1	-1

Table C.2. $(1, 1, .6)$-Maximin model-robust design, for the example in §4.4.1.

C.2 Designs for Example in §4.4.2

Run	x_1	x_2	x_3	x_4	x_5
1	0.6	0.05	0.15	0.15	0.05
2	0.64	0.05	0.05	0.18	0.08
3	0.6	0.05	0.095	0.105	0.15
4	0.64	0.05	0.05	0.25	0.01
5	0.645	0.05	0.095	0.21	0
6	0.55	0.1	0.15	0.1	0.1
7	0.62	0.15	0.05	0.1	0.08
8	0.57	0.15	0.1	0.1	0.08
9	0.57	0.1	0.07	0.11	0.15
10	0.62	0.05	0.15	0.1	0.08
11	0.5	0.15	0.1	0.25	0
12	0.52	0.15	0.15	0.18	0
13	0.6	0.05	0.15	0.2	0
14	0.7	0.07	0.05	0.1	0.08
15	0.5	0.15	0.09	0.11	0.15
16	0.55	0.15	0.05	0.1	0.15
17	0.5	0.15	0.15	0.1	0.1
18	0.54	0.15	0.05	0.25	0.01
19	0.66	0.11	0.05	0.18	0
20	0.5	0.15	0.09	0.18	0.08
21	0.538	0.15	0.1	0.212	0
22	0.7	0.05	0.05	0.2	0
23	0.65	0.05	0.05	0.1	0.15
24	0.62	0.15	0.05	0.14	0.04
25	0.66	0.05	0.11	0.18	0

Table C.3. Maximin model-robust design, for the example in §4.4.2.

Run	x_1	x_2	x_3	x_4	x_5
1	0.5	0.15	0.1	0.25	0
2	0.6	0.05	0.1	0.25	0
3	0.62	0.15	0.05	0.18	0
4	0.5	0.15	0.15	0.1	0.1
5	0.5	0.15	0.15	0.15	0.05
6	0.7	0.07	0.05	0.1	0.08
7	0.52	0.15	0.15	0.18	0
8	0.7	0.05	0.05	0.15	0.05
9	0.55	0.15	0.05	0.25	0
10	0.52	0.15	0.07	0.11	0.15
11	0.57	0.15	0.1	0.1	0.08
12	0.65	0.05	0.05	0.1	0.15
13	0.62	0.05	0.15	0.18	0
14	0.6	0.05	0.09	0.18	0.08
15	0.55	0.1	0.15	0.2	0
16	0.632	0.094	0.094	0.18	0
17	0.538	0.15	0.1	0.212	0
18	0.6	0.05	0.15	0.1	0.1
19	0.54	0.15	0.05	0.18	0.08
20	0.7	0.05	0.05	0.2	0
21	0.585	0.15	0.05	0.1	0.115
22	0.55	0.1	0.09	0.11	0.15
23	0.64	0.05	0.05	0.25	0.01
24	0.66	0.11	0.05	0.1	0.08
25	0.66	0.05	0.11	0.1	0.08

Table C.4. (.9,1,1)-maximin model-robust design, for the example in §4.4.2.

Run	x_1	x_2	x_3	x_4	x_5
1	0.54	0.15	0.05	0.18	0.08
2	0.65	0.05	0.05	0.1	0.15
3	0.5	0.15	0.15	0.1	0.1
4	0.62	0.15	0.05	0.1	0.08
5	0.57	0.15	0.1	0.18	0
6	0.6	0.05	0.09	0.11	0.15
7	0.55	0.15	0.05	0.25	0
8	0.57	0.1	0.15	0.18	0
9	0.64	0.05	0.05	0.25	0.01
10	0.5	0.15	0.09	0.25	0.01
11	0.62	0.15	0.05	0.18	0
12	0.6	0.05	0.15	0.1	0.1
13	0.52	0.15	0.15	0.1	0.08
14	0.7	0.07	0.05	0.1	0.08
15	0.62	0.05	0.15	0.18	0
16	0.66	0.05	0.11	0.1	0.08
17	0.6	0.05	0.09	0.18	0.08
18	0.7	0.07	0.05	0.18	0
19	0.7	0.05	0.05	0.2	0
20	0.5963	0.095	0.0875	0.1606	0.06062
21	0.6	0.05	0.1	0.25	0
22	0.5	0.15	0.15	0.2	0
23	0.5	0.15	0.095	0.105	0.15
24	0.55	0.15	0.05	0.1	0.15
25	0.57	0.1	0.07	0.25	0.01

Table C.5. (.9,1,.5)-maximin model-robust design, for the example in §4.4.2.

Run	x_1	x_2	x_3	x_4	x_5
1	0.64	0.05	0.05	0.18	0.08
2	0.7	0.07	0.05	0.1	0.08
3	0.6	0.1	0.05	0.25	0
4	0.5	0.15	0.15	0.1	0.1
5	0.57	0.15	0.05	0.165	0.065
6	0.7	0.05	0.05	0.2	0
7	0.5	0.15	0.09	0.18	0.08
8	0.57	0.15	0.1	0.1	0.08
9	0.62	0.15	0.05	0.1	0.08
10	0.6	0.05	0.1133	0.1683	0.0683
11	0.6	0.05	0.125	0.1	0.125
12	0.5	0.15	0.1	0.25	0
13	0.62	0.05	0.15	0.1	0.08
14	0.57	0.1	0.15	0.14	0.04
15	0.54	0.15	0.05	0.11	0.15
16	0.66	0.05	0.11	0.18	0
17	0.538	0.15	0.1	0.212	0
18	0.54	0.15	0.05	0.25	0.01
19	0.605	0.09333	0.09333	0.1	0.1083
20	0.62	0.05	0.07	0.25	0.01
21	0.62	0.15	0.05	0.18	0
22	0.52	0.15	0.15	0.18	0
23	0.55	0.1	0.09	0.11	0.15
24	0.65	0.05	0.05	0.1	0.15
25	0.6	0.05	0.15	0.2	0

Table C.6. Model-robust design constructed using MRMF algorithm, for the example in §4.4.2.

Run	x_1	x_2	x_3	x_4	x_5	Run	x_1	x_2	x_3	x_4	x_5
1	0.5	0.15	0.09	0.18	0.08	14	0.6	0.05	0.15	0.1	0.1
2	0.5	0.15	0.11	0.1	0.14	15	0.6	0.1	0.05	0.25	0
3	0.5	0.15	0.15	0.2	0	16	0.6	0.15	0.05	0.1	0.1
4	0.52	0.15	0.07	0.25	0.01	17	0.61	0.1	0.11	0.1	0.08
5	0.52	0.15	0.15	0.1	0.08	18	0.62	0.05	0.15	0.16	0.02
6	0.54	0.15	0.05	0.11	0.15	19	0.62	0.15	0.05	0.18	0
7	0.54	0.15	0.1	0.15	0.06	20	0.64	0.05	0.05	0.19	0.07
8	0.56	0.09	0.09	0.11	0.15	21	0.64	0.05	0.1	0.21	0
9	0.56	0.09	0.15	0.2	0	22	0.65	0.05	0.05	0.1	0.15
10	0.56	0.15	0.05	0.18	0.06	23	0.67	0.1	0.05	0.13	0.05
11	0.56	0.15	0.11	0.18	0	24	0.7	0.05	0.05	0.2	0
12	0.6	0.05	0.1	0.25	0	25	0.7	0.05	0.07	0.1	0.08
13	0.6	0.05	0.12	0.16	0.07						

Table C.7. Design optimal for model (4.13), for the example in §4.4.2.

Run	x_1	x_2	x_3	x_4	x_5	Run	x_1	x_2	x_3	x_4	x_5
1	0.5	0.15	0.09	0.11	0.15	14	0.62	0.15	0.05	0.1	0.08
2	0.5	0.15	0.09	0.11	0.15	15	0.62	0.15	0.05	0.1	0.08
3	0.5	0.15	0.09	0.11	0.15	16	0.62	0.15	0.05	0.1	0.08
4	0.5	0.15	0.09	0.25	0.01	17	0.62	0.15	0.05	0.18	0
5	0.5	0.15	0.09	0.25	0.01	18	0.62	0.15	0.05	0.18	0
6	0.5	0.15	0.09	0.25	0.01	19	0.64	0.05	0.05	0.11	0.15
7	0.5	0.15	0.15	0.1	0.1	20	0.64	0.05	0.05	0.11	0.15
8	0.5	0.15	0.15	0.2	0	21	0.64	0.05	0.05	0.11	0.15
9	0.62	0.05	0.15	0.1	0.08	22	0.64	0.05	0.05	0.25	0.01
10	0.62	0.05	0.15	0.1	0.08	23	0.64	0.05	0.05	0.25	0.01
11	0.62	0.05	0.15	0.1	0.08	24	0.64	0.05	0.05	0.25	0.01
12	0.62	0.05	0.15	0.18	0	25	0.7	0.07	0.05	0.18	0
13	0.62	0.05	0.15	0.18	0						

Table C.8. Design optimal for model (4.11), for the example in §4.4.2.

Run	x_1	x_2	x_3	x_4	x_5	Run	x_1	x_2	x_3	x_4	x_5
1	0.5	0.15	0.09	0.11	0.15	14	0.6	0.05	0.1	0.25	0
2	0.5	0.15	0.09	0.25	0.01	15	0.6	0.05	0.15	0.1	0.1
3	0.5	0.15	0.15	0.1	0.1	16	0.6	0.05	0.15	0.2	0
4	0.5	0.15	0.15	0.2	0	17	0.62	0.05	0.15	0.1	0.08
5	0.54	0.15	0.05	0.18	0.08	18	0.62	0.15	0.05	0.1	0.08
6	0.55	0.15	0.05	0.1	0.15	19	0.62	0.15	0.05	0.18	0
7	0.56	0.09	0.1	0.1	0.15	20	0.64	0.05	0.05	0.11	0.15
8	0.56	0.11	0.15	0.18	0.18	21	0.65	0.05	0.05	0.25	0
9	0.56	0.15	0.11	0.1	0.08	22	0.67	0.05	0.1	0.18	0
10	0.58	0.15	0.05	0.22	0	23	0.69	0.05	0.05	0.1	0.11
11	0.59	0.1	0.05	0.25	0.01	24	0.7	0.07	0.05	0.1	0.08
12	0.59	0.1	0.09	0.16	0.06	25	0.7	0.07	0.05	0.18	0
13	0.6	0.05	0.09	0.18	0.08						

Table C.9. Design optimal for model (4.12), for the example in §4.4.2.

C.3 Designs for Example in §4.4.3

In this section, we give only the maximin and $(.9, 1, 1, 1, .9)$-maximin designs, since the others referred to in Table 4.3 are given in Appendix A, Tables 3.4-A.28.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	0	0.5	0.5	7	1	0	0
2	0	0	1	8	1/3	1/3	1/3
3	0	1	0	9	0	1	0
4	0	0	1	10	5/12	0	7/12
5	1	0	0	11	1	0	0
6	1/3	2/3	0				

Table C.10. Maximin model-robust design, for the example in §4.4.3.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	0	1	0	7	0	0	1
2	0.5	0.5	0	8	1	0	0
3	0	0	1	9	7/12	0	5/12
4	1	0	0	10	1/3	5/12	1/4
5	0	1	0	11	0	0	1
6	0	0.5	0.5				

Table C.11. (.9,1,1,1,.9)-maximin model-robust design, for the example in §4.4.3.

Appendix D

Matrix Algebra Results for Chapter 5

We provide here a collection of results which are necessary to prove Theorem 3 and Corollary 2. The first is well-known and presented without proof.

Lemma 4. *Let* $\mathbf{\Delta}$ *be a block matrix such that*

$$\mathbf{\Delta} = \begin{pmatrix} \mathbf{\Delta}_{11} & \mathbf{\Delta}_{12} \\ \mathbf{\Delta}_{21} & \mathbf{\Delta}_{22} \end{pmatrix}, \tag{D.1}$$

where $\mathbf{\Delta}_{11}$ *is a* $n \times n$ *nonsingular matrix,* $\mathbf{\Delta}_{12}$ *is a* $n \times k$ *matrix,* $\mathbf{\Delta}_{21}$ *is a* $k \times n$ *matrix, and* $\mathbf{\Delta}_{22}$ *is a* $k \times k$ *nonsingular matrix. Then*

$$|\mathbf{\Delta}| = |\mathbf{\Delta}_{11}||\mathbf{\Delta}_{22} - \mathbf{\Delta}_{21}\mathbf{\Delta}_{11}^{-1}\mathbf{\Delta}_{12}| = |\mathbf{\Delta}_{11} - \mathbf{\Delta}_{12}\mathbf{\Delta}_{22}^{-1}\mathbf{\Delta}_{21}||\mathbf{\Delta}_{22}|. \tag{D.2}$$

The next result is a slight generalization of an identity given in Schott [93].

