CONSTRUCTION, ANALYSIS, AND DATA-DRIVEN AUGMENTATION OF SUPERSATURATED DESIGNS

DISSERTATION

Alex J. Gutman,
AFIT-ENC-DS-13-S-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense, or the United States Government.

AFIT-ENC-DS-13-S-02

CONSTRUCTION, ANALYSIS, AND DATA-DRIVEN AUGMENTATION OF SUPERSATURATED DESIGNS

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy (Applied Mathematics)

Alex J. Gutman, BS, MS

September 2013

AFIT-ENC-DS-13-S-02

CONSTRUCTION, ANALYSIS, AND DATA-DRIVEN AUGMENTATION OF SUPERSATURATED DESIGNS

Alex J. Gutman, BS, MS

Approved:

//signed//	September 2013
Edward D. White, III, PhD (Chairman)	Date
//signed//	September 2013
Raymond R. Hill, PhD (Member)	Date
//signed//	September 2013
Dursun A. Bulutoglu, PhD (Member)	Date
//signed//	September 2013
Lt.Col. Richard L. Warr, PhD (Member)	Date

Accepted:

//signed//	September 2013
Dr. Heidi R. Ries Interim Dean, Graduate School of Engineering and Management	Date

AFIT-ENC-DS-13-S-02

Abstract

Screening designs are used in the early stages of industrial and computer experiments to find the most important input factors affecting a system's output. They provide an economical way to remove unimportant factors from further, potentially costly, experimentation. However, when an experiment has a large number of control factors and limited number of available runs, it is infeasible to run a traditional screening design. In these situations, experimenters can use supersaturated designs. A supersaturated design is a fractional factorial design that can screen a set of k factors in n runs, where $k > n - 1$. Unfortunately, they do not always provide definitive results. Improper and incomplete analysis of supersaturated designs can cause an experimenter to misclassify active factors and waste resources in subsequent experiments. In light of these concerns, this research investigates how to construct efficient and effective supersaturated designs, how to analyze such designs, and how to strategically plan follow-up runs to designs.

To Erin

Acknowledgements

Pursuing and completing a PhD is certainly not an individual effort. It takes a team of teachers, mentors, friends, and family - all of whom deserve tremendous thanks. Personally, I am indebted to numerous individuals and organizations who encouraged and enabled my educational pursuit. First, I must thank my employer, Riverside Research. Rather than paying lip service to the importance of education, they make it a real priority and fully support their employees' educational and career development goals.

My decision to undertake this effort was influenced by three individuals: Dr. Stephen Chambal, Dr. Jeff Weir, and Dr. Ray Hill. Four years ago, Dr. Chambal and Dr. Weir told me, "You don't want to work at AFIT for five years and realize you could be finished with your degree." They gave me the final push I needed to begin taking classes. Dr. Chambal and Dr. Hill then gave me the opportunity to work as a lead researcher for the "Science of Test" initiative. They introduced me to the right research topic at the right time, and it put me on a great career path.

My dissertation committee has been an invaluable resource throughout this process. My advisor, Dr. Edward White, gave me the guidance every graduate student needs, all while treating me with the respect of a colleague. I sincerely thank him for his dedication, direction, insight, and positive demeanor. I also thank Lt. Col. Richard Warr for his help with programming in R, Dr. Hill for reviewing the early drafts of my research papers, and Dr. Dursun Bulutoglu for his much-needed instruction and ideas during the final phase of this research. I would also like to thank Dr. Dennis Lin at Penn State for his suggestions to improve Chapter 4.

Friends have made this experience memorable and family has made it meaningful. Thanks to Carl, Paul, Clayton, Cade, Brad, Trevor, Marcus, Jimmy, and others for suffering through this with me. Thanks to my parents for raising me to be a life-long

learner. They taught me how to be a good student, and more importantly, how to be a good person. I love and appreciate them very much. Lastly, I want to thank my beautiful wife for her unwavering support. She took care of our home and lives while I spent the majority of the past few years in our basement doing research. I admire and love her, and no amount of thanks will ever make up for what I owe her.

Alex J. Gutman

Table of Contents

Page

Page

List of Figures

List of Tables

Table Page

Table Page

CONSTRUCTION, ANALYSIS, AND
DATA-DRIVEN AUGMENTATION OF
SUPERSATURATED DESIGNS

I. Introduction

Screening designs are used in the early stages of industrial and computer experiments to find the most important input variables, or factors, affecting a system's output. They provide an economical way to remove unimportant factors from further experimentation. Consider, for instance, a system optimization experiment with six input variables. One common design used in optimization experiments is called a central composite design (CCD) (Myers et al., 2009). A CCD with with six control factors and two center points requires 46 experimental runs. Suppose, however, that two of the input factors had minimal effects on the output. A CCD with four factors would have required only 26 runs. The presence of the two extraneous variables increased the number of runs by 20. A simple two-level fractional factorial design (Montgomery, 2009) could have screened the six original factors into four in only eight runs. Thus, the screening design and CCD would require 12 fewer runs than the original design. When a single experimental run costs thousands of dollars, the benefits of screening are immediately clear.

It is important to separate the factors with large effects on the output (call these *active*) from those with minimal or no effect (*inactive*) as efficiently as possible in order to save most of the experimental budget for more in-depth experiments. In certain cases, the number of factors exceeds the number of runs available to test them. In these situations, experimenters can use supersaturated designs. A super-

saturated design is a fractional factorial design that can screen a set of k factors in n runs, where $k > n - 1$. However, supersaturated designs do not always provide definitive results. Improper and incomplete analysis of supersaturated designs can cause an experimenter to misclassify active factors and waste resources in subsequent experiments.

As test budgets shrink, efficient and effective screening designs will play a more pivotal role in experimentation. Supersaturated designs are one of the few screening techniques equipped to study a large number of factors in a limited number of runs. Despite their problems, they are preferred over naive screening approaches like subjective opinion about what variables are important. Decision makers want statistically defendable results, not best guesses. Therefore, its crucial to understand how to use supersaturated designs and how to improve them.

1.1 Research Objective and Scope

Research on supersaturated designs can generally be divided into two sects: how to construct the designs and how to analyze the designs. Unfortunately, researchers in the construction and analysis areas have not come to an agreement on the best way to do either. Moreover, there is a void in the literature on how to add runs to a supersaturated design. Therefore, this dissertation investigates *how to construct efficient and effective designs, how to analyze such designs, and how to strategically plan follow-up runs to supersaturated designs using information from the initial experiment.*

We proceeded with the following specific goals:

1. *Develop and validate a straightforward method to analyze supersaturated designs.* The analysis of supersaturated designs is difficult because of highly complex aliasing structures in the design matrix. Main effects are partially aliased with each other, creating false positives of important factors. Standard regres-

sion techniques generally do not work well, and techniques that work better are difficult for practitioners to implement. Finding an intermediary analysis method is desirable.

2. *Introduce a data-driven methodology to augment supersaturated designs with follow-up runs to improve screening results.* If a supersaturated design did not provide the experimenter with enough information to comfortably move forward into the next phase of experimentation, additional runs are needed to study the factors in more detail. Adding follow-up runs to supersaturated designs is a relatively new research area. Two papers have been published about adding follow-on runs to supersaturated designs, but they do so independently of the data. A better approach would be to analyze the initial data first, and then prioritize the new runs to clarify discrepancies in the data.

3. *Address concerns with the traditional construction criterion for supersaturated designs, $E(s^2)$, and create designs that are optimal under the resolution-rank criterion.* The $E(s^2)$ measurement of a supersaturated design measures the "orthogonality" of a design. By definition, supersaturated designs cannot be orthogonal, but the desire to reduce a design's $E(s^2)$ has been a major research thrust for 20 years. However, an $E(s^2)$ optimal design may not have the ability to differentiate between two competing models, even in the noiseless case. A different criterion, resolution-rank, was introduced by Deng et al. (1996) but never fully matured. New designs optimal with respect to the resolution-rank criterion will give experimenters a better opportunity to detect a system's active factors.

1.2 Overview and Organization

The remainder of this dissertation follows a scholarly article format. Chapters II, III, IV, and V are self-contained research articles on supersaturated designs. Each contains a literature review of the research relevant to that chapter. The original contribution of each chapter is as follows:

Chapter II gives an overview of large screening experiments and discusses the background and terminology of supersaturated designs. The intent of the chapter is to introduce the reader to supersaturated designs and provide general guidance on how to construct, analyze, and augment a design. Proposed construction and analysis techniques are presented, as well as a novel method to add runs to supersaturated designs. The article was accepted into the *Proceedings of the 2013 Industrial and Systems Engineering Research Conference* (ISERC) and was presented at ISERC in San Jaun, Puerto Rico in May 2013 (Gutman et al., 2013a).

Chapter III is an in-depth exploration of the specific challenges experimenters face when analyzing data from a supersaturated experiment. The introduction shares some components with Chapter II, but the crux of Chapter III is on addressing the inherent difficulties associated with supersaturated designs and developing a straightforward analysis method to mitigate Type I errors (declaring an inactive factor as active). Chapter III also contains a comprehensive simulation study of numerous supersaturated design analysis techniques. The article is currently under review for publication in the *Journal of Quality Technology*.

Chapter IV introduces an original data-driven methodology to add runs to supersaturated designs. In side-by-side comparisons, the proposed Bayesian D-optimal method outperforms the supersaturated design augmentation techniques in the literature. The article has been submitted for publication in *Computational Statistics & Data Analysis* and is currently undergoing its second review.

Chapter V addresses concerns with the $E(s^2)$-optimality criterion for two-level supersaturated designs and introduces a catalogue of new designs with high resolution-rank, a criterion that directly assesses a supersaturated design's ability to detect active factors. Several of the designs presented are shown to be provably optimal. The search for large supersaturated designs with high resolution-rank is aided by binary integer programming and design isomorphism properties. This article will be submitted for publication in the *Journal of Statistical Theory and Practice*. And lastly, Chapter VI reiterates the importance of studying supersaturated designs, summarizes all original research contributions, and provides suggestions for future work.

II. Large Screening Experiments: An Overview of Supersaturated Designs for Practitioners

2.1 Introduction

The influential inputs to a process are not always known in advance. As such, a screening experiment is done to separate factors into the those that are influential and those that are not. Specifically, screening is the process of using statistically designed experiments to find the factors that appear to influence a response variable. Screening is often referred to as "Phase 0" of an experiment to communicate its importance at the very onset of a test (Myers et al., 2009). For large screening experiments, researchers have observed that changes in a response variable are usually caused by a small number of *active* factors - a concept called "effect sparsity" (Box and Meyer, 1986). Finding the active factors in a system directs resources for future experiments because carrying superfluous variables into the more involved Phase I of experimentation can drastically increases costs.

Traditional screening designs, like fractional factorial designs (Montgomery, 2009) and Plackett-Burman (Plackett and Burman, 1946) designs, have more runs than control factors, a necessary condition for standard statistical analysis. However, budget constraints may be too restrictive for this requirement. Many experiments have a large pool of control factors - larger than the number of runs available to analyze them. For example, suppose a weapon system has 40 input factors, but experimenters can only afford to do 20 runs. What is the best way to perform this experiment? To answer this, alternative designs are needed that can screen a set of k variables in n runs, where $k > n - 1$. The three most popular techniques for large screening experiments are (1) group screening, (2) sequential bifurcation, and (3) supersaturated designs. In this paper, we briefly discuss the first two methods before turning our

attention to supersaturated designs. Our goal is to introduce practitioners to basic concepts and technical issues related to the construction, analysis, and augmentation of supersaturated designs.

2.1.1 Group Screening and Sequential Bifurcation.

In group screening (Watson, 1961), g groups of factors are each assigned k factors. For instance, to perform the 40-factor, 20-run military experiment mentioned earlier, an experimenter could put the 40 factors into 8 groups of 5. The experiment proceeds using a two-level design with 8 "group factors", where each group of 5 factors is assigned to a column in a standard design matrix. A $+1$ in the design matrix means each of the 5 factors in the group is set to its high-level, or vice versa for a -1. The objective is to screen out the inactive *groups*, and then regroup the factors from active groups into new groups. See Vine et al. (2008) for a detailed discussion of two-stage group screening. In group screening, certain assumptions must be satisfied. The most limiting assumption is that the sign of all effects must be known in advance. So, when a group is set at its high level ($+1$), all factors in that group must have a positive effect on the response. Otherwise, factors within the same group could cancel each other out. For more on group screening, see Kleijnen (1987) and Morris (2006).

Sequential bifurcation is an extension of group screening, so the same assumptions must hold. As its name suggests, sequential bifurcation is sequential in nature. In the first stage, all factors are placed in a single group at their high level. If the group is active (i.e. contains active factors), it is split (bifurcated) into two subsequent groups, each of which is tested for importance. The process continues until the active factors are found. This technique has been useful in computer simulation experiments, and the reader is referred to Kleijnen et al. (2006) for more information.

2.2 Supersaturated Designs

Supersaturated designs have less restrictive assumptions than group screening and sequential bifurcation because the experimenter does not need to know the sign of the effects in advance. The focus of supersaturated designs is on identifying the important main effects in a linear model. Consider an experiment with k factors and n runs. The underlying main-effect model is represented as:

$$\mathbf{y} = \beta_0\mathbf{1} + \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.1}$$

where $\mathbf{y}$ is the response vector, $\beta_1, \ldots, \beta_k$ are the unknown model parameters, $\mathbf{X} = (\mathbf{1}, \mathbf{x}_1, \ldots, \mathbf{x}_k)$ is the model matrix, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_{n\times n})$ is the error term. Calculating the parameter estimates for traditional designs with $n > k$ is done via ordinary least squares (OLS), $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. But, when $\mathbf{X}$ has more columns than rows, $\mathbf{X}'\mathbf{X}$ is singular and the OLS estimates do not exist. Effect sparsity suggests most of the β_i's in Equation 2.1 are zero, so supersaturated designs are used to remove those negligible factors from further consideration. Then, when the matrix projects to a dimension less then n, it's possible to find an estimable model.

The traditional definition for a supersaturated design is a fractional factorial design in which the number of factors, k, is larger than $n-1$, where n is the number of runs (Montgomery, 2009). The designs were introduced around 1960 (Satterthwaite, 1959; Booth and Cox, 1962), but they did not gain popularity in the statistics field until the early 1990's (Lin, 1993; Wu, 1993; Lin, 1995a). Research on supersaturated designs has been extremely active since that time and generally falls into two camps: how to construct designs and how to analyze the data. Section 2.3 reviews the most popular construction techniques, and Section 2.4 highlights some of the proposed

analysis methods. A novel approach to add follow-up runs to designs is presented in Section 2.5, and Section 2.6 concludes the paper.

2.3 Constructing a Supersaturated Design

Table 1 shows a two-level supersaturated design from Holcomb and Carlyle (2002) to serve as a reference for terms and concepts in this section. Supersaturated designs are constructed to be "optimal" with respect to some criterion. Two prominent optimality criteria with similar performance capabilities (Marley and Woods, 2010) are presented here.

Table 1. Supersaturated design example with 14 factors and 8 runs

	Design Factors													
Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2	−	+	−	−	−	+	+	−	−	−	+	−	+	+
3	+	−	+	−	−	−	+	+	−	−	−	+	−	+
4	+	+	−	+	−	−	−	+	+	−	−	−	+	−
5	−	+	+	−	+	−	−	−	+	+	−	−	−	+
6	−	−	+	+	−	+	−	+	−	+	+	−	−	−
7	−	−	−	+	+	−	+	−	+	−	+	+	−	−
8	+	−	−	−	+	+	−	−	−	+	−	+	+	−

2.3.1 $E(s^2)$ Criterion.

