5-2024

Exploring Application of the Coordinate Exchange to Generate Optimal Designs Robust to Data Loss

Asher Hanson
Utah State University, a02213550@usu.edu

EXPLORING APPLICATION OF THE COORDINATE EXCHANGE TO GENERATE OPTIMAL DESIGNS ROBUST TO DATA LOSS

by

Asher Hanson

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

of

MASTERS OF SCIENCE

in

Statistics

Approved:

Stephen J. Walsh, Ph.D.
Major Professor

Erin Beckman, Ph.D.
Committee Member

KimberLeigh Hadfield, Ph.D.
Committee Member

D. Richard Cutler, Ph.D.
Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan,Utah

2024

ABSTRACT

Exploring Application of the Coordinate Exchange to Generating
Optimal Designs Robust to Data Loss

by

Asher Hanson, Master of Science

Utah State University, 2024

Major Professor: Stephen J. Walsh, Ph.D.
Department: Mathematics and Statistics

The purpose of implementing optimal design of experiments (DoE) is to enhance the efficiency of data extraction while considering the associated resource constraints. DoE's primary objective is to maximize information about experimental factors while also managing costs and time investments. Adopting a suitable DoE allows comparisons and strengthens causal inferences, establishing a clear link between cause and effect. To achieve optimal designs, researchers rely on computational algorithms to tailor their designs to specific experiments. The Coordinate Exchange (CEXCH) Algorithm has gained popularity in recent years and emerged as a valuable tool for generating optimal designs.

Additionally, the importance of incorporating robust characteristics into experimental design has been found to be vital, depending on the cost of resources and time. This attribute ensures reliability in data collection by minimizing the impact of potential data loss. The implementation of robust features becomes particularly crucial in scenarios where the loss of observations may occur, emphasizing the need for a reliable experimental design. With this objective in mind, the thesis focuses on utilizing the CEXCH algorithm to create robust optimal designs. This process unfolds in three distinct phases. First, second-order model designs with two to three factors are generated and compared to those previously published. Second, to employ the algorithm to generate first-order model designs tailored specifically for screening purposes. Finally, to explore designs that are robust to the loss of two observations.

Results obtained during the validation phase indicate that the CEXCH algorithm is adept at producing designs equal to or better than those already published. This conclusion is drawn based on scoring each design using various criteria functions and efficiencies. From that phase, there was reason enough to believe that the CEXCH algorithm could be used for other models to offer robust optimal designs. Designs from the screening experiments also showed that the robust designs generated from the CEXCH were comparatively better in most cases than those that were generated without using a robust criterion. The exploration of designs that were robust to the loss of two observations resulted positively, with new designs equipped for such experiments. Having completed the three phases, it is determined that the CEXCH has the ability to aid in generating designs for researchers interested in having robust optimal designs for their experiments.

(90 pages)

PUBLIC ABSTRACT

Exploring Application of the Coordinate Exchange to Generating Optimal Designs Robust to Data Loss

Asher Hanson

The primary objective of this study is to evaluate the efficacy of the coordinate exchange (CEXCH) algorithm in the generation of robust optimal designs. The assessment involves a comparative analysis, wherein designs produced by the Point Exchange (PEXCH) Algorithm are employed as benchmarks for evaluating the efficiency of CEXCH designs. Three modified criteria, selected from the traditional alphabet criteria pool, are utilized to score each algorithm. To enhance the reliability of the comparative analysis, multiple rounds of validation are conducted, focusing on visual assessments, design scores, and criteria efficiencies. The findings from each round of validation contribute to a comprehensive understanding of the effectiveness of the CEXCH algorithm.

Following the investigation into the effectiveness of CEXCH, the subsequent phase involves the implementation of the algorithm in generating screening designs. Screening experiments are used to extract insightful factors from an extensive list of potential variables.

Exploring insights into the generating designs robust to the loss of two observations, show that the application of the CEXCH algorithm, particularly with the integration of robust criteria, yields designs that compare favorably with alternative criteria and algorithms. These findings highlight the importance of considering the CEXCH algorithm during the preparatory stages of experimental design for researchers seeking robust and effective outcomes.

ACKNOWLEDGMENTS

Taking on a thesis project helped me learn and grow in many ways. While becoming a contributing researcher in the field of optimal design, I also improved as a student, teacher, and person. Thank you, Utah State University, for giving me the chance to become an Aggie and embark on this life-changing experience. I would also like to express my gratitude to my Major Professor, Dr. Stephan Walsh, for accepting me as a student and coaching me through this experience. His sincere desire to help me develop a passion for learning and navigate the highs and lows of research led me to become the optimistic researcher I am today. Alongside my Major Professor, I was fortunate to have a very supportive committee consisting of Dr. KimberLeigh Hadfield and Dr. Erin Beckman. Apart from their interest in my research, both members generously offered insightful advice on school, jobs, and life. This team of professors allowed me to maximize the experience as well as produce work that I am proud of.

I would also like to acknowledge my friends and family. Despite not having much knowledge about statistics or optimal design of experiments, they consistently stood by my side, encouraging me to share my research and explain how this knowledge could benefit others. I have been blessed by this experience, and I plan on cherishing these memories far beyond my academic career and into the new experiences that await.

Asher Hanson

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
REVIEW OF ROBUST OPTIMAL DESIGN

1.1
Introduction

Since the development and implementation of experimental design by Ronald Fisher (1935), the search for optimal designs has gathered a lot of attention in research and application. It is this drive that led Kristine Smith to introduce the field of Optimal Design [1]. The benefit of optimal design lies in its ability to create unique designs with minimal variance and accurate predictions for any given scenario. As scientists design experiments, they want to extract the maximum amount of information about their factors while considering the associated costs and time required for each test. To ensure the adequacy of a design for a specific experiment, scientists must guarantee that it performs optimally and remains robust in the face of potential outcomes that could impact the experiment.

The origins of experimental design can be traced back to its application in answering agricultural questions related to temperature, soil conditions, and rainfall. This distinctive process enables experimenters to specify the factors of interest and design experiments that shed light on the effects of individual factors as well as potential interactions among them. Prior to this methodology, conducting experiments involved repeatedly changing one factor, recording the outcomes, and then conducting the experiment numerous times.

Design of Experiments (DoE) is popular for various reasons, one is its ability to facilitate direct comparison between treatments of interest in a professional and systematic manner. DoE allows research to minimize bias in comparisons and to make inferences about causation, connecting the dots between cause and effect [2]. This critical aspect drives extensive testing worldwide, highlighting the significance of DoE as a topic of interest.

Since DoE development, the incorporation of designed experiments has both spread and become a valuable tool in various fields. In the 1950s, Box and Wilson began experimenting with its methods in the chemical industry. During this period, the idea of treating designs as functions and seeking their optimal conditions emerged. One of their most pivotal contributions was the development of Response Surface Methodology (RSM), which facilitated the comparison of the relationship between responses and quantitative treatment factors.

1.1.1

Brief introduction to optimal design

The field of experimental design has expanded in various ways, thanks to advancements in computer-aided designs. One notable branch is optimal design, a term coined by Kiefer in 1959, which aims to create a design space that achieves the best score based on a numerical criterion [1, 3]. However, the concept and initial work can be attributed to Kristine Smith, a statistician who worked for Karl Pearson in 1918. Smith recognized the impact of residual error variance on a design and focused on creating designs with minimal uncertainty [4]. Initially, these findings were not widely accepted in the statistical community, as optimal design was seen as an abstract method for working with non-convex multidimensional objective functions. It was only through advancements in computing power and further investigation into optimization criteria that the field gained acceptance as a means of creating optimal design spaces [5, 6].

Optimal design is intriguing for various reasons. Joanne R. Wendleburger and George Box were particularly interested in its ability to quantify uncertainty. Creating an experiment is not simply a matter of deciding factors and running it multiple times; it requires an artful design that meets the researchers needs and adapts to unique situations [7]. Experimenters must consider variability associated with errors in the experimental variables, model specification, random errors, and measurement errors [8]. Controlling or minimizing variance is a crucial aspect of all experimentation. Sufficient information is needed to identify the uniqueness in factors and provide quantifiable measurements to explain the differences, which necessitates multiple observations.

Another reason for the growing interest in optimal design is its capacity to create unique designs for each specific situation. Often, it is challenging to fit an experiment into a predefined model design, as each experiment possesses its own distinct characteristics and objectives. "It is always better to create a custom design for the actual problem you want to solve than to force your problem to fit a standard design" [9]. Not every design is intended for the same purpose. Some designs focus on reducing variance in parameter estimates, while others aim to minimize variance associated with prediction responses.

Peter Goos and Bradley Jones, authors of "Optimal Design of Experiments: A Case Study

Approach," provide additional compelling reasons for the power of this field. Early in their book, they discuss the number of runs in an experiment, emphasizing that "For any given number of runs, you can find a design that maximizes the information about the model you want to estimate. In this respect, optimal experimental designs differ completely from full factorial or fractional factorial designs" [10].

With optimal design gaining recognition as a field of study and finding applications in various disciplines, promising areas of research have emerged. In a recent paper by Jensen, one such area of interest is the pursuit of robust designs. The concept of a robust design may seem contradictory to the idea of optimality, which implies a single optimal design. However, a robust design allows for the possibility of experimental mishaps or unexpected factors while still obtaining valuable information. Jensen explains several benefits of studying robustness, the foremost being the ability to conduct experiments without prior knowledge of the exact factors of interest and still derive valuable insights [11]. Although this scenario may not be common, as experiments are often expensive or resource-limited, it would be advantageous to have a design that can yield useful information regardless of the specific experimental setup.

To help researchers find the region of optimal experimentation, Response Surface Methodology (RSM) techniques are devised and implemented. RSM is a statistical technique used within optimal design to explore and model relationships between input variables and output responses. This technique is useful in identifying optimal operating conditions within a given experimental framework [12]. In the field of RSM, there are experiments called screening experiments that often have the characteristic of being saturated, meaning the experiment is designed with only the exact number of observations required to get estimates of all defined factors. In this paper, the concept of designing optimal designs is explored to identify strengths and weaknesses in saturated screening designs.

1.1.2 Modeling and Notation

Let N represent the number of design points and K represent the number of experimental factors. A design point is an $\mathbf{x}' : 1 \times K$ row-vector. We assume all design factors are scaled to range [-1,1] and so the design space is the $\mathcal{X} = [-1, 1]^K$ hypercube. Let $\mathbf{X} : N \times K$ represent the design matrix. While $\mathcal{X}$ denotes the space of candidate *design points* $\mathbf{x}'$, a *design matrix* $\mathbf{X}$ is a collection of N such design points. Thus, the space of all candidate designs is an NK-dimensional hypercube and is denoted:

$$\mathbf{X} \in \bigtimes_{j=1}^{N} \mathcal{X} = \bigtimes_{j=1}^{N} [-1,1]^K = [-1,1]^{NK} = \mathcal{X}^N. \tag{1.1}$$

We consider the second-order linear model which has $p = \binom{K+2}{2}$ linear coefficient parameters. In scalar form, the second-order model is written

$$y = \beta_0 + \sum_{i=1}^{K} \beta_i x_i + \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \beta_{ij} x_i x_j + \sum_{i=1}^{K} \beta_{ii} x_i^2 + \epsilon.$$

Let $\mathbf{F} : N \times p$ represent the model matrix with rows given by the expansion vector $\mathbf{f}'(\mathbf{x}_i') = (1 \quad x_{i1} \quad \dots \quad x_{iK} \quad x_{i1}x_{i2} \quad \dots \quad x_{i(K-1)}x_{iK} \quad x_{i1}^2 \quad \dots \quad x_{iK}^2)$. The model can be written in vector-matrix form as $\mathbf{y} = \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where we impose the standard ordinary least squares assumptions $\boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 \mathbf{I}_N)$ where $\mathcal{N}_N$ denotes the N-dimensional multivariate normal distribution. The ordinary least squares estimator of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = (\mathbf{F}'\mathbf{F})^{-1}\mathbf{F}'\mathbf{y}$ which has variance $\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{F}'\mathbf{F})^{-1}$. The total information matrix for $\boldsymbol{\beta}$, specifically $\mathbf{M}(\mathbf{X}) = \mathbf{F}'\mathbf{F}$, plays an important role in optimal design of experiments—all optimal design objective functions are functions of this matrix.

The practitioner must choose a design from $\mathcal{X}^N$ to implement the experiment in practice. The χ is the space design points live in while $\mathcal{X}^N$ is the space the candidate design matrices live in. An optimality criterion is used to define which candidate designs $\mathbf{X} \in \mathcal{X}^N$ are 'good' designs. An optimization algorithm is required to search $\mathcal{X}^N$ to find the 'best', or optimal, design. Thus, an exact optimal design problem is defined by three components:

1. The number of design points N that can be afforded in the experiment.

2. The structure of the model one wishes to fit (here the second-order model).

3. A criterion which defines an optimal design. This is a function of $\mathbf{M}(\mathbf{X})$.

D-Optimal Exact Design

Let $\mathcal{X}^N$ be the set of all possible exact designs on χ, $\mathbf{X}$ be the set of potential design points, and $\mathbf{x}_i'$ be a design point in $\mathbf{X}$.

$$\mathbf{X}^* := \arg\min_{\mathbf{X} \epsilon \mathcal{X}^N} |\mathbf{M}^{-1}(\mathbf{X})| \tag{1.2}$$

D-Optimal Robust Exact Design

The D criterion seeks a design $\mathbf{X}_R^*$ satisfying

$$\mathbf{X}_R^* := \arg \min_{\mathbf{X} \epsilon \mathcal{X}^N} \max_{\mathbf{x}_i' \epsilon \mathcal{X}} \frac{1}{|\mathbf{M}(\mathbf{X}_{-i})|} \tag{1.3}$$

where M($\mathbf{X}_{-i}$) is the moment matrix of the design with a missing point $\mathbf{x}_i'$.

A-Optimal Exact Design

Let $\mathcal{X}^N$ be the set of all possible exact designs on χ, $\mathbf{X}$ be the set of potential design points, and $\mathbf{x}_i'$ be a design in $\mathbf{X}$. The operation 'tr' denotes the linear algebra trace function.

$$\mathbf{X}^* := \arg \min_{\mathbf{X} \epsilon \mathcal{X}^N} \text{tr}\ \{\mathbf{M}^{-1}(\mathbf{X})\} \tag{1.4}$$

A-Optimal Robust Exact Design

The proposed A-optimal robust exact design satisfies

$$\mathbf{X}_R^* := \arg \min_{\mathbf{X} \epsilon \mathcal{X}^N} \max_{\mathbf{x}_i' \epsilon \mathcal{X}} \text{tr}\{\mathbf{M}^{-1}(\mathbf{X}_{-i})\} \tag{1.5}$$

where M($\mathbf{X}_{-i}$) is the moment matrix of the design with a missing point $\mathbf{x}_i'$.

I-Optimal Exact Design

Let $\mathcal{X}^N$ be the set of all possible exact designs on χ, $\mathbf{X}$ be the set of potential design points, χ be the cuboid region, $\mathbf{x}_i'$ be the design point in $\mathbf{X}$, and V is equal to the volume of the design space, $V = \int_\chi \mathbf{dx}$.

$$\mathbf{X}^* := \arg \min_{\mathbf{X} \epsilon \mathcal{X}^N} \frac{N}{V} \int_\chi \mathbf{f}'(\mathbf{x}_i')\mathbf{M}^{-1}(\mathbf{X})\mathbf{f}(\mathbf{x}_i')\mathbf{dx} \tag{1.6}$$

I-Optimal Robust Exact Design

The I criterion seeks a design $\mathbf{X}_R^*$ satisfying

$$\mathbf{X}_R^* := \arg \min_{\mathbf{X} \epsilon \mathcal{X}^N} \max_{\mathbf{x}_i' \epsilon \mathcal{X}} \left[\frac{1}{V} \int_\chi \mathbf{f}'(\mathbf{x}_i')\mathbf{M}^{-1}(\mathbf{X}_{-i})\mathbf{f}(\mathbf{x}_i')\mathbf{dx} \right] \tag{1.7}$$

where V is equal to the volume of the design space, $V = \int_\chi \mathbf{dx}$.

1.1.3

Brief introduction to robust optimal design

Part of creating robust designs, is anticipating and planning for missing runs. Missing observations are not uncommon in real-world experiments and have the potential of corrupting an experiment [13]. This can significantly impact the reliability of the experiment, particularly when the number of observations is already limited. Given that experiments often involve significant funding or meticulous data collection, each result holds insightful information and should not be undervalued. The suggested approach is to develop designs that can accommodate such situations, deviating slightly from classical designs. Previous to this exploration, researchers would simply have to ensure enough runs were created to allow mistakes to occur or they would live with the statistical consequences.

When studying missing data, there are several categories to consider. One common category is known as Missing at Random (MAR) or Missing Completely at Random (MCAR). This occurs when a faulty run or experiment results in a missing observation. Replicating or replacing these runs can be challenging, leading to missing data in the observations. Another category is Missing Not at Random (MNAR), which is more complex. MNAR arises when the probability of missing data varies for reasons that researchers don't fully understand. In MNAR situations, the likelihood of missing data depends on certain aspects of the experiment, such as equipment, time, or unknown factors. Initially, it is common to assume that the missing data are MAR until advanced statistical methods indicate otherwise and suggest they are MNAR. The study of missing data has led to the exploration of new criteria and algorithms for designing experiments and handling missing observations.

1.1.4

Literature review

One of the earliest recorded works with robust model designs came from Walter Shewhart in the 1920's. Shewhart is popular for working with the Six Sigma Quality strategy and declaring that three sigmas away from the mean requires correction. From that point, many other scientists have worked on creating criterion and algorithms to better perfect designs. One reason robust designs are in such high need is they improve the quality of a product by minimizing the effect of the causes of variation without eliminating the cause. This approach is different from the classical way because with a robust model design it is assumed that some form of mistake or error can take place and that a design that can adapt to such situations is needed. To adapt in this case means to be able to receive close to the same amount of information from a test as if no errors were to happen.

Over the years, new methodologies have emerged to assist in the creation of robust designs. Welch (1983) and Fang and Wiens (2000) conducted experiments with mean squared error, while other scientists have explored Bayesian approaches to protect against potential uncertainties while maintaining a good design for primary terms. HerediaLanger et al. (2004) and Smucker et al. (2011) focused on creating functions, known as criteria, to evaluate the strength and information capacity of a design [14]. Srisuradetchai further developed robust functions for traditional alphabet criterion to create robust designs for all given cases [15]. In his work, he used the point exchange algorithm to cycle through each possible design point and find the optimal design for each criterion.

1.1.5 Conclusions

From the aforementioned research, it can be concluded that finding a robust model for an experiment carries much importance. In this paper, I will further investigate the work that has been done to create interpretable robust designs as well as introduce techniques and steps that prove to create equal to or better designs than have already been developed. For the purpose of this paper, the main focus will be on two-level designs using the coordinate exchange algorithm when the number of runs is small. This field is of most interest as "there is remarkably little done in accessible methods for exact (small-sample) model-robust designs"[14, 11]. With this work we aim to provide this capability to the optimal design research community.

CHAPTER 2
DEVELOPMENT, IMPLEMENTATION, AND VALIDATION OF THE COORDINATE EXCHANGE FOR GENERATING ROBUST OPTIMAL DESIGNS

2.1
Introduction to Candidate Designs

In order to generate robust optimal designs, a process of building various designs and allocating a score is required. The purpose of having an algorithm is to take a design and make alterations while trying to improve a predefined criterion. Since 1972, when the first exchange algorithm was created, many researchers have introduced unique ways to simplify the computational process (Mitchell 1974; Cook and Nachtsheim 1980; Johnson and Nachtsheim 1983; Meyer and Nachtsheim 1995; Atkinson et al. 2007). With several different algorithms to choose from, the task is to select the one that is both computationally efficient and results in the most optimal design. Each algorithm follows a similar initiation process: providing a nonsingular design, scoring the design based on a criterion, exchanging elements between each point of the design space χ, rescoring the design, and comparing the results of the new and old designs. Differences between algorithms can be with respect to the space of elements permitted to exchange into the design or the number of swaps occurring during each exchange.

The algorithm chosen for this project was the coordinate exchange algorithm (CEXCH), designed by Meyer and Nachtsheim in 1995. The CEXCH is unique because it does not require a discrete candidate list of design points. "This candidate-list-free property is a crucial advantage and the reason we use it as the basis for our algorithms" [14]. When the algorithm was initially created, it was intended for creating optimal designs for completely randomized experiments, where the experimental observations were independent. Now, the algorithm has been tuned to adapt to all sorts of experiments, including split-plot, two-way split-plot, mixtures, and many other experiment-specific designs [16]. Another reason why it is a convenient tool is that "the coordinate-exchange algorithm runs in polynomial time, which means that the time it needs to find an optimal design does not explode when the size of the design and the number of factors increases” [10].

2.2

The Coordinate Exchange

CEXCH works as follows: a specified number of design points N, experimental factors K, model, and optimality criterion are chosen. The model is unique to each experiment and allows researchers to decide the factors' order and interactions. To begin, an initial design is created by putting together an NK-dimensional space filled with points randomly chosen from a uniform distribution. The initial design is then evaluated based on the criteria and scored. Next, the algorithm iterates through the rows and columns of the design matrix $\mathbf{X}$. At each coordinate, each element from a list of candidate points is swapped into the design and then evaluated using the criteria. The points swapped into the design range from both ends of the design space and contain multiple points in between endpoints. The algorithm then compares the newly calculated score with the initial score to identify if an improvement occurred. If an improvement was made, then the exchange is made permanent, and the process continues through the rest of the design; if not, then the original value is retained, and the process moves on. This process goes through the entire design and then begins again as long as the design score continues to improve at each iteration. Once the algorithm passes through the full design matrix and no more improvements are found, the exchange process halts, and the design is returned.

Shown below is an outline of how CEXCH works. The space that the algorithm covers is represented by G and signifies a regular sequence on [-1, 1].

Algorithm 1 Element-wise CEXCH Pseudocode

Inputs: $K :=$ number of experimental factors, $N :=$ number of affordable experimental runs, $f :=$ the optimal design criterion

// Randomly instantiate an $N \times K$ design matrix from a uniform distribution

$\mathbf{X} \leftarrow \{x_{ij} \overset{i.i.d}{\sim} U(-1,1) \text{ for } i = 1, \ldots, N, j = 1, \ldots, K\}$

while $\{$*improvements to* f *are found* $\}$ **do**

  for $i = 1, 2, \ldots, N$ **do**

    for $j = 1, 2, \ldots, K$ **do**

      $\mathbf{X} \leftarrow \text{argmin}_{x_{ij} \in \mathrm{G}} f(x_{ij} | \mathbf{X}_{-ij})$ // coordinate proposal is a univariate optimization

    endfor

  endfor

endwhile

Output: $\mathbf{X} :=$ a locally optimal design

Though the algorithm claims to find the design that can no longer be improved through coordinate exchanges, the design may only be a local optimal design. The reason it may be a local optimal design has to do with the fact that at the beginning of the whole process, the initial design was randomly generated, and thus, the algorithm is dependent on those points. "Note that exchange algorithms are heuristics which do not necessarily converge to a global optimum. Thus, the algorithms should be run multiple times from a variety of randomly generated initial designs to find a near-optimal design" [14].

Using the coordinate exchange algorithm to find robust optimal designs is useful because of the explained characteristics above; specifically, the fact that numerous random initial designs will be used to try and find the optimal design. In research done by Patchanok Srisuradetchai (2015), his focus was to find optimal robust designs using a point exchange algorithm. Reasons why CEXCH will prove to find better designs revolves around drawbacks in the point exchange algorithm.

One fault being that the point exchange (PEXCH) algorithm requires the researcher to specify the set of candidate points which can be a setback with large numbers of factors or size of candidate set. Because the algorithm is subject to the defined set of candidate points, it is not allowed to try any number not given. Researchers employing the PEXCH algorithm can generate candidate lists that are more defined by adding more points; however, a potential drawback is the increase in computational runtime [7]. Since CEXCH starts off with randomly chosen points between the bounds of the design, the chance of getting a better optimal design increases. Since the CEXCH starts by generating a design using a uniform distribution, more specific values get to be tested. These randomly generated values carry more decimal places allowing for a finer mesh of points as initial starting points. These defined starting points allow the CEXCH to search more places in the design space. Later on in the paper, the subject of using OPTIM to find EXACT designs along with CEXCH will be explained and why using the two of them together has become an option in finding the global optimal design.