Lemma 5. *Let* $\mathbf{M}$ *be* $n \times n$, $\mathbf{A}$ *be* $n \times k$ *and* $\mathbf{B}$ *be* $k \times n$. *Then*

$$|\mathbf{M} + \mathbf{AB}| = \begin{vmatrix} \mathbf{M} & \mathbf{A} \\ -\mathbf{B} & \mathbf{I}_k \end{vmatrix}. \tag{D.3}$$

Proof. Using basic matrix multiplication, it is true that

$$\begin{pmatrix} \mathbf{M} & \mathbf{A} \\ -\mathbf{B} & \mathbf{I}_k \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{B} & \mathbf{I}_k \end{pmatrix} = \begin{pmatrix} \mathbf{M} + \mathbf{AB} & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}. \tag{D.4}$$

Taking the determinant of both sides, and using the well known property that the

determinant of a product of two matrices is equal to the product of the determinants of the matrices, gives

$$\begin{vmatrix} \mathbf{M} & \mathbf{A} \\ -\mathbf{B} & \mathbf{I}_k \end{vmatrix} \begin{vmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{B} & \mathbf{I}_k \end{vmatrix} = \begin{vmatrix} \mathbf{M}+\mathbf{AB} & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{vmatrix}, \tag{D.5}$$

and by Lemma 4,

$$\begin{vmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{B} & \mathbf{I}_k \end{vmatrix} = |\mathbf{I}_n||\mathbf{I}_k - \mathbf{0}| = 1. \tag{D.6}$$

Thus,

$$\begin{aligned} \begin{vmatrix} \mathbf{M} & \mathbf{A} \\ -\mathbf{B} & \mathbf{I}_k \end{vmatrix} &= \begin{vmatrix} \mathbf{M}+\mathbf{AB} & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{vmatrix} \\ &= |\mathbf{I}_n||\mathbf{M}+\mathbf{AB}-\mathbf{A}\mathbf{I}_k\mathbf{0}| \\ &= |\mathbf{M}+\mathbf{AB}|, \end{aligned}$$

where the second equality follows from another appeal to Lemma 4. □

Another identity simply combines the two previous results and is a slight generalization of Lemma 2.5.1 in Fedorov [41].

Lemma 6. *Let* $\mathbf{M}$ *be a nonsingular* $n \times n$ *matrix, let* $\mathbf{A}$ *be a* $n \times k$ *matrix and let* $\mathbf{B}$ *be an* $k \times n$ *matrix; then*

$$|\mathbf{M}+\mathbf{AB}| = |\mathbf{M}||\mathbf{I}_k + \mathbf{B}\mathbf{M}^{-1}\mathbf{A}|. \tag{D.7}$$

Proof. Lemma 5 gives that

$$|\mathbf{M}+\mathbf{AB}| = \begin{vmatrix} \mathbf{M} & \mathbf{A} \\ -\mathbf{B} & \mathbf{I}_k \end{vmatrix}. \tag{D.8}$$

and then by Lemma 4 we get what we wanted to prove. □

Finally, a useful and well-known result ([41], Lemma 2.6.1) is the Matrix Inversion Lemma.

Lemma 7. *Let* $\mathbf{A}$ *be a* $n \times k$ *matrix and let* $\mathbf{B}$ *be an* $k \times n$ *matrix; then*

$$(\mathbf{I}_n + \mathbf{AB})^{-1} = \mathbf{I}_n - \mathbf{A}(\mathbf{I}_k + \mathbf{BA})^{-1}\mathbf{B}. \tag{D.9}$$

Appendix E

Designs for Chapter 5

E.1 Designs for Example in §5.5.1

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1.73	-1.73	-1.73	11	-1.73	-1.73	1.73
2	-1.73	1.73	-1.73	12	-1.73	-1.73	-1.73
3	1.73	-1.73	1.73	13	0	-1.73	0
4	-1.73	1.73	1.73	14	1.73	-1.73	-1.73
5	-1.73	-1.73	1.73	15	0	1.73	-1.73
6	1.73	1.73	1.73	16	1.73	1.73	-1.73
7	0	1.73	1.73	17	-1.73	0	0
8	-1.73	1.73	0	18	1.73	1.73	0
9	1.73	0	1.73	19	-1.73	-1.73	-1.73
10	0	-1.73	0	20	0	0	-1.73

Table E.1. $\mathcal{D}$-optimal design constructed via MX algorithm, for example in §5.5.1, when $\boldsymbol{\Sigma}$ and $\boldsymbol{\rho}(\mathbf{x})$ are known

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1.680	0	0	11	-1.730	1.730	0.026
2	0	0	0	12	-1.730	1.730	0.026
3	0	0	0	13	-1.729	-1.730	-1.728
4	1.728	-1.729	-1.720	14	-1.729	-1.730	-1.728
5	1.728	-1.729	-1.720	15	-1.730	-0.096	1.730
6	1.729	1.729	1.729	16	1.729	1.724	-1.729
7	1.729	1.729	1.729	17	1.729	1.724	-1.729
8	-1.725	-1.723	1.715	18	-0.154	1.730	-1.730
9	-1.730	1.721	1.729	19	-0.154	1.730	-1.730
10	1.730	-1.729	1.729	20	-0.101	-1.730	1.730

Table E.2. $\mathcal{D}$-optimal design constructed using semi-definite programming [9], for example in §5.5.1, when $\boldsymbol{\Sigma}$ and $\boldsymbol{\rho}(\mathbf{x})$ are known

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1.73	-1.73	1.73	11	1.73	1.73	1.73
2	-1.73	1.73	-1.73	12	1.73	0	-1.73
3	-1.73	-1.73	-1.73	13	0	1.73	0
4	-1.73	1.73	1.73	14	1.73	1.73	1.73
5	0	1.73	0	15	0	-1.73	1.73
6	-1.73	-1.73	0	16	1.73	-1.73	0
7	0	-1.73	1.73	17	-1.73	0	0
8	1.73	1.73	-1.73	18	1.73	-1.73	1.73
9	-1.73	1.73	1.73	19	0	0	-1.73
10	-1.73	1.73	-1.73	20	1.73	-1.73	-1.73

Table E.3. $\mathcal{D}$-optimal design for example in §5.5.1 with $\boldsymbol{\rho}(\mathbf{x})$ known and $\boldsymbol{\Sigma} = \mathbf{I}$.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1.73	1.73	1.73	11	-1.73	1.73	1.73
2	1.73	-1.73	1.73	12	-1.73	-1.73	1.73
3	-1.73	1.73	1.73	13	0	1.73	-1.73
4	-1.73	-1.73	-1.73	14	1.73	1.73	-1.73
5	1.73	-1.73	-1.73	15	-1.73	-1.73	-1.73
6	-1.73	1.73	-1.73	16	-1.73	-1.73	1.73
7	1.73	1.73	1.73	17	1.73	-1.73	-1.73
8	1.73	1.73	0	18	0	0	1.73
9	1.73	-1.73	1.73	19	1.73	0	-1.73
10	0	-1.73	0	20	-1.73	0	0

Table E.4. Model-robust design for the example in §5.5.1, constructed using the SSPS set of possible models for each response.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1.73	1.73	1.73	11 -1.73	-1.73	-1.73	
2	1.73	-1.73	-1.73	12 -1.73	1.73	-1.73	
3	1.73	1.73	-1.73	13 1.73	-1.73	-1.73	
4	-1.73	-1.73	-1.73	14 1.73	1.73	1.73	
5	1.73	-1.73	1.73	15 1.73	-1.73	1.73	
6	-1.73	1.73	-1.73	16 -1.73	1.73	1.73	
7	-1.73	-1.73	1.73	17 0	0	-1.73	
8	-1.73	-1.73	1.73	18 0	1.73	0	
9	1.73	-1.73	1.73	19 -1.73	1.73	1.73	
10	1.73	1.73	-1.73	20 -1.73	0	0	

Table E.5. Model-robust design for the example in §5.5.1, constructed using the SPS set of possible models for each response, assuming $\mathbf{\Sigma}$ is known (in this case, the design is the same if we use $\mathbf{\Sigma} = \mathbf{I}$).

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1.73	1.73	-1.73	11	0	-1.73	0
2	1.73	-1.73	1.73	12	-1.73	-1.73	1.73
3	1.73	1.73	1.73	13	0	1.73	1.73
4	1.73	0	-1.73	14	-1.73	0	-1.73
5	-1.73	1.73	-1.73	15	-1.73	1.73	0
6	1.73	-1.73	1.73	16	1.73	0	0
7	1.73	1.73	0	17	1.73	-1.73	-1.73
8	-1.73	-1.73	1.73	18	-1.73	-1.73	-1.73
9	-1.73	1.73	1.73	19	0	-1.73	-1.73
10	0	1.73	-1.73	20	0	0	1.73

Table E.6. $\mathcal{D}$-optimal design for the example in §5.5.1, constructed assuming the full quadratic model only.

E.2 Designs for Example in §5.5.2

Run	x_1	x_2	Run	x_1	x_2
1	-1	-1	7	1	-1
2	-1	-1	8	1	-1
3	1	1	9	1	1
4	-1	1	10	-1	1
5	0	-1	11	0	1
6	1	0	12	-1	0

Table E.7. Model-robust design for example in §5.5.2, constructed using the SSPS model set (the design constructed using the SPS model set is the same).

Run	x_1	x_2	Run	x_1	x_2
1	0	1	7	-1	0
2	1	-1	8	0	-1
3	1	0	9	1	-1
4	1	1	10	-1	-1
5	-1	1	11	-1	1
6	-1	-1	12	0	0

Table E.8. $\mathcal{D}$-optimal design for the example in §5.5.2, constructed assuming the full quadratic model only.

E.3 Designs for Example in §5.5.3

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1	-1	0	10	1	-1	1
2	-1	-1	-1	11	-1	1	0
3	1	1	-1	12	0	1	0
4	1	1	0	13	0	-1	1
5	1	-1	-1	14	-1	1	-1
6	-1	1	1	15	0	-1	0
7	-1	-1	0	16	1	1	1
8	-1	-1	1	17	0	1	-1
9	0	-1	-1				

Table E.9. $\mathcal{D}$-optimal design for example in §5.5.3, with $\boldsymbol{\rho}(\mathbf{x})$ and $\boldsymbol{\Sigma}$ assumed known.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	1	-1	10	-1	0	1
2	1	1	1	11	0	1	0
3	-1	-1	-1	12	1	-1	1
4	1	-1	1	13	1	0	0
5	-1	1	-1	14	1	1	1
6	1	1	-1	15	1	-1	-1
7	-1	1	1	16	0	0	-1
8	-1	-1	1	17	1	-1	-1
9	-1	-1	0				

Table E.10. Model-robust design for example in §5.5.3, constructed using the SSPS model set.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	-1	1	1	10	-1	-1	-1
2	1	1	1	11	1	-1	0
3	-1	-1	1	12	-1	1	-1
4	1	-1	-1	13	-1	1	-1
5	1	-1	1	14	0	0	0
6	1	1	-1	15	0	1	-1
7	-1	-1	-1	16	-1	-1	1
8	-1	1	1	17	1	0	-1
9	1	1	1				

Table E.11. Model-robust design for example in §5.5.3, constructed using the SPS model set.

Run	x_1	x_2	x_3	Run	x_1	x_2	x_3
1	1	-1	1	10	-1	-1	1
2	1	1	1	11	1	-1	-1
3	-1	1	1	12	1	-1	1
4	1	1	-1	13	-1	0	-1
5	-1	1	-1	14	1	1	0
6	-1	-1	-1	15	-1	1	0
7	-1	0	0	16	0	1	-1
8	1	0	-1	17	0	0	1
9	0	-1	0				

Table E.12. $\mathcal{D}$-optimal design for example in §5.5.3, for quadratic model only.

Appendix F

Updating Formulae for Split-Plot Exchange Algorithms

F.1 Derivation of Convenient Form of the Split-Plot Information Matrix

In this appendix, we will review the derivation of a particular form of the split-plot information matrix, for a given model form. Fist, to calculate the inverse of $\mathbf{W}$, we need a lemma that is given and proven in Miller [78].