In an orthogonal design, the dot product between any two columns, $\mathbf{x}_i'\mathbf{x}_j, i \neq j$ is zero. Therefore, for an orthogonal design matrix $\mathbf{X}_{n\times(k+1)}$, $\mathbf{X}'\mathbf{X} = n\mathbf{I}_{(k+1)\times(k+1)}$, so each main effect can be estimated without bias. This is impossible with supersaturated designs because $\mathbf{X}$ has more columns than rows. Since true orthogonality is impossible, supersaturated designs are constructed to be as "nearly orthogonal as possible" (Booth and Cox, 1962). A measurement of near-orthogonality is the $E(s^2)$ criterion, which has become the standard criterion for all balanced two-level supersat-

urated designs. The idea of the $E(s^2)$-criterion is the make the off-diagonal elements of $\mathbf{X}'\mathbf{X}$, on average, as close to 0 as possible; this effectively makes the design "nearly orthogonal." Denote the (i,j)th element of $\mathbf{X}'\mathbf{X}$ as s_{ij}. $E(s^2)$ is then defined as $E(s^2) = \sum_{i<j} s_{ij}^2/(k(k-1)/2)$, where k is the number of factors in the model. A design with the lowest possible $E(s^2)$ value for a given size is said to be $E(s^2)$-optimal. Researchers have proposed many systematic and computational construction methods to create $E(s^2)$-optimal designs (Nguyen, 1996; Tang and Wu, 1997; Bulutoglu and Cheng, 2004; Ryan and Bulutoglu, 2007; Das et al., 2008), and a great library of designs is available online at: `http://www.iasri.res.in/design/Supersaturated_Design/SSD/Supersaturated.html`

2.3.2 Bayesian D-Optimal Designs.

Another popular criterion used to construct and compare supersaturated designs is Bayesian D-Optimality (Jones et al., 2008). Bayesian D-Optimality can create designs of any size with any number of blocks and can also incorporate categorical variables. Perhaps most important, the designs are computer-generated and construction is implemented in the JMP statistical software, making it easy for practitioners to create a design for their specific problem, as opposed to using a catalogued design which may not have the correct amount of factors or runs.

To create a Bayesian D-Optimal design, assume the common main-effects screening model in Equation 2.1 holds. Let the prior distribution of the parameters be $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \sigma^2\mathbf{R}^{-1})$, where $\mathbf{R}$ is some covariance matrix, and the conditional distribution of $\mathbf{y}$ given $\boldsymbol{\beta}$ be $\mathbf{y}|\boldsymbol{\beta} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$. The posterior distribution for $\boldsymbol{\beta}$ given $\mathbf{y}$ is then $\boldsymbol{\beta}|\mathbf{y} \sim N(\boldsymbol{\beta}^*, \sigma^2\mathbf{D})$, where $\boldsymbol{\beta}^* = (\mathbf{X}'\mathbf{X}+\mathbf{R})^{-1}(\mathbf{X}'\mathbf{y}+\mathbf{R}\boldsymbol{\beta}_0)$ and $\mathbf{D} = (\mathbf{X}'\mathbf{X}+\mathbf{R})^{-1}$. A Bayesian D-Optimal design, $\mathbf{X}_B$, aims to reduce the error variances of the parame-

ter estimates given in $\mathbf{D}$. This is accomplished by constructing a design that satisfies:

$$\mathbf{X}_B = \text{argmax}_{\mathbf{X}} |\mathbf{X}'\mathbf{X} + \mathbf{R}|, \tag{2.2}$$

where $|\cdot|$ is the determinant operator.

Prior information and uncertainty about the parameters must be modeled under the Bayesian paradigm. Jones et al. (2008) suggest using prior information to split models terms into two sets: primary terms and potential terms. Primary terms are assumed active in the true model, whereas potential terms may or may not be active. Using this information, the p_1 primary terms employ a diffuse prior with an arbitrary prior mean and prior variance tending toward infinity. The infinite variance implies no knowledge on the main effects of the primary terms, and the non-zero arbitrary mean implies the main effects are less likely equal to zero. The $p_2 = (k+1) - p_1$ potential terms are given a prior mean zero and finite variance $\tau^2\sigma^2$, where τ represents the expected effect of the factor relative to standard error. The prior information is then reflected in the matrix $\mathbf{R}$. Let $\mathbf{R} = \mathbf{K}/\tau^2$, where

$$\mathbf{K} = \begin{pmatrix} \mathbf{0}_{p_1 \times p_1} & \mathbf{0}_{p_1 \times p_2} \\ \mathbf{0}_{p_2 \times p_1} & \mathbf{I}_{p_2 \times p_2} \end{pmatrix}. \tag{2.3}$$

$\mathbf{R} = \mathbf{K}/\tau^2$ is substituted into Equation 2.2, and the coordinate exchange algorithm (Meyer and Nachtsheim, 1995) is used to create the design. (The algorithm is discussed in detail in Section 2.5.) For supersaturated designs, all k control factors are typically set as potential terms because the experimenter cannot assume they are active *a priori*. The intercept term in the model matrix $\mathbf{X}$ is the only primary term.

2.4 Analyzing a Supersaturated Design

Regardless of the construction method, supersaturated designs are inherently difficult to analyze because $\mathbf{X}'\mathbf{X}$ is singular. Moreover, the correlation structure and interdependencies of a supersaturated design matrix make it hard (sometimes impossible) to find the correct model. Many researchers have investigated this problem and novel analysis methods have been introduced in the literature. It is important to note that there is no accepted "best" way to analyze a supersaturated design, as each method has pros and cons. See Gilmour (2006); Mee (2009); and Georgiou (2012) for detailed literature reviews of proposed methods. In this section, we highlight two methods; first, the Dantzig selector method because it has gained popularity in recent simulation studies. Then, basic regression techniques are discussed (forward regression and all-subsets regression) because practitioners are familiar with these methods.

2.4.1 Dantzig Selector.

The Dantzig selector (Candes and Tao, 2007) has recently been applied to the analysis of supersaturated designs (Phoa et al., 2009). The Dantzig selector searches for active factors in a supersaturated design via linear programming. To find the parameter estimates in Equation 2.1, $\hat{\boldsymbol{\beta}}$, we solve the linear programming problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{p} |\beta_i| \quad \text{s.t.} \quad \|\mathbf{X}'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})\|_\infty \leq \delta \tag{2.4}$$

where $|\cdot|$ is the absolute value, δ is a tuning parameter, and for a vector $\mathbf{x}$, $\|\mathbf{x}\|_\infty = \max |x_i|$. The Dantzig selector performed well when compared to other methods in a simulation study (Marley and Woods, 2010), but it does require a judicious selection of the tuning parameter. If δ is too high, the linear program may set $\hat{\boldsymbol{\beta}} = \mathbf{0}$, declaring

all factors inactive. If δ is too small, the program might overfit the model. When tuned properly, the Dantzig selector is a good analysis method and relatively easy to implement since many software programs, such as R, can solve linear programs.

2.4.2 Basic Regression Methods.

The assumption of the linear model in Equation 2.1 implies basic subset regression techniques may be a good starting point for the analysis of supersaturated designs. Forward and all-subsets have been used with some success (Lin, 1993; Wu, 1993; Abraham et al., 1999), but all-subsets regression is generally preferable. In forward regression, we start with an intercept-only model. The contribution of each of the k main effects is calculated, and whichever variable most improves the model is added. The process repeats to add additional factors until further addition fails to provide sufficient model improvement. In all-subsets regression, a model is fit to every possible combination of factors. For supersaturated designs, this can be a computational issue because the number of possible models in a large data set can be large. Also, users should be cautious when using subset section methods because the results are not always clear. Consider the design in Table 2 having responses generated with the equation $\mathbf{y} = -20\mathbf{x}_1 + 20\mathbf{x}_5 + 20\mathbf{x}_8 + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$.

Table 2. Example Supersaturated Design with Responses

	Design Factors with $\mathbf{y} = -20\mathbf{x}_1 + 20\mathbf{x}_5 + 20\mathbf{x}_8 + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$														
Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{y}$
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	18.629
2	−	+	−	−	−	+	+	−	−	−	+	−	+	+	-20.870
3	+	−	+	−	−	−	+	+	−	−	−	+	−	+	-20.720
4	+	+	−	+	−	−	−	+	+	−	−	−	+	−	-21.328
5	−	+	+	−	+	−	−	−	+	+	−	−	−	+	20.834
6	−	−	+	+	−	+	−	+	−	+	+	−	−	−	19.929
7	−	−	−	+	+	−	+	−	+	−	+	+	−	−	20.482
8	+	−	−	−	+	+	−	−	−	+	−	+	+	−	-20.420

Suppose a forward regression is used to identify the true active factors. The results are in Table 3. Notice, however, that forward regression did not do well choosing the correct parameter estimates. The first factor chosen to be active, $\mathbf{x}_{13}$, is a false effect; it *appears* active because the factor is correlated with real active factors. Forward regression cannot correct the issue. It continues to select inactive factors, and only one active factor, $\mathbf{x}_5$, was detected.

Table 3. Forward Regression Results on Table 2 with Parameter Estimates

Step	β_{13}	β_5	β_{11}	β_{12}	R^2	R^2_{adj}
1	-10.56				0.268	0.156
2	-10.56	10.31			0.523	0.333
3	-10.56	10.31	9.98		0.762	0.584
4	-10.56	13.80	9.98	-6.98	0.850	0.649

Results from all-subsets regression in Table 2 are also troublesome. Although the true model only has three active factors, best three-factor model found with all-subsets regression contains $\mathbf{x}_3, \mathbf{x}_4$, and $\mathbf{x}_8$. Fortunately, the four-factor model identifies the true factors, but in a real situation, the form of the underlying model is obviously unknown. Consequently, the recommendations from such an analysis are very difficult. Nevertheless, basic regression methods *can* be helpful, but are certainly not infallible. For a detailed discussion about the analytical challenges associated with supersaturated designs, please see Gutman et al. (2013b).

Table 4. All-Subsets Results on Table 2

Number	Model Terms	R^2	$AICc$
1	$\mathbf{x}_{13}$	0.2679	80.4661
2	$\mathbf{x}_5, \mathbf{x}_{13}$	0.5232	86.3683
3	$\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_8$	0.9996	49.1094
4	$\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_8, \mathbf{x}_{13}$	0.9999	91.8652

2.5 Augmenting a Design

With all the problems faced by an experimenter using supersaturated designs, one thing is clear: more runs would be helpful. This section reviews some of the basic augmentation techniques for regular designs and then presents a new technique to augment supersaturated designs.

2.5.1 Augmenting Standard Designs.

To make a computer generated standard design ($n > k$), the user specifies the desired number of factors and runs, and a computer algorithm generates a design to optimize some criteria, usually the D-optimality criteria. In regular designs that have fewer factors than runs, a D-optimal design, $\mathbf{X}_D$ is one that maximizes the determinant

$$\text{argmax}_{\mathbf{X}}|\mathbf{X}'\mathbf{X}|. \tag{2.5}$$

D-optimal designs are popular because designs that maximize Equation 2.5 minimize the variance of the design parameters, $\boldsymbol{\beta}$. Orthogonal designs, like factorial and fractional-factorial designs, are D-optimal. For other two-level designs, D-optimal designs can be generated with the coordinate-exchange algorithm (Meyer and Nachtsheim, 1995). The algorithm is summarized in the following steps:

1. For each entry $x_{i,j}$ in the model matrix $\mathbf{X}$, generate a uniform random number from $[-1, 1]$.

2. At $x_{i,j}$, replace the random entry with -1 and calculate $|\mathbf{X}'\mathbf{X}|$.

3. Replace the same entry with $+1$ and calculate $|\mathbf{X}'\mathbf{X}|$.

4. Choose the value $\{-1, +1\}$ that results in the largest determinant.

5. Proceed to the next entry in the matrix and repeat the process.

When the algorithm terminates, all entries in the model matrix $\mathbf{X}$ are ± 1. The process is repeated many times with different starting values for the $x_{i,j}$ entries. After many random starts, e.g. 100, the design with the largest determinant is returned as the D-optimal design, $\mathbf{X}_D$.

Augmenting a design applies similar concepts. To explain how to choose the best possible follow-up runs for an non-saturated experiment, we give an example similar to one presented in Goos and Jones (2011). Designs in this section were created with the JMP$^{\text{TM}}$ 9.0 statistical software. Consider the model matrix for a screening experiment with four factors and eight runs:

$$\mathbf{X}^* = \begin{pmatrix} 1 & -1 & -1 & -1 & -1 \\ 1 & +1 & +1 & +1 & -1 \\ 1 & -1 & +1 & -1 & +1 \\ 1 & +1 & -1 & +1 & +1 \\ 1 & -1 & +1 & +1 & +1 \\ 1 & +1 & +1 & -1 & -1 \\ 1 & +1 & -1 & -1 & +1 \\ 1 & -1 & -1 & +1 & -1 \end{pmatrix} \tag{2.6}$$

It's easy to check the matrix $\mathbf{X}^{*\prime}\mathbf{X}^* = 8\mathbf{I}_5$. So, the design is orthogonal and hence optimal for the main-effect model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon. \tag{2.7}$$

After the initial eight runs, the experimenter may want to estimate two-factor interactions. In this case, the model in Equation 2.7 is insufficient. To estimate the four main effects and six two-factor interactions, the new model becomes

$$\begin{aligned} Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_{12} x_1 x_2 \\ + \beta_{13} x_1 x_3 + \beta_{14} x_1 x_4 + \beta_{23} x_2 x_3 + \beta_{24} x_2 x_4 + \beta_{34} x_3 x_4 + \epsilon. \end{aligned} \tag{2.8}$$

The model matrix for the new model with interaction terms is constructed by adding the interaction columns (i.e. $x_1x_2, x_1x_3, \dots$) to the original model matrix, $\mathbf{X}^*$. This gives the new model matrix, $\mathbf{X}_1$, in Equation 2.9. $\mathbf{X}_1$ has 11 columns: one for the intercept, four for the main effects, and six for the two factor interactions.

$$\mathbf{X}_1 = \left(\begin{array}{c|cccc|cccccc} 1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 \\ 1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 \\ 1 & -1 & +1 & -1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 \\ 1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 \\ 1 & -1 & +1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 \\ 1 & +1 & +1 & -1 & -1 & +1 & -1 & -1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & -1 & -1 \\ 1 & -1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{array}\right) \tag{2.9}$$

The matrix $\mathbf{X}_1$, without follow-up runs, is supersaturated because it does not have enough runs to estimate all 11 effects. Because the matrix has more columns than rows, some columns are linearly dependent. Notice that certain factors in $\mathbf{X}_1$are completely aliased: $x_1 = -x_2x_4$, $x_2 = -x_1x_4$, and $x_4 = -x_1x_2$. Augmenting the design will resolve the linear dependencies. We could, of course, use the fold-over technique (Montgomery, 2009), but this will double the total run size to 16. Adding runs with the D-optimal approach lets us choose the number of additional runs. In this case, augmenting the design with four runs will give a total of 12 runs, which will be enough to estimate all main effects.