2.3

Robust Criterion

As mentioned previously, one of the main benefits of using robust functions is that they take into consideration the possibility of accidents occurring, alluding to incomplete observations. The way robust criteria works can be explained in three steps. First, the function takes in a design matrix and transforms it into a model matrix, which depends on the number of factors and the desired model. Then, in a sequential manner starting from the top row and going down, a row is omitted and scored depending on the set criterion. This value is then stored in a list that holds all scores. The last step involves taking the list of scores and finding the optimal score. Depending on whether the criterion is a maximum or minimum, it determines whether the desired score is the smallest or largest value in the list. The optimal score is the output of the function and is used to explain how robust a design is. Depending on the number of runs and factors, this process can be altered to allow for a robust design that will work well with two or more missing observations.

Algorithm 2 Robust Criterion Pseudocode

Inputs: $\mathbf{X}$:= design matrix, N := number of rows
$\mathbf{F} \leftarrow$ makemodelmatrix($\mathbf{X}$)
det.inf $\leftarrow$ det$|\mathbf{F}'\mathbf{F}|$
if det.inf < .Machine\$double.eps$^{\frac{1}{2}}$:
  criterion.score $\leftarrow 999$
else
  sc $\leftarrow$ //empty list
  for $i = 1, ..., N$ **do**
    // // Matrix $\mathbf{F}_{-i}$ is the result of extracting row i from the matrix $\mathbf{F}$.
    mat $\leftarrow (\mathbf{F}^{'}_{-i}\mathbf{F}_{-i})$
    if det.inf.2 < .Machine\$double.eps$^{\frac{1}{2}}$:
      sc$[i] \leftarrow 999$
    else
      sc$[i] \leftarrow$ criterion(**mat**)
  endfor
criterion.score $\leftarrow$ max(**sc**)
return criterion.score

2.3.1

Analyzing a Robust Design

As explained in Algorithm 2, the robust criterion is going through a design, dropping a point, and then scoring the design. In Figure 2.1, we see each step in dropping a point and scoring the design. In the figure, the **red** dot is the point that is dropped and the black points are the remaining observations. This particular example comes from a Robust I design that will be brought up again in Figure 2.6 and Table 2.5. Above each design is the I score when the red point is dropped. According to the Robust Criterion, once the list of I scores is gathered, the highest value is returned as the robust score. In this case, it is the design that has the I score of 1.654 and is found in row 2 column 2. We learn from this plot, that this design suffers the most when the middle point is lost. The design with the I score of 1.031 (row 3 column 1), has the smallest robust score meaning of all the scenarios of losing an observation, this would be the best case. Researchers may find it helpful to know the importance each point has to ensure that the point(s) is not lost.

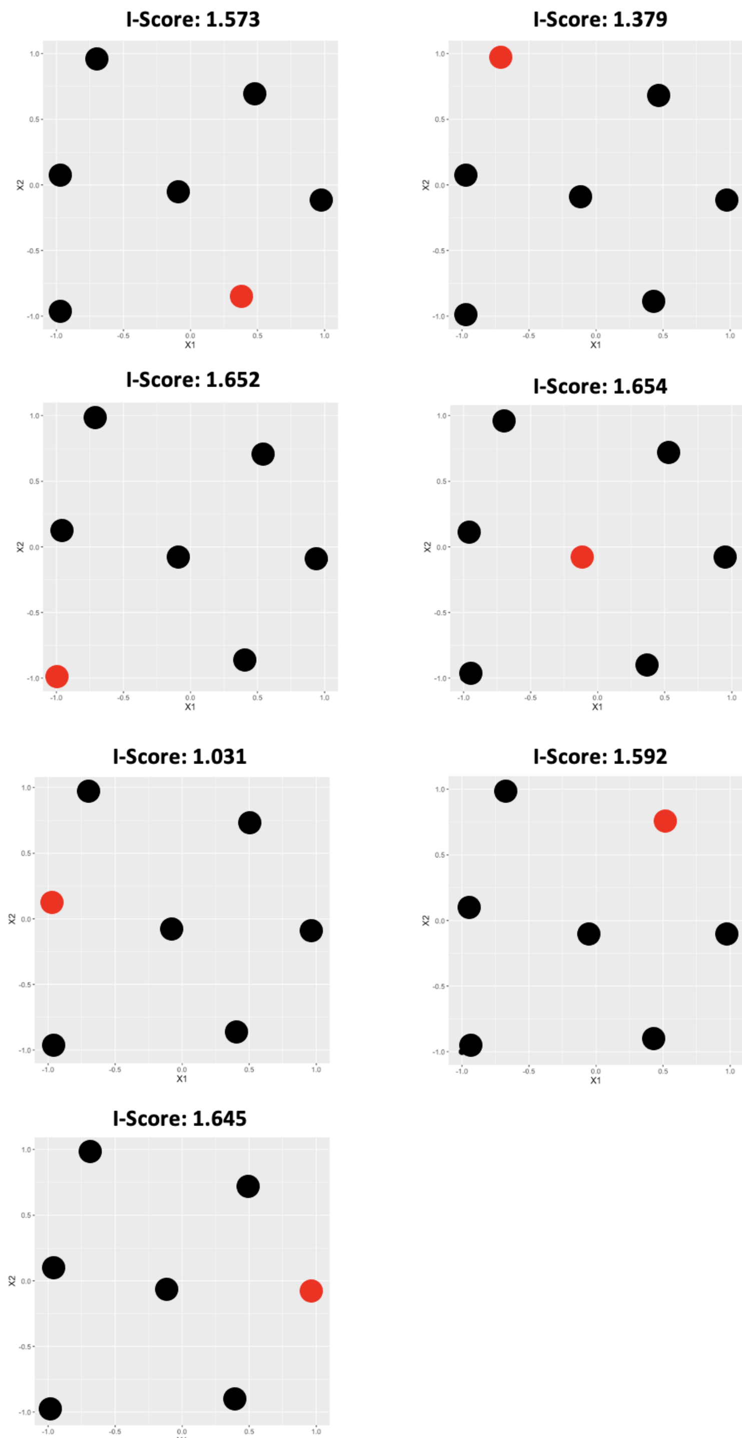

Figure 2.1: The Effect of Drop One in Robust Designs

2.4

Robust Alphabetic Optimality Criteria

To aid in explaining how well a design is, efficiency criteria can be used to mathematically gage how well one would be under a particular criterion. In this study, we are interested in knowing how much information was saved by using a robust design compared to a hypothetical orthogonal design. Srisuradetchai utilized three different forms of efficiency to describe his designs: criterion-efficiency, min criterion-efficiency and leave-m-out criterion-efficiency; each one demonstrating different robust strengths to that design. [15]. Written out below are the formulas and explanations as to how each formula is calculated and its use. To be consistent with notation given previously, $\mathbf{F}$ represents the model matrix and $N \times p$ are the number of observations and parameters in a second-order model. $\mathbf{M}(\mathbf{X}) = \mathbf{F}'\mathbf{F}$ and the $-i$ refers to the absence of an observation in a design which will be shown as a subscript (i = 1, 2, ..., N). Upon removing the ith term, the information matrix will be referred to as $\mathbf{F}'_{-i}\mathbf{F}_{-i}$.

2.4.1

Variations of the D-efficiency

The traditional D*-efficiency is defined as:

$$D^*\text{-efficiency} = \left(\frac{|\mathbf{M}(\mathbf{X})|}{N}\right)^{\frac{1}{p}} \times 100.$$

The minimum of D_i reflects the worst-case scenario of have one missing observation:

$$D_i = \frac{|\mathbf{F}'_{-i}\mathbf{F}_{-i}|^{\frac{1}{p}}}{N-1} \times 100,$$

$$\text{Min } D = \min_{1 \leq i \leq \text{N}} \{D_i\}.$$

Since the Min D is focused on reporting the min efficiency for the $-i$ design, this value is compared to as a lower bound. The higher the Min D, the more robust it is compared to other designs given missing data occurs.

The *leave-m-out* efficiency D-efficiency is defined as:

$$D^{(m)} = \frac{\sum_{t \epsilon T_m} D_i}{\binom{N}{m}},$$

which is the average of all the different scenarios of a design missing m points. T_m is the set of all possible subset designs of size $N - m$, where m is the number of missing points, and D_i is the D-efficiency of subset design $t \epsilon T_m$. The *leave-one-out* efficiency is a very intuitive efficiency to use in this problem space because of its ability to take into consideration all the scored efficiencies of $-i$ in the design and compute an average. For example, when $m = 1$, $D^{(1)}$ is the average of D-efficiencies of all possible designs having one missing point. The absence of even just one observation can greatly affect the efficiency.

2.4.2
Variations of the A-efficiency

The traditional A*-efficiency is defined as:

$$A\text{*-efficiency} = \frac{p}{\mathrm{tr}[\mathbf{M}^{-1}(\mathbf{X})]} \times 100.$$

The minimum of A_i reflects the worst-case scenario of have one missing observation:

$$A_i = \frac{p}{\mathrm{tr}[(N-1)(\mathbf{F}'_{-i}\mathbf{F}_{-i})^{-1}]} \times 100,$$

$$\text{Min } A = \min_{1 \le i \le \mathrm{N}} \{A_i\}$$

The *leave-m-out* efficiency A-efficiency defined as:

$$A^{(m)} = \frac{\sum_{t \epsilon T_m} A_i}{\binom{N}{m}},$$

which is the average of all the different scenarios of a design missing m points. T_m is the set of all possible subset designs of size $N - m$ and A_i is the A-efficiency of subset design $t \epsilon T_m$.

2.4.3
Variations of the I-efficiency

The traditional I*-efficiency is defined as:

$$I\text{*-efficiency} = \frac{V}{N \ \mathrm{tr}[\mathbf{M}^{-1}(\mathbf{X}) \int_{\chi} \mathbf{f}'(\mathbf{x}'_i)\mathbf{f}(\mathbf{x}'_i)\mathbf{dx}]} \times 100.$$

The minimum of I_i reflects the worst-case scenario of have one missing observation:

$$I_i = \frac{V}{(N-1)\ \mathrm{tr}[(\mathbf{F}'_{-i}\mathbf{F}_{-i})^{-1}\int_{\chi}\mathbf{f}'(\mathbf{x}'_i)\mathbf{f}(\mathbf{x}'_i)\mathbf{dx}]} \times 100,$$

$$\text{Min } I = \min_{1\leq i\leq \text{N}}\{I_i\}$$

The *leave-one-out* efficiency I-efficiency defined as:

$$I^{(m)} = \frac{\sum_{t\epsilon T_m} I_i}{\binom{N}{m}},$$

which is the average of all the different scenarios of a design missing m points. T_m is the set of all possible subset designs of size $N-m$ and I_i is the I-efficiency of subset design $t\epsilon T_m$.

2.4.4

Relative Efficiency

Relative Efficiencies are used to compare two designs by dividing one design efficiency score by the other and multiplying by 100. This percentage tell us whether design one or two has a better efficiency score. A relative efficiency larger than 100 indicates that Design 1 is better than Design 2 in terms of optimality.

$$R.E.(D_1, D_2) = (D_1/D_2) * 100$$

2.5

Details

Once the robust criterion has been decided and created, the next step is selecting an algorithm. In this experiment, the coordinate-exchange algorithm (CEXCH) was chosen and implemented using the Julia programming language. Julia was selected for this project due to its quick compilation of functions and its ability to complete tasks swiftly. As explained in section 2.2 of this chapter, CEXCH takes in three parameters and returns a list of values determined by the researcher. I have chosen to include the initial matrix, initial design score, optimal matrix, optimal score, and number of iterations in the returned values. Both the optimal matrix and score are the most valuable results, as they represent the end products of the CEXCH algorithm. Since CEXCH is designed to search for the global optimum, the researcher needs to adjust the algorithm based on whether they are searching for the global maximum or minimum. This choice is determined by the design of the criterion.

2.6

Validation

The following section is dedicated to validating the coordinate exchange results to that of the exact algorithm (EXACT) provided John J. Borkowski [17] and a point exchange algorithm (PEXCH) by Patchanok Srisuradetchai [15]. The EXACT designs are created by using the optimal exact formulas provided in section 1.1.2. While a typical PEXCH initiates by creating a candidate set of points, Srisuradetchai's PEXCH algorithm incorporated a scattered starting design. To have scattered starting design points, the design region was partitioned and then one point was randomly chosen from each partition region. This approach ensured the starting design points were not clustered in one place which might happen in completely random selection of designs with a small sample size. Next, the point design was sequentially filled with points that have the highest prediction variance, which was equivalent to maximizing the determinant according to the rank-one updated formula. In the following sections, graphics and tables will be provided with the intent of showing how the CEXCH was able to achieve equal to or better results than the EXACT and PEXCH via criterion scores, robust criterion scores, and efficiencies table.

2.7

Reading Plots and Efficiency Tables

For each design case in this chapter, a plot and efficiency table will be provided to illustrate the design and allow comparison. Each plot will have the following: a title that specifies the selected criteria, the traditional alphabet criteria score, the robust criteria score, and either a two-dimensional or three-dimensional plot of the design.

The reported scores indicate how well that design did when scored using that criterion. For all criteria presented in this research, smaller the score the better.

Below the graphics, there will be an efficiency table. The efficiencies will be arranged so that the first group will line up with the type of criteria that generated the designs. This means that if the designs were generated using the D and Robust D, then the first group will be about D efficiencies. In the table, there will also be an efficiency that is highlighted **red**. Highlighting that particular value serves two purposes. One, it is the efficiency that best represents the robustness to one data loss for those criteria. Two, this value indicates whether the CEXCH has generated a design that is more robust than the design generated from the PEXCH. If the value is greater than 100, then the CEXCH design found a more robust design. I will only highlight the efficiency that correlates with the Min efficiency of the criteria being used to generate the design as that is the value that indicates robustness. In a real life scenario, a researcher may look across all efficiencies to understand the risk of using a design. If there is a low chance of data loss, using a robust design may not be wise. In this paper, the main focus is to compare optimal robust designs, so when making comparisons, the Min efficiency is going to be my indicator to whether my robust design is better or worse.

Efficiency values tell how well the design would do compared to a hypothetical orthogonal design. So, when reading the table, the higher the value the better. Depending on the section in this paper, the first two to three columns show which algorithm was used to generate the design. There will be occasions when there will be a 0 value in these columns. This indicates that this design is not robust to data loss. The following columns are relative efficiencies comparing design efficiencies between the different algorithms. The middle columns provide efficiencies comparing the CEXCH to the EXACT and the PEXCH to the EXACT. This value indicates whether a robust design is better than using an exact one. The last column compares the robust designs computed from the CEXCH and PEXCH. This value is of importance because I want to know whether the CEXCH algorithm is able to generate better designs than the PEXCH. In the second and third column sections of the table, there may be an N/A value. This value shows up when the calculation may have been trying to divide by 0.

When comparing designs, there are three different possible outcomes, those being Better, Equal, and Worse. A better design will be different in point arrangement, have a smaller robust score, and a higher efficiency. An equal design may have the same or a different point arrangement but will have the same robust score and efficiency, resulting in a 100 relative efficiency score. A worse design will have a different design with a higher robust score and a lower efficiency. The robust and efficiency scores are synonymous in meaning, thus if a design has a higher robust score than another, its efficiency score will be less than the other design.

2.7.1

Two-factor D-Optimal Designs

The 7-Point Design

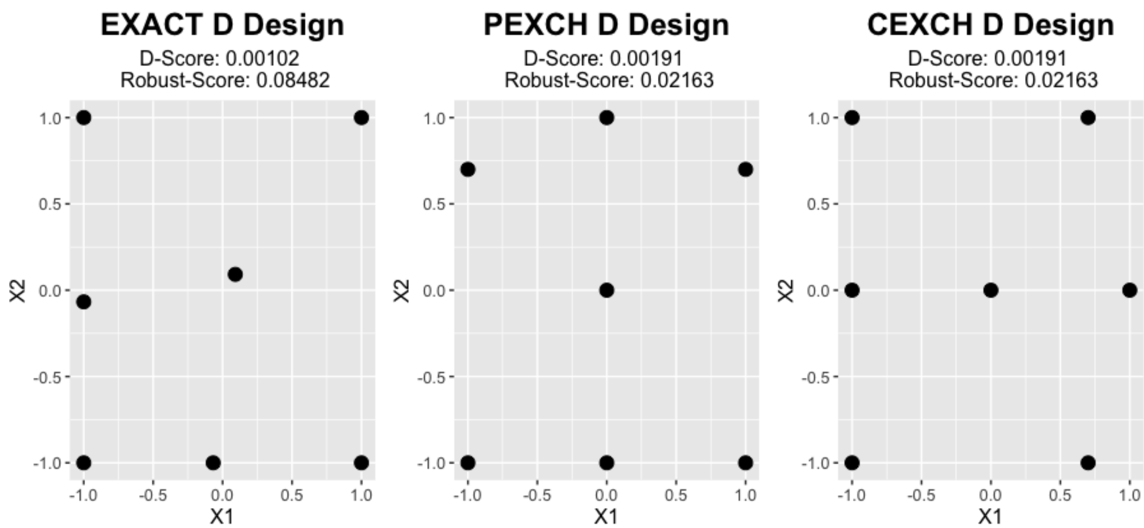

Figure 2.2: Seven-point D-optimal EXACT (left), PEXCH (center), and CEXCH (right) designs for a second-order model in two factors.

Table 2.1: Properties of the 7-point 2-factor D-optimal designs.

Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
D-efficiency	45.029	40.567	40.567	90.091	90.091	100
Min D-efficiency	25.144	31.567	31.567	125.581	125.581	**100**
Leave-1-out D-efficiency	34.789	33.24	33.24	95.547	95.547	100
A-efficiency	26.598	23.534	23.534	88.48	88.48	100
Min A-efficiency	3.128	4.565	4.565	145.94	145.94	100
Leave-1-out A-efficiency	12.509	14.29	14.29	114.238	114.238	100
I-efficiency	21.653	20.147	20.147	93.045	93.045	100
Min I-efficiency	2.783	3.627	3.627	130.327	130.327	100
Leave-1-out I-efficiency	10.405	12.43	12.43	119.462	119.462	100

Results for 7-point 2-factor D-optimal designs are shown in Figure 2.2 and Table 2.1. From the plots, we see two different point arrangements; this is because the CEXCH design is just a 90 degree rotation of the PEXCH design. With these designs, the axes are interchangeable, meaning that though the plots above look different, they are actually the same if the axes were to be switched. From the table, we see that the PEXCH and CEXCH both achieved the Min D-efficiency of 31.567 which is better than the EXACT design with a 25.144.

The 8-Point Design

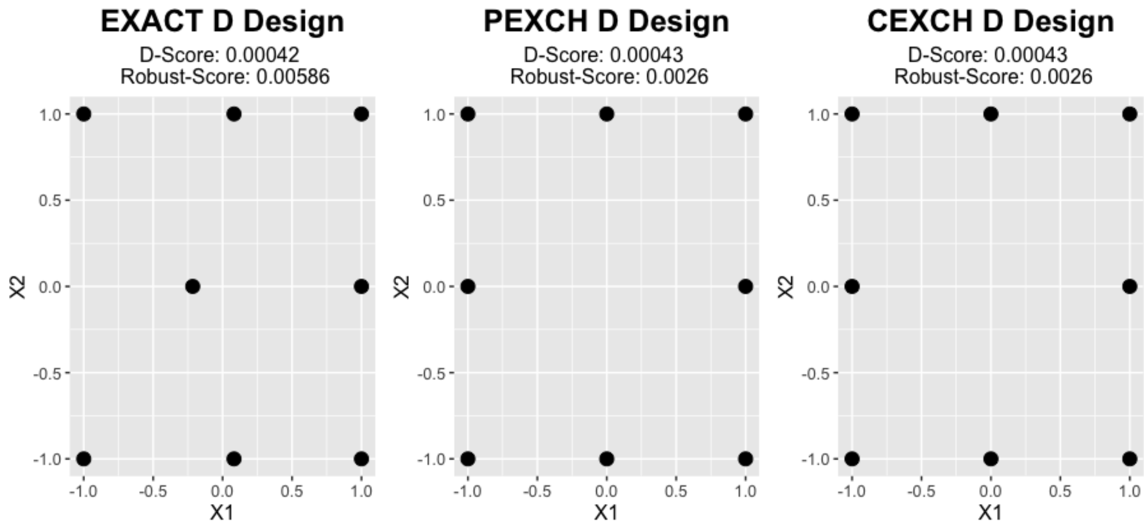

Figure 2.3: Eight-point D-optimal EXACT (left), PEXCH (center), and CEXCH (right) designs for a second-order model in two factors.

Table 2.2: Properties of the 8-point 2-factor D-optimal designs.

Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
D-efficiency	45.616	45.428	45.428	99.588	99.588	100
Min D-efficiency	33.648	38.514	38.514	114.461	114.461	**100**
Leave-1-out D-efficiency	40.38	40.873	40.873	101.221	101.221	100
Leave-2-out D-efficiency	31.035	29.809	29.809	96.05	96.05	100
A-efficiency	28.899	22.5	22.5	77.857	77.857	100
Min A-efficiency	15.192	16.59	16.59	109.202	109.202	100
Leave-1-out A-efficiency	21.134	17.012	17.012	80.496	80.496	100
Leave-2-out A-efficiency	11.325	10.165	10.165	89.757	89.757	100
I-efficiency	22.619	16.791	16.791	74.234	74.234	100
Min I-efficiency	11.943	12.187	12.187	102.043	102.043	100
Leave-1-out I-efficiency	16.91	13.197	13.197	78.043	78.043	100
Leave-2-out I-efficiency	9.223	7.827	7.827	84.864	84.864	100

Results for 8-point 2-factor D-optimal designs are shown in Figure 2.3 and Table 2.2. The plots show that PEXCH and CEXCH found the same design which has all its points along the edges and in the corners. Score wise, both the PEXCH and CEXCH got Min D-efficiency of 38.514 beating the EXACT design which scored a 33.648. From the efficiency table, we see that the PEXCH and CEXCH scored the same values in all fields. We learn that in this case there does exist a robust design that has better robust properties that an EXACT generated design.

2.7.2

Two-factor A-Optimal Designs

The 7-Point Design

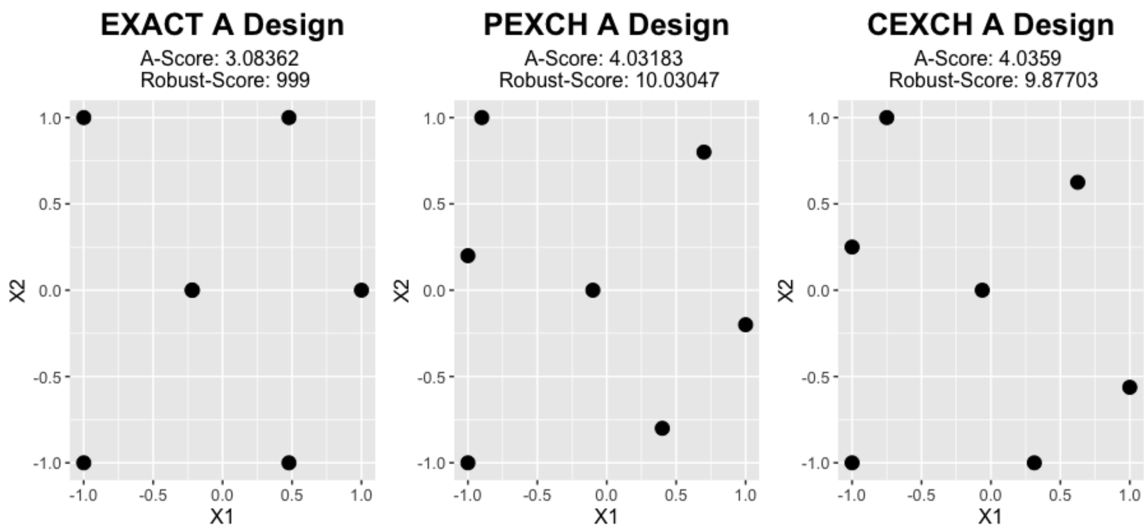

Figure 2.4: Seven-point A-optimal EXACT (left), PEXCH (center), and CEXCH (right) designs for a second-order model in two factors.