Lemma 8. *Let* $\mathbf{G}$ *be a nonsingular square matrix and* $\mathbf{H}$ *be a rank one matrix of the same dimension. Then,*

$$(\mathbf{G}+\mathbf{H})^{-1} = \mathbf{G}^{-1} - \frac{1}{1+g}\mathbf{G}^{-1}\mathbf{H}\mathbf{G}^{-1}$$

where $g = tr(\mathbf{H}\mathbf{G}^{-1})$

Then, as given in Goos and Vandebroek [49], $\mathbf{W}_i^{-1}$ can be calculated by appealing to Lemma 8, noting that $tr(\eta \mathbf{1}_{s_i}\mathbf{1}_{s_i}^{'}) = ds_i$:

$$\mathbf{W}_i^{-1} = \frac{1}{\sigma_\epsilon^2}\left(\mathbf{I}_{s_i} - \frac{\eta}{1+s_i\eta}\mathbf{1}_{s_i}\mathbf{1}_{s_i}^{'}\right)$$

Then, because $\mathbf{W}$ is block diagonal,

$$\mathbf{W}^{-1} = \begin{pmatrix} \mathbf{W}_1^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2^{-1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{W}_b^{-1} \end{pmatrix}$$

Now, we write

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \dots \\ \mathbf{X}_b \end{pmatrix}$$

where $\mathbf{X}_i$ is a $s_i \times p$ matrix:

$$\mathbf{X}_i = \begin{pmatrix} \mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{i1}) \\ \mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{i2}) \\ \dots \\ \mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{is_i}) \end{pmatrix}$$

Then matrix multiplication shows that the information matrix, $\mathbf{M}_{sp}$, is

$$\begin{aligned} \mathbf{M}_{sp} = \mathbf{X}'\mathbf{W}^{-1}\mathbf{X} &= \begin{pmatrix} \mathbf{X}_1' & \mathbf{X}_2' & \dots & \mathbf{X}_b' \end{pmatrix} \begin{pmatrix} \mathbf{W}_1^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2^{-1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{W}_b^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \dots \\ \mathbf{X}_b \end{pmatrix} \\ &= \sum_{i=1}^{b} \mathbf{X}_i' \mathbf{W}_i^{-1} \mathbf{X}_i \end{aligned}$$

It can be shown [49] that

$$\mathbf{X}'\mathbf{V}^{-1}\mathbf{X} = \frac{1}{\sigma_\epsilon^2} \left(\sum_{i=1}^{b} \sum_{j=1}^{s_i} \mathbf{f}(\mathbf{z}_i, \mathbf{x}_{ij}) \mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{ij}) - \sum_{i=1}^{b} \frac{\eta}{1 + s_i \eta} (\mathbf{X}_i' \mathbf{1}_{s_i}) (\mathbf{X}_i' \mathbf{1}_{s_i})' \right) \quad \text{(F.1)}$$

This is convenient as an update formula for the information matrix. For in-

stance, if we want to add a point to the i^{th} whole plot:

$$\begin{aligned}\tilde{\mathbf{M}}_{sp} = \tilde{\mathbf{X}}'\tilde{\mathbf{W}}^{-1}\tilde{\mathbf{X}} = \mathbf{X}'\mathbf{W}^{-1}\mathbf{X} &+ \mathbf{f}(\mathbf{z}_i, \mathbf{x}_{i,s_i+1})\mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{i,s_i+1}) \\ &+ \frac{\eta}{1+s_i\eta}(\mathbf{X}_i'\mathbf{1}_{s_i})(\mathbf{X}_i'\mathbf{1}_{s_i})' \\ &- \frac{\eta}{1+(s_i+1)\eta}(\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i+1})(\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i+1})'\end{aligned} \tag{F.2}$$

where the last two parts together give the impact of the additional observation on the last part of the information matrix in (F.1). We can rewrite (F.2) in the form $\mathbf{R} + \mathbf{STU}$, so that we can use the well-known matrix inversion lemma (see [5]):

$$(\mathbf{R} + \mathbf{STU})^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{S}\left(\mathbf{T}^{-1} + \mathbf{UR}^{-1}\mathbf{S}\right)^{-1}\mathbf{UR}^{-1}$$

where $\mathbf{R}$ is $r \times r$, $\mathbf{S}$ is $r \times t$, $\mathbf{T}$ is $t \times t$ and $\mathbf{U}$ is $t \times r$. For this case, we have

$$\begin{aligned}\tilde{\mathbf{M}} &= \mathbf{M} + \begin{pmatrix}\mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{i,s_i+1}) \\ (\mathbf{X}_i'\mathbf{1}_{s_i})' \\ (\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i+1})'\end{pmatrix}' \begin{pmatrix}1 & 0 & 0 \\ 0 & \frac{d}{1+s_id} & 0 \\ 0 & 0 & \frac{d}{1+(s_i+1)d}\end{pmatrix} \begin{pmatrix}\mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{i,s_i+1}) \\ (\mathbf{X}_i'\mathbf{1}_{s_i})' \\ (\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i+1})'\end{pmatrix} \\ &= \mathbf{R} + \mathbf{STU}\end{aligned}$$

so that

$$\tilde{\mathbf{M}}^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{S}\left(\mathbf{T}^{-1} + \mathbf{UR}^{-1}\mathbf{S}\right)^{-1}\mathbf{UR}^{-1}.$$

F.2 Matrix Results Used in Split-Plot Exchange Algorithms

For reference purposes, we reproduce some results from [5] which speed up our algorithm. Exchanges in the split-plot situation are more complicated than the completely randomized scenario because there are several possible types of exchanges, each having a different impact upon the determinant and inverse of the information matrix. In what follows we give updating formulae for the information matrix, its inverse, and its determinant, for each of the three relevant exchange scenarios.

F.2.1 Updating Formulae for Changes in Easy-to-Change Factors

First, we consider the case in which a design point is exchanged for a candidate point, both having the same whole plot factor levels. The design point is $(\mathbf{z}_i, \mathbf{x}_{ij})$, whereas the candidate point to be exchanged is $(\mathbf{z}_i, \mathbf{x}^*_{ij})$. It is shown in Arnouts and Goos [5] that:

$$\begin{aligned}\tilde{\mathbf{M}}_{sp} = \mathbf{M}_{sp} &- \mathbf{f}(\mathbf{z}_i, \mathbf{x}_{ij})\mathbf{f}'(\mathbf{z}_i, \mathbf{x}_{ij}) + \mathbf{f}(\mathbf{z}_i, \mathbf{x}^*_{ij})\mathbf{f}'(\mathbf{z}_i, \mathbf{x}^*_{ij}) \\ &+ \frac{\eta}{1+\eta s_i}\left(\mathbf{X}_i'\mathbf{1}_{s_i}\right)\left(\mathbf{X}_i'\mathbf{1}_{s_i}\right)' - \frac{\eta}{1+\eta s_i}\left(\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i}\right)\left(\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i}\right)', \end{aligned} \tag{F.3}$$

$$\left|\tilde{\mathbf{M}}_{sp}\right| = \left|\mathbf{M}_{sp}\right| \left|\mathbf{D}_1\right| \left|\mathbf{D}_1^{-1} + \mathbf{U}_1\mathbf{M}_{sp}^{-1}\mathbf{U}_1'\right| \tag{F.4}$$

and

$$\left(\tilde{\mathbf{M}}_{sp}\right)^{-1} = \mathbf{M}_{sp}^{-1} - \mathbf{M}_{sp}^{-1}\mathbf{U}_1'\left(\mathbf{D}_1^{-1} + \mathbf{U}_1\mathbf{M}_{sp}^{-1}\mathbf{U}_1'\right)^{-1}\mathbf{U}_1\mathbf{M}_{sp}^{-1} \tag{F.5}$$

where $\mathbf{D}_1 = diag\left(-1, 1, \frac{\eta}{1+s_i\eta}, -\frac{\eta}{1+s_i\eta}\right)$,

$$\mathbf{U}_1 = \left[\mathbf{f}(\mathbf{z}_i, \mathbf{x}_{ij}), \mathbf{f}(\mathbf{z}_i, \mathbf{x}^*_{ij}), \left(\mathbf{X}_i'\mathbf{1}_{s_i}\right), \left(\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i}\right)\right]',$$

$|\mathbf{D}_1| = \frac{\eta^2}{(1+s_i\eta)^2}$, and $\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i} = \mathbf{X}_i'\mathbf{1}_{s_i} - \mathbf{f}(\mathbf{z}_i, \mathbf{x}_{ij}) + \mathbf{f}(\mathbf{z}_i, \mathbf{x}^*_{ij})$.

F.2.2 Swapping Two Points From Different Whole Plots

Another way to perturb a split-plot design is to exchange existing design points from two different whole plots. To perform such a swap, the whole plot factor levels for both must be the same (otherwise, the result would be whole plots with more than one level of whole plot factors). Note that the change does not affect the quantity $\mathbf{X}'\mathbf{X}$ because of the simple change in ordering of the two design points. However, it does affect whole plot part of the information matrix:

$$\begin{aligned}\tilde{\mathbf{M}}_{sp} = \mathbf{M}_{sp} &+ \frac{\eta}{1+\eta s_i}\left(\mathbf{X}_i'\mathbf{1}_{s_i}\right)\left(\mathbf{X}_i'\mathbf{1}_{s_i}\right)' + \frac{\eta}{1+\eta s_l}\left(\mathbf{X}_l'\mathbf{1}_{s_l}\right)\left(\mathbf{X}_l'\mathbf{1}_{s_l}\right)' \\ &- \frac{\eta}{1+\eta s_i}\left(\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i}\right)\left(\tilde{\mathbf{X}}_i'\mathbf{1}_{s_i}\right)' - \frac{\eta}{1+\eta s_l}\left(\tilde{\mathbf{X}}_l'\mathbf{1}_{s_l}\right)\left(\tilde{\mathbf{X}}_l'\mathbf{1}_{s_l}\right)', \end{aligned} \tag{F.6}$$

$$\left|\tilde{\mathbf{M}}_{sp}\right| = \left|\mathbf{M}_{sp}\right| \left|\mathbf{D}_2\right| \left|\mathbf{D}_2^{-1} + \mathbf{U}_2\mathbf{M}_{sp}^{-1}\mathbf{U}_2^{'}\right| \tag{F.7}$$

and

$$\left(\tilde{\mathbf{M}}_{sp}\right)^{-1} = \mathbf{M}_{sp}^{-1} - \mathbf{M}_{sp}^{-1}\mathbf{U}_2^{'}\left(\mathbf{D}_2^{-1} + \mathbf{U}_2\mathbf{M}_{sp}^{-1}\mathbf{U}_2^{'}\right)^{-1}\mathbf{U}_2\mathbf{M}_{sp}^{-1} \tag{F.8}$$

where $\mathbf{D}_2 = diag\left(\frac{\eta}{1+s_i\eta}, \frac{\eta}{1+s_l\eta}, -\frac{\eta}{1+s_i\eta}, -\frac{\eta}{1+s_l\eta}\right)$,

$$\mathbf{U}_2 = \left[\left(\mathbf{X}_i^{'}\mathbf{1}_{s_i}\right), \left(\mathbf{X}_l^{'}\mathbf{1}_{s_l}\right), \left(\tilde{\mathbf{X}}_i^{'}\mathbf{1}_{s_i}\right), \left(\tilde{\mathbf{X}}_l^{'}\mathbf{1}_{s_l}\right)\right],$$

$|\mathbf{D}_2| = \frac{\eta^4}{(1+s_i\eta)^2(1+s_l\eta)^2}$, $\tilde{\mathbf{X}}_i^{'}\mathbf{1}_{s_i} = \mathbf{X}_i^{'}\mathbf{1}_{s_i} - \mathbf{f}(\mathbf{z}_i, \mathbf{x}_{ij}) + \mathbf{f}(\mathbf{z}_l, \mathbf{x}_{lm})$, and $\tilde{\mathbf{X}}_l^{'}\mathbf{1}_{s_l} = \mathbf{X}_l^{'}\mathbf{1}_{s_l} - \mathbf{f}(\mathbf{z}_l, \mathbf{x}_{lm}) + \mathbf{f}(\mathbf{z}_i, \mathbf{x}_{ij})$.

F.2.3 Updating Formulae for Changes in Hard-to-Change Factors

If the levels of the whole plot factors are changed from z_i to z_i^* for any run ij, the levels of those factors for all runs in whole plot i will have to be changed as well because of the restriction that the whole-plot factor levels be the same within a given whole plot. Arnouts and Goos [5] show that:

$$\begin{aligned}\tilde{\mathbf{M}}_{sp} = \mathbf{M}_{sp} &- \mathbf{X}_i^{'}\mathbf{X}_i + \tilde{\mathbf{X}}_i^{'}\tilde{\mathbf{X}}_i \\ &+ \frac{\eta}{1+\eta s_i}\left(\mathbf{X}_i^{'}\mathbf{1}_{s_i}\right)\left(\mathbf{X}_i^{'}\mathbf{1}_{s_i}\right)^{'} - \frac{\eta}{1+\eta s_i}\left(\tilde{\mathbf{X}}_i^{'}\mathbf{1}_{s_i}\right)\left(\tilde{\mathbf{X}}_i^{'}\mathbf{1}_{s_i}\right)^{'},\end{aligned} \tag{F.9}$$

$$\left|\tilde{\mathbf{M}}_{sp}\right| = \left|\mathbf{M}_{sp}\right| \left|\mathbf{D}_3\right| \left|\mathbf{D}_3^{-1} + \mathbf{U}_3\mathbf{M}_{sp}^{-1}\mathbf{U}_3^{'}\right| \tag{F.10}$$

and

$$\left(\tilde{\mathbf{M}}_{sp}\right)^{-1} = \mathbf{M}_{sp}^{-1} - \mathbf{M}_{sp}^{-1}\mathbf{U}_3^{'}\left(\mathbf{D}_3^{-1} + \mathbf{U}_3\mathbf{M}_{sp}^{-1}\mathbf{U}_3^{'}\right)^{-1}\mathbf{U}_3\mathbf{M}_{sp}^{-1} \tag{F.11}$$

where $\mathbf{D}_3 = diag\left(-\mathbf{I}_{s_i}, \mathbf{I}_{s_i}, \frac{\eta}{1+s_i\eta}, -\frac{\eta}{1+s_i\eta}\right)$,

$$\mathbf{U}_3 = \left[\mathbf{X}_i^{'}, \tilde{\mathbf{X}}_i^{'}, \left(\mathbf{X}_i^{'}\mathbf{1}_{s_i}\right), \left(\tilde{\mathbf{X}}_i^{'}\mathbf{1}_{s_i}\right)\right],$$

$|\mathbf{D}_3| = \frac{\eta^3}{(1+s_i\eta)^3}(-1)^{s_i+1}$, and $\tilde{\mathbf{X}}_i^{'} = \left[\mathbf{f}(\mathbf{z}_i^*, \mathbf{x}_{i1}), \ldots, \mathbf{f}(\mathbf{z}_i^*, \mathbf{x}_{is_i})\right]^{'}$.