Denote the additional runs of the model matrix $\mathbf{X}_2$. The final form of the model matrix $\mathbf{X}$ is given by

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} = \left(\begin{array}{cccccccccccc} 1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 \\ 1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 \\ 1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & -1 & +1 & -1 \\ 1 & +1 & -1 & +1 & +1 & -1 & +1 & +1 & -1 & -1 & +1 \\ 1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 \\ 1 & +1 & +1 & -1 & -1 & +1 & -1 & -1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ 1 & -1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ \hline 1 & . & . & . & . & . & . & . & . & . & . \\ 1 & . & . & . & . & . & . & . & . & . & . \\ 1 & . & . & . & . & . & . & . & . & . & . \\ 1 & . & . & . & . & . & . & . & . & . & . \end{array}\right) \tag{2.10}$$

The entries for the final four runs are chosen to maximize the determinant of the final information matrix $\mathbf{X}'\mathbf{X}$. A D-optimal follow-up design will maximize $|\mathbf{X}'\mathbf{X}| = |\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2|$, where $\mathbf{X}_1$ is fixed. It's important to take into account the entire design matrix. "Not doing so, by just maximizing $|\mathbf{X}_2'\mathbf{X}_2|$, will lead to a follow-up experiment that pays equal attention to the main effects, about which we already have substantial information from the initial experiment," (Goos and Jones, 2011, pp. 63). The coordinate-exchange algorithm constructed the following complete model matrix.

$$\mathbf{X} = \left(\begin{array}{cccccccccccc} 1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 \\ 1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & -1 \\ 1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & -1 & +1 & -1 \\ 1 & +1 & -1 & +1 & +1 & -1 & +1 & +1 & -1 & -1 & +1 \\ 1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 \\ 1 & +1 & +1 & -1 & -1 & +1 & -1 & -1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ 1 & -1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ \hline 1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ 1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ 1 & -1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & +1 \end{array}\right) \tag{2.11}$$

The new matrix is no longer singular, so no columns are completely aliased. Therefore, the combination of the original matrix, $\mathbf{X}_1$ and the follow-up design, $\mathbf{X}_2$, has enough information to estimate all main effects and two-factor interactions in Equation 2.8. If too much time passes between the original and follow-up experiments, a blocking factor can be added to test if the mean response shifted over time.

2.5.2 Augmenting Supersaturated Designs.

Next, the above methods are applied to augment supersaturated designs. There is little research in this area; two recent papers (Gupta et al., 2010, 2012) discuss some theoretical augmentation strategies. For instance, an $E(s^2)$-optimal design can be augmented with additional runs to create a new class of "extended $E(s^2)$-optimal" supersaturated designs with new lower bounds, but this is independent of any analysis on the initial experiment. It may be beneficial to analyze the initial experiment first and use the response data to strategically choose the additional runs. Such a method is presented here.

Let's consider the supersaturated design in Table 2. Suppose the experimenter wants to know the three most important factors. Further investigation of the data with all-subsets regression revealed the two best three-factor models to be:

1. $f_1 = -0.433 + 20.522\mathbf{x}_3 + 20.281\mathbf{x}_4 - 20.841\mathbf{x}_8$ with $R^2 = 0.9996$

2. $f_2 = -0.433 - 20.508\mathbf{x}_1 + 20.295\mathbf{x}_5 + 19.962\mathbf{x}_8$ with $R^2 = 0.9995$

Notice that f_2 is the true underlying model. An experimenter, of course, will not recognize this after the first eight runs because the top two models are essentially identical in terms of the R^2 criterion, which measures how well the function fits the data ($R^2 = 1$ implies perfect fit). Now suppose the experimenter wants to test which model is really generating the response data. An additional run can do this. In

essence, there are two "competing" models, and a new run is required to differentiate the models as much as possible.

One way to differentiate two models to choose an additional run to maximize the distance between the two models' predicted values: $\hat{y_1} = f_1(x_3, x_4, x_8)$ and $\hat{y_2} = f_2(x_1, x_5, x_8)$. Thus, we want to maximize $|\hat{y_1} - \hat{y_2}|$. This criterion is called the Maximum Differences between Predictions (MDP) (Jones et al., 2007). Let S be the set of all factors in the design and M be the set of all factors in the models of interest; i.e. $M = \{x_1, x_3, x_4, x_5, x_8\}$ and $S = \{x_1, x_2, \ldots, x_{14}\}$. Now, suppose the experimenter can add a ninth run. The first objective is to choose the appropriate factor levels for all $x_i \in M$ to maximize the MPD:

$$\max_{\forall x_i \in M} |\hat{y_1} - \hat{y_2}|. \tag{2.12}$$

For all factors not in M, factor levels are chosen to optimize some other design criterion. In Section 2.5.1, runs were added to standard designs to optimize the D-optimality criterion, which will not work for supersaturated designs because the determinant of $\mathbf{X'X}$ is zero when $\mathbf{X}$ has more columns than rows. As such, another criterion is needed. To mimic the goal of $E(s^2)$ in Section 2.3.1, factor levels are chosen to reduce the pairwise correlations of the design matrix. Thus, the second objective is choosing the appropriate factor levels for all $x_i \in S \setminus M$ to minimize:

$$\min_{\forall x_i \in S \setminus M} \sum_{i<j} (\mathbf{x}_i' \mathbf{x}_j)^2. \tag{2.13}$$

Like the augmentation in Section 2.5.1, this procedure is carried out with the coordinate-exchange algorithm from Section 2.5.1. First, the design in Table 2 is augmented with a ninth row of random numbers generated uniformly in the range [-1,1] to give baseline values for $|\hat{y_1} - \hat{y_2}|$ and $\sum_{i<j} (\mathbf{x}_i' \mathbf{x}_j)^2$. Then, for each $x_i \in M$,

test whether a $+1$ or -1 optimizes the MPD in Equation 2.12. This is done in Table 5. The value of x_1 for the ninth run, $x_{1,9}$, was initially set to 0.86. If $x_{1,9} = -1$, the MPD $|\hat{y_1} - \hat{y_2}| = 79.28$. If $x_{1,9} = +1$, the MPD $|\hat{y_1} - \hat{y_2}| = 38.27$. Thus, the factor level for $x_{1,9}$ is set to -1 because it provides the largest separation between models. The process continues by assigning factor levels for the remaining factors in M.

Table 5. Choosing Factor Levels to Optimize MPD

Factors in M	$x_{9,1}$	$x_{9,3}$	$x_{9,4}$	$x_{9,5}$	$x_{9,8}$
U(-1,1)	0.86	-0.53	-0.15	0.52	0.84
$\lvert\hat{y_1} - \hat{y_2}\rvert$ with -1	79.28	88.89	106.20	75.32	40.80
$\lvert\hat{y_1} - \hat{y_2}\rvert$ with $+1$	38.27	47.84	65.64	115.91	122.41
Factor Level	-1	-1	-1	1	1

To choose factor levels for the remaining factors in $S \setminus M$, test whether a $+1$ or -1 optimizes Equation 2.13. In Table 6, $x_{9,2}$ was initially set to -0.54. If $x_{9,2} = -1$, $\sum_{i<j}(\mathbf{x}_i'\mathbf{x}_j)^2 = 471.54$. If $x_{9,2} = +1$, $\sum_{i<j}(\mathbf{x}_i'\mathbf{x}_j)^2 = 478.54$. Therefore, -1 is the best choice for the factor level because it provides a smaller value. The process continues for the remaining variables to choose the factor levels for the ninth run.

The response for the addition run,

$$(-1, -1, -1, -1, 1, -1, +1, +1, +1, +1, +1, -1, +1, +1),$$

is 60.332. The best three-factor model on entire design contained factors $\mathbf{x}_1, \mathbf{x}_5$, & $\mathbf{x}_8$ had $R^2 = 0.9998$, so there is greater evidence that f_2 is the true underlying model. Further, the model factors $\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_8$ now has $R^2 = 0.0461$, which indicates it is not the underlying model. The additional run effectively discriminated between the top two competing models. Note that due to the random start, this run is a local optimum. The process can be repeated several times to find a more suitable run.

The example highlights how using information from the initial supersaturated design can improve model selection. Also, had the original models been incorrect, the

Table 6. Choosing Factor Levels to Optimize Equation 2.13

Factor in $S \setminus M$	$x_{9,2}$	$x_{9,6}$	$x_{9,7}$	$x_{9,9}$	$x_{9,10}$	$x_{9,11}$	$x_{9,12}$	$x_{9,13}$	$x_{9,14}$
U(-1,1)	-0.54	0.65	-0.63	-0.30	0.99	-0.76	0.25	0.71	0.28
$\sum_{i<j}(\mathbf{x}_i'\mathbf{x}_j)^2$ with -1	471.45	460.80	470.61	461.50	493.58	466.54	458.29	492.13	473.00
$\sum_{i<j}(\mathbf{x}_i'\mathbf{x}_j)^2$ with $+1$	478.54	480.58	451.30	461.50	461.58	466.54	490.29	460.13	473.00
Factor Level	-1	-1	1	1	1	1	-1	1	1

additional run would still provide valuable information by removing certain models from contention. To generalize this approach, experimenters can adapt this method in a few ways. First, if more than two models are of interest, the MPD in Equation 2.12 can be replaced with an objective function to maximize the minimum distance between any pair of predicted values: $\max_{\forall x_i \in M}\{\min_{i \neq j} |\hat{y}_i - \hat{y}_j|\}$. And, if the experimenter can add more than one run, all additional runs can be chosen to minimize $\min_{\forall x_i \in S} \sum_{i<j} (\mathbf{x}_i' \mathbf{x}_j)^2$. A formalized augmentation strategy for model discrimination with supersaturated designs is an area for future research. For a more general augmentation strategy using the Bayesian D-optimality criterion, see Gutman et al. (2013c).

2.6 Conclusions

Supersaturated designs can be used in large screening experiments when the number of factors exceeds the number of available runs. However, there seems to be a general confusion about supersaturated designs from the practitioner's view (Gilmour, 2006). Our goal was to give a general overview of the designs for the practitioner's sake. We discussed some construction and analysis methods, outlined some of the issues facing experimenters and analysts who use them, and we also suggested an augmentation strategy to clarify discrepancies in the data analysis.

III. Supersaturated Designs: Analytical Challenges and New Analysis Methods

In the past twenty years, researchers have produced a multitude of new construction and analysis techniques to study supersaturated designs. Unfortunately, supersaturated designs can be a nebulous concept to practitioners. Analysis methods can be confusing and results from such experiments are not always clear. In this paper, we aim to make practitioners more comfortable with these designs by reviewing basic concepts and recent developments from a macro level. We discuss popular construction methods and, via explicit examples, highlight the challenges faced when analyzing data from a supersaturated experiment. Additionally, we present new, easy-to-use analysis methods and perform simulation studies on well-known supersaturated design matrices.

3.1 Introduction

> "We all live in a supersaturated world - there are *always* more variables than we can handle."

A supersaturated design is an experimental design with more factors than runs. The opening quote by statistician Dennis Lin (1995b) candidly suggests that, in actuality, all experiments have more factors than runs because any number of variables may influence a system's response. Prior to running a formal experiment, a subject matter expert must sift through the many possible control factors and remove those not expected to affect the system. In many cases, the remaining k factors can then be placed in a standard screening design with $n > k$ runs to find which are truly active. When this is not possible, experimenters can use supersaturated designs.

Like traditional screening methods, e.g. Plackett-Burman designs (1946) or Resolution III and IV fractions, the first supersaturated designs were constructed to be balanced and two-level. They date back to 1959 when Satterthwaite introduced *random balanced designs*, which randomly assigned balanced columns to a design matrix with more factors than runs. Soon after, Booth and Cox (1962) created the first systematic supersaturated designs. Statisticians, however, did not consider the designs well-suited for experiments. If a matrix has more factors than runs, unbiased estimates of main effects are impossible, and the tradeoff between efficient run-size and biased estimates was deemed to great. Consequently, research on the subject was dormant for more than 30 years until Lin (1993) introduced some new designs and rekindled interest in the field.

Recently, supersaturated designs have become more sophisticated than their two-level roots. Jones et al. (2008) introduced custom computer-generated designs, Sun et al. (2011) discussed optimal mixed-level designs, and Liu and Liu (2011) created designs with a large number of levels. As the construction methods became more adaptive, analysis techniques became more complex. Examples include a contrast-based method (Holcomb et al., 2003), a staged dimensionality reduction (Lu and Wu, 2004), linear-programming via the Dantzig selector (Phoa et al., 2009), and a cluster analysis strategy (Li et al., 2010). Of course, this is not an exhaustive list of either construction or analysis methods. (See Georgiou (2012) for a detailed review.) Researchers continue to do innovative work in both areas, but practitioners are still hesitant to use supersaturated designs (Gilmour, 2006). In our view, there are two reasons for this: 1) the biased main-effect estimates are too worrisome, and 2) supersaturated designs are becoming esoteric because most research is done at a theoretical level. Consequently, practitioners do not fully understand the pros and cons of supersaturated designs or how to use them. Here, we focus on the designs

from a higher level for the practitioner's sake. Our objective is for experimenters to appreciate the value of supersaturated designs but also understood the limitations and risks involved when using them for screening experiments.

3.2 Supersaturated Designs

The traditional definition for a supersaturated design is a fractional factorial design in which the number of factors, k, is larger than $n-1$, where n is the number of runs. While this definition is accurate, it is also limiting. The term *supersaturated* is in reference to the insufficient degrees of freedom needed to estimate all effects, not necessarily just main effects. As Bradley Jones, Principal Research Fellow for JMP, said, "...in some sense, every experiment is a supersaturated design," because if we consider all possible terms in the empirical model - i.e. all two-way interactions, quadratic terms, etc. - the number of terms would be greater than the number of runs available to estimate them (Jones, 2011). For example, a Plackett-Burman design is saturated in its main effects, but if we consider two-way interactions, it becomes supersaturated. The new three-level definitive screening designs by Jones and Nachtsheim (2011) are also supersaturated if we consider main effects, two-factor interactions, and quadratic terms. For this paper, we focus on traditional *main-effect supersaturated designs*, in which the number of independent factors, k, is greater than $n-1$, as opposed to *model-effect supersaturated designs*, where $k < n$, but the number of model terms, p, is greater than $n-1$. If $p = n$, the design is saturated.

Regardless of the type of supersaturated design, main-effect or otherwise, they are inherently difficult to analyze. The crux of the analysis problem is that the model matrix $\mathbf{X}$ has more columns than rows. Therefore, it is rank deficient and unique parameter estimates for the hypothesized model, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, do not exist. Fortunately, researchers have observed that changes in a response variable are usually

caused by a small number of factors, a concept referred to as "effect sparsity" (Box and Meyer, 1986). An analogous concept is the Pareto Principle, or the "law of the vital few." As such, the goal of analyzing a supersaturated design is to separate the few active factors from the many inactive. This is easier said than done. Since main effects are confounded with each other, it can be difficult to find the true important factors. In addition, inactive factors can appear to have an effect and active factors may be hidden by noise.

Despite their problems, supersaturated designs are preferred over naive screening approaches like subjective opinion about which variables are important. Budget constraints sometimes necessitate such designs and researchers have found value in their use. Holcomb et al. (2007) used a supersaturated design in a study of turbine engine development; Rais et al. (2009) analyzed active factors in the preparation of sulfated amides of olive pomace oil fatty acids; Matsuura et al. (2010) applied supersaturated designs to robust parameter design; and Scinto et al. (2011) used a supersaturated design to screen factors for gasoline-powered engine fuel economy.

To familiarize experimenters with supersaturated designs, we proceed with the following outline: In the next section, we briefly review how supersaturated designs are constructed. In §4, we highlight common pitfalls to be aware of when searching for the few active factors. Then we review different ways to analyze the designs and present two straightforward analysis methods. In §5, we take a detailed look at the popular Williams (1968) data set and correct an oversight in simulation studies using the design matrix. We also perform a simulation study on the 138 factor, 24 run matrix from Lin (1995a). We conclude with some practical guidelines and considerations.