Table 2.3: Properties of the 7-point 2-factor A-optimal designs.

Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
A-efficiency	27.797	21.259	21.238	76.479	76.404	99.901
Min A-efficiency	0	9.97	10.125	N/A	N/A	**101.555**
Leave-1-out A-efficiency	7.123	11.888	12.04	166.896	169.03	101.279
D-efficiency	38.535	34.066	33.95	88.403	88.102	99.659
Min D-efficiency	0	23.627	24.26	N/A	N/A	102.679
Leave-1-out D-efficiency	11.444	27.976	28.026	244.46	244.897	100.179
I-efficiency	24.657	20.011	19.851	81.157	80.509	99.2
Min I-efficiency	0	7.446	6.797	N/A	N/A	91.284
Leave-1-out I-efficiency	6.15	11.444	11.634	186.081	189.171	101.66

Results for 7-point 2-factor A-optimal designs are shown in Figure 2.4 and Table 2.3. In this case, the CEXCH generated a design slightly different than the PEXCH; they have a similar structure, but the points are not in the same locations. Looking at the table, we see that the relative efficiency score of CEXCH, PEXCH for Min A-efficiency is 101.555. This means the CEXCH design is 1.555 percent more optimal that the PEXCH design. The table also shows that the EXACT design is not robust to losing data because its Min A-efficiency 0.

The 8-Point Design

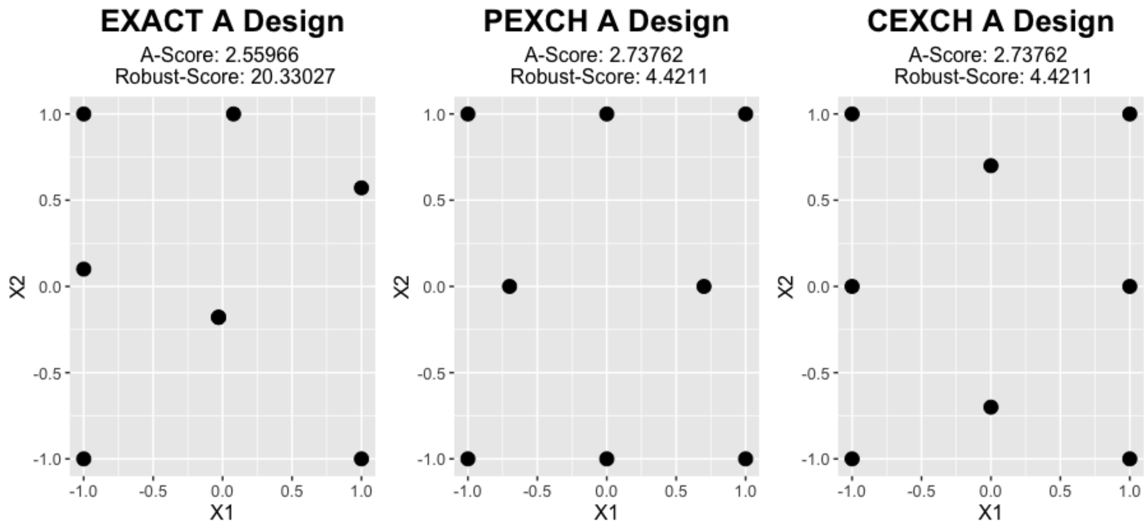

Figure 2.5: Eight-point A-optimal EXACT (left), PEXCH (center), and CEXCH (right) designs for a second-order model in two factors.

Table 2.4: Properties of the 8-point 2-factor A-optimal designs.

Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
A-efficiency	29.301	27.396	27.396	93.499	93.499	100
Min A-efficiency	4.216	19.388	19.388	459.867	459.867	**100**
Leave-1-out A-efficiency	17.945	20.439	20.439	113.898	113.898	100
Leave-2-out A-efficiency	6.434	11.449	11.449	177.945	177.945	100
D-efficiency	41.618	44.039	44.039	105.817	105.817	100
Min D-efficiency	23.972	35.937	35.937	105.817	149.912	100
Leave-1-out D-efficiency	34.885	39.254	39.254	112.524	112.524	100
Leave-2-out D-efficiency	15.691	28.208	28.208	179.772	179.772	100
I-efficiency	25.294	22.569	22.569	89.227	89.227	100
Min I-efficiency	4.32	15.833	15.833	366.505	366.505	100
Leave-1-out I-efficiency	15.542	17.358	17.358	111.684	111.684	100
Leave-2-out I-efficiency	5.438	9.552	9.552	175.653	175.653	100

Results for 8-point 2-factor A-optimal designs are shown in Figure 2.5 and Table 2.4. From the plots, we see that PEXCH and CEXCH generated the same design which focused on symmetry and having a point in each corner. The table explains that PEXCH and CEXCH designs both have a Min A-efficiency of 19.388 which is better than the EXACT design score of 4.216. This robust generated design appears to do well in all efficiencies compared to a non-robust design. This conclusion comes from looking over the efficiency values in the R.E.(P, E) and R.E.(C, E) columns. In the different A-efficiencies provided, the robust design also did well in Leave-1-out A-efficiency and Leave-2-out A-efficiency.

2.7.3

Two-factor I-Optimal Designs

The 7-Point Design

Figure 2.6: Seven-point I-optimal EXACT (left), PEXCH (center), and CEXCH (right) designs for a second-order model in two factors.

Table 2.5: Properties of the 7-point 2-factor I-optimal designs.

Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
I-efficiency	24.907	21.147	19.66	84.904	78.934	92.968
Min I-efficiency	0	9.094	10.075	N/A	N/A	**110.787**
Leave-1-out I-efficiency	6.185	12.059	11.375	194.972	183.913	94.328
D-efficiency	38.326	37.194	31.893	98.998	83.215	84.057
Min D-efficiency	0	25.408	21.754	N/A	N/A	85.619
Leave-1-out D-efficiency	11.382	31.042	26.362	272.729	231.611	84.924
A-efficiency	27.426	23.619	20.027	86.119	73.022	84.792
Min A-efficiency	0	7.221	8.17	N/A	N/A	113.142
Leave-1-out A-efficiency	7.046	13.194	11.457	187.255	162.603	86.835

Results for 7-point 2-factor I-optimal designs are shown in Figure 2.6 and Table 2.5. In this scenario, we find that each algorithm generated a different design. From the table we see from the highlighted 110.787 that the design generated by the CEXCH algorithm is 10.787 percent more efficient than the PEXCH design. The CEXCH design has a Min I-efficiency of 10.075 and the PEXCH design has a 9.094. The EXACT design has a Min I-efficiency of 0, meaning it is not robust to losing data.

The 8-Point Design

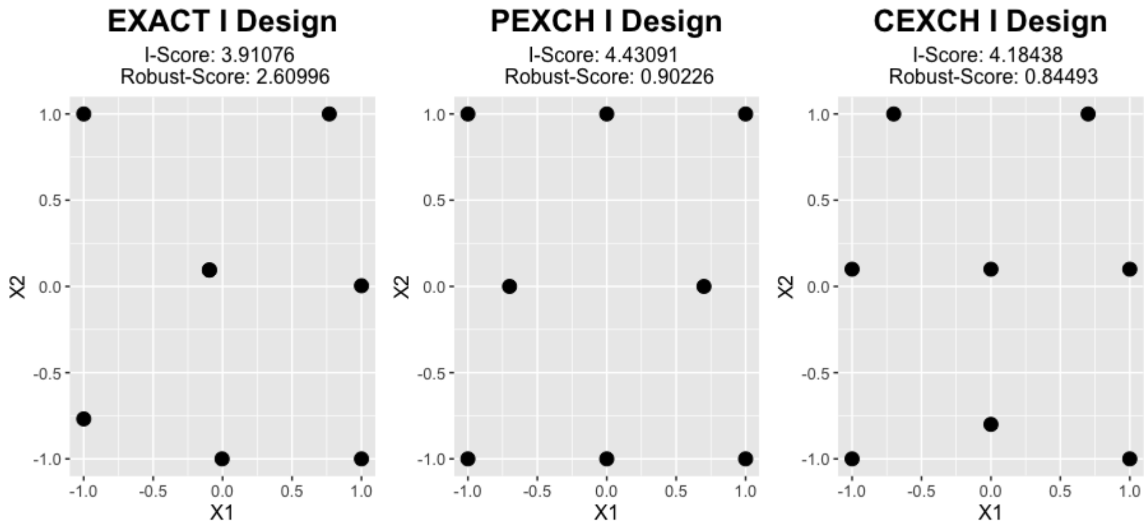

Figure 2.7: Eight-point I-optimal EXACT (left), PEXCH (center), and CEXCH (right) designs for a second-order model in two factors.

Table 2.6: Properties of the 8-point 2-factor I-optimal designs.

Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
I-efficiency	25.57	22.569	23.898	88.264	93.461	105.889
Min I-efficiency	5.474	15.833	16.907	289.24	308.86	**106.783**
Leave-1-out I-efficiency	15.899	17.358	18.048	109.177	113.517	103.975
Leave-2-out I-efficiency	5.601	9.552	9.647	170.541	172.237	100.995
D-efficiency	0.919	44.039	4 0.582	107.625	99.176	92.15
Min D-efficiency	26.183	35.937	31.978	137.253	122.133	88.983
Leave-1-out D-efficiency	34.482	39.254	36.178	113.839	104.919	92.164
Leave-2-out D-efficiency	15.538	28.208	27.707	181.542	178.318	98.224
A-efficiency	28.999	27.396	27.444	94.472	94.638	100.175
Min A-efficiency	5.861	19.388	17.271	330.797	294.677	89.081
Leave-1-out A-efficiency	18	20.388	20.4	113.55	113.333	99.809
Leave-2-out A-efficiency	6.489	11.449	10.696	176.437	164.833	93.423

Results for 8-point 2-factor I-optimal designs are shown in Figure 2.7 and Table 2.6. From the plots, we see that all three algorithms produced different designs. The PEXCH and CEXCH found designs that have interesting symmetrical properties. In the R.E.(C, P) column of the table, we learn that the CEXCH design outperformed the PEXCH design in all I-efficiencies. This is the first case analyzed so far that has that characteristic. In the R.E(C, E) column, we find that the CEXCH design does better than the EXACT design in Min-I efficiency, Leave-1-out efficiency, and Leave-2-out efficiency. From the results, I would conclude that the CEXCH algorithm was able to find a new design that performs better to loss of data than other algorithms according to my returned efficiencies.

2.8

Validation Conclusion

Table 2.7: Chapter Summary for 2 Factor N = 7 and 8 scores.

Experiment Design	Robust - D	Robust - A	Robust - I
2-Factor N = 7	Equal	Better	Better
2-Factor N = 8	Equal	Equal	Better

By validating CEXCH with Borkowski [17] and Srisuradetchai [15], we now have a good understanding of how well the algorithm compares to what has already been published. From seeing the side-by-side plots and efficiency tables, and summary 2.7 it can be said that the CEXCH algorithm has the ability to perform equal to or better than PEXCH; this can be attributed to CEXCH's ability to search through numerous random designs to find the optimal design. And though the EXACT design is helpful in many response surface design searches, its ability to prescribe a design that is robust to missing observations is often weak and not helpful. In the next chapter we will use this knowledge to discuss designs for bigger factors (K) and observations (N).

CHAPTER 3

FURTHER VALIDATION OF THE COORDINATE EXCHANGE FOR GENERATING ROBUST OPTIMAL DESIGNS WITH $K = 3$ AND EXTENSION TO UNADDRESSED PROBLEMS.

3.1

Introduction

In the following sections, we will present the efficiencies and plots for designs generated using the algorithms EXACT, PEXCH, and CEXCH. Each section will begin with an overview of the efficiencies for each design, followed by an explanation and a 3D plot illustrating the design. Designs generated using PEXCH are not included due to the absence of reported data by Srisuradetchai [15]. However, he did report the efficiencies for each design, which will be presented in the tables. The following sections are specific to designs with $K = 3$, and N ranges from 11 to 13. The choice of starting at 11 is to ensure there are enough runs to estimate all the parameters in a second-order model, including interactions. The tables and 3D plots for each design will be in the Appendix at the end of the chapter. The 3D plots are helpful in demonstrating the uniqueness of each criterion and the variability among algorithms.

3.1.1

Three-factor D-Optimal Designs

From Table 3.3, starting with $N = 11$, CEXCH-generated designs performed better overall for efficiencies related to D. In the other categories, CEXCH was very close or slightly behind. In the Figure 3.1 for Robust-D, with $K = 3$ and $N = 11$, it's evident that the point on the bottom face of the cube is not at the center but is slightly shifted outward. This movement in the point might be one of the contributing factors to the robust design. Continuing in the table, when $N = 12$, designs resulting from CEXCH and PEXCH are similar, often times only differing by a hundredth. Similarly, in the figure 3.1, the point on the bottom face is shifted away from the center. It's noteworthy that CEXCH and PEXCH outperformed EXACT in terms of Min-D efficiency, while in other aspects, the

EXACT algorithm performed better. In this case, determining whether a robust design is necessary becomes crucial for researchers when choosing the optimal design.

3.1.2

Three-factor A-Optimal Designs

Referring to Table 3.4, starting with $N = 11$, the CEXCH-generated design outperformed the PEXCH design in all cases and surpassed the EXACT design in all instances related to robustness. Identifying the specific point in the plot contributing to these superior efficiency scores is more challenging, but it remains intriguing to observe the arrangement of the design in comparison to previously examined cases. In the section for $N = 12$, CEXCH designs excelled in efficiencies related to A. In other efficiency measures, EXACT and PEXCH designs alternated in achieving higher results. In the final table, representing $N = 13$, CEXCH designs surpassed both EXACT and PEXCH designs in all efficiency categories, except for I-efficiency.

3.1.3

Three-factor I-Optimal Designs

From Table 3.5, starting with $N = 11$, the CEXCH-generated design performed better than the PEXCH design and was better than the EXACT design with efficiencies concerning robustness. It is interesting to compare the plots and observe the distinct locations of the points. In the table for $N = 12$, the CEXCH design outperformed the PEXCH design in all efficiency categories related to A and I. However, for D-efficiency and I-efficiency, the EXACT design reported higher values but significantly lower than CEXCH in other aspects. In the last table for $N = 13$, CEXCH performed very similarly to PEXCH, often differing by only a hundredth.

3.2

Section Conclusion

Table 3.1: Chapter Summary for 3 Factor N = 11, 12 and 13.

Experiment Design	Robust - D Score	Robust - A Score	Robust - I Score
3-Factor N = 11	Better	Better	Better
3-Factor N = 12	Equal	Better	Better
3-Factor N = 13	Equal	Better	Better

Table 3.1 provides a summary view of how the CEXCH produced robust optimal designs compared to the PEXCH. The designs for this experiment can be found in Figure 3.3 in the Appendix. From the results, only with the Robust D criteria was there ever a time when the designs were equal in score. In all other cases, CEXCH designs were better. This is further validation in a higher dimension that this algorithm can generate robust optimal designs.

3.3

Comparing Three-factor Designs when $N = (14, 15 \text{ and } 16)$.

Compared to the section above, the following three-factor designs are only of generated EXACT and CEXCH designs. At this point in paper, I am convinced that CEXCH performs equal to and often times better than PEXCH. As number of observations increases, it will be interesting to observe if robust design can still perform better than EXACT design across the board of efficiencies. In the following sections we will observe second order models when $K = 3$ and $N = 14$, 15, and 16. The tables and plots for each experiment will be found in the Appendix of this chapter.

3.3.1

Three-factor D-Optimal Designs with $N = (14, 15 \text{ and } 16)$.

Referring to Table 3.6, it is evident that for $N = 14$, the design generated by CEXCH performed significantly better than that generated by EXACT. The appearance of N/A values in the table indicates that, during the calculation of efficiency, a singular matrix was encountered. Frequent N/A responses can make comparisons challenging, but in this instance, we can still observe that CEXCH created a robust overall design. Examining the plots, we can see the differences between the designs produced by EXACT and CEXCH. CEXCH's design shows clear symmetry compared to the more random appearance of the EXACT design. In the subsequent table where $N = 15$, CEXCH performed better compared to the design generated by the EXACT algorithm, especially in terms of efficiencies related to A. Once again, when looking at the plots, the unique structures of the two designs contribute to the efficiency scores. In the table for $N = 16$, CEXCH's design excelled in all categories except for two, indicating its high quality as a design.

3.3.2

Three-factor A-Optimal Designs with N = (14, 15 and 16).

From Table 3.7, starting with $N = 14$, the CEXCH-generated design is identical to the EXACT design. However, when $N = 15$, the CEXCH design performs better, achieving higher efficiency scores in all categories. It's interesting to see how effectively the CEXCH design outperformed the EXACT design, given that both were intended to excel under the A-criterion. In the last case, when $N = 16$, the CEXCH design once again performs well, securing higher efficiency scores than the EXACT design. While N/A values are present, it's worth noting that these designs were not specifically optimized for scenarios with two missing points.

3.3.3

Three-factor I-Optimal Designs with N = (14, 15 and 16).

Referring to Table 3.8, at $N = 14$, the design generated by CEXCH outperforms the EXACT design in all relative efficiency measures for all criteria. In this scenario, it's helpful to examine the 3D plots to observe the distinct orientation of points. The CEXCH design displays greater symmetry, with the exception of two points on the top (see Figure 3.6). In the following case at $N = 15$, the CEXCH design once again surpasses the EXACT design in all efficiency scores. In the plot, the EXACT design achieved optimality by having two points share a location, the CEXCH design assigns a point to the center of each of the six faces of the cube. In the last table, when $N = 16$, the CEXCH design achieves higher efficiencies than the EXACT design. Unlike the CEXCH design, which has 16 unique positions for all its points, the EXACT design has two points sharing a location on the cube.

3.4

Conclusion

Table 3.2: Chapter Summary for 3 Factor N = 14, 15 and 16 scores.

Experiment Design	Robust - D	Robust - A	Robust - I
3-Factor N = 11	Better	Equal	Better
3-Factor N = 12	Better	Better	Better
3-Factor N = 13	Better	Better	Better

Having gone through various scenarios of N for the second order model with $K = 3$, it can be concluded that the CEXCH algorithm consistently produces designs that are equal to or better than those generated by the other algorithms. Table 3.2 shows that in eight of the nine scenarios, using a robust optimal criterion proved helpful in finding a more optimal design than simply using an EXACT design. This discovery provides a solid foundation for further research aimed at finding the algorithm's specific strengths and limitations on a case-by-case basis.

With the reliability of CEXCH as a design generator established and verified, the next chapter will focus on creating new first order designs using models with varying factors, K, and observations N. The focus will continue to be with the D and A criterion but in higher dimensions. The K factor will include factors 2 and 5 with N being dependent on the number of factors.

Appendix

Table 3.3: Properties of 3-factor D-optimal designs for N = (11, 12 and 13).

11-point D-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
D-efficiency	44.769	41.889	42.375	93.567	94.653	101.160
Min D-efficiency	0	34.823	35.144	N/A	N/A	**100.922**
Leave-1-out D-efficiency	29.091	36.016	36.418	123.805	125.186	101.116
A-efficiency	21.39	22.592	21.598	105.619	100.972	95.600
Min A-efficiency	0	5.6	5.079	N/A	N/A	90.696
Leave-1-out A-efficiency	13.244	13.404	12.938	101.208	97.690	96.523
I-efficiency	11.73	12.724	11.794	108.474	100.546	92.691
Min I-efficiency	0	3.38	2.78	N/A	N/A	82.249
Leave-1-out I-efficiency	7.164	7.645	7.151	106.714	99.819	93.538

12-point D-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
D-efficiency	44.986	44.921	44.388	99.856	98.671	98.813
Min D-efficiency	38.956	38.974	38.974	100.046	100.046	**100.000**
Leave-1-out D-efficiency	40.639	40.593	40.187	99.887	98.888	99.000
Leave-2-out D-efficiency	29.909	30.441	30.252	101.779	101.147	99.379
A-efficiency	22.364	22.273	21.892	99.593	97.889	98.289
Min A-efficiency	7.642	7.731	7.693	101.165	100.667	99.508
Leave-1-out A-efficiency	16.688	16.637	16.54	99.694	99.113	99.417
Leave-2-out A-efficiency	8.155	8.165	8.237	100.123	101.006	100.882
I-efficiency	11.978	11.901	11.98	99.357	100.017	100.664
Min I-efficiency	3.446	3.506	3.47	101.741	100.696	98.973
Leave-1-out I-efficiency	8.99	8.927	9.102	99.299	101.246	101.960
Leave-2-out I-efficiency	4.449	4.449	4.591	100.000	103.192	103.192

13-point D-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
D-efficiency	46.391	45.443	45.443	97.957	97.957	100.000
Min D-efficiency	38.326	40.259	40.259	105.044	105.044	**100.000**
Leave-1-out D-efficiency	43.12	42.223	42.223	97.920	93.282	100.000
Leave-2-out D-efficiency	38.555	37.583	37.583	97.479	97.479	100.000
A-efficiency	26.948	24.149	24.149	89.613	89.613	100.000
Min A-efficiency	14.335	12.425	12.425	86.676	86.676	100.000
Leave-1-out A-efficiency	21.986	19.788	19.788	90.003	90.003	100.000
Leave-2-out A-efficiency	15.204	13.676	13.676	89.950	89.950	100.000
I-efficiency	13.751	12.511	12.511	90.982	90.982	100.000
Min I-efficiency	8.327	5.868	5.868	70.265	70.470	100.000
Leave-1-out I-efficiency	11.409	10.356	10.356	90.770	90.770	100.000
Leave-2-out I-efficiency	8.004	7.244	7.244	90.505	90.505	100.000

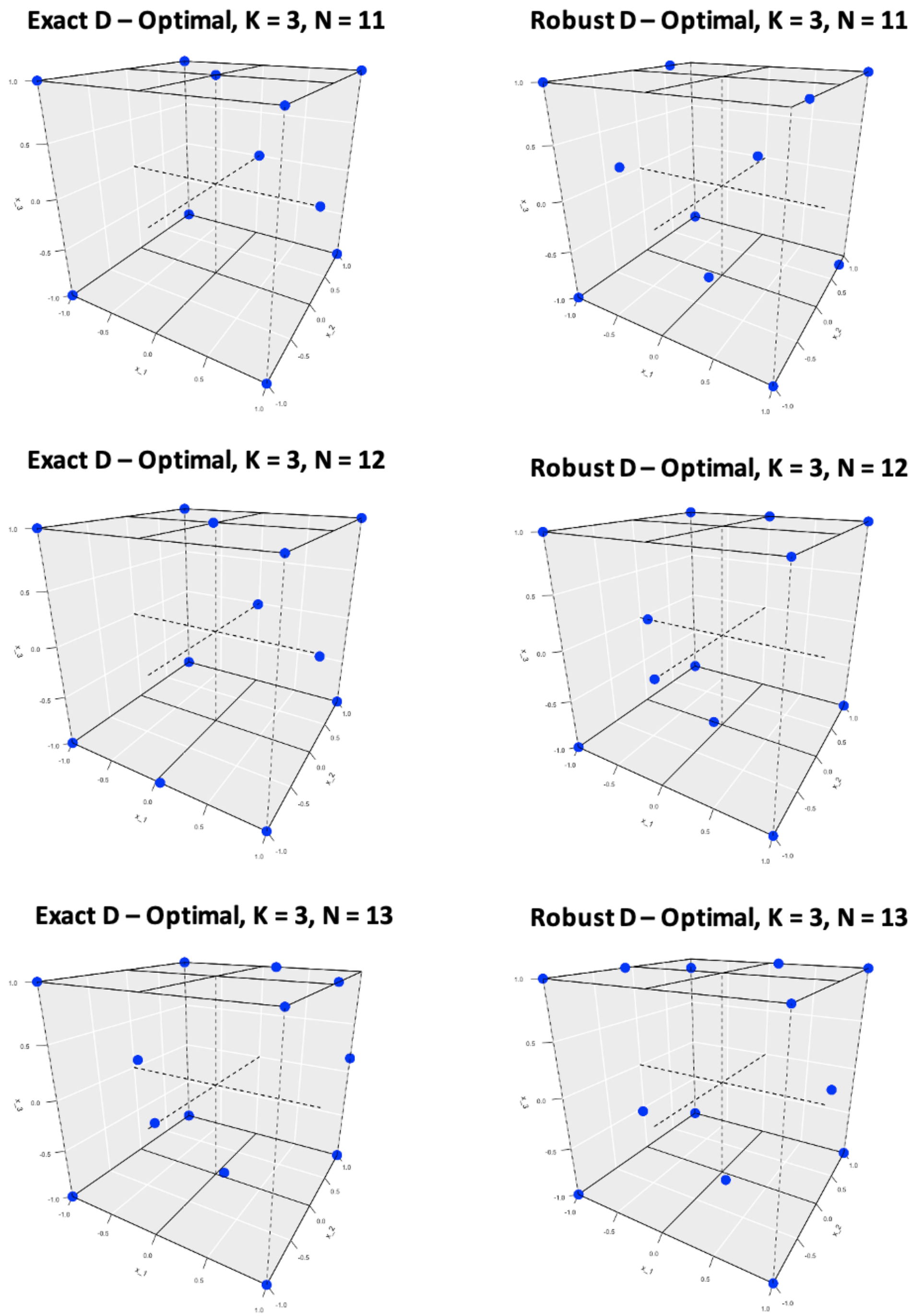

Figure 3.1: Exact and Robust I - Optimal Designs for K = 11, 12, 13

Table 3.4: Properties of 3-factor A-optimal designs for N = (11, 12 and 13).