Appendix G

$\mathcal{D}$-optimal Split-Plot Algorithm of Goos and Vandebroek [50]

Here we give the original algorithm of [50], generally using the same notation as in §6.3.2. For the algorithm, we must specify 1) The candidate list, C; 2) the number of whole plots, b; 3) the size of each whole plot, $k_1, k_2, \ldots, k_b$; and 4) the model form $\mathbf{f}'(\mathbf{z}, \mathbf{x})$. Let $\mathbf{M}^*_{sp}$ be the information matrix for the best design found so far. The algorithm is as follows.

1. Set $|\mathbf{M}^*_{sp}| = 0$, $t_c = 1$.
2. Determine p_w, the number of coefficients for whole plot factors only.
3. Determine p_s, the number of coefficients for subplot factors only.
4. Set $\mathbf{M}_{sp} = \omega\mathbf{I}$ (where ω is a small constant) and calculate $\mathbf{M}^{-1}_{sp}$; set $H_i = \emptyset$.
5. Construct initial design
   (a) Randomly assign p_w unique whole-plot factor settings to p_w whole plots.
   (b) Randomly assign $b - p_w$ levels of the whole-plot factors to the rest of the whole plots.
   (c) Randomly choose u ($1 \leq u \leq p$).
   (d) Do u times:
      i. Randomly select $i \in B$ (select a whole plot at random)

ii. Randomly select $j \in C_i$ (select a candidate point with the i^{th} setting of whole plot, at random)

iii. If $\#H_i < k_i$, then $H_i = H_i \cup j$; otherwise, go back to step i.

iv. Update $\mathbf{M}_{sp}^{-1}$ via (F.1).

(e) Do $n - u$ times:

i. Set $l = 1$.

ii. Determine $j \in C$ with the l^{th} biggest prediction variance via (6.5).

iii. Find i, where $i \in B$, $j \in C_i$, and $\#H_i < k_i$ (find a nonfull whole plot into which j can be inserted). If no such i exists, set $l = l + 1$ and return to step ii.

iv. $H_i = H_i \cup j$.

v. Update $\mathbf{M}_{sp}^{-1}$ via (F.1).

6. Compute $\mathbf{M}_{sp}$ and $|\mathbf{M}_{sp}|$ for the initial design. If $|\mathbf{M}_{sp}| = 0$, go back to step 4. Otherwise, continue.

7. Set $\nu = 0$.

8. Evaluate design point exchanges (swapping design points with candidate points, where whole plot factors settings are the same):

(a) Set $\delta = 1$.

(b) $\forall i \in B$, $\forall j \in H_i$, $\forall k \in C_i$, $j \neq k$:

i. Determine the effect $\delta_{jk}^i = |\mathbf{M}_{jk,sp}^i|/|\mathbf{M}_{sp}|$ of exchanging, in the i^{th} whole plot, points j and k, using (F.4).

ii. if $\delta_{jk}^i > \delta$, then $\delta = \delta_{jk}^i$ and store i, j, and k.

9. If $\delta > 1 + \epsilon$, then go to step 10; otherwise, go to step 11.

10. Perform the best exchange:

(a) $H_i = H_i \setminus j \cup k$.

(b) Update $\mathbf{M}_{sp}^{-1}$ via (F.5) and $|\mathbf{M}_{sp}|$ via (F.4).

(c) Set $\nu = 1$.

11. Evaluate interchanges of points within whole plots with the same factor levels

    (a) Set $\delta = 1$.

    (b) $\forall i, j \in B$, $i < j$, $\mathbf{z}_i = \mathbf{z}_j$, $\forall k \in H_i$, $\forall l \in H_j$, $k \neq l$:

        i. Determine the effect $\delta_{ik}^{jl} = |\mathbf{M}_{ik,sp}^{jl}|/|\mathbf{M}_{sp}|$ of moving k to whole plot j (from whole plot i) and l to whole plot i (from whole plot j), via (F.7).

        ii. If $\delta_{ik}^{jl} > \delta$, then $\delta = \delta_{ik}^{jl}$ and store i, j, k, and l.

12. If $\delta > 1 + \epsilon$, go to step 13; otherwise, go to step 14.

13. Perform the best interchange:

    (a) $H_i = H_i \setminus k \cup l$.

    (b) $H_j = H_j \setminus l \cup k$.

    (c) Update $\mathbf{M}_{sp}^{-1}$ and $|\mathbf{M}_{sp}|$ via (F.7).

    (d) Set $\nu = 1$.

14. Evaluate exchanges of whole-plot factor settings:

    (a) Set $\delta = 1$.

    (b) $\forall i \in B$, $\forall j \in P$, $\mathbf{z}_i \neq \mathbf{z}_j$:

        i. Determine the effect $\delta_{ij} = |\mathbf{M}_{ij,sp}|/|\mathbf{M}_{sp}|$ of exchanging $\mathbf{z}_i$ by $\mathbf{z}_j$ in the i^{th} whole plot, via (F.10).

        ii. If $\delta_{ij} > \delta$, then $\delta = \delta_{ij}$ and store i and j.

15. If $\delta > 1 + \epsilon$, go to step 16; otherwise, go to step 17.

16. Perform best exchange:

    (a) Update H_i and C_i.

    (b) Update $\mathbf{M}_{sp}^{-1}$ via (F.11) and $|\mathbf{M}_{sp}|$ via (F.10).

    (c) Set $\nu = 1$.

17. If $\nu = 1$, go to step 7.

18. If $|\mathbf{M}_{sp}| > |\mathbf{M}^*_{sp}|$, then $|\mathbf{M}^*_{sp}| = |\mathbf{M}_{sp}|$; also, update the design ξ_{nb}.

19. If $t_c < t$, then $t_c = t_c + 1$, and go back to step 4; otherwise, STOP.

Notes: 1) This algorithm, including Step 5 which generates the initial design, generalizes Goos and Vandebroek [50] but follows its basic outline. 2) In 13(c), we recalculate $\mathbf{M}^{-1}$ by directly taking its inverse, because of numerical issues associated with the updated formula in (F.8). This should have a minimal effect on the speed of the algorithm because it occurs only once per iteration. 3) In steps 9, 12, and 15, we have changed the original algorithm of Goos and Vandebroek [50] slightly by requiring that, for any considered exchange/interchange, the increase in the minimum efficiency is greater than ϵ (instead of 0, as it was originally), to encourage algorithmic stability.

Notes: 1) In step 13(c), we update $\mathbf{M}^{-1}$ by recalculating the information matrix after the interchange and directly taking its inverse, because of numerical inaccuracies incurred by using (F.8). 2) In steps 9, 12, and 15, as with the maximin version of this algorithm, we have changed the original algorithm slightly by requiring that, for any considered exchange/interchange, the multiplicative difference in the determinant of the information matrix is greater than ϵ (instead of 1, as it was originally). Without this adjustment, we found that the algorithm was unstable.

Appendix H

Designs for Chapter 6

H.1 Designs for Example in §6.4.1

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	-1	1	0	1	25	-1	1	-1	-1
2	-1	1	1	-0.5	26	-1	1	1	1
3	-1	1	-1	1	27	-1	1	1	-1
4	-1	1	-1	-1	28	-1	1	-1	1
5	-1	-1	1	-1	29	-1	-1	-1	-1
6	-1	-1	-1	-0.5	30	-1	-1	1	1
7	-1	-1	-1	1	31	-1	-1	-1	1
8	-1	-1	1	1	32	-1	-1	1	-1
9	1	1	-1	1	33	1	-1	1	1
10	1	1	1	-1	34	1	-1	-1	-1
11	1	1	-1	-1	35	1	-1	0.5	-1
12	1	1	1	1	36	1	-1	-1	1
13	1	0	1	-1	37	1	1	-1	-1
14	1	0	-1	-1	38	1	1	1	1
15	1	0	-1	1	39	1	1	1	-1
16	1	0	1	1	40	1	1	-1	1
17	-1	1	-0.5	-1	41	1	-1	1	1
18	-1	1	1	-1	42	1	-1	-1	1
19	-1	1	-1	1	43	1	-1	-1	-1
20	-1	1	1	1	44	1	-1	1	-1
21	0	1	-1	-1	45	-1	-1	0	1
22	0	1	1	1	46	-1	-1	-1	-1
23	0	1	1	-1	47	-1	-1	1	0.5
24	0	1	-1	0.5	48	-1	-1	1	-1

Table H.1. $\mathcal{F}_1$-maximin model-robust split-plot design, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	1	-1	-1	1	25	0	1	-1	-1
2	1	-1	1	-1	26	0	1	1	-1
3	1	-1	-1	-1	27	0	1	-1	1
4	1	-1	1	1	28	0	1	1	1
5	-1	0	0	0	29	1	-1	1	-1
6	-1	0	1	-1	30	1	-1	-1	0
7	-1	0	-1	-1	31	1	-1	-0.5	1
8	-1	0	1	1	32	1	-1	1	1
9	1	-1	1	-0.5	33	1	1	1	1
10	1	-1	-1	1	34	1	1	-1	1
11	1	-1	-1	-1	35	1	1	-1	-1
12	1	-1	1	1	36	1	1	1	-1
13	-1	1	0	1	37	-1	-1	0	-1
14	-1	1	1	-1	38	-1	-1	1	-1
15	-1	1	-1	1	39	-1	-1	-1	1
16	-1	1	-1	-1	40	-1	-1	1	1
17	-0.5	-1	-1	1	41	-1	-1	1	-1
18	-0.5	-1	-1	-1	42	-1	-1	-1	1
19	-0.5	-1	0	1	43	-1	-1	-1	-1
20	-0.5	-1	1	0	44	-1	-1	1	1
21	-1	1	-1	1	45	1	1	1	-0.5
22	-1	1	1	1	46	1	1	-1	-1
23	-1	1	1	-1	47	1	1	1	1
24	-1	1	-1	-1	48	1	1	-1	1

Table H.2. $(.9, .9, 1)$-$\mathcal{F}_1$-maximin model-robust split-plot design, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	-1	0.5	1	0	25	-1	-0.5	-1	-1
2	-1	0.5	-1	1	26	-1	-0.5	-1	1
3	-1	0.5	-1	-1	27	-1	-0.5	1	1
4	-1	0.5	1	-1	28	-1	-0.5	1	-1
5	-1	-1	1	1	29	-1	-1	-1	1
6	-1	-1	1	-1	30	-1	-1	-1	-0.5
7	-1	-1	-0.5	1	31	-1	-1	1	0.5
8	-1	-1	-1	-1	32	-1	-1	1	-1
9	1	-1	0.5	1	33	1	1	1	-1
10	1	-1	0	-1	34	1	1	-1	1
11	1	-1	-1	0.5	35	1	1	-1	-1
12	1	-1	1	-1	36	1	1	1	1
13	-0.5	1	-1	-1	37	1	-1	1	1
14	-0.5	1	0.5	-1	38	1	-1	1	-1
15	-0.5	1	-1	1	39	1	-1	-1	1
16	-0.5	1	1	1	40	1	-1	-1	-1
17	0.5	-1	1	-0.5	41	1	0.5	-1	-1
18	0.5	-1	-1	1	42	1	0.5	1	1
19	0.5	-1	1	1	43	1	0.5	1	-0.5
20	0.5	-1	-1	-1	44	1	0.5	-0.5	1
21	-1	1	1	-1	45	0.5	1	-0.5	-1
22	-1	1	-1	0.5	46	0.5	1	1	1
23	-1	1	1	1	47	0.5	1	-1	1
24	-1	1	-1	-1	48	0.5	1	1	-1

Table H.3. $\mathcal{F}_2$-maximin model-robust split-plot design, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	-0.5	-1	-1	1	25	1	0	0	0
2	-0.5	-1	-1	-1	26	1	0	-1	1
3	-0.5	-1	0	-1	27	1	0	1	-1
4	-0.5	-1	1	0	28	1	0	1	1
5	-1	-0.5	-1	0	29	-0.5	1	0	-1
6	-1	-0.5	0	1	30	-0.5	1	1	1
7	-1	-0.5	1	-1	31	-0.5	1	1	0
8	-1	-0.5	1	1	32	-0.5	1	-1	1
9	-1	0.5	-1	-1	33	0	1	-1	0
10	-1	0.5	0	1	34	0	1	1	1
11	-1	0.5	1	-0.5	35	0	1	-1	-1
12	-1	0.5	-1	1	36	0	1	1	-1
13	1	-1	-1	-1	37	1	-1	1	1
14	1	-1	1	-1	38	1	-1	-1	-1
15	1	-1	1	1	39	1	-1	-1	1
16	1	-1	-1	1	40	1	-1	1	-1
17	-1	1	1	-1	41	1	1	-1	-1
18	-1	1	-1	1	42	1	1	0.5	1
19	-1	1	-1	-1	43	1	1	-1	1
20	-1	1	1	1	44	1	1	1	-1
21	-1	-1	-1	1	45	1	1	-1	-1
22	-1	-1	1	-1	46	1	1	1	1
23	-1	-1	1	1	47	1	1	1	-0.5
24	-1	-1	-1	-1	48	1	1	-1	1