Table 7. Supersaturated Design Example with 8 Runs and 14 Factors

	Design Factors													
Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2	−	+	−	−	−	+	+	−	−	−	+	−	+	+
3	+	−	+	−	−	−	+	+	−	−	−	+	−	+
4	+	+	−	+	−	−	−	+	+	−	−	−	+	−
5	−	+	+	−	+	−	−	−	+	+	−	−	−	+
6	−	−	+	+	−	+	−	+	−	+	+	−	−	−
7	−	−	−	+	+	−	+	−	+	−	+	+	−	−
8	+	−	−	−	+	+	−	−	−	+	−	+	+	−

3.3 Choosing a Supersaturated Design

Booth and Cox (1962) proposed a systematic construction of balanced designs and created the first supersaturated designs to be "as nearly orthogonal as possible." Their measurement of near-orthogonality was $E(s^2)$, and it became the standard criterion for all balanced two-level supersaturated designs. It is helpful to explain the criterion with an example. Consider the two-level supersaturated design in Table 7, which appeared in Holcomb and Carlyle (2002), and the main-effects model,

$$\mathbf{y} = \beta_0\mathbf{1} + \beta_1\mathbf{x}_1 + \cdots + \beta_{14}\mathbf{x}_{14} + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.1}$$

where $\mathbf{y}$ is the response vector, $\beta_0 \ldots \beta_{14}$ are the unknown model parameters, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_8)$ is the error term. In order calculate the Ordinary Least Squares equation, $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, it is necessary to find $(\mathbf{X}'\mathbf{X})^{-1}$. As mentioned, $\mathbf{X}$ is rank deficient, so $\mathbf{X}'\mathbf{X}$ is singular. However, the structure of $\mathbf{X}'\mathbf{X}$ contains useful information and is key to characterizing a supersaturated design. The design in Table 7 has $\mathbf{X}'\mathbf{X}$ equal to:

$$\mathbf{X}'\mathbf{X} = \begin{array}{c} \begin{array}{cccccccccccccc} \mathbf{1} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 & \mathbf{x}_8 & \mathbf{x}_9 & \mathbf{x}_{10} & \mathbf{x}_{11} & \mathbf{x}_{12} & \mathbf{x}_{13} & \mathbf{x}_{14} \end{array} \\ \begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & -4 & 4 \\ 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 & 4 & 4 & 0 & 4 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & -4 & 4 & 4 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & -4 & 4 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & -4 & 4 & 4 & 0 & 4 \\ 0 & 4 & 0 & 4 & 4 & -4 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 4 & 4 & -4 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 & 4 & -4 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 4 & 0 & 4 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 4 & -4 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{pmatrix} \end{array} \tag{3.2}$$

In an orthogonal design, all off-diagonal elements are 0, signifying each main effect is independent of all others. For a supersaturated design, this is not possible. However, the idea of the $E(s^2)$ criterion is to make the off-diagonal elements, on average, as close to 0 as possible, effectively making the design "nearly orthogonal." If we denote the (i, j)th element of $\mathbf{X}'\mathbf{X}$ as s_{ij}, then we can define $E(s^2)$ as

$$E(s^2) = \sum_{i<j} s_{ij}^2/(k(k-1)/2), \tag{3.3}$$

where k is the number of factors in the model. Because $E(s^2) > 0$ whenever the number of factors exceeds the number of runs, it is helpful to have lower bounds on $E(s^2)$ for a given design size to know whether or not a design is $E(s^2)$-optimal. See Nguyen (1996), Tang and Wu (1997), Bulutoglu and Cheng (2004), Ryan and Bulutoglu (2007), and Das et al. (2008) for more information on $E(s^2)$ lower bounds and $E(s^2)$-optimal designs.

Another popular criterion to construct and compare supersaturated designs is Bayesian D-Optimality. This criterion can be used on regular designs, as in Du-Mouchel and Jones (1994), but Jones et al. (2008) adapted the technique to create

supersaturated designs. This criterion and its construction method are more versatile than $E(s^2)$ because Bayesian D-Optimal designs can be any size with any number of blocks and can incorporate categorical variables. They can also handle interactions. We omit the technical details for brevity, but note their construction is implemented in the JMP statistical software, making it easy for practitioners to create a design. An overview of other design criteria and construction methods can be found in Gilmour (2006), but $E(s^2)$ and Bayesian D-Optimal designs are the most prevalent. Both are also easy to construct in software or find online. To make a Bayesian D-Optimal design in JMP, start the DOE Custom Design, change the "Estimability" of potential model effects from "Necessary" to "If Possible," and specify a desired number of runs. A catalog of various other designs, like $E(s^2)$-optimal, is available online at:

`http://www.iasri.res.in/design/Supersaturated_Design/SSD/Supersaturated.html`

For experimenters, this means it is relatively easy to find a suitable design matrix. Thus, the question "How should supersaturated designs be constructed?" is more thoroughly answered than "How should supersaturated designs be analyzed?". Researchers have optimality criteria they are trying to reach and construct designs to meet it. Moreover, Marley and Woods (2010) found little difference in the screening performance of $E(s^2)$ and Bayesian D-Optimal Designs. The challenging part, regardless of construction, is analyzing the results.

3.4 Analyzing a Supersaturated Design

When analyzing a main-effect supersaturated design, we assume the underlying model is linear with only main effects:

$$\mathbf{y} = \beta_0\mathbf{1} + \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{3.4}$$

where $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{I}_n)$ and most of the β_i's are negligible by effect sparsity. Because the model is assumed linear, it seems natural to analyze experimental data with traditional subset selection methods from linear regression. Indeed, these were the first methods used to analyze supersaturated designs. Wu (1993) used forward regression to find active factors, while Lin (1993) suggested stepwise regression and later used normal (or half-normal) plots in addition to stepwise regression (1995b). Abraham et al. (1999) advocated all-subsets regression when possible because they found all-subsets outperformed stepwise regression. Yet, they "urged caution" in using supersaturated designs because analytical recommendations are not always clear. Even if the matrix is "optimal" with respect to some design criterion, the singularity of $\mathbf{X}'\mathbf{X}$ makes supersaturated designs tricky to analyze with any method. The few active and many inactive factors are tangled together by a complex aliasing structure, which is quantified in $\mathbf{X}'\mathbf{X}$. For example, scaling $\mathbf{X}'\mathbf{X}$ matrix in Equation 3.2 by 1/8 gives us the complete aliasing structure of the design in Table 7.

$$\frac{1}{8}\mathbf{X}'\mathbf{X} = \begin{array}{c} \begin{array}{ccccccccccccccc} \mathbf{1} & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & \beta_8 & \beta_9 & \beta_{10} & \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & .5 & 0 & 0 & -.5 & .5 & .5 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & .5 & 0 & 0 & -.5 & .5 & .5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & .5 & 0 & .5 & 0 & 0 & -.5 & .5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & .5 & .5 & 0 & .5 & 0 & 0 & -.5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -.5 & .5 & .5 & 0 & .5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -.5 & .5 & .5 & 0 & .5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -.5 & .5 & .5 & 0 & .5 \\ 0 & .5 & 0 & .5 & .5 & -.5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .5 & 0 & .5 & .5 & -.5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .5 & 0 & .5 & .5 & -.5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -.5 & 0 & 0 & .5 & 0 & .5 & .5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & .5 & -.5 & 0 & 0 & .5 & 0 & .5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & .5 & .5 & -.5 & 0 & 0 & .5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & .5 & .5 & -.5 & 0 & 0 & .5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \tag{3.5}$$

The rows (or columns) of $\frac{1}{8}\mathbf{X}'\mathbf{X}$ in Equation 3.5 reveal that every factor estimate is biased by four other factors, each with a ± 0.5 correlation. For instance, the estimate of factor 1 is correlated with factors 8, 11, 12, and 13. For a generic supersaturated

design, the rows of $\frac{1}{n}\mathbf{X}'\mathbf{X}$ give the biased estimates for each main effect,

$$E(\hat{\beta}_i) = \beta_i + \sum_{j \neq i} \rho_{ij}\beta_j, \tag{3.6}$$

where ρ_{ij} is the correlation between factors $\mathbf{x}_i$ and $\mathbf{x}_j$. We refer to a factor's biased estimate equation as its *aliasing chain*. Although the aliasing chains complicate analysis, basic subset selection methods are still used for their simplicity and familiarity to experimenters, and most are available in basic statistical software programs.

3.4.1 Inherent Difficulties.

In this section, we show how common analysis methods can fail when analyzing data from a supersaturated design. We highlight four specific causes of analysis difficulties:

1. Inactive factors are inflated by active factors.
2. Active factors are hidden by noise and the aliasing structure.
3. Many models explain the data well.
4. The assumption of effect sparsity does not hold.

For each of the four analytical challenges, we will construct an example model and analyze the responses with tradition regression methods to highlight how the methods can fail. For consistency, we refer to the 14 factor, 8 run $E(s^2)$-optimal design shown in Table 7.

First, some preliminaries. There are different thoughts on how many active factors a supersaturated design can detect; Holcomb et al. (2003) suggest a maximum of $n/2$, while Marley and Woods (2010) suggest $n/3$. In either case, notice how the ratio of active factors is dictated by the number of runs, n, and not by the total number of

factors, k. In many supersaturated designs, the factor-to-run ratio is so large that if we searched a fraction of k, there is a possibility we would be searching for more active factors than runs. If this was the case, many different models would fit the data perfectly. For example, consider the design in Table 7, and suppose we generated an arbitrary response, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. If we fit a model with factors $\mathbf{x}_1, \ldots, \mathbf{x}_7$ and the intercept, we create an invertible model matrix $\mathbf{X}^*$. This will generate an estimate, $\boldsymbol{\beta}^*$, that will fit the data exactly because $\mathbf{y} = \mathbf{X}^*\boldsymbol{\beta}^*$ implies $\boldsymbol{\beta}^* = \mathbf{X}^{*-1}\mathbf{y}$. This is true of any supersaturated design if there exists an invertible submatrix made from $n-1$ factor columns of the original design matrix. Hence, the number of active factors is limited to a fraction of n and not k. For our examples, we will search for up to $n/2 = 4$ active factors. Also, forward, stepwise, and all-subsets regression require user-input that can affect model selection. While many criteria are used to build and compare models, we will focus on R^2, R^2_{adj}, and the corrected Akaike's Information Criterion, $AICc$. We analyzed the data using the following options in the statistical software JMP:

- JMP's Half Normal Plot (v9). In a Half Normal plot, effects that look like random noise will roughly fall in a straight line. Any effect that considerably deviates from the line is identified as active.
- Forward regression in 4 steps with the minimum $AICc$ stopping rule.
- Stepwise regression (aka "Mixed" direction in JMP) with p-value to enter and p-value to leave set to 0.1.
- All-Subsets regression (aka "All Possible Models" in JMP) for up to $n/2 = 4$ factors.

It is also worthwhile to define a *contrast*. For balanced two-level designs, a factor contrast is the sum of the responses when a factor is at its high level (+1), minus the

sum of the responses when the factor is at its low level (-1). More simply, a contrast vector, $\mathbf{C}$, can be calculated as $\mathbf{C} = \mathbf{X}'\mathbf{y}$ (Holcomb et al., 2003). In an orthogonal design, the contrast vector clearly identifies the effects of each factor, without bias. In a supersaturated design, there is danger in analyzing just the contrast vector. Ideally, an active factor will have a large contrast to signify the response variable changed substantially when the factor changed, but this is not always the case.

3.4.1.1 Example 1: Inflated Inactive Factors.

Table 8. Supersaturated Design with Inflated Inactive Factor

	Design Factors with $\mathbf{y} = 3\mathbf{x}_1 + 15\mathbf{x}_8 + 10\mathbf{x}_9 + 20\mathbf{x}_{11} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$														
Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{y}$
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	47.232
2	−	+	−	−	−	+	+	−	−	−	+	−	+	+	-7.432
3	+	−	+	−	−	−	+	+	−	−	−	+	−	+	-13.096
4	+	+	−	+	−	−	−	+	+	−	−	−	+	−	8.690
5	−	+	+	−	+	−	−	−	+	+	−	−	−	+	-27.867
6	−	−	+	+	−	+	−	+	−	+	+	−	−	−	22.397
7	−	−	−	+	+	−	+	−	+	−	+	+	−	−	11.387
8	+	−	−	−	+	+	−	−	−	+	−	+	+	−	-40.700
C	3.6	40.6	56.7	178.8	-20.5	42.4	75.6	129.8	78.3	1.5	146.6	9.0	15.0	-2.9	

A common source for error when analyzing a supersaturated design is the presence of a false effect - an inactive factor whose parameter estimate is inflated because it's aliased with active factors. Consider the design in Table 8 with responses $\mathbf{y} = 3\mathbf{x}_1 + 15\mathbf{x}_8 + 10\mathbf{x}_9 + 20\mathbf{x}_{11} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$. Notice that factors $\mathbf{x}_4$, $\mathbf{x}_8$, and $\mathbf{x}_{11}$ have the largest contrasts in Table 8, yet the Half Normal Plot in Figure 1 suggests $\mathbf{x}_4$, $\mathbf{x}_9$, and $\mathbf{x}_{14}$ are active. The initial analysis is already unclear because the two methods do not coincide and, in both cases, an inactive factor, $\mathbf{x}_4$ appears to have the largest impact on the response. Investigating further, we fit the first four factors with forward regression to get the results in Table 9.

Factors $\mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_9$, and $\mathbf{x}_{14}$ were selected as "active", but only $\mathbf{x}_9$ is truly active. The rest are false positives with inflated effects due to the aliasing structure in the design. In Step 1, $\mathbf{x}_4$ was flagged as active, and if we examine its aliasing chain, we

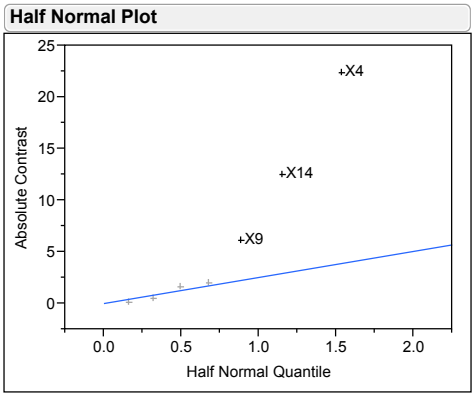

Figure 1. Half Normal Plot of Main-Effect Estimates for the Design in Table 8

Table 9. Forward Regression Results on Table 8 with Parameter Estimates

Step	β_4	β_{14}	β_9	β_6	R^2	R^2_{adj}
1	22.35				0.714	0.666
2	29.56	14.41			0.937	0.911
3	34.55	19.91	-7.49		0.990	0.983
4	33.31	16.29	-5.63	2.48	0.996	0.990

can see why this is true. From Equations 3.5 and 3.6, the estimate for β_4 is really

$$E(\hat{\beta}_4) = \beta_4 + \frac{1}{2}\beta_8 + \frac{1}{2}\beta_9 + \frac{1}{2}\beta_{11} - \frac{1}{2}\beta_{14}. \tag{3.7}$$

$\hat{\beta}_4$ is large because it is pulling information from three *real* effects, $\beta_8 = 15, \beta_9 = 10$, and $\beta_{11} = 20$. Their combined effect in the aliasing chain creates a false effect in $\hat{\beta}_4$ greater than any one of the true parameter estimates, including β_{11} which is 20 times larger than the noise level. Notice in Step 1, $\hat{\beta}_4 = 22.35 \approx \frac{1}{2}15 + \frac{1}{2}9 + \frac{1}{2}20$ (noise causes the difference). Moreover, once a false effect is flagged, it is less likely for the true active factors in its aliasing chain to register because the active factors essentially lost their information to $\hat{\beta}_4$. Then, in Step 2, the only inactive factor in $\mathbf{x}_4$'s aliasing chain, $\mathbf{x}_{14}$, was put into the model to counteract the false explanatory power of β_4. Analyzing the data with stepwise regression yielded similar results, as $\mathbf{x}_4, \mathbf{x}_9$, and $\mathbf{x}_{14}$ were flagged as active.