11-point A-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
A-efficiency	28.891	17.477	18.364	60.493	63.563	105.075
Min A-efficiency	0.014	8.488	9.021	60628.571	64435.714	**106.279**
Leave-1-out A-efficiency	8.905	9.413	10.009	105.705	112.398	106.332
D-efficiency	40.887	28.963	29.392	70.837	71.886	101.481
Min D-efficiency	16.02	22.511	23.453	140.518	146.398	104.185
Leave-1-out D-efficiency	28.041	24.643	25.16	87.882	89.726	102.098
I-efficiency	16.44	11.132	11.335	67.713	68.948	101.824
Min I-efficiency	0.008	4.33	4.645	54125.000	58062.500	107.275
Leave-1-out I-efficiency	5.332	6.067	6.334	113.785	118.792	104.401

12-point A-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
A-efficiency	28.909	23.703	34.939	81.992	120.859	147.403
Min A-efficiency	4.737	14.962	15.498	315.854	327.169	**103.582**
Leave-1-out A-efficiency	16.971	17.118	17.432	100.866	102.716	101.834
Leave-2-out A-efficiency	6.378	8.082	8.356	126.717	131.013	103.390
D-efficiency	42.177	40.504	38.857	96.033	92.128	95.934
Min D-efficiency	30.313	33.023	31.966	108.940	105.453	96.799
Leave-1-out D-efficiency	36.464	36.412	35.088	99.857	96.226	96.364
Leave-2-out D-efficiency	27.108	29.527	28.06	108.924	103.512	95.032
I-efficiency	16.383	13.382	14.067	81.682	85.863	105.119
Min I-efficiency	2.653	7.5	8.173	282.699	308.066	108.973
Leave-1-out I-efficiency	9.83	9.861	10.443	100.315	106.236	105.902
Leave-2-out I-efficiency	3.792	4.761	5.107	125.554	134.678	107.267

13-point A-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
A-efficiency	29.669	25.099	25.994	84.597	87.613	103.566
Min A-efficiency	0.124	18.794	19.661	15156.452	15855.645	**104.613**
Leave-1-out A-efficiency	19.684	20.283	21.119	103.043	107.290	104.122
Leave-2-out A-efficiency	10.312	14.038	14.465	136.133	140.273	103.042
D-efficiency	40.272	39.079	40.471	97.038	100.494	103.562
Min D-efficiency	19.314	33.747	35.058	174.728	181.516	103.885
Leave-1-out D-efficiency	34.726	36.225	37.565	104.317	108.175	103.699
Leave-2-out D-efficiency	27.086	32.358	33.354	119.464	123.141	103.078
I-efficiency	17.083	14.683	14.638	85.951	85.688	99.694
Min I-efficiency	0.068	10.01	10.033	14720.588	14754.412	100.230
Leave-1-out I-efficiency	11.598	11.987	12.044	103.354	103.845	100.476
Leave-2-out I-efficiency	6.171	8.381	8.34	135.813	135.148	99.511

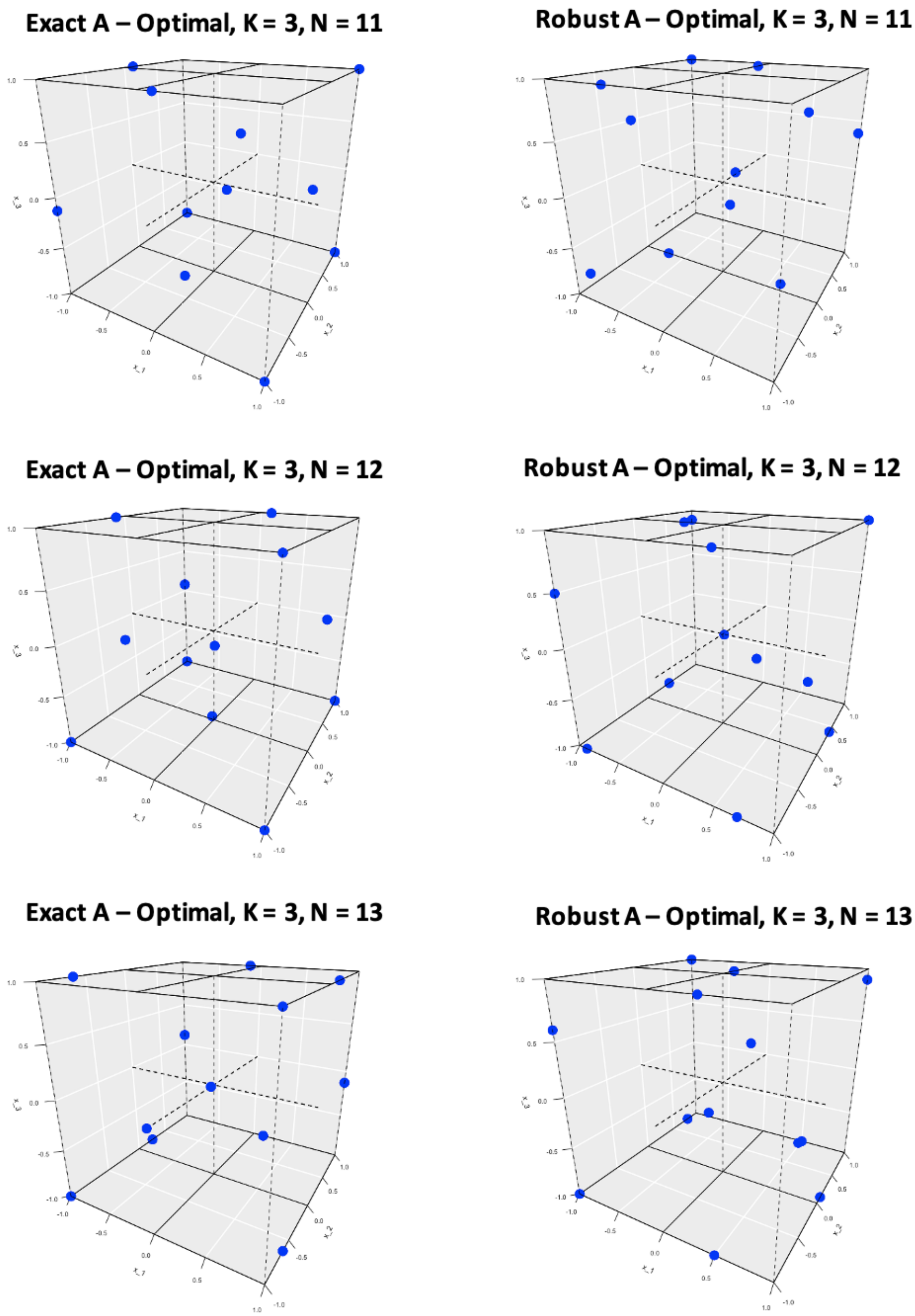

Figure 3.2: Exact and Robust I - Optimal Designs for K = 11, 12, 13

Table 3.5: Properties of 3-factor I-optimal designs for N = (11, 12 and 13).

11-point I-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
I-efficiency	16.528	11.132	11.335	67.352	68.581	101.824
Min I-efficiency	0.066	4.33	4.645	6560.606	7037.879	**107.275**
Leave-1-out I-efficiency	5.286	6.067	6.334	114.775	119.826	104.401
D-efficiency	40.969	28.963	29.392	70.695	71.742	101.481
Min D-efficiency	19.951	22.511	23.453	112.831	117.553	104.185
Leave-1-out D-efficiency	29.47	24.643	25.16	83.621	85.375	102.098
A-efficiency	26.339	17.477	18.364	66.354	69.722	105.075
Min A-efficiency	0.119	8.488	9.021	7132.773	7580.672	106.279
Leave-1-out A-efficiency	9.661	9.413	10.009	97.433	103.602	106.332

12-point I-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
I-efficiency	17.018	13.382	14.067	78.634	82.660	105.119
Min I-efficiency	0.045	7.5	8.173	16666.667	18162.222	**108.973**
Leave-1-out I-efficiency	7.74	9.861	10.443	127.403	134.922	105.902
Leave-2-out I-efficiency	1.727	4.761	5.107	275.680	295.715	107.267
D-efficiency	39.435	40.504	38.857	102.711	98.534	95.934
Min D-efficiency	18.571	33.023	31.966	177.820	172.129	96.799
Leave-1-out D-efficiency	30.386	36.412	35.088	119.832	115.474	96.364
Leave-2-out D-efficiency	9.413	29.527	28.06	313.683	298.098	95.032
A-efficiency	28.623	23.703	34.939	82.811	122.066	147.403
Min A-efficiency	0.078	14.962	15.498	19182.051	19869.231	103.582
Leave-1-out A-efficiency	12.659	17.118	17.432	135.224	137.704	101.834
Leave-2-out A-efficiency	2.842	8.082	8.356	284.377	294.018	103.390

13-point I-optimal						
Criteria Evaluated	EXACT	PEXCH	CEXCH	R.E.(P, E)	R.E.(C, E)	R.E.(C, P)
I-efficiency	17.083	14.683	14.793	85.951	86.595	100.749
Min I-efficiency	0.068	10.01	11.644	14720.588	17123.529	**116.324**
Leave-1-out I-efficiency	11.598	11.987	12.041	103.354	103.820	100.450
Leave-2-out I-efficiency	6.171	8.381	8.397	135.813	136.072	100.191
D-efficiency	40.272	39.079	36.987	97.038	91.843	94.647
Min D-efficiency	19.314	33.747	31.789	174.728	164.590	94.198
Leave-1-out D-efficiency	34.726	36.225	34.331	104.317	98.863	94.772
Leave-2-out D-efficiency	27.086	32.358	33.354	119.464	123.141	103.078
A-efficiency	29.669	25.099	24.59	84.597	82.881	97.972
Min A-efficiency	0.124	18.794	17.69	15156.452	14266.129	94.126
Leave-1-out A-efficiency	19.684	20.283	19.758	103.043	100.376	97.412
Leave-2-out A-efficiency	10.312	14.038	13.641	136.133	132.283	97.172

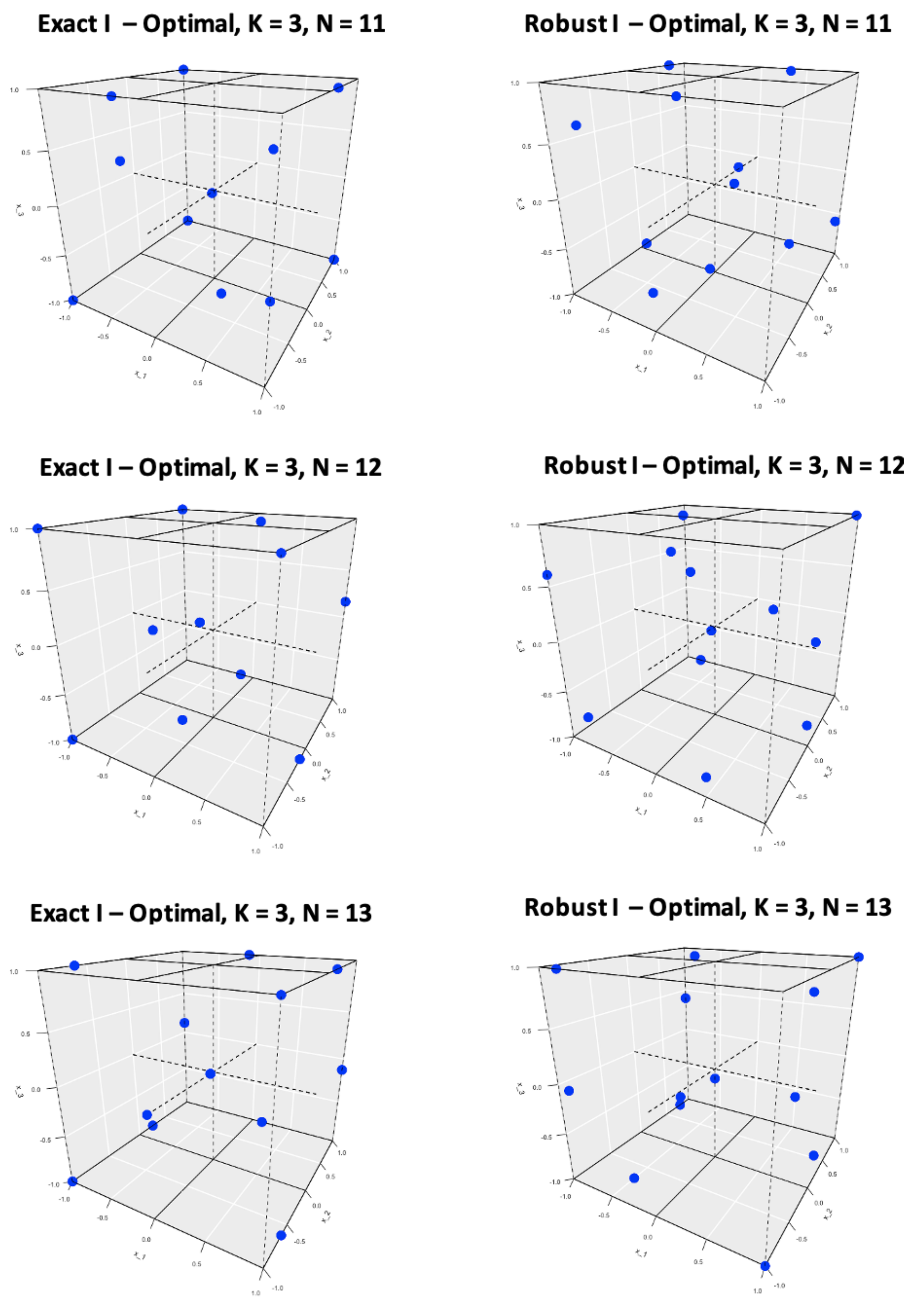

Figure 3.3: Exact and Robust I - Optimal Designs for K = 11, 12, 13

Table 3.6: Properties of 3-factor D-optimal designs for N = (14, 15 and 16).

14-point D-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
D-efficiency	46.326	46.304	99.953
Min D-efficiency	41.056	42.453	**103.403**
Leave-1-out D-efficiency	43.729	43.759	100.069
Leave-2-out D-efficiency	40.384	37.856	93.740
A-efficiency	26.349	31.056	117.864
Min A-efficiency	18.267	26.076	142.749
Leave-1-out A-efficiency	22.655	27.087	119.563
Leave-2-out A-efficiency	17.838	N/A	N/A
I-efficiency	13.401	17.143	127.923
Min I-efficiency	8.72	14.348	164.541
Leave-1-out I-efficiency	11.645	14.977	128.613
Leave-2-out I-efficiency	9.275	N/A	N/A

15-point D-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
D-efficiency	44.413	45.834	103.200
Min D-efficiency	39.27	41.922	**106.753**
Leave-1-out D-efficiency	42.436	43.707	102.995
Leave-2-out D-efficiency	39.694	39.044	98.362
A-efficiency	20.68	29.648	143.366
Min A-efficiency	13.008	24.663	189.599
Leave-1-out A-efficiency	18.453	26.444	143.305
Leave-2-out A-efficiency	15.753	22.004	139.681
I-efficiency	10.177	16.354	160.696
Min I-efficiency	5.417	13.604	251.135
Leave-1-out I-efficiency	9.185	14.096	153.468
Leave-2-out I-efficiency	7.952	12.096	152.113

16-point D-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
D-efficiency	43.859	44.884	102.337
Min D-efficiency	38.944	42.288	**108.587**
Leave-1-out D-efficiency	42.167	43.299	102.685
Leave-2-out D-efficiency	40.149	41.334	102.952
A-efficiency	20.157	23.499	116.580
Min A-efficiency	16.892	15.665	92.736
Leave-1-out A-efficiency	18.379	21.479	116.867
Leave-2-out A-efficiency	16.242	18.949	116.667
I-efficiency	9.254	11.645	125.837
Min I-efficiency	7.245	6.995	96.549
Leave-1-out I-efficiency	8.481	10.735	126.577
Leave-2-out I-efficiency	7.544	9.566	126.803

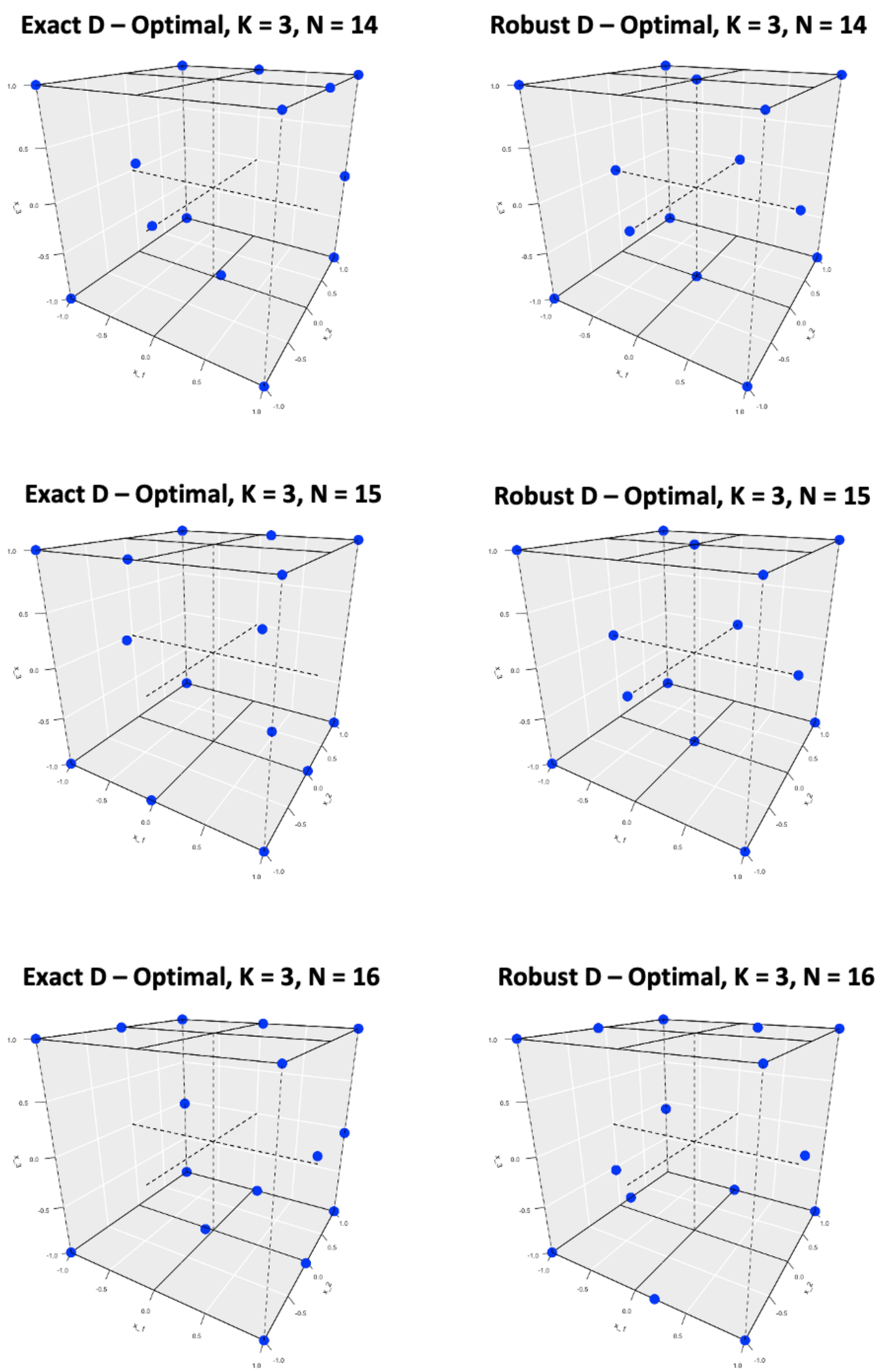

Figure 3.4: Exact and Robust A - Optimal Designs for N = 14, 15 and 16

Table 3.7: Properties of 3-factor A-optimal designs for N = (14, 15 and 16).