Table H.4. $(.8, .8, 1, .5)$-$\mathcal{F}_2$-maximin model-robust split-plot design, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	-1	-1	-1	-1	25	0	-1	0	0
2	-1	-1	1	-1	26	0	-1	0	0
3	-1	-1	-1	1	27	0	-1	0	0
4	-1	-1	1	1	28	0	-1	0	0
5	1	-1	-1	-1	29	0	1	0	0
6	1	-1	1	-1	30	0	1	0	0
7	1	-1	-1	1	31	0	1	0	0
8	1	-1	1	1	32	0	1	0	0
9	-1	1	-1	-1	33	0	-1	0	
10	-1	1	1	-1	34	0	0	1	0
11	-1	1	-1	1	35	0	0	0	-1
12	-1	1	1	1	36	0	0	0	1
13	1	1	-1	-1	37	0	0	0	0
14	1	1	1	-1	38	0	0	0	0
15	1	1	-1	1	39	0	0	0	0
16	1	1	1	1	40	0	0	0	0
17	-1	0	0	0	41	0	0	0	0
18	-1	0	0	0	42	0	0	0	0
19	-1	0	0	0	43	0	0	0	0
20	-1	0	0	0	44	0	0	0	0
21	1	0	0	0	45	0	0	0	0
22	1	0	0	0	46	0	0	0	0
23	1	0	0	0	47	0	0	0	0
24	1	0	0	0	48	0	0	0	0

Table H.5. Design used by Vining et al. [103], for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	-1	1	1	1	25	-1	-1	-1	1
2	-1	1	1	-1	26	-1	-1	1	-1
3	-1	1	-1	-1	27	-1	-1	-1	-1
4	-1	1	-1	1	28	-1	-1	1	1
5	1	-1	1	-1	29	-1	1	-1	1
6	1	-1	1	-1	30	-1	1	1	-1
7	1	-1	-1	1	31	-1	1	1	-1
8	1	-1	-1	1	32	-1	1	-1	1
9	1	1	1	1	33	1	1	1	-1
10	1	1	-1	-1	34	1	1	-1	-1
11	1	1	-1	-1	35	1	1	-1	1
12	1	1	1	1	36	1	1	1	1
13	1	-1	1	1	37	-1	1	1	-1
14	1	-1	-1	1	38	-1	1	-1	1
15	1	-1	1	-1	39	-1	1	1	1
16	1	-1	-1	-1	40	-1	1	-1	-1
17	-1	-1	1	1	41	1	1	-1	-1
18	-1	-1	-1	-1	42	1	1	1	1
19	-1	-1	-1	-1	43	1	1	-1	1
20	-1	-1	1	1	44	1	1	1	-1
21	1	-1	1	-1	45	-1	-1	-1	1
22	1	-1	1	1	46	-1	-1	-1	-1
23	1	-1	-1	-1	47	-1	-1	1	1
24	1	-1	-1	1	48	-1	-1	1	-1

Table H.6. Optimal split-plot design for model (6.9) assuming $\eta = 1$, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	1	1	1	-1	25	-1	-1	-1	-1
2	1	1	-1	-1	26	-1	-1	-1	1
3	1	1	1	1	27	-1	-1	1	-1
4	1	1	-1	1	28	-1	-1	1	1
5	-1	-1	1	1	29	-1	1	1	1
6	-1	-1	-1	1	30	-1	1	-1	1
7	-1	-1	1	-1	31	-1	1	-1	-1
8	-1	-1	-1	-1	32	-1	1	1	-1
9	-1	1	1	1	33	1	1	1	-1
10	-1	1	1	-1	34	1	1	1	1
11	-1	1	-1	-1	35	1	1	-1	-1
12	-1	1	-1	1	36	1	1	-1	1
13	1	-1	-1	1	37	1	-1	1	-1
14	1	-1	1	1	38	1	-1	-1	-1
15	1	-1	-1	-1	39	1	-1	1	1
16	1	-1	1	-1	40	1	-1	-1	1
17	1	1	-1	1	41	1	-1	-1	-1
18	1	1	1	1	42	1	-1	-1	1
19	1	1	1	-1	43	1	-1	1	1
20	1	1	-1	-1	44	1	-1	1	-1
21	-1	-1	1	-1	45	-1	1	1	-1
22	-1	-1	-1	1	46	-1	1	-1	1
23	-1	-1	-1	-1	47	-1	1	1	1
24	-1	-1	1	1	48	-1	1	-1	-1

Table H.7. Optimal split-plot design for model (6.10) assuming $\eta = 1$, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	0	-1	-1	0	25	1	1	0	-1
2	0	-1	-1	1	26	1	1	1	0
3	0	-1	1	1	27	1	1	-1	-1
4	0	-1	0	-1	28	1	1	-1	1
5	-1	0	-1	0	29	-1	1	-1	1
6	-1	0	1	-1	30	-1	1	-1	-1
7	-1	0	-1	-1	31	-1	1	1	-1
8	-1	0	0	1	32	-1	1	1	
9	1	1	-1	-1	33	0	-1	1	
10	1	1	1	-1	34	0	1	0	0
11	1	1	1	1	35	0	1	1	1
12	1	1	-1	1	36	0	1	1	-1
13	1	0	0	1	37	-1	-1	0	-1
14	1	0	1	-1	38	-1	-1	-1	-1
15	1	0	-1	0	39	-1	-1	-1	1
16	1	0	1	1	40	-1	-1	1	0
17	-1	-1	1	1	41	-1	1	0	0
18	-1	-1	1	-1	42	-1	1	-1	-1
19	-1	-1	-1	1	43	-1	1	1	1
20	-1	-1	-1	-1	44	-1	1	1	-1
21	1	-1	-1	-1	45	1	-1	1	1
22	1	-1	0	1	46	1	-1	-1	-1
23	1	-1	1	-1	47	1	-1	1	-1
24	1	-1	1	0	48	1	-1	-1	1

Table H.8. Optimal split-plot design for model (6.11) assuming $\eta = 1$, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	0	-1	-0.5	-1	25	0	0	-0.5	1
2	0	-1	0.5	1	26	0	0	-1	-1
3	0	-1	-1	1	27	0	0	1	-1
4	0	-1	1	-0.5	28	0	0	1	1
5	-1	-1	-0.5	1	29	-1	0	1	-0.5
6	-1	-1	1	0.5	30	-1	0	-1	1
7	-1	-1	1	-1	31	-1	0	0.5	1
8	-1	-1	-1	-0.5	32	-1	0	-0.5	-1
9	0.5	0.5	-1	1	33	0.5	1	-1	0
10	0.65	0.17	-1	-0.3	34	0.5	1	1	-1
11	0.5	0	1	0.5	35	0.5	1	1	0
12	0.5	0	0	-1	36	0.5	1	0	1
13	1	-0.5	-1	0	37	1	-1	-0.5	0.5
14	1	-0.5	-0.5	1	38	1	-1	-1	-1
15	1	-0.5	0	-1	39	1	-1	1	1
16	1	-0.5	1	0	40	1	-1	1	-1
17	-1	-1	1	1	41	-1	1	-1	-1
18	-1	-1	-1	-1	42	-1	1	1	1
19	-1	-1	-1	0.5	43	-1	1	1	-1
20	-1	-1	0.5	-0.5	44	-1	1	-0.5	0.5
21	1	1	-1	-1	45	1	0.5	-0.5	-0.5
22	1	1	-1	1	46	1	0.5	0.5	0.5
23	1	1	0.5	-0.5	47	1	0.5	-1	1
24	1	1	1	1	48	1	0.5	1	-1

Table H.9. Optimal split-plot design for model (6.12) assuming $\eta = 1$, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	-1	-1	-1	-1	25	-1	-1	-1	1
2	-1	-1	-1	-1	26	-1	-1	-1	1
3	-1	-1	1	1	27	-1	-1	1	-1
4	-1	-1	1	1	28	-1	-1	1	-1
5	1	1	-1	-1	29	-1	1	-1	1
6	1	1	1	1	30	-1	1	-1	1
7	1	1	-1	-1	31	-1	1	1	-1
8	1	1	1	1	32	-1	1	1	-1
9	1	-1	-1	1	33	-1	-1	1	1
10	1	-1	1	-1	34	-1	-1	-1	1
11	1	-1	1	1	35	-1	-1	-1	-1
12	1	-1	-1	-1	36	-1	-1	1	-1
13	1	-1	1	-1	37	1	1	1	1
14	1	-1	-1	-1	38	1	1	1	-1
15	1	-1	-1	1	39	1	1	-1	1
16	1	-1	1	1	40	1	1	-1	-1
17	-1	1	-1	1	41	-1	1	1	1
18	-1	1	1	-1	42	-1	1	-1	-1
19	-1	1	-1	-1	43	-1	1	1	-1
20	-1	1	1	1	44	-1	1	-1	1
21	1	1	1	-1	45	1	-1	1	1
22	1	1	-1	1	46	1	-1	-1	-1
23	1	1	-1	-1	47	1	-1	1	-1
24	1	1	1	1	48	1	-1	-1	1

Table H.10. Optimal split-plot design for model (6.9) assuming $\eta = 5.65$, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	1	1	-1	1	25	-1	1	1	1
2	1	1	1	1	26	-1	1	-1	1
3	1	1	-1	-1	27	-1	1	1	-1
4	1	1	1	-1	28	-1	1	-1	-1
5	1	1	1	1	29	1	-1	-1	-1
6	1	1	1	-1	30	1	-1	-1	1
7	1	1	-1	1	31	1	-1	1	1
8	1	1	-1	-1	32	1	-1	1	-1
9	1	-1	-1	-1	33	-1	-1	-1	-1
10	1	-1	-1	1	34	-1	-1	1	1
11	1	-1	1	-1	35	-1	-1	1	-1
12	1	-1	1	1	36	-1	-1	-1	1
13	-1	-1	1	-1	37	-1	1	1	1
14	-1	-1	-1	-1	38	-1	1	-1	-1
15	-1	-1	-1	1	39	-1	1	-1	1
16	-1	-1	1	1	40	-1	1	1	-1
17	-1	1	1	1	41	-1	-1	1	1
18	-1	1	1	-1	42	-1	-1	1	-1
19	-1	1	-1	1	43	-1	-1	-1	1
20	-1	1	-1	-1	44	-1	-1	-1	-1
21	1	1	1	-1	45	1	-1	1	-1
22	1	1	1	1	46	1	-1	1	1
23	1	1	-1	-1	47	1	-1	-1	1
24	1	1	-1	1	48	1	-1	-1	-1

Table H.11. Optimal split-plot design for model (6.10) assuming $\eta = 5.65$, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	1	-1	1	1	25	1	-1	-1	-1
2	1	-1	-1	1	26	1	-1	0	1
3	1	-1	1	-1	27	1	-1	-1	1
4	1	-1	-1	-1	28	1	-1	1	0
5	0	1	1	-1	29	-1	-1	1	1
6	0	1	0	-1	30	-1	-1	1	-1
7	0	1	1	1	31	-1	-1	-1	0
8	0	1	-1	0	32	-1	-1	0	-1
9	-1	1	1	0	33	1	1	1	1
10	-1	1	-1	1	34	1	1	1	-1
11	-1	1	-1	-1	35	1	1	-1	-1
12	-1	1	0	1	36	1	1	-1	1
13	-1	-1	1	1	37	1	0	1	1
14	-1	-1	-1	-1	38	1	0	0	0
15	-1	-1	-1	1	39	1	0	1	-1
16	-1	-1	1	-1	40	1	0	-1	-1
17	-1	0	-1	1	41	-1	1	1	-1
18	-1	0	1	0	42	-1	1	-1	-1
19	-1	0	0	1	43	-1	1	-1	1
20	-1	0	-1	-1	44	-1	1	1	1
21	0	-1	0	0	45	1	1	-1	-1
22	0	-1	-1	1	46	1	1	1	1
23	0	-1	1	-1	47	1	1	1	-1
24	0	-1	1	1	48	1	1	-1	1

Table H.12. Optimal split-plot design for model (6.11) assuming $\eta = 5.65$, for the example in §6.4.1.

Run	z_1	z_2	x_1	x_2	Run	z_1	z_2	x_1	x_2
1	-0.5	1	-1	1	25	0.5	1	-0.5	1
2	-0.5	1	1	-0.5	26	0.5	1	-1	0
3	-0.5	1	0.5	1	27	0.5	1	1	1
4	-0.5	1	-0.5	-1	28	0.5	1	0.5	-1
5	-1	0.5	-1	-0.5	29	1	-1	0	1
6	-1	0.5	0.5	-1	30	1	-1	-0.5	-1
7	-1	0.5	-0.5	1	31	1	-1	-1	1
8	-1	0.5	1	0.5	32	1	-1	1	0
9	1	1	0.5	0.5	33	-1	-1	-1	-1
10	1	1	-1	-1	34	-1	-1	1	-1
11	1	1	-1	1	35	-1	-1	-1	0.5
12	1	1	1	-1	36	-1	-1	0	1
13	1	-1	1	1	37	-1	-0.5	0.5	0
14	1	-1	1	-1	38	-1	-0.5	-0.5	-1
15	1	-1	-1	-1	39	-1	-0.5	-1	1
16	1	-1	-0.5	0	40	-1	-0.5	1	1
17	0	-1	1	0.5	41	1	0	1	-0.5
18	0	-1	-1	-0.5	42	1	0	1	1
19	0	-1	0.5	-1	43	1	0	-1	0.5
20	0	-1	-1	1	44	1	0	0	-1
21	-1	1	-0.5	0	45	0	0	-1	1
22	-1	1	1	-1	46	0	0	-1	-1
23	-1	1	1	1	47	0	0	1	1
24	-1	1	-1	-1	48	0	0	1	-1

Table H.13. Optimal split-plot design for model (6.12) assuming $\eta = 5.65$, for the example in §6.4.1.