The position of the active factors in $\mathbf{x}_4$'s aliasing chain is to blame here. $\mathbf{x}_4$'s aliasing chain contains three active factors, and with correlations of ± 0.5, three active factors are guaranteed to inflate $\hat{\beta}_4$ with a false effect larger than at least one true effect. Without loss of generality, suppose $\mathbf{x}_8, \mathbf{x}_9$, and $\mathbf{x}_{14}$ are active with positive effects, and $\min\{\beta_8, \beta_9, \beta_{14}\} = \beta_8$. Then, $E(\hat{\beta}_4) = \beta_4 + \frac{1}{2}\beta_8 + \frac{1}{2}\beta_9 + \frac{1}{2}\beta_{11} - \frac{1}{2}\beta_{14} \geq \frac{1}{2}\beta_8 + \frac{1}{2}\beta_8 + \frac{1}{2}\beta_8 = \frac{3}{2}\beta_8 > \beta_8$. This can be problematic for sequential procedures, especially when the false effect is greater than all effects in the aliasing chain.

Table 10. All-Subsets Results on Table 8

Number	Model Terms	R^2	$AICc$
1	$\mathbf{x}_4$	0.7139	77.0955
2	$\mathbf{x}_4, \mathbf{x}_{14}$	0.9365	74.3824
3	$\mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{11}$	0.9929	75.4935
4	$\mathbf{x}_1, \mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{11}$	0.9994	112.237

All-subsets regression may be a viable option to avoid this pitfall. We performed all-subsets using JMP and searched for the best models with up to 4 factors. The results are shown in Table 10; notice how the best four-factor model correctly identified all active factors. However, also notice the best two-factor model has two false effects with a reasonably high R^2 value. One might expect the factors appearing in the best two-factor model to appear in the best three or four-factor model, but the aliasing structure of supersaturated designs can prevent this. Fortunately, because supersaturated designs are screening experiments, the main goal is to select a subset of factors for follow-up runs. In this case, a practitioner could easily carry the six factors identified from all-subsets into the next phase of testing, thereby reducing the number of factors from 14 to 6, which is an effective screen.

However, it is important to highlight an overriding issue with these methods, and that is the apparent inability of R^2 or R^2_{adj} to differentiate models. The $AICc$ criterion seems to do a better job, but only at comparing models of the same size. For instance, review the R^2 and R^2_{adj} values from Example 1. The forward regression

results in Table 9 show $R^2 = 0.996$ and $R^2_{adj} = 0.990$ *for the wrong model*. The correct model, found via all-subsets, had slightly higher values: $R^2 = 0.999$ and $R^2_{adj} = 0.999$. Unfortunately, an experimenter would most likely never catch the mistake because the R^2 and R^2_{adj} values for the wrong model are close to one. While the models have similar R^2 and R^2_{adj} values, the correct model has a noticeably smaller $AICc$ value: 112.2372 compared to 127.8558. Be careful, though, not to choose designs based on $AICc$ alone. Lower $AICc$ values are ideal, but it is best to compare the $AICc$ values of models with the same number of variables. Otherwise, you might underfit the model. In Table 10, the correct model has the highest $AICc$ value when compared to the smaller models, but it has the lowest $AICc$ value of all models with four factors.

3.4.1.2 Example 2: Hidden Active Factors.

When an inactive factor appears to have the largest effect on the response, forward, stepwise regression, and Half-Normal plots can perform poorly. In this example, we will see how mistakes can still occur if an active factors does have the largest effect on the response. A complex aliasing structure can not only inflate the effect of an inactive factor, but it can also completely hide the true effect of an active factor, making it indistinguishable from noise.

Table 11. Supersaturated Design with Hidden Active Factor

	Design Factors with $\mathbf{y} = 9\mathbf{x}_3 + 10\mathbf{x}_4 - 20\mathbf{x}_9 + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$														
Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{y}$
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-0.110
2	−	+	−	−	−	+	+	−	−	−	+	−	+	+	0.264
3	+	−	+	−	−	−	+	+	−	−	−	+	−	+	19.447
4	+	+	−	+	−	−	−	+	+	−	−	−	+	−	-18.966
5	−	+	+	−	+	−	−	−	+	+	−	−	−	+	-20.654
6	−	−	+	+	−	+	−	+	−	+	+	−	−	−	38.108
7	−	−	−	+	+	−	+	−	+	−	+	+	−	−	-20.293
8	+	−	−	−	+	+	−	−	−	+	−	+	+	−	-0.354
C	2.6	-76.4	76.1	0.0	-80.3	78.4	1.2	79.5	-117.5	36.5	38.5	-0.1	-35.8	0.5	

Let's revisit the 8 factor, 14 run design but with responses generated with the equation $\mathbf{y} = 9\mathbf{x}_3 + 10\mathbf{x}_4 - 20\mathbf{x}_9 + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$. The design, responses, and

contrasts are in Table 11. First, notice that the largest contrast, in absolute value, belongs to an active factor, $\mathbf{x}_9$, which is ideal. The Half-Normal Plot in Figure 2 also detected $\mathbf{x}_9$. However, the contrast of $\mathbf{x}_4$ is 0.0 even though it has a larger effect than $\mathbf{x}_3$. So, superficially, $\mathbf{x}_4$ has no effect because there is negligible differences in the response variables as $\mathbf{x}_4$ changes from its low level to high level.

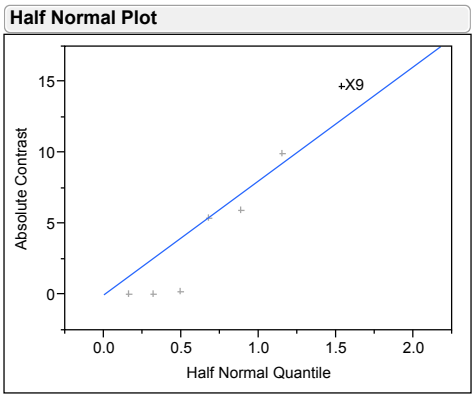

Figure 2. Half Normal Plot of Main-Effect Estimates for the Design in Table 11

Here, an active factor, $\mathbf{x}_4$ was cancelled out because it is aliased with another active factor, $\mathbf{x}_9$, having an opposite effect. The aliasing chain for $\mathbf{x}_4$ in Equation 3.7 reveals that the large positive value of β_9 effectively hides $\hat{\beta}_4$ because $E(\hat{\beta}_4) = \beta_4 + \frac{1}{2}\beta_9 = 10 + \frac{1}{2}(-20) = 0$. Again, the contrasts and Half-Normal Plot do not give much useful information. Forward and stepwise regression, unfortunately, also fail. Table 12 shows the results with forward regression, and $\mathbf{x}_9$ was selected as active in the first step. In Step 2, however, $\mathbf{x}_8$ registered as active because $E(\hat{\beta}_8) = \beta_8 + \frac{1}{2}\beta_1 + \frac{1}{2}\beta_3 + \frac{1}{2}\beta_4 - \frac{1}{2}\beta_5$, and $\beta_3 = 9$ and $\beta_4 = 10$ are inflating its estimate. Notice $\hat{\beta}_8 = 9.94 \approx \frac{1}{2}\beta_3 + \frac{1}{2}\beta_4 = \frac{1}{2}9 + \frac{1}{2}10 = 9.5$. Forward regression does not recover, and continues to select false effects. Stepwise only selected factors $\mathbf{x}_9$ and $\mathbf{x}_8$. Ideally, once factor $\mathbf{x}_9$ is chosen and the sum of squares are adjusted for the most dominant factor, the previously suppressed variable $\mathbf{x}_4$ would show its effect. However, the presence of noise and the aliasing chains prevent this.

Table 12. Forward Regression Results on Table 11 with Parameter Estimates

Step	β_9	β_8	β_1	β_5	R^2	R^2_{adj}
1	-14.68				0.570	0.498
2	-14.68	9.94			0.831	0.763
3	-14.68	13.04	-6.19		0.907	0.837
4	-19.28	19.16	-9.26	9.18	1.000	1.000

All-subsets regression in Table 13, on the other hand, correctly identifies the correct three-factor model with $\mathbf{x}_3$, $\mathbf{x}_4$, and $\mathbf{x}_9$. The best four-factor model though does not contain any of these factors. Recommendations are difficult in this case because different models seemingly explain the data well, but they do not share any factors in common. The experimenter would have to choose which factors to study in follow-up experiments. $\mathbf{x}_9$ is on obvious choice, but seven other factors appeared at some point in the analysis. A conservative approach would be eliminating the six factors that never appeared in any of methods.

Table 13. All-Subsets Results on Table 11

Number	Model Terms	R^2	$AICc$
1	$\mathbf{x}_9$	0.5699	75.4434
2	$\mathbf{x}_8, \mathbf{x}_9$	0.8309	77.3087
3	$\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_9$	0.9994	51.4485
4	$\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_{13}$	0.9999	88.1677

3.4.1.3 Example 3: Model Indiscrimination.

Supersaturated designs cannot always discriminate models, even if the design is $E(s^2)$-optimal and the model is sparse. For this example, we used our 14 factor, 8 run $E(s^2)$-optimal design and generated responses with the equation $\mathbf{y} = -20\mathbf{x}_1 + 20\mathbf{x}_5 + 20\mathbf{x}_8 + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$. First, notice the pattern in the contrast vector $\mathbf{C}$ in Table 14. All contrasts are either close to ± 80 or 0. This confuses the Half-Normal Plot in Figure 2, so no factors appear important. Stepwise also failed to identify any

active factors. This is peculiar because the effects are significantly larger than the noise.

Table 14. Model Indiscriminate Supersaturated Design

	Design Factors with $\mathbf{y} = -20\mathbf{x}_1 + 20\mathbf{x}_5 + 20\mathbf{x}_8 + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$														
Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{y}$
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	18.629
2	−	+	−	−	−	+	+	−	−	−	+	−	+	+	-20.870
3	+	−	+	−	−	−	+	+	−	−	−	+	−	+	-20.720
4	+	+	−	+	−	−	−	+	+	−	−	−	+	−	-21.328
5	−	+	+	−	+	−	−	−	+	+	−	−	−	+	20.834
6	−	−	+	+	−	+	−	+	−	+	+	−	−	−	19.929
7	−	−	−	+	+	−	+	−	+	−	+	+	−	−	20.482
8	+	−	−	−	+	+	−	−	−	+	−	+	+	−	-20.420
C	-84.2	-2.0	80.8	78.9	82.5	-2.0	-1.5	-3.5	80.7	81.4	79.8	-0.6	-84.5	-0.8	

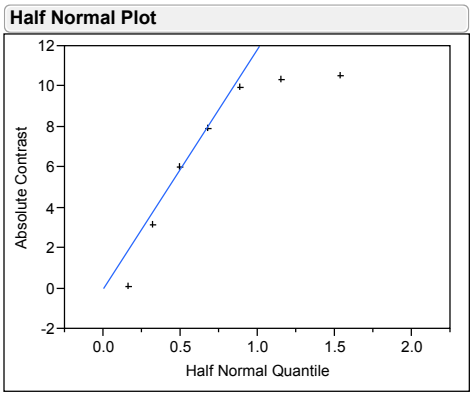

Figure 3. Half Normal Plot of Main-Effect Estimates for the Design in Table 14

The results for forward regression are in Table 15 and only one true active factor was detected, $\mathbf{x}_5$. Factor $\mathbf{x}_{13}$ was chosen first because it had the largest contrast in absolute value. The remaining factors, $\mathbf{x}_{11}$ and $\mathbf{x}_{12}$ were selected because they are aliased with real effects. Also note that the final model has a low R^2_{adj} compared to the high values we have seen. Results from all-subsets regression in Table 16 are also troublesome. The best three-factor model identifies $\mathbf{x}_3$, $\mathbf{x}_4$, and $\mathbf{x}_8$ as active factors. Although the four-factor model identifies the true factors, we expect with only three nonzero parameters, all-subsets regression would identify the correct three-factor model. Investigation showed the second best three-factor model was the true underlying model with $\mathbf{x}_1$, $\mathbf{x}_5$, and $\mathbf{x}_9$. Additionally, with and $R^2 = 0.9995$ and

$AICc = 49.3771$, the two best models are essentially identical. To examine this further, suppose now the response vector is noiseless (i.e. $\mathbf{y} = -20\mathbf{x}_1 + 20\mathbf{x}_5 + 20\mathbf{x}_8$). So,

$$\mathbf{y}' = (20 \quad -20 \quad -20 \quad -20 \quad 20 \quad 20 \quad 20 \quad -20).$$

Table 15. Forward Regression Results on Table 14 with Parameter Estimates

Step	β_{13}	β_5	β_{11}	β_{12}	R^2	R^2_{adj}
1	-10.56				0.268	0.156
2	-10.56	10.31			0.523	0.333
3	-10.56	10.31	9.98		0.762	0.584
4	-10.56	13.80	9.98	-6.98	0.850	0.649

Table 16. All-Subsets Results on Table 14

Number	Model Terms	R^2	$AICc$
1	$\mathbf{x}_{13}$	0.2679	80.4661
2	$\mathbf{x}_5, \mathbf{x}_{13}$	0.5232	86.3683
3	$\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_8$	0.9996	49.1094
4	$\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_8, \mathbf{x}_{13}$	0.9999	91.8652

Using all-subsets regression, we can find two different models that fit the data *exactly*:

1. $\mathbf{y} = -20\mathbf{x}_1 + 20\mathbf{x}_5 + 20\mathbf{x}_8$

2. $\mathbf{y} = 20\mathbf{x}_3 + 20\mathbf{x}_4 - 20\mathbf{x}_8$

If we cannot differentiate between two models in the noiseless case, we certainly cannot expect to differentiate between them in the presence of noise. This occurs because the factors in the above models are linearly dependent. If we construct a matrix using the factor columns of the competing models, we get $\mathbf{X}^* = [\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_8]$, which has a rank of 4. $\mathbf{X}^*$ has 5 columns, so it is rank deficient and $\mathbf{X}^*\boldsymbol{\beta} = \mathbf{y}$ has multiple least squares solutions. In this case, $\hat{\boldsymbol{\beta}}_1{}' = (-20 \quad 0 \quad 0 \quad 20 \quad 20)$ and

$\hat{\boldsymbol{\beta}}_2{}' = (0 \quad 20 \quad 20 \quad 0 \quad -20)$. This phenomenon motivates the design criterion called resolution-rank from Deng et al. (1999), although a discussion of this criterion is beyond the scope of this paper. It is an interesting area for future research because in this case, no method can correctly find the active factors, although an experimenter could analyze the six terms in the competing models in later experiments. Also, in a real experiment, the presence of noise would likely make it impossible to detect if two models could generate the same noiseless response vector.

3.4.1.4 Example 4: Effect Sparsity Does Not Hold.

For our final example, we will examine what happens if the assumption of effect sparsity does not hold. Specifically, we generated responses with a six factor model, $\mathbf{y} = 15\mathbf{x}_1 + 8\mathbf{x}_3 + 10\mathbf{x}_5 + 14\mathbf{x}_9 + 12\mathbf{x}_{12} + 11\mathbf{x}_{14}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$. While this is less than half of the total number of factors, the number of active factors is greater the $n/2 = 4$ factors we are trying to identify. At best, an experimenter can hope to detect the most active factors and carry them into the next phase of experimentation.