14-point A-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
A-efficiency	31.056	31.056	100.000
Min A-efficiency	26.076	26.076	**100.000**
Leave-1-out A-efficiency	27.087	27.087	100.000
Leave-2-out A-efficiency	N/A	N/A	N/A
D-efficiency	46.304	46.304	100.000
Min D-efficiency	42.453	42.453	100.000
Leave-1-out D-efficiency	43.759	43.759	100.000
Leave-2-out D-efficiency	37.856	37.856	100.000
I-efficiency	17.143	17.143	100.000
Min I-efficiency	14.348	14.348	100.000
Leave-1-out I-efficiency	14.977	14.977	100.000
Leave-2-out I-efficiency	N/A	N/A	N/A

15-point A-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
A-efficiency	29.572	31.291	105.813
Min A-efficiency	24.979	27.263	**109.144**
Leave-1-out A-efficiency	25.934	27.827	107.299
Leave-2-out A-efficiency	20.915	23.137	110.624
D-efficiency	43.826	44.716	102.031
Min D-efficiency	39.445	40.844	103.547
Leave-1-out D-efficiency	41.554	42.542	102.378
Leave-2-out D-efficiency	38.537	39.699	103.015
I-efficiency	17.095	18.136	106.089
Min I-efficiency	14.132	15.97	113.006
Leave-1-out I-efficiency	14.935	16.097	107.780
Leave-2-out I-efficiency	11.973	13.316	111.217

16-point A-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
A-efficiency	29.412	31.055	105.586
Min A-efficiency	23.358	27.122	**116.114**
Leave-1-out A-efficiency	26.106	28.126	107.738
Leave-2-out A-efficiency	N/A	24.425	N/A
D-efficiency	43.528	44.262	101.686
Min D-efficiency	39.273	40.219	102.409
Leave-1-out D-efficiency	41.542	42.429	102.135
Leave-2-out D-efficiency	37.783	40.179	106.341
I-efficiency	16.129	17.766	110.149
Min I-efficiency	7.245	15.431	212.988
Leave-1-out I-efficiency	8.481	16.087	189.683
Leave-2-out I-efficiency	8.807	13.966	158.578

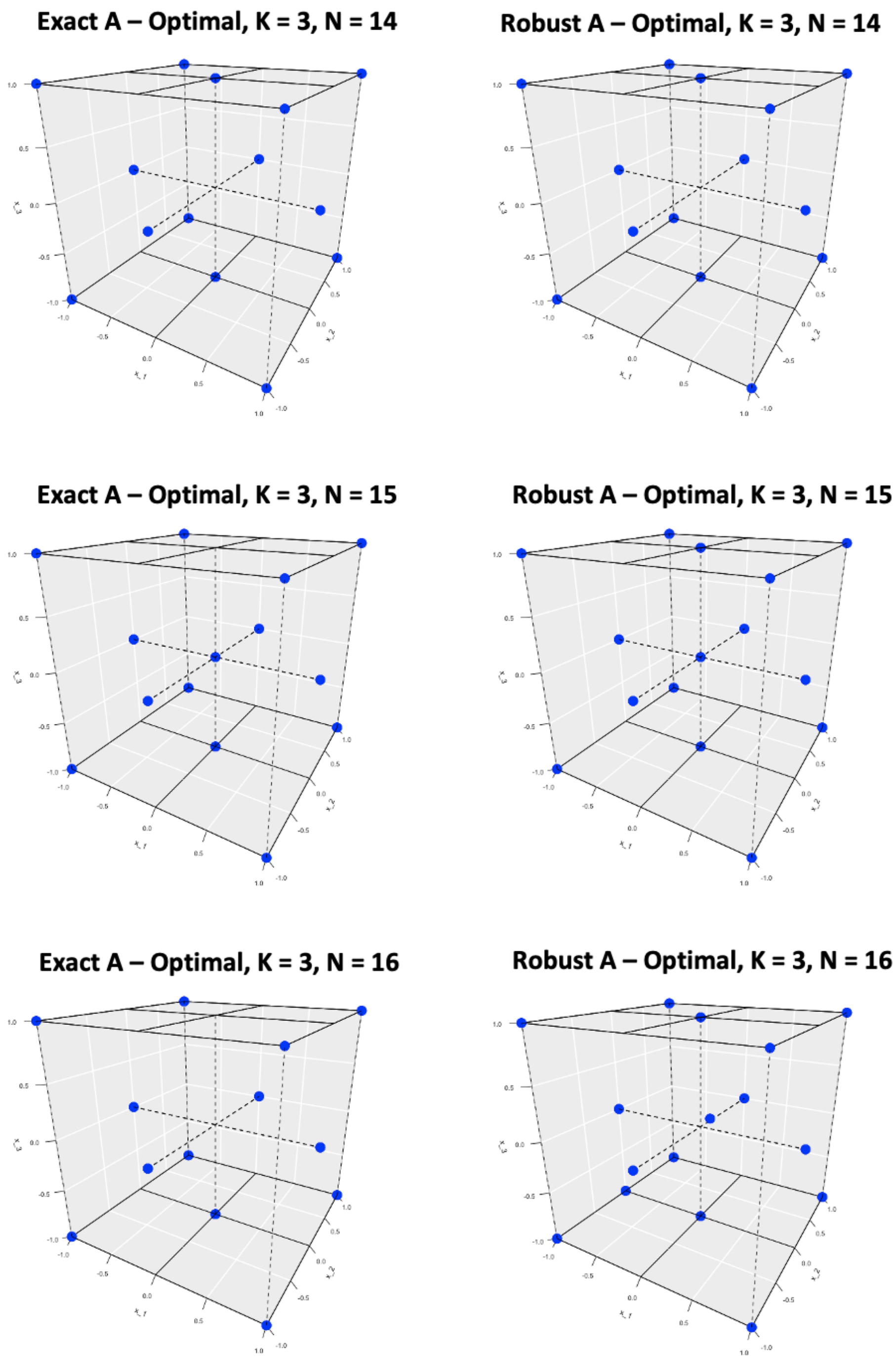

Figure 3.5: Exact and Robust A - Optimal Designs for N = 14, 15 and 16

Table 3.8: Properties of 3-factor I-optimal designs for N = (14, 15 and 16).

14-point I-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
I-efficiency	17.021	16.662	97.891
Min I-efficiency	0.245	14.083	**5748.163**
Leave-1-out I-efficiency	12.46	14.442	115.907
Leave-2-out I-efficiency	8.133	11.269	138.559
D-efficiency	38.618	43.044	111.461
Min D-efficiency	21.612	38.31	177.263
Leave-1-out D-efficiency	34.755	40.604	116.829
Leave-2-out D-efficiency	30.276	37.05	122.374
A-efficiency	29.091	28.76	98.862
Min A-efficiency	0.432	23.619	5467.361
Leave-1-out A-efficiency	21.062	24.906	118.251
Leave-2-out A-efficiency	13.597	19.574	143.958

15-point I-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
I-efficiency	17.095	18.136	106.089
Min I-efficiency	14.132	15.97	**113.006**
Leave-1-out I-efficiency	14.935	16.097	107.780
Leave-2-out I-efficiency	11.973	13.316	111.217
D-efficiency	43.826	44.716	102.031
Min D-efficiency	39.445	40.844	103.547
Leave-1-out D-efficiency	41.554	42.542	102.378
Leave-2-out D-efficiency	38.537	39.699	103.015
A-efficiency	29.572	31.291	105.813
Min A-efficiency	24.979	27.263	109.144
Leave-1-out A-efficiency	25.934	27.827	107.299
Leave-2-out A-efficiency	20.915	23.137	110.624

16-point I-optimal			
Criteria Evaluated	EXACT	CEXCH	R.E.(C, E)
I-efficiency	17.388	18.333	105.435
Min I-efficiency	14.037	15.862	**113.001**
Leave-1-out I-efficiency	15.365	16.432	106.944
Leave-2-out I-efficiency	12.723	13.959	109.715
D-efficiency	42.193	42.987	101.882
Min D-efficiency	37.807	39.12	103.473
Leave-1-out D-efficiency	40.209	41.09	102.191
Leave-2-out D-efficiency	37.671	38.698	102.726
A-efficiency	29.07	30.65	105.435
Min A-efficiency	24.594	26.961	109.624
Leave-1-out A-efficiency	25.838	27.587	106.769
Leave-2-out A-efficiency	21.536	23.573	109.459

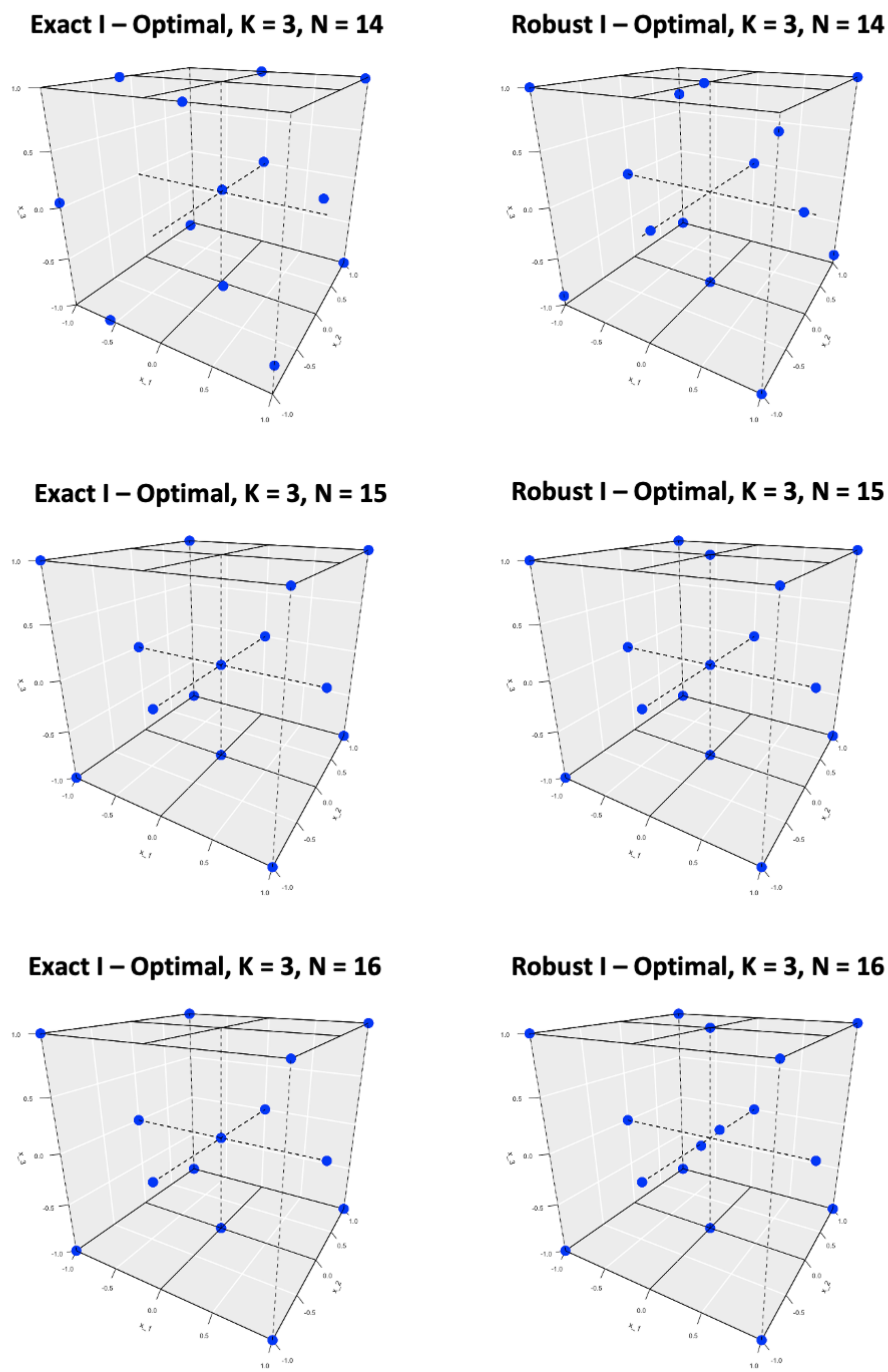

Figure 3.6: Exact and Robust I - Optimal Designs for K = 14, 15 and 16

CHAPTER 4

GENERATING ROBUST DESIGNS IN SCREENING EXPERIMENTS

In research, it is common to attempt to identify factors that will be significant in an experiment before utilizing all of the company's time and resources. One approach is to conduct a screening experiment. "The screening analysis aims to identify the few factors that drive most of the process variation, often according to a linear model comprising main effects and interaction effects" [12]. Once a screening experiment is complete, researchers can then focus on preparing an experiment that will focus in on the effects of factors that were previously proven to be significant.

In the following sections, designs will be created using both the D and A traditional alphabet criteria and robust criterion as done in the previous chapters. Screening designs are not typically created with the intention of making predictions, but rather getting the best parameter estimates. For that reason, the following sections will focus solely on D and A designs and not I. The results will determine whether linear models consisting of just main effects or main effects and interaction effects are more efficient when created using a robust criterion or if the traditional alphabet criterion will suffice. All designs will be computed using the coordinate exchange algorithm subject to the criterion being used.

In the section about two factors, first designs will be presented from just the main effects models. Directly after will be two factor models with main effects and interaction. As mentioned, oftentimes screening is done to help narrow down the number of factors presented, so generating models for a two-factor model isn't the most practical. The reason it is being done here is to first demonstrate the effect of robust designs on the simplest case possible and then to provide comparison to higher factor models. To conclude the chapter, designs will be presented for five factor models. In that section, both designs for main effects and main effects with interaction will be introduced and compared.

4.1

Two-factor Designs: Main Effects Model

In the following section, a main effects model will be used to generate robust designs. The formula for such a model can be explained by the following:

$$y = \beta_0 + \sum_{i=1}^{2} \beta_i x_i + \epsilon \tag{4.1}$$

The following two factor designs will experiment with N starting at 4 and going to 11. Four is the minimum number of observations required to create a robust design given that in the model has three parameters.

4.1.1

Two-factor Design with $N = 4$

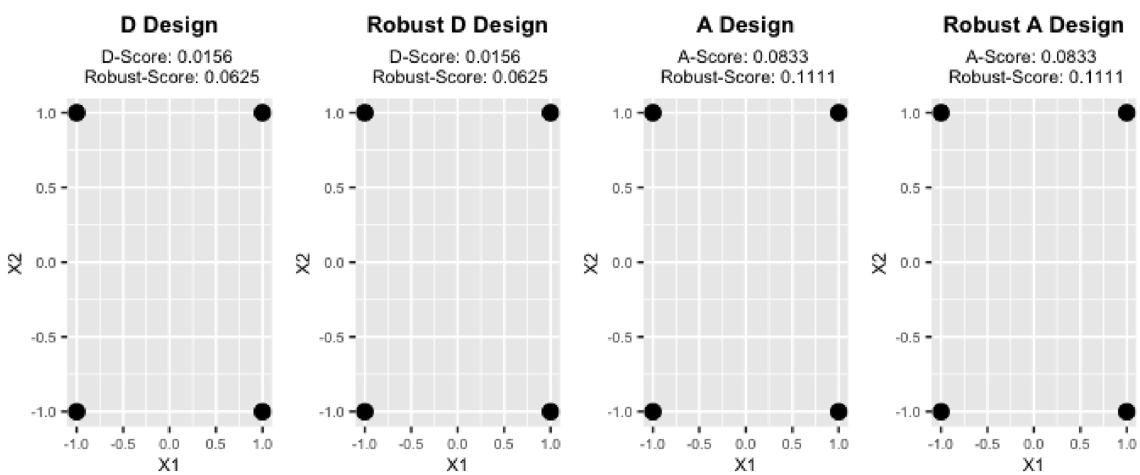

Figure 4.1: 4-point design for a main effects model with two factors.

Table 4.1: Properties of the 4-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	100	100	100	100	100	100
Min D-efficiency	83.995	83.995	83.995	83.995	**100**	100
Leave-1-out D-efficiency	83.995	83.995	83.995	83.995	100	100
A-efficiency	100	23.534	100	88.48	100	100
Min A-efficiency	66.667	66.667	66.667	66.667	100	**100**
Leave-1-out A-efficiency	66.667	66.667	66.667	66.667	100	100

Results for 4-point 2-factor designs are shown in Figure 4.1 and Table 4.1 and were generated to satisfy a first order model with only main effects. The plots and robust scores indicate that all four criteria generated the same design. Each design has its points in the corners to try and capture as much information possible about the response space.

4.1.2

Two-factor Design with $N = 5$

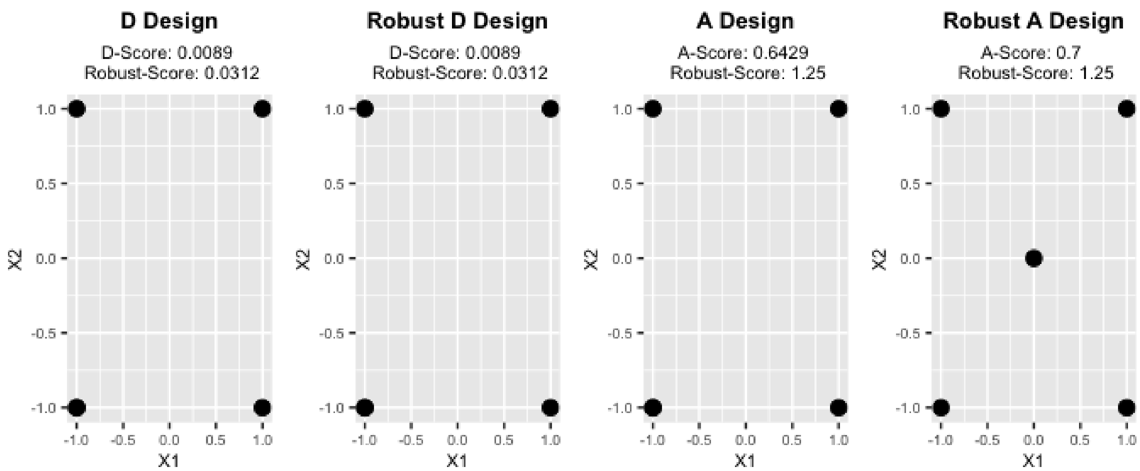

Figure 4.2: 5-point design for a main effects model with two factors.

Table 4.2: Properties of the 5-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	96.406	96.406	96.406	86.177	100	89.39
Min D-efficiency	79.37	79.37	79.37	72.112	**100**	90.855
Leave-1-out D-efficiency	87.622	87.622	87.622	77.69	100	88.665
Leave-2-out D-efficiency	58.796	58.796	58.796	54.763	100	88.665
A-efficiency	93.333	93.333	93.333	88.48	100	94.800
Min A-efficiency	60	60	60	66.667	100	**111.112**
Leave-1-out A-efficiency	76	76	76	68	100	89.474
Leave-2-out A-efficiency	46.667	46.667	46.667	40	100	85.714

Results for 5-point 2-factor designs are shown in Figure 4.2 and Table 4.2 and were generated to satisfy a first order model with only main effects. From the plots, we see that the D, Robust D, and A Designs are all the same. The D related designs both found designs with robust scores of 0.0312. The A related designs both have a robust score of 1.25, but found different designs. And though they have the same robust score when compared to the Min A-efficiency, we learn that the Robust A Design achieves a higher efficiency.

4.1.3

Two-factor Design with $N = 6$

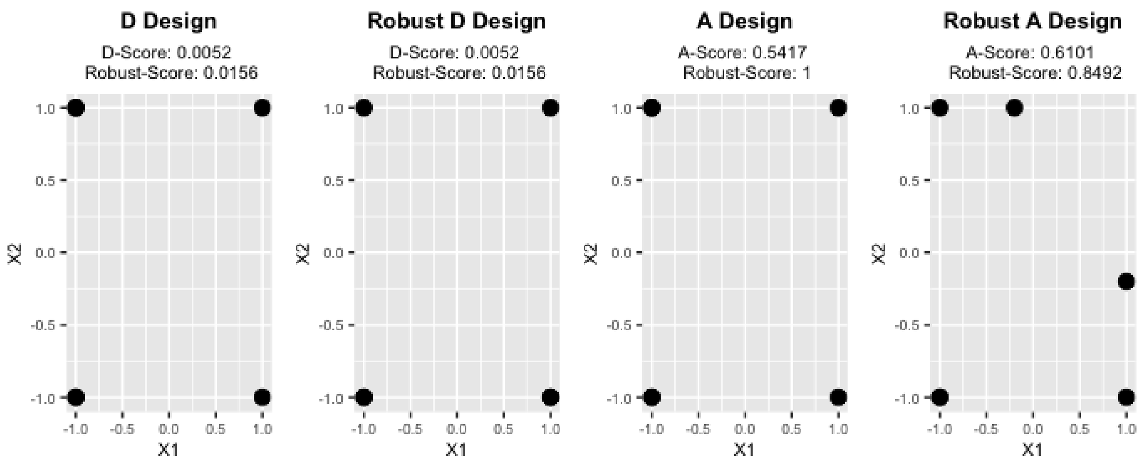

Figure 4.3: 6-point design for a main effects model with two factors.

Table 4.3: Properties of the 6-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	96.15	96.15	96.15	85..664	100	89.094
Min D-efficiency	80	80	80	78.566	**100**	98.208
Leave-1-out D-efficiency	90.937	90.937	90.937	81.536	100	89.662
Leave-2-out D-efficiency	79.58	79.58	79.58	72.642	100	91.282
A-efficiency	92.308	92.308	92.308	81.536	100	88.782
Min A-efficiency	60	60	60	81.953	100	**117.752**
Leave-1-out A-efficiency	82.222	82.222	82.222	73.741	100	89.685
Leave-2-out A-efficiency	66.667	66.667	66.667	59.891	100	89.835

Results for 6-point 2-factor designs are shown in Figure 4.3 and Table 4.3 and were generated to satisfy a first order model with only main effects. From the plots, we see that the D, Robust D, and A criteria generated the same design. The design they generated favors having all the observations in the corners of the space. The Robust A Design did not favor a symmetric corner focused design, and instead has its points in a few corners and along the center of two sides. From the table, we see that the Robust A Design outperforms the A Design from the R.E. (R.A., A) score of 117.752.

4.1.4

Two-factor Design with N = 7

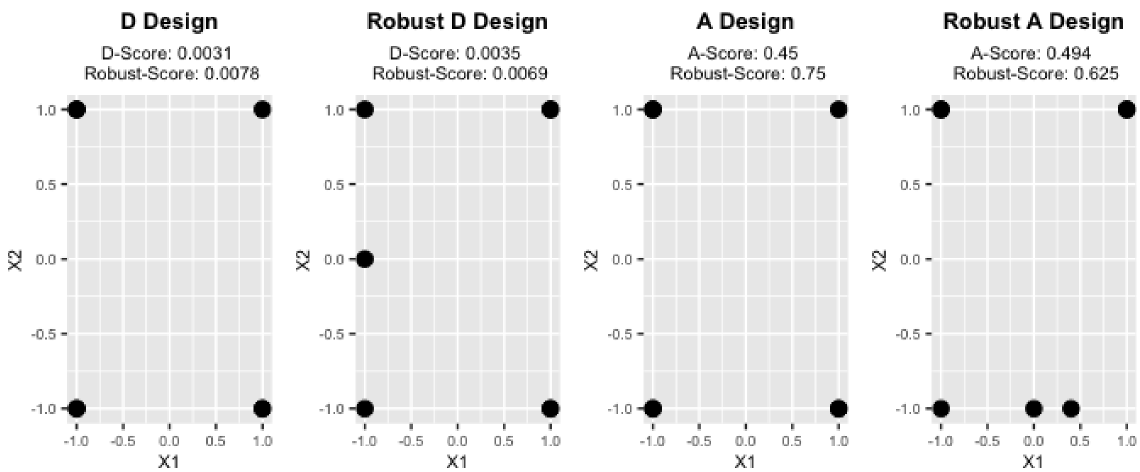

Figure 4.4: 7-point design for a main effects model with two factors.

Table 4.4: Properties of the 7-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	97.713	94.341	97.713	93.316	96.549	95.5
Min D-efficiency	83.995	87.358	83.995	86.739	**104.004**	103.267
Leave-1-out D-efficiency	94.414	91.246	94.414	90.271	96.645	95.612
Leave-2-out D-efficiency	89.375	86.43	89.375	85.291	96.705	95.654
A-efficiency	95.238	93.506	95.238	92.369	98.181	96.988
Min A-efficiency	66.667	80	66.667	80	119.999	**119.999**
Leave-1-out A-efficiency	88.645	87.108	88.645	86.08	98.266	97.106
Leave-2-out A-efficiency	79.048	77.751	79.048	76.834	98.359	97.199

Results for 7-point 2-factor designs are shown in Figure 4.4 and Table 4.4 and were generated to satisfy a first order model with only main effects. In these scenario, the D and A criterion generated the same design, having all of its points in the corners of the space. The Robust D and Robust A both found designs that have some points in the corners but also a point in the center of a side. The Robust D got a robust score of 0.0069 and the Robust A got a 0.625, both of which are smaller and thus more robust-optimal than the traditional D and A designs. From the table in the furthest two columns, we find 104.004 for R.E. (R.D., D) and 119.999 for R.E. (R.A., A), further validating the conclusion that the a designs is more efficient in producing robust design subject to one data loss.

4.1.5 Two-factor Design with $N = 8$

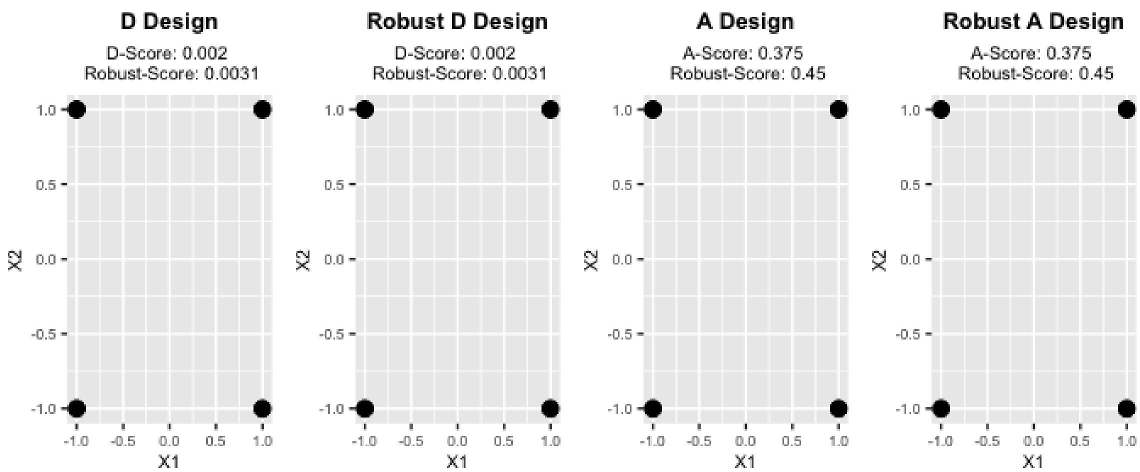

Figure 4.5: 8-point design for a main effects model with two factors.