H.2 Designs for Example in §6.4.2

Run	z_1	z_2	x_1	x_2	x_3
1	-1	1	0	0	1
2	-1	1	0.5	0.5	0
3	-1	1	1	0	0
4	-1	1	0	1	0
5	1	-1	1	0	0
6	1	-1	0.5	0	0.5
7	1	-1	0	1	0
8	1	-1	0	0	1
9	1	1	1	0	0
10	1	1	0	0.5	0.5
11	1	1	0.5	0	0.5
12	1	1	0	1	0
13	-1	1	0	0.5	0.5
14	-1	1	1	0	0
15	-1	1	0	1	0
16	-1	1	0	0	1
17	1	1	0.5	0	0.5
18	1	1	0	1	0
19	1	1	0	0	1
20	1	1	1	0	0
21	-1	-1	0.5	0.5	0
22	-1	-1	0	0	1
23	-1	-1	0	1	0
24	-1	-1	1	0	0
25	-1	-1	0	1	0
26	-1	-1	0.5	0.5	0
27	-1	-1	0	0	1
28	-1	-1	1	0	0

Table H.14. $\mathcal{F}_1$-maximin model-robust design, for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	-1	0	0.5	0.5
2	-1	-1	0	1	0
3	-1	-1	0	0	1
4	-1	-1	1	0	0
5	1	1	0	1	0
6	1	1	0	0	1
7	1	1	1	0	0
8	1	1	0	0	1
9	-1	1	0.5	0	0.5
10	-1	1	0	0	1
11	-1	1	1	0	0
12	-1	1	0	1	0
13	-1	-1	0	1	0
14	-1	-1	0.5	0.5	0
15	-1	-1	0	0	1
16	-1	-1	1	0	0
17	-1	1	0	0	1
18	-1	1	0	1	0
19	-1	1	0	0.5	0.5
20	-1	1	1	0	0
21	1	-1	0	0	1
22	1	-1	1	0	0
23	1	-1	0.5	0	0.5
24	1	-1	0	1	0
25	1	-1	0	0	1
26	1	-1	0.5	0.5	0
27	1	-1	0	1	0
28	1	-1	1	0	0

Table H.15. $(.7, .9, 1, 1, 1, 1)$-$\mathcal{F}_1$-maximin model-robust design, for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	1	0	0	1
2	1	1	1	0	0
3	1	1	0.5	0	0.5
4	1	1	0	1	0
5	-1	1	0.5	0.5	0
6	-1	1	0	1	0
7	-1	1	0	0	1
8	-1	1	1	0	0
9	1	1	1	0	0
10	1	1	0	1	0
11	1	1	0	0.5	0.5
12	1	1	0	0	1
13	0	-1	0.5	0.5	0
14	0	-1	0.5	0	0.5
15	0	-1	0	1	0
16	0	-1	0	0	1
17	1	-1	1	0	0
18	1	-1	0	1	0
19	1	-1	1	0	0
20	1	-1	0	0	1
21	-1	-1	1	0	0
22	-1	-1	2/3	1/6	1/6
23	-1	-1	0	0	1
24	-1	-1	0	1	0
25	-1	0	0	0.5	0.5
26	-1	0	0	0	1
27	-1	0	1	0	0
28	-1	0	0	1	0

Table H.16. $(1, 1, 1, 1, 1, 1, .5, .5, .5, .5, .5, .5)$-$\mathcal{F}_2$-maximin design, for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	1	1	0	0
2	-1	1	0	1	0
3	-1	1	0	0	1
4	-1	1	1/3	1/3	1/3
5	1	-1	1	0	0
6	1	-1	0	1	0
7	1	-1	0	0	1
8	1	-1	1/3	1/3	1/3
9	1	1	0.5	0.5	0
10	1	1	0.5	0	0.5
11	1	1	0	0.5	0.5
12	1	1	1/3	1/3	1/3
13	-1	-1	0.5	0.5	0
14	-1	-1	0.5	0	0.5
15	-1	-1	0	0.5	0.5
16	-1	-1	1/3	1/3	1/3
17	0	0	1/3	1/3	1/3
18	0	0	1/3	1/3	1/3
19	0	0	1/3	1/3	1/3
20	0	0	1/3	1/3	1/3
21	0	0	1/3	1/3	1/3
22	0	0	1/3	1/3	1/3
23	0	0	1/3	1/3	1/3
24	0	0	1/3	1/3	1/3
25	0	0	1/3	1/3	1/3
26	0	0	1/3	1/3	1/3
27	0	0	1/3	1/3	1/3
28	0	0	1/3	1/3	1/3

Table H.17. Design from Kowalski et al. [68], for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	0	0.5	0.5	0
2	1	0	0.5	0	0.5
3	1	0	0	0	1
4	1	0	0	1	0
5	1	1	0	0	1
6	1	1	0	0.5	0.5
7	1	1	0	1	0
8	1	1	1	0	0
9	0	1	0	1	0
10	0	1	1/3	1/3	1/3
11	0	1	0	0	1
12	0	1	1	0	0
13	-1	-1	0	1	0
14	-1	-1	0.5	0	0.5
15	-1	-1	1	0	0
16	-1	-1	0	0	1
17	1	-1	0	0.5	0.5
18	1	-1	0	1	0
19	1	-1	0	0	1
20	1	-1	1	0	0
21	-1	-1	0	1	0
22	-1	-1	0.5	0.5	0
23	-1	-1	0	0	1
24	-1	-1	1	0	0
25	-1	1	0	0	1
26	-1	1	0	0.5	0.5
27	-1	1	1	0	0
28	-1	1	0	1	0

Table H.18. $\mathcal{F}_2$-maximin model-robust design, for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	-1	0	0	1
2	1	-1	0	1	0
3	1	-1	1	0	0
4	1	-1	1	0	0
5	1	1	1	0	0
6	1	1	1/3	1/3	1/3
7	1	1	0	1	0
8	1	1	0	0	1
9	1	1	0	1	0
10	1	1	0.5	0	0.5
11	1	1	1	0	0
12	1	1	0	0	1
13	-1	0	0	0.5	0.5
14	-1	0	0	0	1
15	-1	0	0	1	0
16	-1	0	1	0	0
17	-1	1	0	0.5	0.5
18	-1	1	0	1	0
19	-1	1	1	0	0
20	-1	1	0	0	1
21	-1	-1	1	0	0
22	-1	-1	0.5	0.5	0
23	-1	-1	0	1	0
24	-1	-1	0	0	1
25	0	-1	0.5	0	0.5
26	0	-1	0.5	0.5	0
27	0	-1	0	1	0
28	0	-1	0	0	1

Table H.19. $(1, 1, 1, 1, 1, 1, .9, .9, .9, .9, .9, .9)$-$\mathcal{F}_2$-maximin design, for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	0	0.5	0	0.5
2	1	0	1	0	0
3	1	0	0	0	1
4	1	0	0	1	0
5	-1	-1	0	0	1
6	-1	-1	0	0.5	0.5
7	-1	-1	1	0	0
8	-1	-1	0	1	0
9	-1	1	0	0.5	0.5
10	-1	1	0	0	1
11	-1	1	1	0	0
12	-1	1	0	1	0
13	-1	-1	0.5	0	0.5
14	-1	-1	0	1	0
15	-1	-1	1	0	0
16	-1	-1	0	0	1
17	1	1	0	1	0
18	1	1	1/3	1/3	1/3
19	1	1	1	0	0
20	1	1	0	0	1
21	1	-1	0.5	0.5	0
22	1	-1	1	0	0
23	1	-1	0	0	1
24	1	-1	0	1	0
25	0	1	1	0	0
26	0	1	0	1	0
27	0	1	0.5	0.5	0
28	0	1	0	0	1

Table H.20. $\mathcal{F}_3$-maximin model-robust design, for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	-1	1	0	0
2	-1	-1	0	0	1
3	-1	-1	0.5	0	0.5
4	-1	-1	0	1	0
5	-1	1	0	1	0
6	-1	1	0	0.5	0.5
7	-1	1	0	0	1
8	-1	1	1	0	0
9	1	-1	0.5	0	0.5
10	1	-1	1	0	0
11	1	-1	0	0	1
12	1	-1	0	1	0
13	0	1	0	0.5	0.5
14	0	1	0.5	0.5	0
15	0	1	0.5	0	0.5
16	0	1	0	1	0
17	1	1	1	0	0
18	1	1	0	0	1
19	1	1	1/3	1/3	1/3
20	1	1	0	1	0
21	1	0	0.5	0.5	0
22	1	0	0	0	1
23	1	0	0	0.5	0.5
24	1	0	1	0	0
25	-1	-1	1	0	0
26	-1	-1	0	0	1
27	-1	-1	0.5	0.5	0
28	-1	-1	0	1	0

Table H.21. $(.7, 1, .9)$-$\mathcal{F}_3$-maximin model-robust design, for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	1	0	0	1
2	-1	1	1	0	0
3	-1	1	1	0	0
4	-1	1	0	1	0
5	0	1	0	0	1
6	0	1	1	0	0
7	0	1	0	0	1
8	0	1	0	1	0
9	-1	0	1	0	0
10	-1	0	0	0	1
11	-1	0	1	0	0
12	-1	0	0	1	0
13	-1	0	1	0	0
14	-1	0	0	0	1
15	-1	0	0	1	0
16	-1	0	0	1	0
17	1	-1	1	0	0
18	1	-1	1	0	0
19	1	-1	0	1	0
20	1	-1	0	0	1
21	0	1	0	0	1
22	0	1	0	1	0
23	0	1	1	0	0
24	0	1	0	0	1
25	0	1	0	1	0
26	0	1	1	0	0
27	0	1	0	1	0
28	0	1	0	0	1

Table H.22. Design optimal for model (6.14), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	-1	0.5	0	0.5
2	-1	-1	0	1	0
3	-1	-1	0	0	1
4	-1	-1	0	0.5	0.5
5	0	-1	0	0.5	0.5
6	0	-1	1	0	0
7	0	-1	0.5	0.5	0
8	0	-1	0	0	1
9	0	1	0.5	0	0.5
10	0	1	1	0	0
11	0	1	0	1	0
12	0	1	0	0.5	0.5
13	1	1	0.5	0	0.5
14	1	1	0	0	1
15	1	1	0.5	0.5	0
16	1	1	0	1	0
17	-1	0	1	0	0
18	-1	0	0	1	0
19	-1	0	0	0.5	0.5
20	-1	0	0	0	1
21	1	0	0.5	0	0.5
22	1	0	0.5	0.5	0
23	1	0	0	1	0
24	1	0	1	0	0
25	-1	1	0.5	0	0.5
26	-1	1	0	0	1
27	-1	1	0	0.5	0.5
28	-1	1	0.5	0.5	0

Table H.23. Design optimal for model (6.15), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	1	1	0	0
2	1	1	0	1	0
3	1	1	0	0	1
4	1	1	0	0	1
5	-1	1	0	0	1
6	-1	1	0	1	0
7	-1	1	1	0	0
8	-1	1	0	0	1
9	1	-1	0	0	1
10	1	-1	1	0	0
11	1	-1	0	1	0
12	1	-1	1	0	0
13	-1	-1	0	1	0
14	-1	-1	0	1	0
15	-1	-1	1	0	0
16	-1	-1	0	0	1
17	1	1	0	0	1
18	1	1	0	1	0
19	1	1	0	1	0
20	1	1	1	0	0
21	-1	1	0	1	0
22	-1	1	0	0	1
23	-1	1	1	0	0
24	-1	1	0	1	0
25	-1	-1	0	1	0
26	-1	-1	0	0	1
27	-1	-1	1	0	0
28	-1	-1	0	0	1

Table H.24. Design optimal for model (6.16), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	1	0	1	0
2	1	1	1	0	0
3	1	1	1	0	0
4	1	1	0	0	1
5	-1	1	0	1	0
6	-1	1	1	0	0
7	-1	1	0	0	1
8	-1	1	1	0	0
9	-1	-1	0	1	0
10	-1	-1	0	1	0
11	-1	-1	1	0	0
12	-1	-1	0	0	1
13	1	1	0	0	1
14	1	1	0	0	1
15	1	1	1	0	0
16	1	1	0	1	0
17	1	-1	0	1	0
18	1	-1	0	0	1
19	1	-1	1	0	0
20	1	-1	1	0	0
21	-1	1	1	0	0
22	-1	1	0	0	1
23	-1	1	0	0	1
24	-1	1	0	1	0
25	1	-1	0	0	1
26	1	-1	0	0	1
27	1	-1	0	1	0
28	1	-1	1	0	0

Table H.25. Design optimal for model (6.17), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	-1	0	1	0
2	1	-1	1	0	0
3	1	-1	0.5	0	0.5
4	1	-1	0	0.5	0.5
5	1	-1	0	0	1
6	1	-1	0	1	0
7	1	-1	0.5	0.5	0
8	1	-1	1	0	0
9	-1	1	0	0	1
10	-1	1	0	1	0
11	-1	1	0.5	0	0.5
12	-1	1	0.5	0.5	0
13	-1	-1	1	0	0
14	-1	-1	0	0	1
15	-1	-1	0.5	0.5	0
16	-1	-1	0	0.5	0.5
17	-1	1	0	0.5	0.5
18	-1	1	1	0	0
19	-1	1	0	1	0
20	-1	1	0	0	1
21	1	1	1	0	0
22	1	1	0	1	0
23	1	1	0	0.5	0.5
24	1	1	0	0	1
25	-1	-1	0.5	0	0.5
26	-1	-1	0	1	0
27	-1	-1	0	0	1
28	-1	-1	1	0	0