Table 17. Supersaturated Design with Too Many Active Factors

	Design Factors with $\mathbf{y} = 15\mathbf{x}_1 + 8\mathbf{x}_3 + 10\mathbf{x}_5 + 14\mathbf{x}_9 + 12\mathbf{x}_{12} + 11\mathbf{x}_{14}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$														
Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{y}$
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	70.598
2	−	+	−	−	−	+	+	−	−	−	+	−	+	+	-48.215
3	+	−	+	−	−	−	+	+	−	−	−	+	−	+	22.313
4	+	+	−	+	−	−	−	+	+	−	−	−	+	−	-10.052
5	−	+	+	−	+	−	−	−	+	+	−	−	−	+	17.304
6	−	−	+	+	−	+	−	+	−	+	+	−	−	−	-53.190
7	−	−	−	+	+	−	+	−	+	−	+	+	−	−	1.700
8	+	−	−	−	+	+	−	−	−	+	−	+	+	−	4.040
C	169.3	54.8	109.6	13.6	182.8	-58.0	88.3	54.8	154.6	73.0	-62.7	192.8	28.2	119.5	

With six active factors, many models will likely explain the data well. Additionally, the design tangles the many real effects in such a way that the contrasts in Table 17 look like noise because all contrasts are relatively large compared to the other examples. The Half-Normal Plot in Figure 4 therefore fails to separate any active factors from inactive. In Table 18, forward regression chooses a real effect in Step 1, $\mathbf{x}_{12}$, but then falsely selects $\mathbf{x}_2$ in Step 2 because it is aliased with true factors $\mathbf{x}_9$,

$\mathbf{x}_{12}$, and $\mathbf{x}_{14}$. A correct factor is chosen in Step 3, but the final model from forward regression only identified 2 out of 6 real effects. Stepwise did not do any better and only chose factors $\mathbf{x}_{12}$, $\mathbf{x}_2$, and $\mathbf{x}_3$.

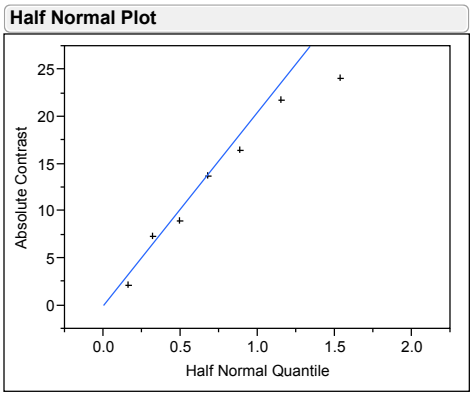

Figure 4. Half Normal Plot of Main-Effect Estimates for the Design in Table 17

Table 18. Forward Regression Results on Table 17 with Parameter Estimates

Step	β_{12}	β_2	β_3	β_7	R^2	R^2_{adj}
1	24.10				0.420	0.324
2	36.70	25.20			0.765	0.671
3	36.70	25.20	13.69		0.901	0.826
4	44.01	28.85	13.69	-10.97	0.959	0.904

All-subsets regression found similar results to forward and stepwise regression for models with 1-3 factors. The best four-factor model in Table 19 identified $\mathbf{x}_5, \mathbf{x}_7, \mathbf{x}_8$, and $\mathbf{x}_{11}$ as active factors, but only $\mathbf{x}_5$ has an effect. This example suggests that if effect sparsity does not hold, supersaturated designs are not a useful screening method.

Table 19. All-Subsets Results on Table 17

Number	Model Terms	R^2	$AICc$
1	$\mathbf{x}_{12}$	0.4204	88.1878
2	$\mathbf{x}_2, \mathbf{x}_{12}$	0.7650	90.2988
3	$\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_{12}$	0.9007	102.071
4	$\mathbf{x}_5, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_{11}$	0.9606	150.670

3.4.2 Summary of Difficulties.

The assumption of a linear model makes subset regression techniques a starting point for the analysis of supersaturated designs, but our examples show that these methods have multiple failure points. Granted, we chose specific models that would fail to warn what can happen. Nevertheless, the examples show how challenging analysis can be. They show it is risky to analyze factors individually, either by contrasts or visually in a Half-Normal Plot, because active effects can be hidden or inactive factors may appear important. Also, forward and stepwise regression are affected if a large false effect is selected, as in Example 1. Moreover, R^2 and R^2_{adj} do not differentiate competing models well, and no analysis method can resolve Example 3 in which two different parameter vectors generate the same response data. The challenges likely worsen with more noise in the responses.

Whatever method is used, the practitioner is often worried about Type I and Type II errors. Type I errors, also known as false positives, occur when an inactive factor is declared active. Type II errors, or false negatives, occur when a active factor is not found. In a screening experiment, it's more important to avoid Type II errors than Type I because future runs can further separate active and inactive factors, but they won't be able to detect a factor if it's not tested. We stress that, in many cases, Type I and Type II errors occur in tandem. In Example 1, a Type I error (selecting a false effect, $\mathbf{x}_4$) caused Type II errors because the active factors lost their effect estimates to a inactive factor. Thus, if the Type I error is reduced from the first step, we will likely decrease the risk of Type II error.

3.4.3 Overview of Analysis Methods.

Many researchers have investigated the supersaturated design analysis problem, and a number of novel analysis methods, both Bayesian and frequentist, have been

introduced in the literature. Westfall et al. (1998) proposed a resampling procedure and adjusted p-values with forward regression to control Type I errors, Beattie et al. (2002) applied a two-stage Bayesian analysis, Li and Lin (2002) used an iterative ridge regression to identify active factors based on a penalized least squares, Yamada (2004) analyzed stepwise Type II errors, and Zhang et al. (2006) introduced a partial least squares approach which combined elements of principle component analysis and canonical correlation with multiple regression. More recently, Georgiou (2008) combined singular value decomposition (SVD), principle component analysis, and regression to analyze supersaturated designs, Scinto et al. (2011) applied a Bayesian Variable Assessment to find active factors, and Edwards and Mee (2011) suggested a global model test to identity potential models. For a comprehensive review of analytical techniques with technical details, please see Gupta and Kohli (2008).

3.4.4 New Analysis Methods: Variants of Forward Regression.

As shown in our examples, standard regression techniques generally do not work well for the analysis of supersaturated designs, and techniques that work better are difficult for practitioners to implement. Many can be hard to understand and program because they require tuning parameters or knowledge of multivariate analysis methods. Finding an intermediary analysis method is desirable for practitioners. Our goal in this section is to develop useful analysis methods using convenient, familiar regression techniques that have nice properties. Lu and Wu (2004) had the same goal, although their staged-dimensionality reduction was geared towards designs constructed from an orthogonal base. The methods we propose can be applied to any design.

Practitioners tend to rely on familiar regression techniques, and while all-subsets regression is the benchmark technique for supersaturated designs, it is computation-

ally expensive. We need a faster way to search the model space for supersaturated designs, particulary for large designs. Standard forward regression is an option, but as we've seen, the presence of a false effect causes forward regression to deviate from the correct model. For example, in Example 1, the inflated inactive factor $\mathbf{x}_4$ was flagged as active and forward regression could not correct itself. Had we known, *a priori*, that $\hat{\beta}_4$ was a false positive, we could have omitted it from the model and used forward regression to find the correct factors $\mathbf{x}_{11}, \mathbf{x}_9, \mathbf{x}_8$, and $\mathbf{x}_1$ in succession. But without prior knowledge about the design, we would not know which factor to omit.

Therefore, we suggest fitting a model with forward regression in s steps to identify a set of potential model factors $\{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_s}\}$. Since any one of the s variables chosen may be a false effect, ignore each factor and restart forward regression on all other $k-1$ factors. One false positive can invalidate forward regression, and fitting a model without that factor can preclude this from happening. Removing a variable will make sure it is not falsely selected at any point and reduce the risk of Type I errors. We refer to the procedure as forward regression with omission, and it will perform forward regression $s+1$ total times. The experimenter can then compare the models based on the $AICc$ criterion and choose the best factors for additional experiments. The choice of s is up to user, though we suggest an integer between $n/3$ and $n/2$.

If the experimenter would rather automate the variable selection, we suggest the modified AIC ($mAIC$) from Phoa et al. (2009) as the stopping rule. The traditional AIC and $AICc$ tend to overfit supersaturated designs if automated. The $mAIC$ criterion, however, enforces a stiff penalty on model complexity. It is defined as

$$mAIC = n\log(RSS/n) + 2p^2, \tag{3.8}$$

where n is the number of runs, $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ is the residual sum of squares, and p is the number of terms in the model. Forward regression with omission would then be carried out on the number of factors selected with $mAIC$. When models are small, this method gives only a portion of the models found with all-subsets regression. For large designs, all-subsets is infeasible and forward regression with omission is a straightforward way to create many models using forward regression.

3.5 Simulation Studies

In this section, we compare the performance of forward regression with omission to other analysis methods in the literature. The two most popular supersaturated designs used for comparing analysis techniques are Lin's (1993) half-fraction of a 24 factor, 28 run design from Williams (1968), and Lin's (1995a) 138 factor, 24 run design which was used to study AIDS incidence rates.

3.5.1 Williams's Rubber Data.

The original Williams (1968) design was a Plackett-Burman type design that studied 24 factors of a rubber making process in 28 runs. The full design also appears in Box and Draper (1987). Lin (1993) used his half-fraction construction method to create a 28 factor, 14 run supersaturated design of the original matrix to see if he could draw the same conclusions as Williams in half the runs. The design, shown in Table 20, has since become the de facto supersaturated design to compare analysis methods. The matrix, however, contained an error; the columns for factors 13 and 16 were identical. Further, an extensive study of the original design by Sundberg (2008) revealed an outlier in the responses which likely caused misleading results when comparing methods on the raw data.

Table 20. Half Fraction of Williams's (1968) Data as reported in Lin (1993), without Factor 16

Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{x}_{17}$	$\mathbf{x}_{18}$	$\mathbf{x}_{19}$	$\mathbf{x}_{20}$	$\mathbf{x}_{21}$	$\mathbf{x}_{22}$	$\mathbf{x}_{23}$	$\mathbf{x}_{24}$	$\mathbf{y}$
1	+	+	+	−	−	−	+	+	+	+	+	−	+	−	−	+	−	−	+	−	−	−	+	133
2	+	−	−	−	−	−	+	+	+	−	−	−	+	+	+	−	+	−	−	+	+	−	−	62
3	+	+	−	+	+	−	−	−	−	+	−	+	+	+	+	+	−	−	−	−	+	+	−	45
4	+	+	−	+	−	+	−	−	−	+	+	−	+	−	+	−	+	+	+	−	−	−	−	52
5	−	−	+	+	+	+	−	+	+	−	−	−	+	−	+	+	−	−	+	−	+	+	+	56
6	−	−	+	+	+	+	+	−	+	+	+	−	−	+	+	+	+	+	+	+	+	−	−	47
7	−	−	−	−	+	−	−	+	−	+	−	+	+	+	−	+	+	+	+	+	−	−	+	88
8	−	+	+	−	−	+	−	+	−	+	−	−	−	−	−	−	−	+	−	+	+	+	−	193
9	−	−	−	−	−	+	+	−	−	−	+	+	−	−	+	+	+	−	−	−	−	+	+	32
10	+	+	+	+	−	+	+	+	−	−	−	+	−	+	+	+	−	+	−	+	−	−	+	53
11	−	+	−	+	+	−	−	+	+	−	+	−	−	+	−	−	+	+	−	−	−	+	+	276
12	+	−	−	−	+	+	+	−	+	+	+	+	+	−	−	−	−	+	−	+	+	+	+	145
13	+	+	+	+	+	−	+	−	+	−	−	+	−	−	−	−	+	−	+	+	−	+	−	130
14	−	−	+	−	−	−	−	−	−	−	+	+	−	+	−	−	−	−	+	−	+	−	−	127

Nevertheless, the design matrix has appeared in many simulation studies, whereby researchers chose a truth model, generated response data with random noise, and searched for the truth model with their proposed analysis method. Unfortunately, a problem occurred when different authors removed one of the duplicate columns in Lin's half fraction. To create the 23 factor, 14 run supersaturated design for the simulation studies, some authors removed factor 16 and kept the labeling of the other factors the same, as in Abraham et al. (1999), Li and Lin (2003), Lu and Wu (2004), Zhang et al. (2006), Li et al. (2010), Scinto et al. (2011). Others (Beattie et al. (2002) and Phoa et al. (2009)) removed factor 13 and relabeled the remaining factors. As a result, intended comparisons of some analysis techniques were inadvertently run on different models.

In the twelve models below, the * indicates cases where analysis methods were mistakenly compared. Cases 6, 8, and 9 were intended to be the same model: $\mathbf{y} = -15\mathbf{x}_1 + 12\mathbf{x}_5 - 8\mathbf{x}_9 + 6\mathbf{x}_{13} - 2\mathbf{x}_{17} + \boldsymbol{\epsilon}$. Beattie et al. (2002) originally proposed the model, and they applied it to the design matrix without factor 13 and all other factors relabeled. Li and Lin (2003) intended to study the same model, but they applied it to the design matrix without factor 16 and no relabeling. Therefore, $\mathbf{x}_{17}$ in Beattie et al.

was $\mathbf{x}_{18}$ in Li and Lin. In Case 9, Lu and Wu (2004) compared their analysis method to Beattie et al. but listed the coefficient of $\mathbf{x}_{17}$ as -12 instead of -2. Here, we correct the models to allow direct comparisons with our methods. All models are written in terms of the matrix in Table 20 with factor 16 deleted. The remaining factor labels are not changed. The simulation models below are ordered chronologically by when they appeared in the literature. For all models, $\boldsymbol{\epsilon} \sim N(0, \mathbf{I}_n)$.

- Case 1: $\mathbf{y} = 5\mathbf{x}_2 + 10\mathbf{x}_7 + 20\mathbf{x}_{13} + \boldsymbol{\epsilon}$
- Case 2: $\mathbf{y} = 14\mathbf{x}_2 + 20\mathbf{x}_7 + 20\mathbf{x}_{13} + \boldsymbol{\epsilon}$
- Case 3: $\mathbf{y} = 20\mathbf{x}_2 + 20\mathbf{x}_7 + 20\mathbf{x}_{13} + \boldsymbol{\epsilon}$
- Case 4: $\mathbf{y} = 10\mathbf{x}_1 + \boldsymbol{\epsilon}$
- Case 5: $\mathbf{y} = -15\mathbf{x}_1 + 8\mathbf{x}_5 - 6\mathbf{x}_9 + 3\mathbf{x}_5\mathbf{x}_9 + \boldsymbol{\epsilon}$
- Case 6*: $\mathbf{y} = -15\mathbf{x}_1 + 12\mathbf{x}_5 - 8\mathbf{x}_9 + 6\mathbf{x}_{14} - 2\mathbf{x}_{18} + \boldsymbol{\epsilon}$
- Case 7: $\mathbf{y} = -15\mathbf{x}_1 + 8\mathbf{x}_5 - 2\mathbf{x}_9 + \boldsymbol{\epsilon}$
- Case 8*: $\mathbf{y} = -15\mathbf{x}_1 + 12\mathbf{x}_5 - 8\mathbf{x}_9 + 6\mathbf{x}_{13} - 2\mathbf{x}_{17} + \boldsymbol{\epsilon}$
- Case 9*: $\mathbf{y} = -15\mathbf{x}_1 + 12\mathbf{x}_5 - 8\mathbf{x}_9 + 6\mathbf{x}_{13} - 12\mathbf{x}_{17} + \boldsymbol{\epsilon}$
- Case 10: $\mathbf{y} = -15\mathbf{x}_2 + 12\mathbf{x}_5 - 8\mathbf{x}_{13} + 6\mathbf{x}_{14} - 2\mathbf{x}_{17} + \boldsymbol{\epsilon}$
- Case 11: $\mathbf{y} = -15\mathbf{x}_2 + 8\mathbf{x}_5 - 6\mathbf{x}_{13} + 3\mathbf{x}_5\mathbf{x}_{13} + \boldsymbol{\epsilon}$
- Case 12: $\mathbf{y} = -15\mathbf{x}_2 + 8\mathbf{x}_5 - 2\mathbf{x}_{13} + \boldsymbol{\epsilon}$

Forward regression with omission was applied with both the automatic stopping rule using $mAIC$ (also referred to as F. Omission v1) and a defined stopping rule with $s = 5$ (F. Omission v2). We simulated each model 1000 times and the results

are summarized in Tables 21 and 22, along with previous results from other authors. Note that not all authors ran their analysis methods on the same models, nor did they all use the same criteria to compare their results. *True Model Identification Rate*, or True Model IR % in the tables, is how often the model was detected exactly, with no Type I errors. For our two methods, True Model IR only applies for F. Omission v1 because F. Omission v2 searches for the 5 most important factors and is not intended to find an *exact* model but rather a model with 5 factors. *Smallest Effect Identification Rate* (Smallest Effect IR % in the tables) was used by some authors to indicate how often their analysis method detected the smallest effect in the model. The metric is similar to the *Active Factor Identification Rate* (Active Factor IR % in the tables), which signifies how often the analysis methods detected all of the active factors. While this does allow for Type I errors, we find this to be the most useful metric to compare screening methods because inactive factors can be removed with later experiments. For our analysis methods, Smallest Effect IR and Active Factor IR were identical, so if we missed a factor, it was the one with the smallest effect. This is not necessarily the case, so the two metrics are separated in the tables for clarity and match each author's original classification.