Table 4.5: Properties of the 8-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	100	100	100	100	100	100
Min D-efficiency	97.713	97.713	97.713	97.713	**100**	100
Leave-1-out D-efficiency	97.713	97.713	97.713	97.713	100	100
Leave-2-out D-efficiency	94.414	94.414	94.414	94.414	100	100
A-efficiency	100	100	100	100	100	100
Min A-efficiency	95.238	95.238	95.238	95.238	100	**100**
Leave-1-out A-efficiency	95.238	95.238	95.238	95.238	100	100
Leave-2-out A-efficiency	88.645	88.645	88.645	88.645	100	100

Results for 8-point 2-factor designs are shown in Figure 4.5 and Table 4.5 and were generated to satisfy a first order model with only main effects. From the plots and the table, we find that all four criteria found the same design. In this case, it must be favorable to be symmetric and have all observations in the corners.

4.1.6

Two-factor Design with $N = 9$

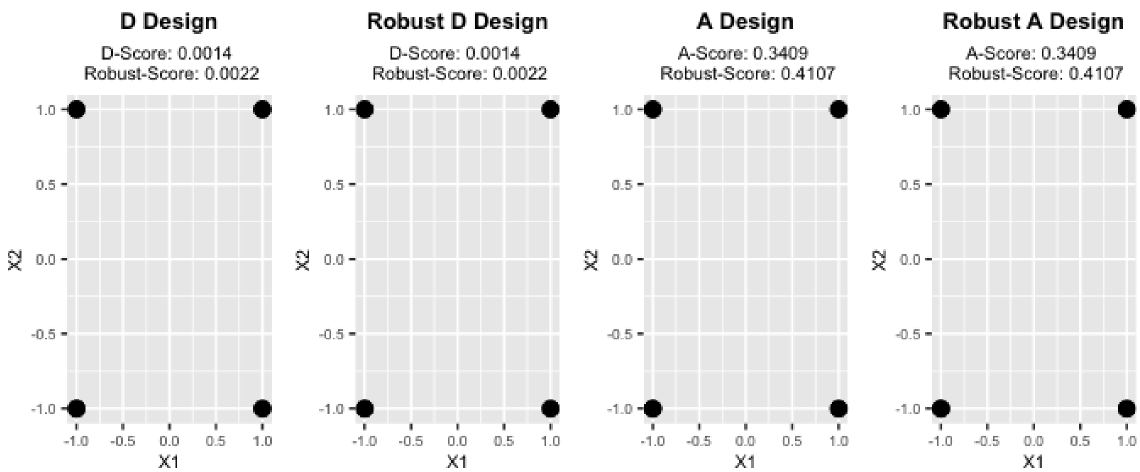

Figure 4.6: 9-point design for a main effects model with two factors.

Table 4.6: Properties of the 9-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	98.844	98.844	98.844	98.844	100	100
Min D-efficiency	95.647	95.647	95.647	95.647	**100**	100
Leave-1-out D-efficiency	97.098	97.098	97.098	97.098	100	100
Leave-2-out D-efficiency	94.72	94.72	94.72	94.72	100	100
A-efficiency	97.778	97.778	97.778	97.778	100	100
Min A-efficiency	91.304	91.304	91.304	91.304	100	**100**
Leave-1-out A-efficiency	94.203	94.203	94.203	94.203	100	100
Leave-2-out A-efficiency	89.484	89.484	89.484	89.484	100	100

Results for 9-point 2-factor designs are shown in Figure 4.6 and Table 4.6 and were generated to satisfy a first order model with only main effects. From the plots and the table, we find that all four criteria found the same design. In this case, it is again favorable to be symmetric and have all observations in the corners.

4.1.7

Two-factor Design with $N = 10$

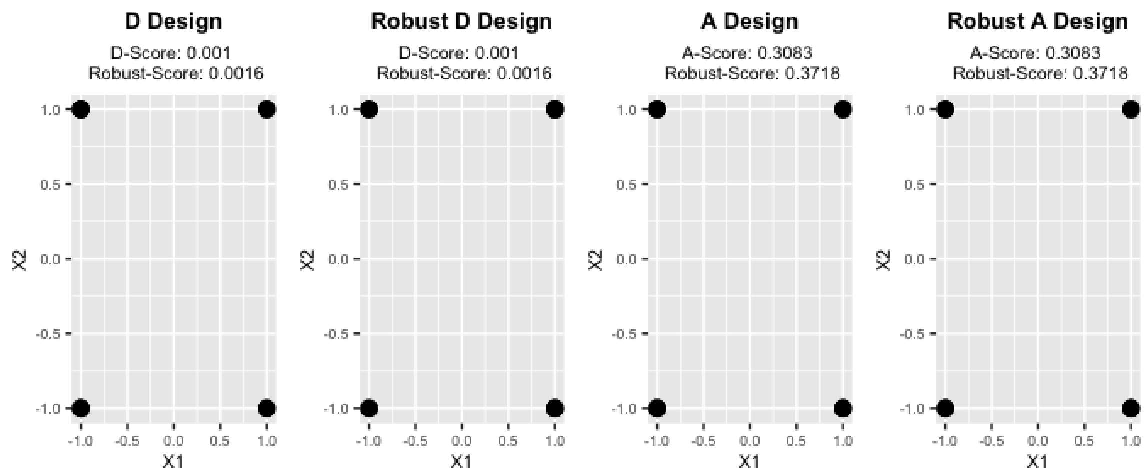

Figure 4.7: 10-point design for a main effects model with two factors.

Table 4.7: Properties of the 10-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	98.648	98.648	98.648	98.648	100	100
Min D-efficiency	94.948	94.948	94.948	94.948	**100**	100
Leave-1-out D-efficiency	97.285	97.285	97.285	97.285	100	100
Leave-2-out D-efficiency	95.509	95.509	95.509	95.509	100	100
A-efficiency	97.297	97.297	97.297	97.297	100	100
Min A-efficiency	89.655	89.655	89.655	89.655	100	**100**
Leave-1-out A-efficiency	94.529	94.529	94.529	94.529	100	100
Leave-2-out A-efficiency	90.999	90.999	90.999	90.999	100	100

Results for 10-point 2-factor designs are shown in Figure 4.7 and Table 4.7 and were generated to satisfy a first order model with only main effects. From the plots and the table, we find that all four criteria have the same design. Once again, we see that it is must be favorable to be symmetric and have all observations in the corners.

4.1.8

Two-factor Design with $N = 11$

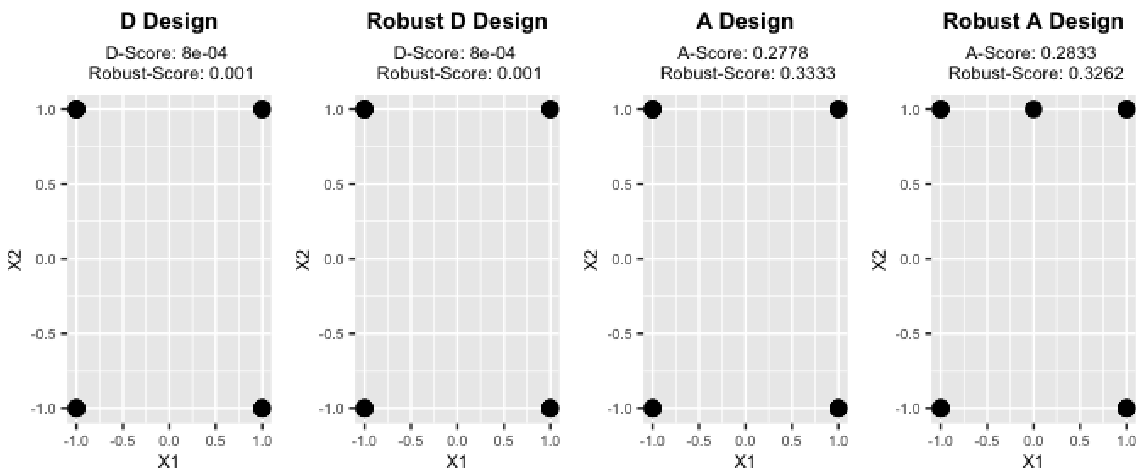

Figure 4.8: 11-point design for a main effects model with two factors.

Table 4.8: Properties of the 11-point 2-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	99.116	99.116	99.116	96.605	100	97.467
Min D-efficiency	95.244	95.244	95.244	94.354	**100**	99.066
Leave-1-out D-efficiency	98.029	98.029	98.029	95.549	100	97.47
Leave-2-out D-efficiency	96.661	96.661	96.661	94.219	100	97.474
A-efficiency	98.182	98.182	98.182	96.257	100	98.039
Min A-efficiency	90	90	90	91.971	100	**102.19**
Leave-1-out A-efficiency	95.971	95.971	95.971	94.088	100	98.038
Leave-2-out A-efficiency	93.225	93.225	93.225	91.395	100	98.037

Results for 11-point 2-factor designs are shown in Figure 4.8 and Table 4.8 and were generated to satisfy a first order model with only main effects. The plots show that D, Robust D, and A Designs are the same; a design that has all its observations in the corners. The Robust A Design is different, having moved a point from a corner to the center of one side. This change allows the Robust A Design to find a design with a Min A-efficiency of 91.971, beating the A Design's efficiency of 90.

4.2

Two-factor Designs: Main Effects with Interaction Model

In this section, two-factor designs were generated from a model that includes main effects and interaction. This time there will be four parameters, which changes the starting N point to 5. In this section, designs for $N = 5$ up to 11 will be presented and analyzed. The model for the following designs is written as:

$$y = \beta_0 + \sum_{i=1}^{2} \beta_i x_i + \sum_{i=1}^{1} \sum_{j=i+1}^{2} \beta_{ij} x_i x_j + \epsilon \quad (4.2)$$

4.2.1

Two-factor Design with $N = 5$

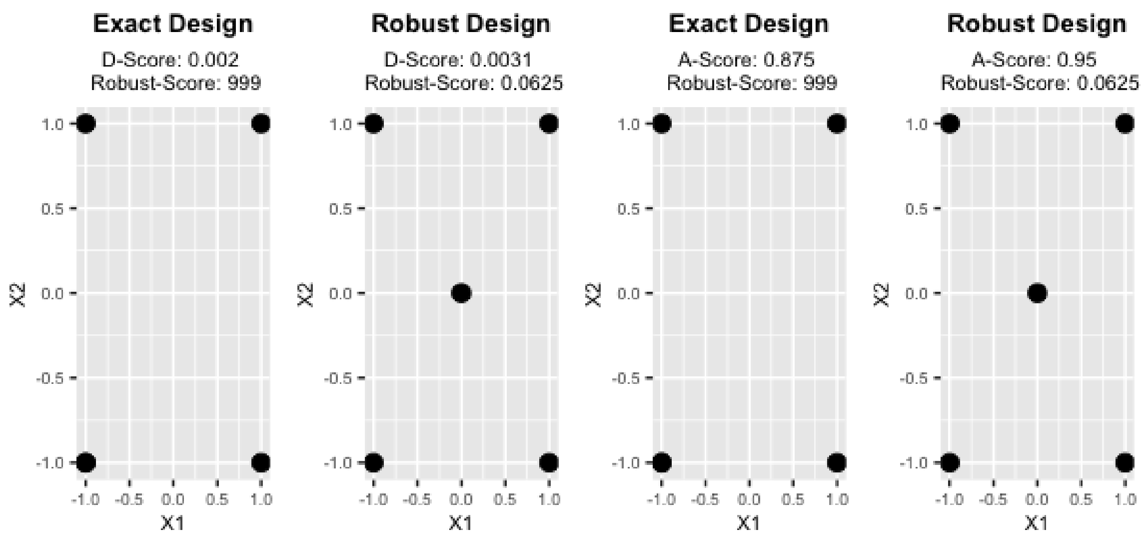

Figure 4.9: 5-point design for a 2-factor main effects with interaction model.

Table 4.9: Properties of the 5-point 2-factor optimal designs: Main Effects and Interaction

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	95.137	84.59	95.137	84.59	88.914	88.914
Min D-efficiency	0	50	0	50	**N/A**	N/A
Leave-1-out D-efficiency	40.003	60	40.003	60	149.989	149.989
A-efficiency	91.429	84.211	91.429	84.211	92.105	92.105
Min A-efficiency	0	18.182	0	18.182	N/A	**N/A**
Leave-1-out A-efficiency	40	24.545	40	24.545	86.363	86.363

Results for 5-point 2-factor designs are shown in Figure 4.9 and Table 4.9 and were generated to satisfy a first order model for main effects and interaction. From the plots, we see that the D and A designs are the same as well as the Robust D and Robust A Designs. The robust designs both moved one of their points from a corner and placed it in the center. From the table, we see that the D and A designs both reported 0 for the Min-efficiency while Robust D and Robust A got a 50 for Min D-efficiency and a 18.182 for Min A-efficiency.

4.2.2

Two-factor Design with $N = 6$

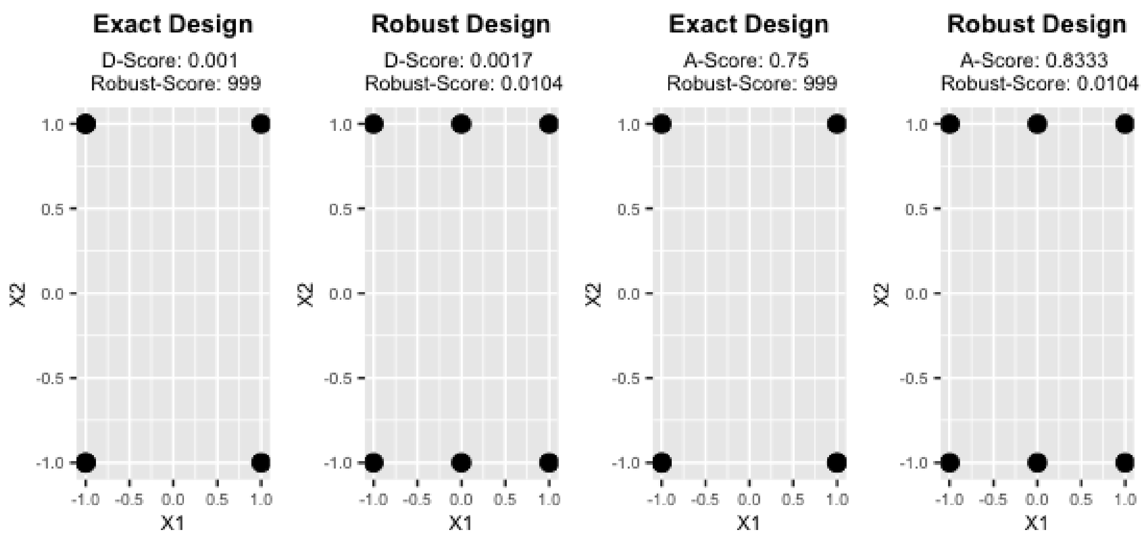

Figure 4.10: 6-point design for a 2-factor main effects with interaction model.

Table 4.10: Properties of the 6-point 2-factor optimal designs: Main Effects and Interaction

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	94.281	81.65	94.281	81.65	86.603	86.603
Min D-efficiency	0	62.603	0	62.603	**N/A**	N/A
Leave-1-out D-efficiency	63.424	71.247	63.424	71.247	112.334	112.334
Leave-2-out D-efficiency	26.671	38.856	26.67	38.856	145.686	149.692
A-efficiency	88.889	80	88.889	80	90	90
Min A-efficiency	0	41.739	0	41.739	N/A	**N/A**
Leave-1-out A-efficiency	60.952	56.917	60.952	56.917	93.38	93.38
Leave-2-out A-efficiency	26.667	28.889	26.667	28.889	108.332	108.332

Results for 6-point 2-factor designs are shown in Figure 4.10 and Table 4.10 and were generated to satisfy a first order model for main effects and interaction. In this scenario, the D and A designs both reported the same design that has all its points in the corners. The Robust D and Robust A designs are also same but instead of having all the observations in the corners, there are points in the center of opposing sides. From the table, we find that the non-robust designs are not robust to data loss because of their reported 0 in the Min-efficiency sections.

4.2.3

Two-factor Design with $N = 7$

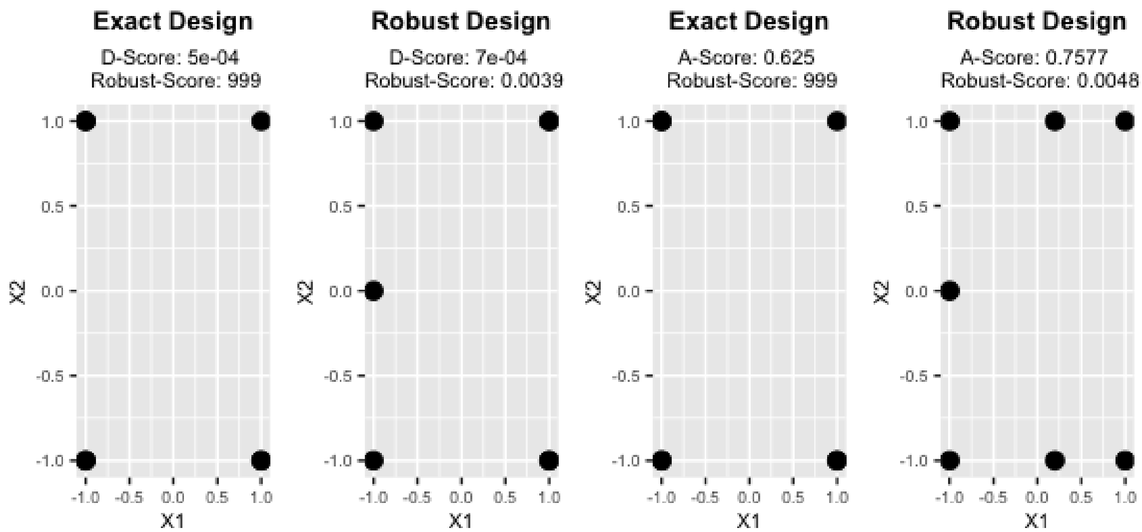

Figure 4.11: 7-point design for a 2-factor main effects with interaction model.

Table 4.11: Properties of the 7-point 2-factor optimal designs: Main Effects and Interaction

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	96.102	89.433	96.102	77.273	93.06	80.407
Min D-efficiency	0	66.667	0	63.175	**N/A**	N/A
Leave-1-out D-efficiency	80.812	82.652	80.812	71.189	102.277	88.092
Leave-2-out D-efficiency	54.364	60.612	54.364	58.153	111.493	106.97
A-efficiency	91.429	85.714	91.429	75.421	93.749	82.491
Min A-efficiency	0	38.095	0	48.184	N/A	**N/A**
Leave-1-out A-efficiency	76.19	71.703	76.19	61.877	94.111	81.214
Leave-2-out A-efficiency	52.245	50.292	52.245	44.443	96.262	85.067

Results for 7-point 2-factor designs are shown in Figure 4.11 and Table 4.11 and were generated to satisfy a first order model for main effects and interaction. From the plots, we learn that the D and A criterion both generated the same design. The Robust D design differs by having a point along one of the sides. The Robust A design has all seven of its points in the corners and in the middle of the sides, except for two points. From the efficiency table, I see that the D and A designs both have 0 for the Min-efficiency sections, which tells us that they are not robust to data loss. The Robust D design got a Min D-efficiency of 66.667 and the Robust A design got a Min A-efficiency of 48.184.

4.2.4

Two-factor Design with $N = 8$

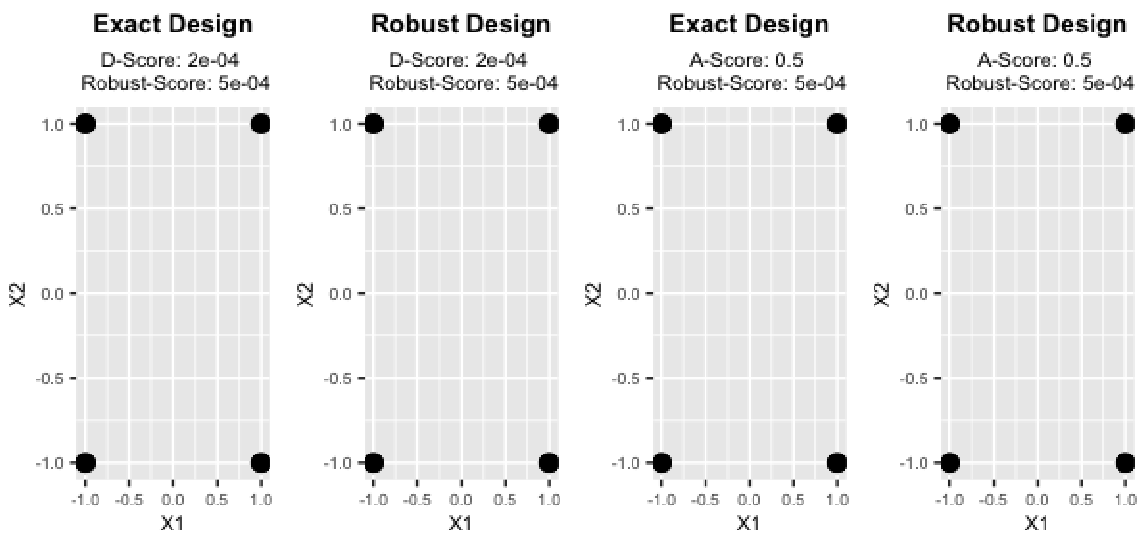

Figure 4.12: 8-point design for a 2-factor main effects with interaction model.

Table 4.12: Properties of the 8-point 2-factor optimal designs: Main Effects and Interaction

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	100	100	100	100	100	100
Min D-efficiency	96.102	96.102	96.102	96.102	**100**	100
Leave-1-out D-efficiency	96.102	96.102	96.102	96.102	100	100
Leave-2-out D-efficiency	80.812	80.812	80.812	80.812	100	100
A-efficiency	100	100	100	100	100	100
Min A-efficiency	91.429	91.429	91.429	91.429	100	**100**
Leave-1-out A-efficiency	91.429	91.429	91.429	91.429	100	100
Leave-2-out A-efficiency	76.19	76.19	76.19	76.19	100	100

Results for 8-point 2-factor designs are shown in Figure 4.12 and Table 4.12 and were generated to satisfy a first order model for main effects and interaction. From the plots and the table, we find that all four criteria have the same design. In this case, it must be favorable to be symmetric and have all observations in the corners.

4.2.5

Two-factor Design with $N = 9$

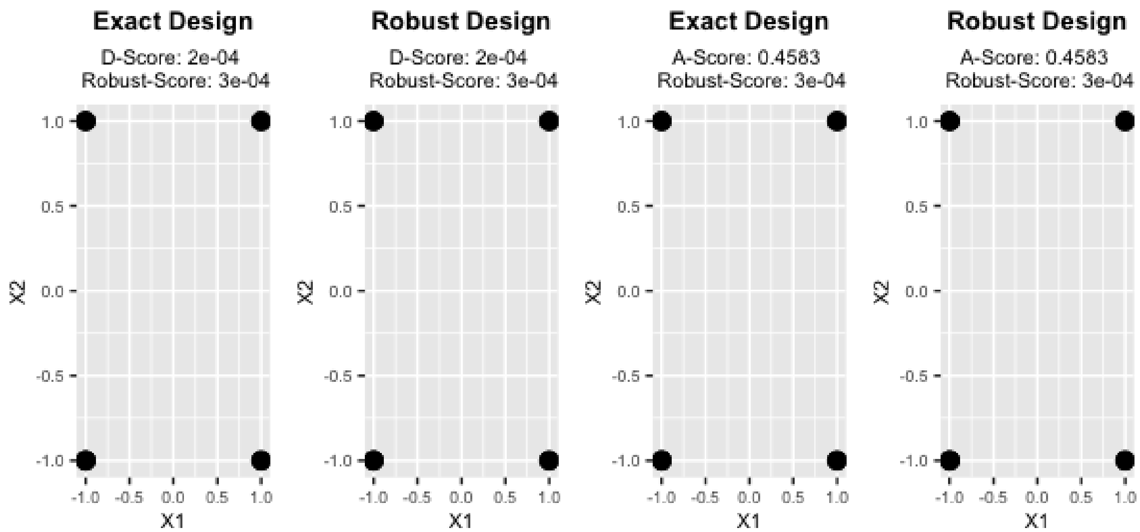

Figure 4.13: 9-point design for a 2-factor main effects with interaction model.