Table H.26. Design optimal for model (6.18), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	1	0	0	1
2	1	1	0	1	0
3	1	1	0	0.5	0.5
4	1	1	1	0	0
5	-1	1	1	0	0
6	-1	1	0.5	0	0.5
7	-1	1	0	0.5	0.5
8	-1	1	0	1	0
9	1	-1	0	0	1
10	1	-1	0	0.5	0.5
11	1	-1	0.5	0.5	0
12	1	-1	1	0	0
13	1	1	0.5	0	0.5
14	1	1	0.5	0.5	0
15	1	1	0	1	0
16	1	1	0	0	1
17	-1	1	1	0	0
18	-1	1	0	0	1
19	-1	1	0.5	0.5	0
20	-1	1	0	1	0
21	1	-1	0	0	1
22	1	-1	0	1	0
23	1	-1	0.5	0	0.5
24	1	-1	1	0	0
25	-1	-1	0.5	0.5	0
26	-1	-1	0	1	0
27	-1	-1	0	0	1
28	-1	-1	1	0	0

Table H.27. Design optimal for model (6.19), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	1	0	0.5	0.5
2	-1	1	1/3	1/3	1/3
3	-1	1	0	1	0
4	-1	1	1	0	0
5	1	1	0	1	0
6	1	1	1/3	1/3	1/3
7	1	1	0.5	0	0.5
8	1	1	0	0	1
9	-1	1	0	0	1
10	-1	1	1	0	0
11	-1	1	0.5	0	0.5
12	-1	1	0	0.5	0.5
13	0	1	0	1	0
14	0	1	1	0	0
15	0	1	0.5	0.5	0
16	0	1	0	0	1
17	-1	0	0.5	0	0.5
18	-1	0	0.5	0.5	0
19	-1	0	0	0.5	0.5
20	-1	0	0	1	0
21	1	1	0	0	1
22	1	1	0.5	0.5	0
23	1	1	0	0.5	0.5
24	1	1	1/3	1/3	1/3
25	-1	0	1	0	0
26	-1	0	0.5	0.5	0
27	-1	0	0.5	0	0.5
28	-1	0	1/3	1/3	1/3

Table H.28. Design optimal for model (6.20), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	-1	1	0	0
2	1	-1	0	0	1
3	1	-1	0	0	1
4	1	-1	0	1	0
5	-1	0	0	0	1
6	-1	0	0	1	0
7	-1	0	0	0	1
8	-1	0	1	0	0
9	1	1	1	0	0
10	1	1	0	0	1
11	1	1	0	0	1
12	1	1	0	1	0
13	1	0	0	1	0
14	1	0	0	0	1
15	1	0	1	0	0
16	1	0	0	1	0
17	0	-1	0	1	0
18	0	-1	1	0	0
19	0	-1	1	0	0
20	0	-1	0	0	1
21	-1	1	1	0	0
22	-1	1	0	1	0
23	-1	1	1	0	0
24	-1	1	0	0	1
25	-1	-1	0	0	1
26	-1	-1	0	1	0
27	-1	-1	1	0	0
28	-1	-1	0	1	0

Table H.29. Design optimal for model (6.21), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	-1	1	0	0
2	-1	-1	0	1	0
3	-1	-1	0.5	0	0.5
4	-1	-1	0	0.5	0.5
5	1	-1	0	0	1
6	1	-1	0	1	0
7	1	-1	0.5	0.5	0
8	1	-1	0.5	0	0.5
9	1	-1	0	0	1
10	1	-1	0	1	0
11	1	-1	1/3	1/3	1/3
12	1	-1	1	0	0
13	1	1	1	0	0
14	1	1	0	0.5	0.5
15	1	1	0	1	0
16	1	1	0	0	1
17	-1	1	1/3	1/3	1/3
18	-1	1	0	0	1
19	-1	1	0	1	0
20	-1	1	1	0	0
21	-1	-1	1	0	0
22	-1	-1	0	1	0
23	-1	-1	0	0	1
24	-1	-1	0.5	0.5	0
25	-1	1	0	0.5	0.5
26	-1	1	1/3	1/3	1/3
27	-1	1	0.5	0.5	0
28	-1	1	0.5	0	0.5

Table H.30. Design optimal for model (6.22), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	-1	1	0.5	0	0.5
2	-1	1	0	1	0
3	-1	1	1	0	0
4	-1	1	0	0.5	0.5
5	1	1	1	0	0
6	1	1	0	0	1
7	1	1	0.5	0.5	0
8	1	1	0	0.5	0.5
9	1	1	0.5	0	0.5
10	1	1	0	1	0
11	1	1	0	0	1
12	1	1	1	0	0
13	-1	1	0	1	0
14	-1	1	0	0	1
15	-1	1	1	0	0
16	-1	1	0.5	0.5	0
17	-1	-1	1	0	0
18	-1	-1	1/3	1/3	1/3
19	-1	-1	0	1	0
20	-1	-1	0	0	1
21	1	-1	0	1	0
22	1	-1	1	0	0
23	1	-1	0	0	1
24	1	-1	1/3	1/3	1/3
25	-1	-1	0.5	0	0.5
26	-1	-1	0	0.5	0.5
27	-1	-1	0.5	0.5	0
28	-1	-1	1/3	1/3	1/3

Table H.31. Design optimal for model (6.23), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	-1	1	0	0
2	1	-1	0	0	1
3	1	-1	0	1	0
4	1	-1	0.5	0	0.5
5	1	0	0	1	0
6	1	0	0.5	0	0.5
7	1	0	0	0	1
8	1	0	0.5	0.5	0
9	-1	-1	1	0	0
10	-1	-1	0	1	0
11	-1	-1	0.5	0	0.5
12	-1	-1	0	0	1
13	1	1	0	1	0
14	1	1	0	0.5	0.5
15	1	1	0	0	1
16	1	1	1	0	0
17	-1	1	1	0	0
18	-1	1	0	1	0
19	-1	1	0	0	1
20	-1	1	0.5	0.5	0
21	0	1	0.5	0	0.5
22	0	1	0	1	0
23	0	1	0	0.5	0.5
24	0	1	1	0	0
25	0	-1	1	0	0
26	0	-1	0	0.5	0.5
27	0	-1	0.5	0.5	0
28	0	-1	0	0	1

Table H.32. Design optimal for model (6.24), for the example in §6.4.2.

Run	z_1	z_2	x_1	x_2	x_3
1	1	1	1	0	0
2	1	1	0	0	1
3	1	1	0.5	0.5	0
4	1	1	0	1	0
5	0	1	0	0	1
6	0	1	1	0	0
7	0	1	1/3	1/3	1/3
8	0	1	0	0.5	0.5
9	1	0	0	1	0
10	1	0	0	0	1
11	1	0	0.5	0	0.5
12	1	0	1/3	1/3	1/3
13	-1	-1	0.5	0.5	0
14	-1	-1	0	1	0
15	-1	-1	0	0	1
16	-1	-1	1	0	0
17	0	0	1/3	1/3	1/3
18	0	0	0.5	0	0.5
19	0	0	0	0.5	0.5
20	0	0	0.5	0.5	0
21	-1	1	0	1	0
22	-1	1	0.5	0	0.5
23	-1	1	1	0	0
24	-1	1	0	0	1
25	1	-1	0	0.5	0.5
26	1	-1	0	0	1
27	1	-1	0	1	0
28	1	-1	1	0	0

Table H.33. Design optimal for model (6.25), for the example in §6.4.2.

Bibliography

[1] V. K. Agboto and C. J. Nachtsheim. Bayesian model robust designs. Technical report, University of Minnesota, 2005.

[2] F. T. Anbari and J. M. Lucas. Super-efficient designs: How to run your experiment for higher efficiency and lower cost. In *ASQC 48th Annual Quality Congress Proceedings*, 1994.

[3] F. T. Anbari and J. M. Lucas. Design and running super-efficient experiments: Optimum blocking with one hard-to-change factor. *Journal of Quality Technology*, 40(1):31–45, 2008.

[4] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. New York: Wiley, 1984.

[5] H. Arnouts and P. Goos. Update formulas for split-plot and block designs. Technical Report 2008-022, University of Antwerp, Faculty of Applied Economics, Department of Mathematics, Statistics, and Actuarial Sciences, December 2008.

[6] H. Arnouts, P. Goos, and B. Jones. Design and analysis of industrial strip-plot experiments. *Quality and Reliability Engineering International*, 26:127–136, 2010.

[7] S. P. Aspery and S. Macchietto. Designing robust optimal dynamic experiments. *Journal of Process Control*, 12(4):545–556, 2002.

[8] A. B. Atashgah and A. Seifi. Application of semi-definite programming to the design of multi-response experiments. *IIE Transactions*, 39:7:763–769, 2007.

[9] A. B. Atashgah and A. Seifi. Optimal design of multi-response experiments using semi-definite programming. *Optimization and Engineering*, 10(1):75–90, 2009.

[10] A. C. Atkinson, A. N. Donev, and R. D. Tobias. *Optimum Experimental Designs, with SAS*. Oxford University Press, 2007.

[11] M. P. F. Berger and F. E. S. Tan. Robust designs for linear mixed effects models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(4):569–581, 2004.

[12] D. Bingham and R. R. Sitter. Minimum-aberration two-level fractional factorial split-plot designs. *Technometrics*, 41(1):62–70, 1999.

[13] D. R. Bingham and H. A. Chipman. Incorporating prior information in optimal design for model selection. *Technometrics*, 49(2):155–163, 2007.

[14] D. R. Bingham and R. R. Sitter. Design issues in fractional factorial split-plot experiments. *Journal of Quality Technology*, 33(1):2–15, 2001.

[15] W. Bischoff. On d-optimal designs for linear models under correlated observations with an application to a linear model with multiple response. *Journal of Statistical Planning and Inference*, 37:69–80, 1993.

[16] W. Bischoff. Determinant formulas with applications to designing when the observations are correlated. *Annals of the Institute of Statistical Mathematics*, 47(2):385–399, 1995.

[17] S. Bisgaard. The design and analysis of $2^{k-p} \times 2^{q-r}$ split plot experiments. *Journal of Quality Technology*, 32(1):39–56, 2000.

[18] G. E. P. Box and N. R. Draper. A basis for the selection of a response surface design. *Journal of the American Statistical Association*, 54(287):622–654, 1959.

[19] G. E. P. Box and N. R. Draper. The bayesian estimation of common parameters from several responses. *Biometrika*, 52(3/4):355–365, 1965.

[20] M. J. Box and N. R. Draper. Estimation and design criteria for multiresponse non-linear models with non-homogenous variance. *Applied Statistics*, 21(1):13–24, 1972.

[21] F. Chang, M. L. Huang, D. K. J. Lin, and H. Yang. Optimal designs for dual response polynomial regression models. *Journal of Statistical Planning and Inference*, 93:309–322, 2001.

[22] S. I. Chang. Some properties of multi-response d-optimal designs. *Journal of Mathematical Analysis and Applications*, 184:256–262, 1994.

[23] S. I. Chang. An algorithm to generate near d-optimal designs for multiple-response surface models. *IIE Transactions*, 29:1073–1081, 1997.

[24] Y.-J. Chang and W. I. Notz. *Handbook of Statistics, Vol. 13*, chapter Model Robust Designs, pages 1055–1098. Elsevier Science B. V., 1996.

[25] H. Chipman, M. Hamada, and C. F. J. Wu. A bayesian variable selection approach for analyzing designed experiments with complex aliasing. *Technometrics*, 39:372–381, 1997.

[26] R. D. Cook and C. J. Nachtsheim. A comparison of algorithms for constructing exact d-optimal designs. *Technometrics*, 22:315–324, 1980.

[27] R. D. Cook and C. J. Nachtsheim. An analysis of the k-exchange algorithm. Technical report, University of Minnesota, Dept. of Operations and Management Science, 1982.

[28] R. D. Cook and C. J. Nachtsheim. Model-robust, linear-optimal designs. *Technometrics*, 24:49–54, 1982.

[29] J. Cornell. *Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data*. Wiley-Interscience, 2nd edition, s 1990.

[30] C. Daniel. Parallel fractional replicates. *Technometrics*, 2(2):263–268, 1960.

[31] H. Dette. A generalization of $\mathcal{D}$- and $\mathcal{D}_1$-optimal design in polynomial regression. *Annals of Statistics*, 18:1784–1804, 1990.

[32] H. Dette and T. Franke. Constrained d- and d_1-optimal designs for polynomial regression. *The Annals of Statistics*, 28(6):1702–1727, 2000.

[33] H. Dette and T. Franke. Robust designs for polynomial regression by maximizing a minimum of $\mathcal{D}$- and $\mathcal{D}_1$-efficiencies. *The Annals of Statistics*, 29(4):1024–1049, 2001.

[34] N. R. Draper and W. G. Hunter. Design of experiments for parameter estimation in multiresponse situations. *Biometrika*, 53:525–533, 1966.

[35] N. R. Draper and W. G. Hunter. The use of prior distributions in the design of experiments for parameter estimation in non-linear situations: Multiresponse case. *Biometrika*, 54(3/4):662–665, 1967.

[36] N. R. Draper and J. A. John. Response surface designs where levels of some factors are difficult to change. *Australian and New Zealand Journal of Statistics*, 40(4):487–495, 1998.