As in Scinto et al. (2011), the search for any model with an interaction term was deemed successful as long as the selection method identified the active factors involved in the main effects and interaction term. We did not search for specific interaction terms, as this would require the inclusion of $\binom{24}{2} = 276$ model terms. The interactions are meant to complicate the noise structure. Also, Scinto et al. analyzed models with two different tuning parameters for their Bayesian Variable Assessment method, but we are only reporting their best results in Tables 21 and 22.

In Cases 1-3, forward regression with omission detected the active factors every time. The best models found with all-subsets, as reported by Abraham et al., also

Table 21. Comparison of Results on the William's Design Matrix

True Model	True Model IR %	Smallest Effect IR %	Active Factor IR %
1. $\mathbf{y} = 5\mathbf{x}_2 + 10\mathbf{x}_7 + 20\mathbf{x}_{13} + \boldsymbol{\epsilon}$			
Forward $k = 5$ in Abraham et al.			100
All-Subsets $k = 5$ in Abraham et al.			100
F. Omission v1	67.4		100
F. Omission v2			100
2. $\mathbf{y} = 14\mathbf{x}_2 + 20\mathbf{x}_7 + 20\mathbf{x}_{13} + \boldsymbol{\epsilon}$			
Forward $k = 5$ in Abraham et al.			0
All-Subsets $k = 5$ in Abraham et al.			100
F. Omission v1	23.2		100
F. Omission v2			100
3. $\mathbf{y} = 20\mathbf{x}_2 + 20\mathbf{x}_7 + 20\mathbf{x}_{13} + \boldsymbol{\epsilon}$			
Forward $k = 5$ in Abraham et al.			0
All-Subsets $k = 5$ in Abraham et al.			100
F. Omission v1	26		100
F. Omission v2			100
4. $\mathbf{y} = 10\mathbf{x}_1 + \boldsymbol{\epsilon}$			
Beattie et al.	61		98
Li and Lin	75.6	100	
Lu and Wu	53		100
Zhang et al.	61	100	
Phoa et al.	99.4	100	
Li et al.	89.2		100
Scinto et al. (threshold = 0.50)	94		100
Forward $k = 5$			100
All-Subsets $k = 5$			100
F. Omission v1	0		100
F. Omission v2			100
5. $\mathbf{y} = -15\mathbf{x}_1 + 8\mathbf{x}_5 - 6\mathbf{x}_9 + 3\mathbf{x}_5\mathbf{x}_9 + \boldsymbol{\epsilon}$			
Beattie et al.	46.5		81-97
Lu and Wu	42		100
Li et al.	66.3		96.1
Scinto et al. (threshold = 0.50)	98		100
Forward $k = 5$			99.4
All-Subsets $k = 5$			98.6
F. Omission v1	83.8		99.9
F. Omission v2			99.4
6*. $\mathbf{y} = -15\mathbf{x}_1 + 12\mathbf{x}_5 - 8\mathbf{x}_9 + 6\mathbf{x}_{14} - 2\mathbf{x}_{18} + \boldsymbol{\epsilon}$			
Beattie et al.	40.7		75-94
Phoa et al.	79.1	91.2	
Forward $k = 5$			100
All-Subsets $k = 5$			98.2
F. Omission v1	89.1		93.5
F. Omission v2			100

Table 22. Comparison of Results on the William's Design Matrix, Continued

True Model	True Model IR %	Smallest Effect IR %	Active Factor IR %
7. $\mathbf{y} = -15\mathbf{x}_1 + 8\mathbf{x}_5 - 2\mathbf{x}_9 + \boldsymbol{\epsilon}$			
Li and Lin	74.7	98.5	
Zhang et al.	76.4	97.7	
Phoa et al.	84.4	85.3	
Li et al.	80.2	99.1	
Forward $k = 5$			99.9
All-Subsets $k = 5$			97.5
F. Omission v1	76.2		99
F. Omission v2			99.9
8*. $\mathbf{y} = -15\mathbf{x}_1 + 12\mathbf{x}_5 - 8\mathbf{x}_9 + 6\mathbf{x}_{13} - 2\mathbf{x}_{17} + \boldsymbol{\epsilon}$			
Li and Lin	69.7	99.4	
Zhang et al.	73.6	95	
Li et al.	85.2		99.2
Scinto et al. (threshold = 0.35)	92		97
Forward $k = 5$			98
All-Subsets $k = 5$			99.7
F. Omission v1	85.6		98.2
F. Omission v2			98
9*. $\mathbf{y} = -15\mathbf{x}_1 + 12\mathbf{x}_5 - 8\mathbf{x}_9 + 6\mathbf{x}_{13} - 12\mathbf{x}_{17} + \boldsymbol{\epsilon}$			
Lu and Wu	53		100
Forward $k = 5$			96.8
All-Subsets $k = 5$			100
F. Omission v1	53.8		100
F. Omission v2			96.8
10. $\mathbf{y} = -15\mathbf{x}_2 + 12\mathbf{x}_5 - 8\mathbf{x}_{13} + 6\mathbf{x}_{14} - 2\mathbf{x}_{17} + \boldsymbol{\epsilon}$			
Li et al.	86		99.7
Forward $k = 5$			100
All-Subsets $k = 5$			100
F. Omission v1	85.6		95.4
F. Omission v2			100
11. $\mathbf{y} = -15\mathbf{x}_2 + 8\mathbf{x}_5 - 6\mathbf{x}_{13} + 3\mathbf{x}_5\mathbf{x}_{13} + \boldsymbol{\epsilon}$			
Li et al.	72.1		98.5
Forward $k = 5$			100
All-Subsets $k = 5$			100
F. Omission v1	19.6		99.4
F. Omission v2			100
12. $\mathbf{y} = -15\mathbf{x}_2 + 8\mathbf{x}_5 - 2\mathbf{x}_{13} + \boldsymbol{\epsilon}$			
Li et al.	86.9		99.8
Forward $k = 5$			100
All-Subsets $k = 5$			99.3
F. Omission v1	68.4		99
F. Omission v2			100

detected the three active factors, though our omission technique required fewer fitted models. Also, for Cases 2 and 3, standard forward regression failed to detect any of the true factors because they inflated the value of $\hat{\beta}_1$. The aliasing structure of the design matrix shows that every factor is aliased with all factors with a $\pm\frac{3}{7}$ or $\pm\frac{1}{7}$ correlation. The aliasing chain for $\mathbf{x}_1$ includes $\frac{3}{7}\beta_2 + \frac{3}{7}\beta_7 + \frac{3}{7}\beta_{13}$, so the active factor create an effect in $\hat{\beta}_1$ greater than any true factor effect (refer to Example 1 in §3.4.1.)

In Case 4, v1 and v2 detected the active factor every time. In Case 5, both found the factors involved in the model more than 99% of the time. In 6 and 7, forward regression with omission v2 performed better than any previously published results, as it found the active factors 100% and 99.9% of the time, respectively. For Case 8, both omission techniques indified the active factors more than 98% of the time, though this is lower than results from Li et al. (2010). In the remaining cases, 9-12, at least one version of forward regression with omission detected the active factors every time. Results show that our method performs favorably to all current methods with respect to this design matrix. Additionally, our methods are easy to implement and do not require tuning parameters.

Notice that Tables 21 and 22 also include simulation results from basic forward regression in 5 steps (based on the $AICc$ criterion) and all-subsets regression for models with 5 factors. We ran these methods on Cases 4-12 to test the adequacy of the models for comparing new analysis methods. An oversight is that basic regression methods outperform many of the novel methods proposed in the literature. This is not due to inconsequential analysis methods; rather, the models themselves do not present the analytical challenges outlined in §3.4.1. In short, the models are not interesting in the context of comparing analysis methods for supersaturated design, with the exceptions of Cases 2 and 3. In 2 and 3, forward regression selected a false effect first and never recovered. Our method was able to correct this. Other methods

may detect the true factors as well, but we did not perform the other analysis methods on models not reported by the original authors.

3.5.2 Lin's AIDS Incidence Design.

In light of our concerns with the simulation models on the Williams matrix, we tested forward regression with omission on a different, but still well-known, supersaturated design. Lin (1995a) analyzed a 138 factor, 24 run two-level supersaturated design to identify factors affecting the AIDS incidence rate per 100,000 people. We defer to Lin, Mee (2009), and Edwards and Mee (2011) for discussions about the analysis of the raw data. Here, we focus on a simulation study using the design matrix. The first 23 factors form a Hadamard matrix, shown in Table 23. As stated in Mee (2009), the remaining factors, $\mathbf{x}_{24} - \mathbf{x}_{138}$, were generated from the following two-factor interactions:

- $\mathbf{x}_{24} - \mathbf{x}_{45}$: $\mathbf{x}_{22+i} = \mathbf{x}_1 * \mathbf{x}_i$, $i = 2, \ldots, 23$
- $\mathbf{x}_{46} - \mathbf{x}_{66}$: $\mathbf{x}_{43+i} = \mathbf{x}_2 * \mathbf{x}_i$, $i = 3, \ldots, 23$
- $\mathbf{x}_{67} - \mathbf{x}_{86}$: $\mathbf{x}_{63+i} = \mathbf{x}_3 * \mathbf{x}_i$, $i = 4, \ldots, 23$
- $\mathbf{x}_{87} - \mathbf{x}_{105}$: $\mathbf{x}_{82+i} = \mathbf{x}_4 * \mathbf{x}_i$, $i = 5, \ldots, 23$
- $\mathbf{x}_{106} - \mathbf{x}_{123}$: $\mathbf{x}_{100+i} = \mathbf{x}_5 * \mathbf{x}_i$, $i = 6, \ldots, 23$
- $\mathbf{x}_{124} - \mathbf{x}_{138}$: $\mathbf{x}_{117+i} = \mathbf{x}_6 * \mathbf{x}_i$, $i = 7, \ldots, 21$

For this simulation, we built five truth models: two have 8 active factors, two have 10, and one has 12. For each model, the location of active factors was chosen randomly and each active factor was assigned a random integer β value between 3 and 50. All other β values were set to 0. We wanted models with different magnitudes

Table 23. First 23 Factors of Lin's (1995) AIDS Incidence Design, as reported by Mee (2009)

Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{x}_{16}$	$\mathbf{x}_{17}$	$\mathbf{x}_{18}$	$\mathbf{x}_{19}$	$\mathbf{x}_{20}$	$\mathbf{x}_{21}$	$\mathbf{x}_{22}$	$\mathbf{x}_{23}$	$\mathbf{y}$	$ln(\mathbf{y})$
1	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	22.61	3.12
2	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	14.26	2.66
3	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	58.42	4.07
4	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	24.59	3.20
5	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	10.28	2.33
6	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	188.46	5.24
7	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	22.68	3.12
8	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	22.90	3.13
9	+	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	52.04	3.95
10	+	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	381.61	5.94
11	−	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	16.22	2.79
12	−	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	108.59	4.69
13	+	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	98.05	4.59
14	+	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	53.13	3.97
15	−	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	83.41	4.42
16	−	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	13.59	2.61
17	+	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	242.96	5.49
18	−	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	663.93	6.50
19	+	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	57.95	4.06
20	−	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	177.49	5.18
21	−	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	40.22	3.69
22	−	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	52.23	3.96
23	−	+	+	+	+	+	−	+	−	+	+	−	−	+	+	−	−	+	−	+	−	−	−	53.50	3.98
24	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	2463.24	7.81

of active factors, and at a noise level of $\sigma = 1$, we have a random spread of small, medium, and large active effects. The five simulation models are:

- Case 1: $\mathbf{y} = 4\mathbf{x}_7 + 20\mathbf{x}_{27} + 43\mathbf{x}_{57} + 29\mathbf{x}_{65} + 49\mathbf{x}_{70} + 38\mathbf{x}_{91} + 28\mathbf{x}_{112} + 49\mathbf{x}_{136} + \boldsymbol{\epsilon}$
- Case 2: $\mathbf{y} = 31\mathbf{x}_{27} + 42\mathbf{x}_{28} + 24\mathbf{x}_{96} + 27\mathbf{x}_{98} + 19\mathbf{x}_{102} + 8\mathbf{x}_{108} + 15\mathbf{x}_{110} + 36\mathbf{x}_{121} + \boldsymbol{\epsilon}$
- Case 3: $\mathbf{y} = 10\mathbf{x}_5 + 17\mathbf{x}_{49} + 15\mathbf{x}_{74} + 23\mathbf{x}_{79} + 11\mathbf{x}_{81} + 50\mathbf{x}_{90} + 38\mathbf{x}_{92} + 24\mathbf{x}_{112} + 18\mathbf{x}_{132} + 7\mathbf{x}_{133} + \boldsymbol{\epsilon}$
- Case 4: $\mathbf{y} = 45\mathbf{x}_{20} + 11\mathbf{x}_{34} + 5\mathbf{x}_{47} + 8\mathbf{x}_{53} + 35\mathbf{x}_{95} + 25\mathbf{x}_{99} + 41\mathbf{x}_{106} + 39\mathbf{x}_{107} + 10\mathbf{x}_{126} + 47\mathbf{x}_{130} + \boldsymbol{\epsilon}$
- Case 5: $\mathbf{y} = 31\mathbf{x}_{18} + 33\mathbf{x}_{25} + 35\mathbf{x}_{41} + 16\mathbf{x}_{53} + 18\mathbf{x}_{58} + 48\mathbf{x}_{64} + 26\mathbf{x}_{65} + 21\mathbf{x}_{75} + 40\mathbf{x}_{89} + 11\mathbf{x}_{95} + 28\mathbf{x}_{102} + 13\mathbf{x}_{114} + \boldsymbol{\epsilon}$

We generated response data with each model using $\boldsymbol{\epsilon} \sim N(0, \mathbf{I}_n)$. In this design, the factor-to-run ratio is large, much larger than the guidelines proposed by Marley

and Woods (2010). They recommend the factor-to-run ratio should be less than 2. Because our ratio is larger, we searched for the maximum recommended active factors, $n/2 = 12$. Note with a design this size, all-subsets regression is computationally impractical. Edwards and Mee (2011) performed all-subsets regression for up to 6 factors on this design, and it took over 7 hours to run. The estimated all-subsets regression for 7 factors would take days, as it would require $\sum_{i=1}^{7} \binom{138}{i} \approx 117 \times 10^9$ fitted models.