Table 4.13: Properties of the 9-point 2-factor optimal designs: Main Effects and Interaction

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	98.372	98.372	98.372	98.372	100	100
Min D-efficiency	93.06	93.06	93.06	93.06	**100**	100
Leave-1-out D-efficiency	95.374	95.374	95.374	95.374	100	100
Leave-2-out D-efficiency	85.871	85.871	85.871	85.871	100	100
A-efficiency	96.97	96.97	96.97	96.97	100	100
Min A-efficiency	85.714	85.714	85.714	85.714	100	**100**
Leave-1-out A-efficiency	90.476	90.476	90.476	90.476	100	100
Leave-2-out A-efficiency	80.224	80.224	80.224	80.224	100	100

Results for 9-point 2-factor designs are shown in Figure 4.13 and Table 4.13 and were generated to satisfy a first order model for main effects and interaction. From the plots and the table, we find that all four criteria have the same design. In this case, whether generated by a robust criterion or a traditional one, the same design is found.

4.2.6

Two-factor Design with $N = 10$

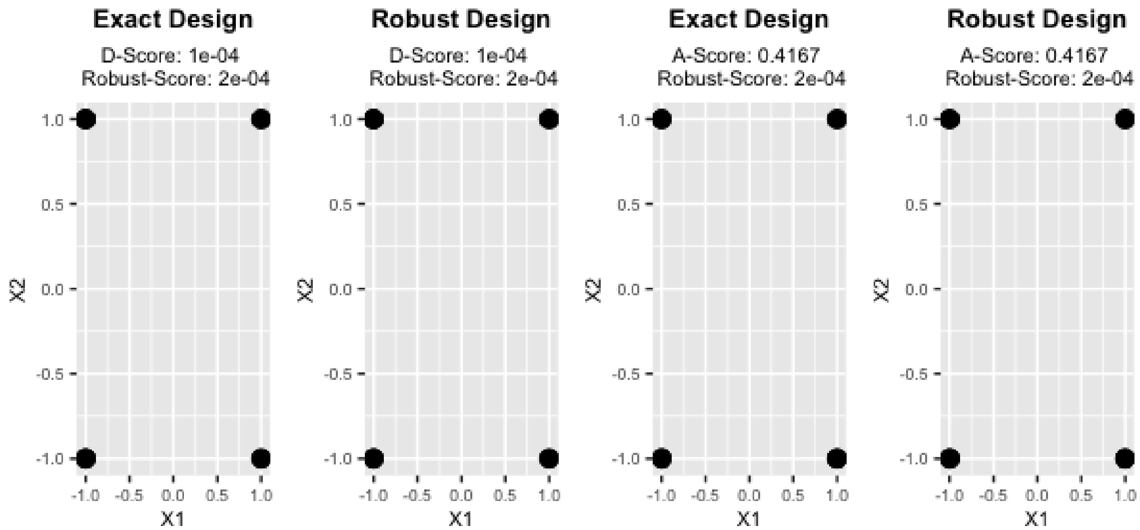

Figure 4.14: 10-point design for a 2-factor main effects with interaction model.

Table 4.14: Properties of the 10-point 2-factor optimal designs: Main Effects and Interaction

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	97.98	97.98	97.98	97.98	100	100
Min D-efficiency	91.545	91.545	91.545	91.545	**100**	100
Leave-1-out D-efficiency	95.641	95.641	95.641	95.641	100	100
Leave-2-out D-efficiency	89.738	89.738	89.738	89.738	100	100
A-efficiency	96	96	96	96	100	100
Min A-efficiency	82.051	82.051	82.051	82.051	100	**100**
Leave-1-out A-efficiency	91.002	91.002	91.002	91.002	100	100
Leave-2-out A-efficiency	83.81	83.81	83.81	83.81	100	100

Results for 10-point 2-factor designs are shown in Figure 4.14 and Table 4.14 and were generated to satisfy a first order model for main effects and interaction. From the plots and the table, we find that all four criteria have the same design. In this case, the best design is one that has all its points in the corners of the design space.

4.2.7

Two-factor Design with $N = 11$

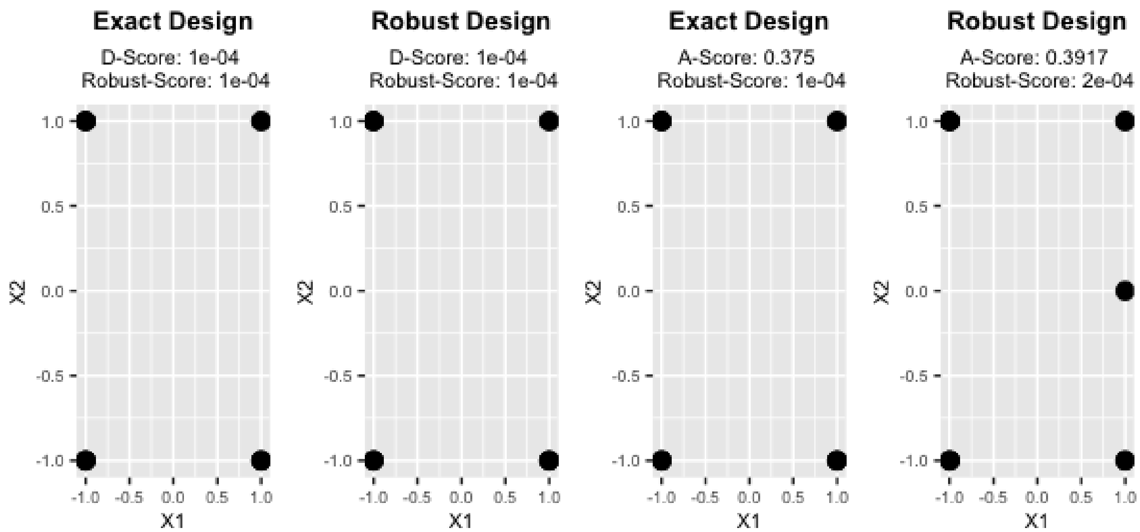

Figure 4.15: 11-point design for a 2-factor main effects with interaction model.

Table 4.15: Properties of the 11-point 2-factor optimal designs: Main Effects and Interaction

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	98.575	98.575	98.575	94.183	100	95.545
Min D-efficiency	91.18	91.18	91.18	89.218	**100**	97.848
Leave-1-out D-efficiency	96.743	96.743	96.743	92.412	100	95.523
Leave-2-out D-efficiency	93.232	93.232	93.232	89.948	100	96.478
A-efficiency	96.97	96.97	96.97	92.843	100	95.744
Min A-efficiency	80	80	80	82.5	100	**103.125**
Leave-1-out A-efficiency	93.091	93.091	93.091	89.077	100	95.688
Leave-2-out A-efficiency	87.883	87.883	87.883	84.057	100	95.646

Results for 11-point 2-factor designs are shown in Figure 4.15 and Table 4.15 and were generated to satisfy a first order model for main effects and interaction. From the plots and the table, we find that D, Robust D, and A criteria found the same design. The Robust A Design differs because it has a point in the center of one of the sides. And from the table, we find that this change is advantageous because it gives the Robust A Design a Min A-efficiency of 82.5 which is bigger than the A Design which has an efficiency of 80.

4.3

Five-factor Designs: Main Effects Model

In this section, five-factor designs are analyzed from a main effects model. In this model, there are six parameters, so the starting N value is 7 and will go up to 11. The model for the following designs is:

$$y = \beta_0 + \sum_{i=1}^{5} \beta_i x_i + \epsilon \tag{4.3}$$

For each design, a table of criterion scores and efficiencies will be presented and analyzed.

4.3.1

Five-factor Design with $N = 7$

Table 4.16: Criterion Scoring of the 7-point 5-factor optimal designs: Main Effects

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	0.0002	0.0002	1.1625	1.4081
Robust Criterion (D/A)	0.0002	0.0002	3	3.1831

Table 4.17: Properties of the 7-point 5-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	83.874	83.874	83.874	68.643	100	81.841
Min D-efficiency	66.667	66.667	66.667	53.977	**100**	80.965
Leave-1-out D-efficiency	69.142	69.142	69.142	57.394	100	83.009
A-efficiency	73.333	77.482	73.333	62.688	105.085	84.993
Min A-efficiency	33.333	34.409	33.333	32.521	103.228	**97.564**
Leave-1-out A-efficiency	39.456	41.384	39.456	35.554	104.886	90.111

Results for 7-point 5-factor designs are reported via robust scores in Table 4.16 and efficiency scores in Table 4.17. These designs were generated to satisfy a first order model for only main effects. The table shows that the D criterion and Robust D generated the same design. The Robust A design was not able to generate a design better than the A criterion as we see in the robust scores and Min A-efficiency. It is possible that if more designs were generated, then the robust criterion would eventually find the same design as the traditional alphabet criterion.

4.3.2

Five-factor Design with $N = 8$

Table 4.18: Criterion Scoring of the 8-point 5-factor optimal designs: Main Effects

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	3.815e-06	3.815e-06	0.75	0.75
Robust Criterion (D/A)	1.526e-05	1.526e-05	1.125	1.125

Table 4.19: Properties of the 8-point 5-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	100	100	100	100	100	100
Min D-efficiency	90.709	90.709	90.709	90.709	**100**	100
Leave-1-out D-efficiency	90.709	90.709	90.709	90.709	100	100
Leave-2-out D-efficiency	47.997	47.997	47.997	47.997	100	100
A-efficiency	100	100	100	100	100	100
Min A-efficiency	76.19	76.19	76.19	76.19	100	**100**
Leave-1-out A-efficiency	76.19	76.19	76.19	76.19	100	100
Leave-2-out A-efficiency	38.095	38.095	38.095	38.095	100	100

Results for 8-point 5-factor designs are reported via robust scores in Table 4.18 and efficiency scores in Table 4.19. These designs were generated to satisfy a first order model for only main effects. We see from the robust scores, that the D and Robust D design both got a score of 1.562e-05; the A and Robust A also both got the same score of 1.125. From the efficiency table we see that for both relative efficiencies, the Min relative efficiency is 100, meaning that the robust criteria did not find a better robust design.

4.3.3

Five-factor Design with $N = 9$

Table 4.20: Criterion Scoring of the 9-point 5-factor optimal designs: Main Effects

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	2.180e-06	2.466e-06	0.696	0.725
Robust Criterion (D/A)	8.719e-06	8.719e-06	1.071	1.035

Table 4.21: Properties of the 9-point 5-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	97.578	95.594	97.578	92.917	97.967	95.223
Min D-efficiency	87.129	87.129	87.129	84.682	**100**	97.192
Leave-1-out D-efficiency	90.643	89.195	90.643	86.579	98.403	95.516
Leave-2-out D-efficiency	64.733	74.484	64.733	74.251	115.063	114.703
A-efficiency	95.726	93.453	95.726	91.916	97.626	96.02
Min A-efficiency	70	71.587	70	72.466	102.267	**103.523**
Leave-1-out A-efficiency	78.333	77.726	78.333	76.131	99.225	97.189
Leave-2-out A-efficiency	52.525	55.54	52.525	54.064	105.74	102.93

Results for 9-point 5-factor designs are reported via robust scores in Table 4.20 and efficiency scores in Table 4.21. These designs were generated to satisfy a first order model for only main effects. We see from the robust scores, that the D and Robust D design both got a score of 8.719e-06. The A and Robust A Designs are different and, in this case, the Robust A Design is better because it has a robust score of 1.035 which is smaller than the A Design with a score of 1.071. The efficiency table also supports this conclusion because the R.E. (R.A., A) is 103.523, showing that the robust design is more robust to data loss.

4.3.4

Five-factor Design with $N = 10$

Table 4.22: Criterion Scoring of the 10-point 5-factor optimal designs: Main Effects

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	1.272e-06	1.356e-06	0.656	0.656
Robust Criterion (D/A)	3.815e-06	3.391e-06	0.819	0.819

Table 4.23: Properties of the 10-point 5-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	96.075	95.047	95.047	95.047	98.93	100
Min D-efficiency	88.889	90.651	90.651	90.651	**101.982**	100
Leave-1-out D-efficiency	91.375	90.651	90.651	90.651	99.208	100
Leave-2-out D-efficiency	77.153	76.98	76.98	76.98	99.776	100
A-efficiency	92.903	91.525	91.525	91.525	98.517	100
Min A-efficiency	76.19	81.356	81.356	81.356	106.78	**100**
Leave-1-out A-efficiency	81.771	81.356	81.356	81.356	99.482	100
Leave-2-out A-efficiency	64.528	65.46	65.46	65.46	101.444	100

Results for 10-point 5-factor designs are reported via robust scores in Table 4.22 and efficiency scores in Table 4.23. These designs were generated to satisfy a first order model for only main effects.

The table shows that the D and Robust D designs have different robust score, 3.815e-06 and 3.391e-06. The Robust A and Alphabet A have the same robust score. In this case, the Robust D generated a different design that achieves a better Min D-efficiency score. This tells me that the Robust-D design is more robust to one missing observation. The Robust A and Alphabet A designs achieved all the same efficiency scores further stating that they have the same design.

4.3.5

Five-factor Design with $N = 11$

Table 4.24: Criterion Scoring of the 11-point 5-factor optimal designs: Main Effects

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	6.698e-07	6.698e-07	0.583	0.583
Robust Criterion (D/A)	1.507e-06	1.507e-06	0.708	0.708

Table 4.25: Properties of the 11-point 5-factor optimal designs: Main Effects

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	97.189	97.189	97.189	97.189	100	100
Min D-efficiency	93.393	93.393	93.393	93.393	**100**	100
Leave-1-out D-efficiency	93.729	93.729	93.729	93.729	100	100
Leave-2-out D-efficiency	87.885	87.885	87.885	87.885	100	100
A-efficiency	93.506	93.506	93.506	93.506	100	100
Min A-efficiency	84.706	84.706	84.706	84.706	100	**100**
Leave-1-out A-efficiency	85.668	85.668	85.668	85.668	100	100
Leave-2-out A-efficiency	75.046	75.046	75.046	75.046	100	100

Results for 11-point 5-factor designs are reported via robust scores in Table 4.24 and efficiency scores in Table 4.25. These designs were generated to satisfy a five-factor model with only main effects. From the robust scores, I see that each set of criteria found a design with the same score. The D and Robust D have a score of 1.507e-06 and the A and Robust A reported 0.708. The efficiency table shows that all the relative efficiencies are 100. This tells me that all four criteria generated the same design.

4.4

Five-factor Designs: Main Effects with Interaction Model

In this section, five-factor designs from a model with main effects and interactions will presented. Since this model is including all combinations of factors, there will be a lot more parameters compared to the two-factor designs. In this case, there are sixteen parameters, so the starting N

will be seventeen, and in this section, designs for $N = 17$, 18, 19, and 20 will be analyzed. Similar to the last section, because there are five factors, there isn't a possible way to illustrate the design. For that reason each section will have a table with the design criterion scores and the efficiency table.

$$y = \beta_0 + \sum_{i=1}^{5} \beta_i x_i + \sum_{i=1}^{4} \sum_{j=i+1}^{5} \beta_{ij} x_i x_j + \epsilon \quad (4.4)$$

4.4.1

Five-factor Design with $N = 17$

Table 4.26: Criterion Scoring of the 17-point 5-factor optimal designs: Main Effects and Interactions

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	2.711e-20	3.392e-20	1.352	3.3553
Robust Criterion (D/A)	3.469e-18	4.493e-18	6.939	7.928

Table 4.27: Properties of the 17-point 5-factor optimal designs: Main Effects and Interactions

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	98.285	96.915	80.914	38.835	95.723	47.995
Min D-efficiency	77.111	75.875	66.01	33.051	**95.796**	50.07
Leave-1-out D-efficiency	82.463	81.505	69.722	24.4	96.299	49.339
A-efficiency	97.154	95.751	69.636	28.05	94.572	40.281
Min A-efficiency	11.268	10.904	14.412	12.614	99.583	**87.524**
Leave-1-out A-efficiency	32.52	32.481	28.371	14.525	98.462	51.197

Results for 17-point 5-factor designs are reported via robust scores in Table 4.26 and efficiency scores in Table 4.27. These designs were generated to satisfy a five-factor model with main effects and interactions. The table shows that for both D and A, the traditional alphabet achieved better scores. I hypothesize that if more runs were completed, the robust criterion could find the same designs, but for the few thousand coordinate exchange runs completed, they did not. In this case, the alphabet designs outperformed the robust ones in all efficiencies.

4.4.2

Five-factor Design with $N = 18$

Table 4.28: Criterion Scoring of the 18-point 5-factor optimal designs: Main Effects and Interactions

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	1.355e-20	1.355e-20	0.938	1.549
Robust Criterion (D/A)	8.674e-19	8.674e-19	4.844	2.412

Table 4.29: Properties of the 18-point 5-factor optimal designs: Main Effects and Interactions

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	96.934	96.934	96.934	63.734	100	65.75
Min D-efficiency	79.143	79.143	79.143	58.215	**100**	73.557
Leave-1-out D-efficiency	86.287	86.287	86.287	58.788	100	68.131
Leave-2-out D-efficiency	58.01	58.01	57.976	50.549	100	87.19
A-efficiency	94.815	94.815	94.815	57.383	100	60.521
Min A-efficiency	19.431	19.431	19.431	39.017	100	**200.798**
Leave-1-out A-efficiency	50.838	50.838	50.838	40.404	100	79.476
Leave-2-out A-efficiency	18.974	18.974	18.974	19.33	100	101.876

Results for 18-point 5-factor designs are reported via robust scores in Table 4.28 and efficiency scores in Table 4.29. These designs were generated to satisfy a five-factor model with main effects and interactions. The table shows that the D designs and Robust D both found designs with the same robust score. As for the A type designs, Robust A reported a better robust score. The D and Robust D must have found the same design because they also share all the same efficiencies. The reported R.E. (R.A., A) is 200.798, which means the Robust A Design is a better design in the face of data loss than the traditional A Design.

4.4.3

Five-factor Design with $N = 19$

Table 4.30: Criterion Scoring of the 19-point 5-factor optimal designs: Main Effects and Interactions

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	6.776e-21	1.237e-20	0.906	1.283
Robust Criterion (D/A)	2.891e-19	1.992e-19	3.479	1.810

Table 4.31: Properties of the 19-point 5-factor optimal designs: Main Effects and Interactions

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	95.898	92.357	95.898	70.89	96.308	73.922
Min D-efficiency	80.059	81.945	80.059	65.409	**102.356**	81.701
Leave-1-out D-efficiency	87.954	85.211	87.954	66.598	96.881	75.719
Leave-2-out D-efficiency	74.19	72.31	74.19	60.834	97.455	81.757
A-efficiency	92.922	87.555	92.922	65.635	94.224	70.635
Min A-efficiency	25.549	46.564	25.549	49.209	182.254	**192.215**
Leave-1-out A-efficiency	61.335	60.112	61.335	51.615	98.006	84.153
Leave-2-out A-efficiency	34.867	35.221	34.867	34.42	101.015	98.718

Results for 19-point 5-factor designs are reported via robust scores in Table 4.30 and efficiency scores in Table 4.31. The table shows that in both cases, the robust designs have better robust scores than the traditional alphabet designs. The robust scores indicate that the Robust D and Robust A Designs are more robust than the D and A because their scores are smaller. Both relative efficiency scores shown in the last two columns are greater than 100, the values being 102.356 and 192.215. This means that the generated robust designs are more robust to data loss than the traditional designs.

4.4.4

Five-factor Design with $N = 20$

Table 4.32: Criterion Scoring of the 20-point 5-factor optimal designs: Main Effects and Interactions

Criteria	D	Robust D	A	Robust A
Alphabet Criterion (D/A)	3.388e-21	3.388e-21	0.875	1.023
Robust Criterion (D/A)	3.614e-20	3.614e-20	1.448	1.438

Table 4.33: Properties of the 20-point 5-factor optimal designs: Main Effects and Interactions

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	95.137	95.137	95.137	84.378	100	88.681
Min D-efficiency	86.372	86.372	86.372	77.951	**100**	90.25
Leave-1-out D-efficiency	89.216	89.216	89.216	79.937	100	89.599
Leave-2-out D-efficiency	77.995	77.995	77.995	74.412	100	94.929
A-efficiency	91.429	91.429	91.429	78.203	100	85.534
Min A-efficiency	58.16	58.16	58.16	58.561	100	**100.689**
Leave-1-out A-efficiency	69.265	69.265	69.265	63.756	100	92.059
Leave-2-out A-efficiency	47.722	47.722	47.722	47.775	100	100.111

Results for 10-point 5-factor designs are reported via robust scores in Table 4.32 and efficiency scores in Table 4.33. The table shows that the D designs and Robust D both found designs with the same robust score. The Robust A criteria generated a design with a robust score of 1.438 which is smaller than the A Design's robust score of 1.448. In the efficiency table, we see that the D and Robust D Designs both have the same efficiencies; this means they are the same design. The Robust A Design got a Min A-efficiency of 58.561 which is better than the Min A-efficiency of the A Design of 58.16.

4.5

Screening Result Summary

Table 4.34: Robust D Designs

Design Settings	Better	Equal	Worse
2-factor: Main effect (8 design scenarios) N = 4, . . . , 11	1	7	0
2-factor: Main effects with Interaction (7 design scenarios) N = 5, . . . , 11	3	4	0
5-factors: Main effects (5 design scenarios) N = 7, . . . , 11	1	4	0
5-factors: Main effects with Interactions (4 design scenarios) N = 17, . . . , 20	1	2	1

Table 4.35: Robust A Designs

Design Settings	Better	Equal	Worse
2-factor: Main effect (8 design scenarios) N = 4, . . . , 11	4	4	0
2-factor: Main effects with Interaction (7 design scenarios) N = 5, . . . , 11	4	3	0
5-factors: Main effects (5 design scenarios) N = 7, . . . , 11	1	3	1
5-factors: Main effects with Interactions (4 design scenarios) N = 17, . . . , 20	3	0	1

Tables 4.34 and 4.35 explain how well screening designs generated using a robust optimal criterion compared to those created using an EXACT approach. Of the forty-eight combined scenarios of D and A, using a robust criterion proved helpful in finding the more robust design in eighteen cases. In twenty-seven of them, equal robust designs were found, and in three scenarios the EXACT approach found a more robust design. In cases where the Robust Design computed a score less than the traditional, it is possible that given more runs, the CEXCH algorithm would find the same design. It would appear, depending on the researcher and probability of data loss, choosing a Robust Optimal Design may prove insightful in certain screening design cases.

CHAPTER 5
INTRODUCTION TO CREATING ROBUST DESIGNS USING A ROBUST DROP TWO CRITERION.

In chapters two and three, the robust designs generated were to be optimal if one given observation was removed. In this chapter, several designs were created to be robust to two missing observations. The following examples belong to two-factor models with N going from 8 to 10. In each section, plots and criterion scores will be presented followed by an analysis of the information.

5.0.1

2-factor Design with $N = 8$

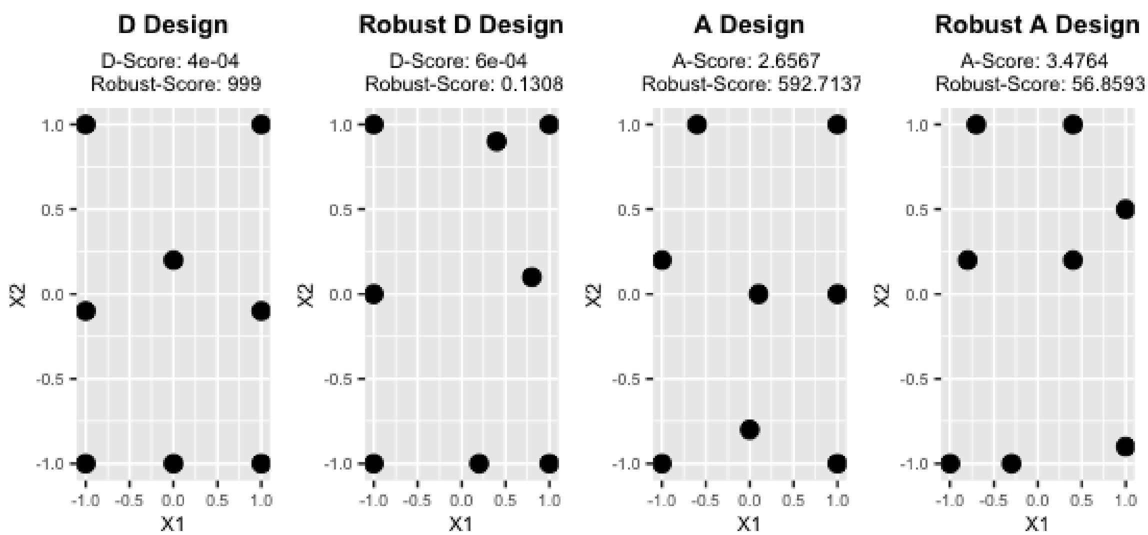

Figure 5.1: 8-point design for a second order model.