[37] W. DuMouchel and B. Jones. A simple bayesian modification of d-optimal designs to reduce dependence on an assumed model. *Technometrics*, 36(1):37–47, 1994.

[38] M. Emmett, P. Goos, and E. Stillman. D-optimal designs for multiple polynomial responses. Technical report, University of Sheffield, 2007.

[39] Z. Fang and D. P. Wiens. Integer-valued, minimax robust designs for estimation and extrapolation in heteroscedastic, approximate linear models. *Journal of the American Statistical Association*, 95(451):807–818, 2000.

[40] Z. Fang and D. P. Wiens. Robust regression designs for approximate polynomial models. *Journal of Statistical Planning and Inference*, 117:305–321, 2003.

[41] V. V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, NY, 1972.

[42] S. E. Frisbee and J. W. McGinity. Influence of nonionic surfactants on the physical and chemical properties of a biodegradable pseudolatex. *European Journal of Pharmaceutics and Biopharmaceutics*, 40:355–363, 1994.

[43] J. Ganju and J. M. Lucas. Detecting randomization restrictions caused by factors. *Journal of Statistical Planning and Inference*, 81:129–140, 1999.

[44] J. Ganju and J. M. Lucas. Randomized and random run order experiments. *Journal of Statistical Planning and Inference*, 133:199–210, 2004.

[45] P. Goos. *The Optimal Design of Blocked and Split-plot Experiments*. Springer, 2002.

[46] P. Goos. Optimal versus orthogonal and equivalent-estimation design of blocked and split-plot experiments. *Statistica Neerlandica*, 60:361–378, 2006.

[47] P. Goos and A. N. Donev. Tailor-made split-plot designs for mixture and process variables. *Journal of Quality Technology*, 39(4):326–339, 2007.

[48] P. Goos, A. Kobilinsky, T. E. O'Brien, and M. Vandebroek. Model-robust and model-sensitive designs. *Computational Statistics & Data Analysis*, 49:201–216, 2005.

[49] P. Goos and M. Vandebroek. Optimal split-plot designs. *The Journal of Quality Technology*, 33(4):436–450, 2001.

[50] P. Goos and M. Vandebroek. D-optimal split-plot designs with given numbers and sizes of whole plots. *Technometrics*, 45(3):235–245, 2003.

[51] P. Goos and M. Vandebroek. Outperforming completely randomized designs. *Journal of Quality Technology*, 36(1):12–26, 2004.

[52] J. A. Heinsman and D. C. Montgomery. Optimization of a household product formulation using a mixture experiment. *Quality Engineering*, 7:583–600, 1995.

[53] G. Heo, B. Schmuland, and D. P. Wiens. Restricted minimax robust designs for misspecified regression models. *The Canadian Journal of Statistics*, 29(1):117–128, 2001.

[54] A. Heredia-Langner, D. C. Montgomery, W. M. Carlyle, and C. M. Borror. Model-robust optimal designs: A genetic algorithm approach. *Journal of Quality Technology*, 36(3):263–279, 2004.

[55] P. Huang, D. Chen, and J. O. Voelkel. Minimum-aberration two-level split-plot designs. *Technometrics*, 40(4):314–326, 1998.

[56] H. M. Huizenga, D. J. Heslenfeld, and P. C. M. Molenaar. Optimal measurement conditions for spatiotemporal eeg/meg source analysis. *Psychometrika*, 67(2):299–313, 2002.

[57] L. Imhof and W. K. Wong. A graphical method for finding maximin efficiency designs. *Biometrics*, 56(1):113–117, 2000.

[58] M. E. Johnson and C. J. Nachtsheim. Some guidelines for constructing exact d-optimal designs on convex design spaces. *Technometrics*, 25:271–277, 1983.

[59] B. Jones and P. Goos. A candidate-set-free algorithm for generating d-optimal split-plot designs. *Journal of the Royal Statistical Society, Series C: Applied Statistics*, 56:347–364, 2007.

[60] B. Jones and P. Goos. D-optimal design of split-split-plot experiments. *Biometrika*, 96(1):67–82, 2009.

[61] B. Jones and C. J. Nachtsheim. Split-plot designs: What, why, and how. *Journal of Quality Technology*, 41(4):340–361, 2009.

[62] H. L. Ju and J. M. Lucas. l^k factorial experiments with hard-to-change and easy-to-change factors. *Journal of Quality Technology*, 34(4):411–421, 2002.

[63] A. I. Khuri and J. A. Cornell. *Response Surfaces: Designs and Analyses*. M. Dekker, 1996.

[64] J. Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society, Series B*, 21:272–304, 1959.

[65] J. Kiefer and J. Wolfowitz. Optimum designs in regression problems. *Annals of Mathematical Statistics*, 30:271–294, 1959.

[66] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.

[67] W. B. Kim and N. R. Draper. Choosing a design for straight line fits to two correlated responses. *Statistica Sinica*, 4:275–280, 1994.

[68] S. M. Kowalski, J. A. Cornell, and G. G. Vining. Split-plot designs and estimation methods for mixture experiments with process variables. *Technometrics*, 44(1):72–79, 2002.

[69] O. Krafft and M. Schaefer. D-optimal designs for a multivariate regression model. *Journal of Multivariate Analysis*, 42:130–140, 1992.

[70] V. G. Kurotschka and R. Schwabe. *Madan Puri Festschrift*, chapter The Reduction of Design Problems for Multivariate Experiments to Univariate Possibilities and their Limitations, pages 193–204. VSP, 1996.

[71] E. Läuter. Experimental design in a class of models. *Statistics*, 5:379–398, 1974.

[72] J. D. Letsinger, R. H. Myers, and M. Lentner. Response surface methods for bi-randomization structures. *Journal of Quality Technology*, 28:4:381–397, 1996.

[73] W. Li and C. J. Nachtsheim. Model-robust factorial designs. *Technometrics*, 42(4):345–352, 2000.

[74] S. X. Liu and D. P. Wiens. Robust designs for approximate polynomial regression. *Journal of Statistical Planning and Inferences*, 64:369–381, 1997.

[75] X. Liu and R. X. Yue. $\mathcal{P}$-optimal robust designs for multiresponse approximately linear regression. *Applied Mathematics - A Journal of Chinese Universities*, 23(2):168–174, 2008.

[76] J. M. Lucas and H. L. Ju. Split plotting and randomization in industrial experiments. In *ASQC Quality Congress Transactions*, 1992.

[77] R. K. Meyer and C. J. Nachtsheim. The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, 37(1):60–69, 1995.

[78] K. S. Miller. On the inverse of the sum of matrices. *Mathematics Magazine*, 54(2):67–72, 1981.

[79] T. J. Mitchell. An algorithm for the construction of d-optimal experimental designs. *Technometrics*, 16:203–210, 1974.

[80] G. Montepiedra and V. V. Fedorov. Minimum bias designs with constraints. *Journal of Statistical Planning and Inference*, 63:97–111, 1997.

[81] G. Montepiedra and A. B. Yeh. A two-stage strategy for the construction of d-optimal experimental designs. *Communications in Statistics - Simulation and Computations*, 27(2):377–401, 1998.

[82] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, Inc., 1991.

[83] A. Neff. *Bayesian two-stage designs under model uncertainty*. PhD thesis, Virginia Polytechnic Institute and State University, 1996.

[84] P. A. Parker, S. M. Kowalski, and G. G. Vining. Construction of balanced equivalent estimation second-order split-plot designs. *Technometrics*, 49(1):56–65, 2007.

[85] G. F. Piepel. Programs for generating extreme vertices and centroids of linearly constrained experimental regions. *Journal of Quality Technology*, 20:2:125–139, 1988.

[86] G. F. Piepel, S. K. Cooley, and B. Jones. Construction of a 21-component layered mixture experiment design using a new mixture coordinate-exchange algorithm. *Quality Engineering*, 17:579–594, 2005.

[87] F. Pukelsheim. *Optimal Design of Experiments*. John Wiley & Sons, Inc., 1993.

[88] F. Pukelsheim and J. L. Rosenberger. Experimental designs for model discrimination. *Journal of the American Statistical Association*, 88(422):642–649, 1993.

[89] R. Rajagopal and E. del Castillo. Model-robust process optimization using bayesian model averaging. *Technometrics*, 47(2):152–163, 2005.

[90] R. G. Roy, S. N. and J. N. Srivastava. *Analysis and Design of Certain Quantitative Multiresponse Experiments*. Pergamon Press, New York, NY, 1971.

[91] A. Ruggoo and M. Vandebroek. Bayesian sequential $\mathcal{D}$-$\mathcal{D}$ optimal model-robust designs. *Computational Statistics & Data Analysis*, 47:655–673, 2004.

[92] SAS. SAS Institute Inc., Version 9.1.3, Cary, NC, USA, 2000-2004.

[93] J. R. Schott. *Matrix Analysis for Statistics*. John Wiley & Sons, Inc., 1997.

[94] H. K. Shah, D. C. Montgomery, and W. M. Carlyle. Response surface modeling and optimization in multiresponse experiments using seemingly unrelated regressions. *Quality Engineering*, 16(3):387–397, 2004.

[95] B. J. Smucker, E. del Castillo, and J. L. Rosenberger. Exchange algorithms for constructing model-robust experimental designs. Technical Report 09-04, The Pennsylvania State University, 2009.

[96] R. D. Snee. Computer-aided design of experiments–some practical experiences. *Journal of Quality Technology*, 17:222–236, 1985.

[97] S. M. Stigler. Optimal experimental design for polynomial regression. *Journal of the American Statistical Association*, 66(334):311–318, 1971.

[98] D. X. Sun. *Estimation capacity and related topics in experimental designs*. PhD thesis, University of Waterloo, 1993.

[99] L. A. Trinca and S. G. Gilmour. Multistratum response surface designs. *Technometrics*, 43(1):25–33, 2001.

[100] P.-W. Tsai and S. G. Gilmour. A general criterion for factorial designs under model uncertainty. *Technometrics (to appear)*, 2010.

[101] C. L. Tseo, J. C. Deng, J. A. Cornell, A. I. Khuri, and R. H. Schmidt. Effect of washing treatment on quality of minced mullet flesh. *Journal of Food Science*, 48:163–167, 1983.

[102] G. G. Vining and S. M. Kowalski. Exact inference for response surface designs within a split-plot structure. *Journal of Quality Technology*, 40(4):394–406, 2008.

[103] G. G. Vining, S. M. Kowalski, and D. C. Montgomery. Response surface designs within a split-plot structure. *Journal of Quality Technology*, 37(2):115–129, 2005.

[104] D. F. Webb, J. M. Lucas, and J. J. Borkowski. Factorial experiments when factor levels are not necessarily reset. *Journal of Quality Technology*, 36(1):1–11, 2004.

[105] W. J. Welch. A mean squared error criterion for the design of experiments. *Biometrika*, 70(1):205–213, 1983.

[106] D. P. Wiens. Robust minimax designs for multiple linear regression. *Linear Algebra Applications*, 127:327–340, 1990.

[107] D. P. Wiens. Minimax designs for approximately linear regression. *Journal of Statistical Planning and Inference*, 31(3):353–371, 1992.

[108] D. P. Wiens. Robust designs for approximately linear regression: M-estimated parameters. *Journal of Statistical Planning and Inference*, 40(1):135–160, 1994.

[109] D. P. Wiens. Minimax robust designs and weights for approximately specified regression models with heteroscedastic errors. *Journal of the American Statistical Association*, 93:1440–1450, 1998.

[110] D. P. Wiens. Robust weights and designs for biased regression models: Least squares and generalized m-estimation. *Journal of Statistical Planning and Inference*, 83(2):395–412, 2000.

[111] D. P. Wiens and J. Zhou. Minimax regression designs for approximately linear models with autocorrelated errors. *Journal of Statistical Planning and Inference*, 55(1):95–106, 1996.

[112] M. M. C. Wijesinha. *Design of experiments for multiresponse models*. PhD thesis, University of Florida, 1984.

[113] F. Yates. Complex experiments, with discussion. *Journal of the Royal Statistical Society, Series B*, 2:181–223, 1935.

[114] R. X. Yue. Model-robust designs in multiresponse situations. *Statistics & Probability Letters*, 58:369–379, 2002.

[115] A. Zellner. An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American Statistical Association*, 57:348–368, 1962.

[116] A. Zellner. Estimators for seemingly unrelated regression equations: Some exact finite sample results. *Journal of the American Statistical Association*, 58:977–992, 1963.

[117] A. Zellner and D. S. Huang. Further properties of efficient estimators for seemingly unrelated regression equations. *International Economic Review*, 3:300–313, 1962.

Vita

Byran J. Smucker

Byran Jay Smucker was born in Albany, OR on July 13, 1982. He graduated from Brownsville Mennonite School in 1999, attended Linn-Benton Community College, and graduated *summa cum laude* from Oregon State University in 2005 with a B.S. in Industrial Engineering. In August, 2005 he entered the Department of Statistics at The Pennsylvania State University, acquired an M.S. in Statistics and Operations Research in 2007, and will graduate in August, 2010 with a Ph.D in Statistics and Operations Research. The same month, he will begin an assistant professor position in the Statistics Department at Miami University at Oxford, OH. Mr. Smucker is the son of Steve and Bonnie Smucker, with five siblings (Randy, Jessica, Justin, Stephanie, and Trevin). He is married to Amy Smucker, with a son Xavier.