Table 5.1: Properties of the 8-point 2-factor optimal designs

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	45.612	43.094	42.265	36.669	94.48	86.76
Min D-efficiency	33.418	34.114	32.543	26.706	102.083	82.064
Min D-efficiency: Drop 2	0	23.392	12.515	17.017	**Inf**	135.973
Leave-1-out D-efficiency	40.346	38.5	37.442	32.728	95.425	87.41
Leave-2-out D-efficiency	30.978	30.779	29.296	26.179	99.358	89.36
A-efficiency	28.898	22.857	28.23	21.574	79.095	76.422
Min A-efficiency	14.898	14.648	14.932	11.561	98.322	77.424
Min A-efficiency: Drop 2	0	0.738	0.169	1.759	Inf	**1040.828**
Leave-1-out A-efficiency	21.083	17.04	20.564	16.111	80.823	78.346
Leave-2-out A-efficiency	11.253	9.843	10.485	8.778	87.47	83.72

Results for 8-point 2-factor designs robust to two data loss are shown in Figure 5.1 and Table 5.1. From the figure, I can see that each design is different in point arrangement. The D Design is the only one that has a clear symmetric design. The Robust D Design has its points closer to the edges of the space and has a close resemblance to the CEXCH 8-point drop-1-design discussed in chapter 2 (Figure 2.5). The A and Robust A Designs differ in designs, indicating that a robust to two data loss structure has a strong effect on the design. From the reported robust scores and the Min Efficiencies, the robust designs did much better. The Robust D Design got an efficiency of 23.392 and the D Design got 0, informing us that its design is not suitable for the loss of two data points. The Robust A Design has a Min A-efficiency of 1.759 while the A Design has a 0.169. In both scenarios, the Robust generated designs are better than the traditional.

5.0.2

2-factor Design with $N = 9$

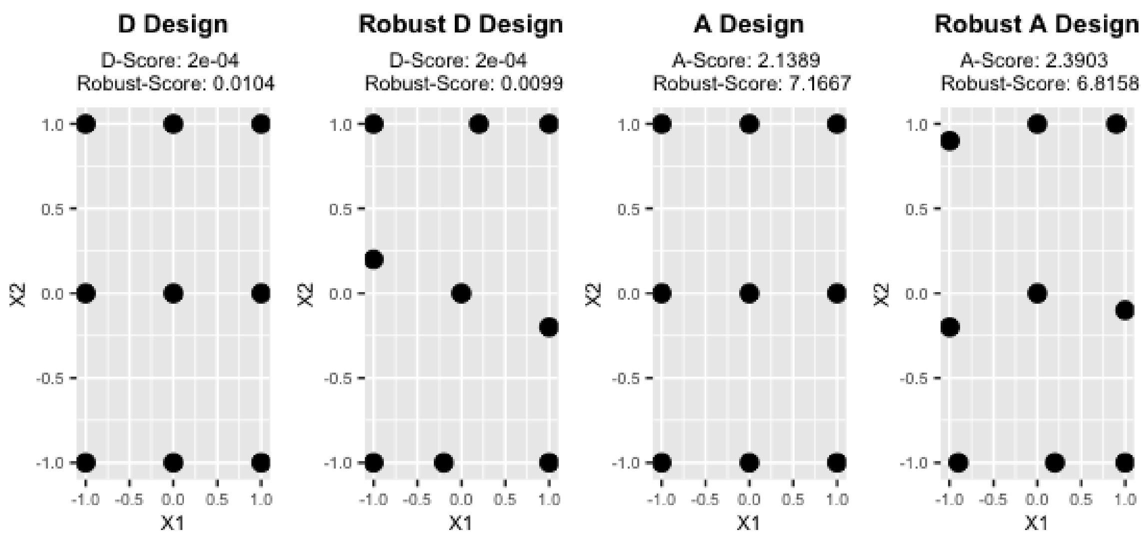

Figure 5.2: 9-point design for a second order model.

Table 5.2: Properties of the 9-point 2-factor optimal designs

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	46.224	45.602	46.224	42.736	98.654	92.454
Min D-efficiency	39.581	39.294	39.581	36.11	99.275	91.231
Min D-efficiency: Drop 2	30.569	30.821	30.569	29.731	**100.824**	97.259
Leave-1-out D-efficiency	42.829	42.306	42.829	39.683	98.779	92.655
Leave-2-out D-efficiency	38.165	37.669	38.165	35.428	98.7	92.829
A-efficiency	31.169	29.815	31.169	27.89	95.656	89.48
Min A-efficiency	22.5	20.794	22.5	17.978	92.418	79.902
Min A-efficiency: Drop 2	11.96	12.292	11.96	12.576	102.776	**105.151**
Leave-1-out A-efficiency	25.968	24.949	25.968	23.405	96.076	90.13
Leave-2-out A-efficiency	19.269	18.473	19.269	17.476	95.869	90.695

Results for 9-point 2-factor designs robust to two data loss are shown in Figure 5.2 and Table 5.2. Looking at the plots, the D and A Designs found the same design, each having points in the corners, in the center, and in the center of each edge. The Robust D and A Designs are similar to the D and A except for the position of the points along the edges. The Robust D Design has taken its points along the edges and moved them all counterclockwise a little. From the table, I find that the robust designs are more efficient in the Min Efficiency: Drop 2 field. This conclusion comes from the 100.824 in the R.E. (R.D., D) column and the 105.151 in the R.E. (R.A., A) column.

5.0.3

2-factor Design with $N = 10$

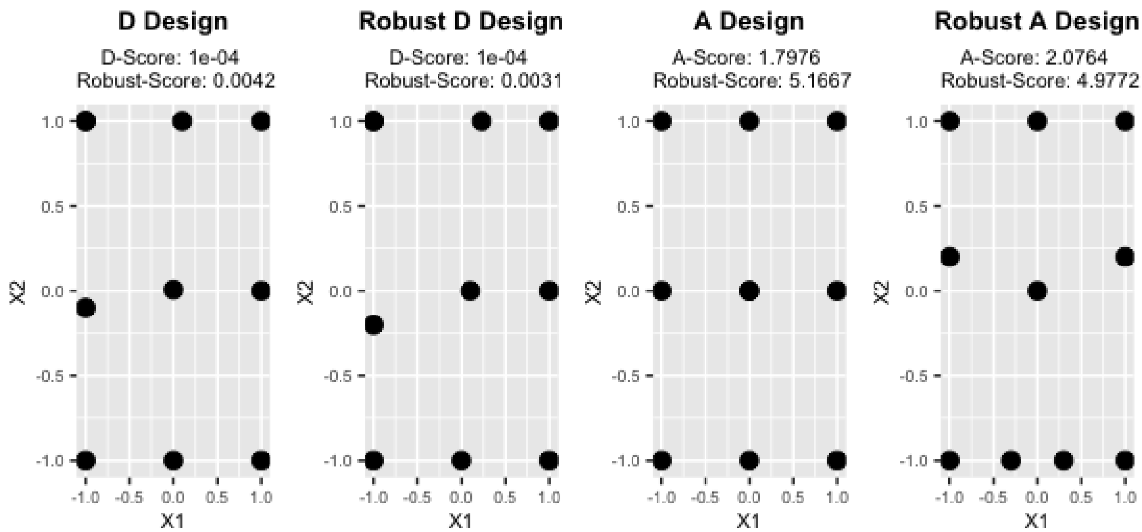

Figure 5.3: 10-point design for a second order model.

Table 5.3: Properties of the 10-point 2-factor optimal designs

Criteria Evaluated	D	Robust D	A	Robust A	R.E.(R.D., D)	R.E.(R.A., A)
D-efficiency	45.983	45.801	44.781	44.464	99.604	99.292
Min D-efficiency	39.299	38.987	38.125	38.78	99.206	101.718
Min D-efficiency: Drop 2	31.084	32.815	30.023	32.114	**105.569**	106.965
Leave-1-out D-efficiency	43.468	43.355	42.082	41.994	99.74	99.791
Leave-2-out D-efficiency	40.075	40.006	38.514	38.733	99.828	100.569
A-efficiency	29.321	28.594	33.377	28.896	97.521	86.575
Min A-efficiency	20.696	19.736	25.564	20.449	95.361	79.991
Min A-efficiency: Drop 2	12.193	13.848	14.516	15.069	113.573	**103.81**
Leave-1-out A-efficiency	25.739	25.204	28.767	25.304	97.921	87.962
Leave-2-out A-efficiency	21.081	20.704	22.972	20.763	98.212	90.384

Results for 10-point 2-factor designs robust to two data loss are shown in Figure 5.3 and Table 5.3. From the plots, I see a similar pattern of having the points spread out in the space with a point in the center for each design. The A and Robust A design both have symmetric structures down the center. The efficiency table indicates that in both cases, the robust designs reported better Min Efficiency: Drop 2. The D Design got a 31.084 and the Robust D Design achieved a 32.815. The A Design got a 14.516 and the Robust A got a 15.069. In this scenario, choosing a robust design specifically designed to be robust to two data loss proved better than the traditional approach. However, it each case, if the design isn't going to lose two points, it may prove advantageous to go with the traditional design. It would be important for the researcher to understand the experiment well along with the pros and cons in choosing one of these robust designs.

5.1

Chapter Summary

Table 5.4: Designs Robust to Loss of 2 Data Points Summary

Design Settings	Robust D Criteria	Robust A Criteria
2-factor: N = 8	Better	Better
2-factor: N = 9	Better	Better
2-factor: N = 10	Better	Better

Table 5.4 is a summary of the designs observed in the chapter. Results show that using a modified robust criterion is helpful in finding designs that are robust to loss of 2 observations compared to traditional EXACT models. In application, it would be best to ensure that an experiment doesn't lose any observations, but depending on the experiment, it may be necessary to accept that managing all factors will be difficult and designs robust to loss of more than one observation is necessary.

CHAPTER 6
CONCLUSION

In this paper, principles of optimal robust design were used to explore efficiency in algorithm selection. The algorithm of particular interest is the coordinate exchange algorithm (CEXCH). Unlike other algorithms, CEXCH can generate optimal designs without relying on a discrete candidate list of design points. Its ability to start with a randomly generated design allows for testing of more coordinate points in search for the best optimal design.

Chapter one introduced robust criteria by using the D, A, and I criterion from the traditional alphabet criterion. These criteria were designed to be functional even with data loss. In chapter two, variations of criteria efficiency were presented, offering varying perspectives on design effectiveness and robustness. With the understanding of algorithms and robust criteria, the second half of the chapter focused on validating results against previously published research.

To validate the functionality of the coordinate exchange algorithm in computing optimal robust designs, comparisons were made with the works of John J. Borkowski and Patchanok Srisuradetchai. Borkowski's EXACT designs were not optimized for robustness, while Srisuradetchai's PEXCH designs utilized robust criteria. The validation process involved presenting and analyzing plots for each method—EXACT, PEXCH, and CEXCH. From the results produced in chapter two, evidence showed that the CEXCH could create designs equal to or better in comparison to other designs.

In chapter three, validation was extended to three factors, represented in three-dimensional plots. The scenarios ran showed that the CEXCH could generate robust designs equal to or better even with in the change of factors and observations. With this evidence supporting the use of the CEXCH algorithm in searching for optimal robust deigns, the focus was changed to generating designs in a screening setting.

Chapter four was specific to screening designs and began by presenting designs for a first-order model with main effects for two factors. In this chapter, designs were computed using the traditional alphabet criteria and the robust criteria and the sequentially scored and compared. Results found that robust designs were equal to or better than traditional criteria. The next step was comparing models that also had the interaction term. Results from these designs again supported the efficiency of optimal robust designs in handling data loss.

The last section of chapter four focused on comparing designs with a higher factor, specifically five. In cases where the model was just a first order with main effects, robust criteria sometimes failed to outperform traditional criteria in all cases but still managed to provide several cases when the robust design was more suitable. For models with main effects and interactions, robust criteria consistently computed designs superior to traditional criteria.

Chapter five introduced robust criteria specializing in generating designs robust to losing two data points. Three scenarios demonstrated that using robust criteria increased efficiency for designs facing potential loss of two observations. Understanding the potential use of coordinate exchange to compute robust optimal designs holds great value in the field of research where every observation is in some way expensive, either by time or funding. Future research may involve generating more designs with different factor settings or models, potentially revealing patterns indicating optimum cases for robust designs. The patterns I am referring to may come from design shape. During my search for optimal screening designs, there were two cases where the CEXCH algorithm struggled to find an optimal robust design that was more robust than the EXACT design. These two cases were: $N = 7$ 5-factors with only main effects and $N = 17$ 5-factors with main effects and interactions. As always, maybe if given more runs, the CEXCH would find the same design as the EXACT algorithm, but maybe there is something occurring at different N settings.

In recent research, Stallrich suggested that "generally A-optimal designs tend to push variances closer to their minimum possible value" and thus have the ability to correctly classify factors as active/inert [12]. Stallrich then provided examples and analysis of why, when constructing screening designs, being A-optimal has better characteristics than D-optimal. With this knowledge, I would suggest that if one were to pursue further robust screening designs, A-optimal is a promising option.

REFERENCES

[1] S. J. Walsh and J. J. Borkowski, "Improved g-optimal designs for small exact response surface scenarios: Fast and efficient generation via particle swarm optimization," *Mathematics*, vol. 10, no. 22, 2022. doi: https://doi.org/10.3390/math10224245.

[2] G. W. Oehlert, *A first course in design and analysis of experiments.* New York, NY: W.H. Freeman, Mar. 2000.

[3] S. J. Walsh, L. Lu, and C. M. Anderson-Cook, "I-optimal or g-optimal: Do we have to choose?," *Quality Engineering*, vol. 0, no. 0, pp. 1–22, 2023. doi: https://doi.org/10.1080/08982112.2023.2194963.

[4] A. C. Atkinson, "The usefulness of optimum experimental designs," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 59–76, 1996. doi: https://doi.org/10.1111/j.2517-6161.1996.tb02067.x.

[5] S. J. Walsh, *Development and applications of particle swarm optimization for constructing optimal experimental designs.* Ph.D. Thesis. Montana State University., 2021. doi: https://scholarworks.montana.edu/xmlui/handle/1/16309.

[6] S. J. Walsh and J. J. Borkowski, "Generating exact optimal designs via particle swarm optimization: Assessing efficacy and efficiency via case study," *Quality Engineering*, vol. 35, no. 2, pp. 304–323, 2022. doi: https://doi.org/10.1080/08982112.2022.2127364.

[7] S. J. Walsh, T. B. Bolten, and J. J. Borkowski, "Generating optimal designs with user-specified pure replication structure," *Quality and Reliability Engineering International.*, 2024.

[8] J. R. Wendelberger, "Uncertainty in designed experiments," *Quality Engineering*, vol. 22, no. 2, pp. 88–102, 2010. doi: https://doi.org/10.1080/08982110903510420.

[9] R. Johnson, D. Montgomery, and B. Jones, "An expository paper on optimal design," *Quality Engineering*, vol. 23, pp. 287–301, 07 2011. doi: https://doi.org/10.1080/08982112.2011.576203.

[10] P. Goos and B. Jones, *Optimal Design of Experiments: A Case study approach.* John Wiley and Sons, Ltd., 2011.

[11] W. A. Jensen, "Open problems and issues in optimal design," *Quality Engineering*, vol. 30, no. 4, pp. 583–593, 2018. doi: `https://doi.org/10.1080/08982112.2018.1517884`.

[12] J. Stallrich, K. Allen-Moyer, and B. Jones, "D- and a-optimal screening designs," *Technometrics*, vol. 0, no. 0, pp. 1–10, 2023. doi: `https://doi.org/10.1080/00401706.2023.2183262`.

[13] B. J. Smucker, W. Jensen, Z. Wu, and B. Wang, "Robustness of classical and optimal designs to missing observations," *Computational Statistics and Data Analysis*, vol. 113, pp. 251–260, 2017. doi: `https://doi.org/10.1016/j.csda.2016.12.001`.

[14] B. J. Smucker, E. del Castillo, and J. L. Rosenberger, "Model-robust two-level designs using coordinate exchange algorithms and a maximin criterion," *Technometrics*, vol. 54, no. 4, pp. 367–375, 2012. doi: `https://doi.org/10.1080/00401706.2012.694774`.

[15] P. Srisuradetchai, "Robust response surface designs against missing observations," *Montana State University*, p. 285, 2015.

[16] A. Strouwen and P. Goos, "A note on the output of a coordinate-exchange algorithm for optimal experimental design," *Chemometrics and Intelligent Laboratory Systems*, vol. 192, p. 103819, 2019. doi: `https://doi.org/10.1016/j.chemolab.2019.103819`.

[17] J. Borkowski, "Using a genetic algorithm to generate small exact response surface designs," *Journal of Probability and Statistical Science*, vol. 1, Feb. 2003.

APPENDIX: DESIGN CATALOG

All designs generated can be found on the author's GitHub page and at the link:
https://github.com/Ellwood12/Robust-Optimal-Designs

APPENDIX: JULIA CODE

```
using LinearAlgebra, Distributions, NLopt, DataFrames, CSV, Plots, Combinatorics

function genDesign(N, K)
    u = rand(Uniform(-1, 1), (N * K))
    X = reshape(u, N, K)
    return X
end

function D_crit(X)
    N = size(X,1)
    # THE CURRENT MODEL IS A SECOND ORDER WITH TWO FACTORS AND INTERACTION
    F_mat = hcat(fill(1, N), X, X[:,1] .* X[:,2], X.^2)
    det_int = det(*(F_mat', F_mat))
    limit =  eps()^(2)
    if det_int < limit
D_score = typemax(Float64)
    else
         D_score = 1/ (det(*(F_mat',F_mat)))
    end
    return D_score
end

function A_crit(X)
    N = size(X,1)
    # THE CURRENT MODEL IS A SECOND ORDER WITH TWO FACTORS AND INTERACTION
    F_mat = hcat(fill(1, N), X, X[:,1] .* X[:,2], X.^2)
    det_int = det(*(F_mat', F_mat))
    if det_int < sqrt(eps())
        A_score = typemax(Float64)
    else
         A_score = tr(inv(*(F_mat',F_mat)))
    end
    return A_score
end

function I_crit(X)
    N = size(X,1)
    # THE CURRENT MODEL IS A SECOND ORDER WITH TWO FACTORS AND INTERACTION
    F_mat = hcat(fill(1, N), X, X[:,1] .* X[:,2], X.^2)
    det_int = det(*(F_mat', F_mat))
    if det_int < sqrt(eps())
         I_score = typemax(Float64)
    else
         I_score = N * tr(*(K_2, inv(*(F_mat', F_mat))))
```

```
    end
    return I_score
end

function rb_D(X)
    N = size(X,1)
    # THE CURRENT MODEL IS A SECOND ORDER WITH TWO FACTORS AND INTERACTION
    F_mat = hcat(fill(1, N), X, X[:,1] .* X[:,2], X.^2)
    det_int = det(*(F_mat', F_mat))
    if det_int <  eps()^(2)
         D_score = typemax(Float64)
    else
         rows = size(F_mat, 1)
         sc = Vector{Float64}(undef, rows)
         for i in 1:rows
             mat = F_mat[setdiff(1:size(F_mat, 1), i), :]
             det_inf2 = det(*(mat', mat))
             if det_inf2 <  eps()^(2)
                 sc[i] = typemax(Float64)
             else
                sc[i] = 1/ (det(*(mat',mat)))
             end
         end
         D_score = maximum(sc)
    end
    return D_score
end

function rb_A(X)
    N = size(X,1)
    # THE CURRENT MODEL IS A SECOND ORDER WITH TWO FACTORS AND INTERACTION
    F_mat = hcat(fill(1, N), X, X[:,1] .* X[:,2], X.^2)
    det_int = det(*(F_mat', F_mat))
    if det_int < eps()^(2)
         A_score = typemax(Float64)
    else
         rows = size(F_mat, 1)
         sc = Vector{Float64}(undef, rows)
         for i in 1:rows
             mat = F_mat[setdiff(1:size(F_mat, 1), i), :]
             det_inf2 = det(*(mat', mat))
             if det_inf2 < sqrt(eps())
                 sc[i] = typemax(Float64)
             else
                 sc[i] = tr(inv(*(mat', mat)))
             end
         end
         A_score = maximum(sc)
    end
    return A_score
end

K_2 = [ 1 0 0 0 (1/3) (1/3);
  0 (1/3) 0 0 0 0;
```

```
  0 0 (1/3) 0 0 0;
  0 0 0 (1/9) 0 0;
  (1/3) 0 0 0 (1/5) (1/9);
  (1/3) 0 0 0 (1/9) (1/5)]

K_3 = [ 1 0 0 0 0 0 0 (1/3) (1/3) (1/3);
0 (1/3) 0 0 0 0 0 0 0 0;
0 0 (1/3) 0 0 0 0 0 0 0;
0 0 0 (1/3) 0 0 0 0 0 0;
0 0 0 0 (1/9) 0 0 0 0 0;
0 0 0 0 0  (1/9) 0 0 0 0;
0 0 0 0 0 0  (1/9) 0 0 0;
(1/3) 0 0 0 0 0 0 (1/5) (1/9) (1/9);
(1/3) 0 0 0 0 0 0 (1/9) (1/5) (1/9);
(1/3) 0 0 0 0 0 0 (1/9) (1/9) (1/5)]

function rb_I(X)
    N = size(X,1)
    # THE CURRENT MODEL IS A SECOND ORDER WITH TWO FACTORS AND INTERACTION
    F_mat = hcat(fill(1, N), X, X[:,1] .* X[:,2], X.^2)
    det_int = det(*(F_mat', F_mat))
    if det_int < sqrt(eps())
         I_score = typemax(Float64)
   else
         rows = size(F_mat, 1)
         sc = Vector{Float64}(undef, rows)
         for i in 1:rows
             mat = F_mat[setdiff(1:size(F_mat, 1), i), :]
             det_inf2 = det(*(mat', mat))
             if det_inf2 < sqrt(eps())
                 sc[i] = typemax(Float64)
             else
# IF THE MODEL WERE TO HAVE THREE FACTORS, WHICH K_2 TO K_3
                 sc[i] = tr(*(K_2, inv(*(mat', mat))))
             end
         end
         I_score = maximum(sc)
    end
    return I_score
end

function cexch(N, K, crit)
    X_init = genDesign(N, K)
    X_t = deepcopy(X_init)
    improvement = true
    n_iter = 0
    points = range(-1, stop = 1, step = .1)
    np = length(points) + 1
    while improvement
        n_iter = n_iter + 1
        crit_iter = crit(X_t)
        crit_t = NaN
        for i in 1:N
            for j in 1:K
```

```
                point_matrix = Vector{Any}(undef, np)
                point_list = fill(NaN, np)
                point_matrix[np] = deepcopy(X_t)
                point_list[np] = crit(X_t)
                for k in eachindex(points)
                    X = deepcopy(X_t)
                    X[i, j] = points[k]
                    point_matrix[k] = deepcopy(X)
                    point_list[k] = crit(X)
                end
                ind_best = argmin(point_list)
                X_t = deepcopy(point_matrix[ind_best])
                crit_t = deepcopy(point_list[ind_best])
            end
        end

        if crit_t == crit_iter
            improvement = false
        end
    end
    return Dict("initDesign" => X_init, "init_crit" => crit(X_init),
                "optDesign" => X_t, "opt_crit" => crit(X_t),
                "niter" => n_iter)
end
```