Table 24. Simulation Results on Lin Matrix

Step:	1	2	3	4	5	6	7	8	9	10	11	12	R^2_{adj}	AIC	
1. $\mathbf{y} = 4\mathbf{x}_7 + 20\mathbf{x}_{27} + 43\mathbf{x}_{57} + 29\mathbf{x}_{65} + 49\mathbf{x}_{70} + 38\mathbf{x}_{91} + 28\mathbf{x}_{112} + 49\mathbf{x}_{136} + \boldsymbol{\epsilon}$															
Forward	**70**	**91**	**112**	**136**	**57**	**65**	**27**	**7**	130	88	19	121	1.000	-38.884	✓
Omission	**70**	**91**	**112**	**136**	**57**	**65**	**27**	**7**	130	88	19	121	1.000	-38.884	✓
2. $\mathbf{y} = 31\mathbf{x}_{27} + 42\mathbf{x}_{28} + 24\mathbf{x}_{96} + 27\mathbf{x}_{98} + 19\mathbf{x}_{102} + 8\mathbf{x}_{108} + 15\mathbf{x}_{110} + 36\mathbf{x}_{121} + \boldsymbol{\epsilon}$															
Forward	**28**	**98**	85	**96**	11	**121**	41	51	**27**	**108**	**110**	59	1.000	-12.781	
Omission	**28**	**98**	**27**	**102**	**121**	**96**	**110**	**108**	45	26	14	132	1.000	-51.755	✓
3. $\mathbf{y} = 10\mathbf{x}_5 + 17\mathbf{x}_{49} + 15\mathbf{x}_{74} + 23\mathbf{x}_{79} + 11\mathbf{x}_{81} + 50\mathbf{x}_{90} + 38\mathbf{x}_{92} + 24\mathbf{x}_{112} + 18\mathbf{x}_{132} + 7\mathbf{x}_{133} + \boldsymbol{\epsilon}$															
Forward	**90**	**112**	**92**	**79**	**132**	**49**	**74**	4	135	56	**5**	122	0.998	69.501	
Omission	**90**	**112**	**92**	**79**	**132**	**49**	**74**	**81**	**5**	**133**	69	33	1.000	-18.360	✓
4. $\mathbf{y} = 45\mathbf{x}_{20} + 11\mathbf{x}_{34} + 5\mathbf{x}_{47} + 8\mathbf{x}_{53} + 35\mathbf{x}_{95} + 25\mathbf{x}_{99} + 41\mathbf{x}_{106} + 39\mathbf{x}_{107} + 10\mathbf{x}_{126} + 47\mathbf{x}_{130} + \boldsymbol{\epsilon}$															
Forward	**130**	**106**	**95**	103	**99**	10	88	111	110	24	5	6	0.997	95.498	
Omission	**130**	61	25	41	76	108	33	**99**	17	4	**53**	84	0.998	82.434	
5. $\mathbf{y} = 31\mathbf{x}_{18} + 33\mathbf{x}_{25} + 35\mathbf{x}_{41} + 16\mathbf{x}_{53} + 18\mathbf{x}_{58} + 48\mathbf{x}_{64} + 26\mathbf{x}_{65} + 21\mathbf{x}_{75} + 40\mathbf{x}_{89} + 11\mathbf{x}_{95} + 28\mathbf{x}_{102} + 13\mathbf{x}_{114} + \boldsymbol{\epsilon}$															
Forward	**41**	82	**102**	104	119	**95**	4	7	11	73	138	87	0.995	103.567	
Omission	**41**	82	**102**	119	122	5	129	58	55	73	84	63	0.998	85.395	

Analysis results from forward regression and forward regression with omission are compared in Table 24, which details what factors were selected at each step. A ✓signifies that every active factor was identified. For the first model, regular forward regression found all 8 active factors before choosing 4 noise factors to fit the desired 12-factor model. Forward regression with omission generated $s + 1 = 13$ models, one of which was the original forward regression model. This had the lowest AIC value, so the same model was detected. For Case 2 with 8 active factors, forward regression detects 2 active factors, $\mathbf{x}_{28}$ and $\mathbf{x}_{98}$, before selecting a false effect, $\mathbf{x}_{85}$. This selection alters the path of forward selection, and although it detected more true factors, it fails to select all 8. Our method, however, fits a model without each factor identified with forward regression in case it was a false effect. When a model

is fit without $\mathbf{x}_{85}$, forward regression chooses another active factor, $\mathbf{x}_{27}$, in step 3. It then continues to select the remaining 5 active factors. The AIC value of the model improved from -12.781 to -51.755, though each model had an $R^2_{adj} = 1.000$. A similar situation happens in Case 3 with the 10 factor model. Forward regression detected a false effect in step 8, but once that factor was removed, our method found all 10 active factors. The AIC dropped from 69.501 to -18.360. The R^2_{adj} values were again nearly identical, further indicating the criterion's inability to differentiate models on supersaturated designs.

Cases 4 and 5 highlight the difficulty of analyzing supersaturated designs with a large factor-to-run ratio. In both cases, our method fit better models in terms of AIC and R^2_{adj}, but detected *less* active factors. The aliasing structure of the design has active and inactive factors tangled together in such a way that many models will fit the data well, at least with respect to standard regression criteria. However, when all-subsets is computationally infeasible, forward regression with omission provides more potential models than forward regression alone, and in some cases, it accurately identifies all active factors.

3.6 Conclusions & Discussion

3.6.1 Conclusions.

Although research on supersaturated designs is extensive, their application is less widespread. In this paper, we presented basic supersaturated design concepts, showed how to find or construct designs, reviewed common analytical challenges, and introduced easy-to-use analysis methods. Our intent is to better familiarize practitioners with supersaturated designs. We clarified the differences between main-effect and model-effect supersaturated designs and walked through examples of four common analysis pitfalls: inflated inactive factors, hidden active factors, model indiscrimina-

tion, and wrong assumptions. We stress that our examples should not serve as a deterrent to experimenters considering supersaturated designs, but rather serve as an impetus for an in-depth look at the analysis of such designs. It is important to be honest about their shortcomings because analysis is not trivial. Many methods are available to analyze the designs, yet no method is consistently better than all the others. Basic regression methods are viable options, but be aware there are times when a slight variation of these methods, i.e. our proposed forward regression with omission, is more useful.

Moreover, we compiled the most comprehensive summary of analysis techniques on simulation models using Lin's half-fraction of Williams design matrix. Inconsistencies in the literature caused by the duplicate column have been resolved. We provided an updated and corrected table of simulation results, and ultimately concluded that we need a new gold standard to compare analysis methods. Researchers would benefit from simulation models that actually present problems for basic techniques. We also presented a new simulation study using the 138 factor, 24 run Lin matrix, and showed that forward regression with omission is a useful method to identify active factors, especially when all-subsets regression is infeasible.

3.6.2 Discussion.

With the documented concerns of supersaturated designs, a practitioner might inquire if it ever makes sense to use a supersaturated design. In reality, using the designs may be unavoidable. To assume they will never be used is naive because when screening problems have more factors than runs, it's important to find a way to statistically analyze the data. A statistician can apply the different analysis methods presented in this paper and test the robustness of the selected active factors. Whatever method is implemented, follow-up runs are required. A potential area for future

work is how to strategically plan the follow-up runs to a supersaturated experiment. Another interesting area for future work is a more direct connection between construction and analysis methods because construction methods "appear to be unrelated to the way in which the data are analyzed"(Gilmour, 2006). While researchers produce new methodologies and tools to alleviate these concerns, we encourage practitioners to consider using supersaturated designs but to also be aware of their risks.

IV. Augmenting Supersaturated Designs with Bayesian D-Optimality

A methodology is developed to add runs to existing supersaturated designs. The technique uses information from the analysis of the initial experiment to choose the best possible follow-up runs. After analysis of the initial data, factors are classified into one of three groups: primary, secondary, and potential. Runs are added to maximize a Bayesian D-optimality criterion to increase the information gained about those factors. Simulation results show the method can outperform existing supersaturated design augmentation strategies that add runs without analyzing the initial response variables.

4.1 Introduction

Screening designs are used in the early stages of industrial and computer experiments to discover which input factors have major effects on a system's output. A screening experiment is intended to remove the negligible, or inactive, factors from further experiments, allowing the investigator to focus on the important, or active, factors. In a large set of factors, relatively few are likely to be active, a concept called effect sparsity (Box and Meyer, 1986). Traditional screening methods for k factors, like two-level 2^{k-p} fractional factorial (Box et al., 2005) or Plackett-Burman designs (Plackett and Burman, 1946), require at least $k+1$ experimental runs to separate the few active factors from the many inactive. But, when k is large or experimental runs are prohibitively expensive, the experimenter requires alternative designs that can screen k factors in $n < k+1$ runs. Supersaturated designs (SSDs) are one such technique.

SSDs were introduced by Satterthwaite (1959) and Booth and Cox (1962) but did not receive considerable attention until Lin (1993) and Wu (1993) renewed interest in the field, which continues today. The focus of an SSD is on identifying the active main effects in a linear model. Consider an experiment with k factors and n runs. The underlying linear main-effect model is represented as:

$$\mathbf{y} = \beta_0\mathbf{1} + \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}; \tag{4.1}$$

where $\mathbf{y}$ is the response vector, $\beta_0, \ldots, \beta_k$ are the $p = k+1$ unknown model parameters, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_n)$ is the error term. The model matrix $\mathbf{X}$ equals $(\mathbf{1}|\mathbf{S})$, where $\mathbf{1}$ is an $n \times 1$ column of 1's and $\mathbf{S} = (\mathbf{x}_1|\ldots|\mathbf{x}_k)$ is the design matrix. The rows of $\mathbf{S}$ contain the k factor level settings for the n experimental runs. For clarity, we adopt the notation in Gupta et al. (2010) and let $\text{SSD}(n, k) = \mathbf{S}$ represent an SSD with n runs and k factors.

An $\text{SSD}(n, k)$ with model matrix $\mathbf{X}$ is typically constructed to optimize a criterion that minimizes the bias of the parameter estimates. For two-level designs, in which factor levels are coded as ± 1, the most popular criterion is $E(s^2)$. Denote the (i, j)th element of $\mathbf{X}'\mathbf{X}$ as s_{ij}. $E(s^2)$ is defined as $E(s^2) = \sum_{i<j} s_{ij}^2/(k(k-1)/2)$. A small $E(s^2)$ implies the average correlations between factor columns are as small as possible. (See Nguyen (1996), Bulutoglu and Cheng (2004), Suen and Das (2010), and references therein for more on $E(s^2)$-optimal designs.) Another popular construction technique is based on the Bayesian D-optimality criterion by Jones et al. (2008), discussed in Sections 4.2 and 4.3. An overview of other design criteria for SSDs, including criteria for designs with more than two levels, can be found in Lin (2003).

Regardless of the construction method, the analysis of SSDs is rather challenging. Since $n < k + 1$, $\mathbf{X}'\mathbf{X}$ is singular and the ordinary least squares estimates, $\mathbf{b} =$

$(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, cannot be calculated. Due to effect sparsity, most of the β_i terms in (4.1) are assumed to be zero, but choosing which factors to remove from the model is difficult. The most common analytical challenges associated with SSDs were documented in Gutman et al. (2013b), and we refer the reader to Gupta and Kohli (2008) and Georgiou (2012) for reviews of proposed analysis methods. Note, however, that no method is infallible. There is a tradeoff between the economy of a design and the information gained from the experiment. The experimenter risks classifying an inactive factor as active (Type I error), or worse, classifying an active factor as inactive (Type II error). For this reason, screening designs are not intended to be utilized for an "all-encompassing" experiment, but rather as the first stage in a sequence of experiments (Box, 1992). This is especially pertinent with SSDs because the original analysis results are not always definitive, a consequence of the inability to simultaneously estimate all main-effects.

Adding follow-up runs to a design is a useful way to clarify or confirm initial results and guide the next phase of experimentation. The notion of sequential experimentation is a well-established approach in experimental design: Box (1992) provided general guidelines to consider, and traditional augmentation strategies like fold-over designs and the addition of center points are described in most experimental design textbooks (e.g. Montgomery (2009) and Wu and Hamada (2000)). However, the idea of augmenting SSDs has only recently been explored. Consider the following.

Suppose after running an SSD(n_1, k), the experimenter can afford n_2 more runs to resolve ambiguities. What is the best way to augment the original design to reduce uncertainty and get the most information out of the final SSD$(n_1 + n_2, k)$? This is a relatively new research area. Two papers by Gupta et al. (2010) and Gupta et al. (2012) describe methods to add rows to two-level and s-level designs, respectively. With Gupta et al.'s method, $E(s^2)$-optimal designs are augmented with additional

runs to create a new class of "extended $E(s^2)$-optimal" designs. Suen and Das (2010) use a similar approach to add or remove one row from an existing $E(s^2)$-optimal design to make a new $E(s^2)$-optimal design. However, in the current methods, there is no effort to analyze the initial results before adding runs. After running an SSD(n_1, k), an experimenter should have *some* useful information about the process. Indeed, that is the motivation for running the experiment in the first place.

The focus of this paper is to present an alternative approach to the extended-$E(s^2)$ augmentation technique presented in Gupta et al. (2010). Our goal is to take the information gained from the initial design, SSD(n_1, k), identify and classify factors of interest, and prioritize the additional n_2 runs to get the most information from the final design, SSD$(n_1 + n_2, k)$. Specifically, we propose an SSD augmentation strategy using the Bayesian D-optimality criterion from DuMouchel and Jones (1994) and Jones et al. (2008). Our approach has several benefits over current methods:

1. It uses information from the first n_1 runs to strategically plan the n_2 follow-up runs;

2. It can augment any SSD built from any construction method or optimality criterion;

3. It can add any number of runs; and

4. It uses the Coordinate-Exchange Algorithm (Meyer and Nachtsheim, 1995), a polynomial-time algorithm.

Like Gupta et al. (2010), we assume additional runs become available *after* the first experiment and that n_2 is provided by a decision maker. This is inherently different than a two-stage design where an experimenter purposefully partitions the allotted screening budget into two parts. SSDs are used when resources are heavily

constrained, so had all the runs been available in the screening budget from the beginning, the experimenter would likely have chosen a design to accommodate all runs.

The next section reviews the relevant background of three key concepts: Bayesian D-optimality, the Coordinate-Exchange Algorithm, and algorithmic augmentation strategies for standard designs. Section 4.3 presents our approach to augment SSDs using information from the initial runs. Section 4.4 compares the performance of Bayesian D-optimal augmented designs with extended $E(s^2)$-optimal designs by highlighting examples where using information from the first runs leads to better recommendations than adding runs to maintain $E(s^2)$-optimality. We conclude with a discussion in Section 4.5.

4.2 Preliminaries

4.2.1 Bayesian D-Optimality.

D-optimality is a popular design criterion for traditional designs with an assumed $n \times k$ model matrix $\mathbf{X}$ with $n > k$. The goal of D-optimality is to reduce the error variances of the least squared estimates, given by $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. This is accomplished by maximizing the determinant of $\mathbf{X}'\mathbf{X}$, denoted $|\mathbf{X}'\mathbf{X}|$ (Myers et al., 2009). Unfortunately, D-optimality is not always model-robust because the design may be 'optimal' to the wrong model. To reduce dependency on one model, researchers have proposed alternative optimality criteria under the Bayesian paradigm. A Bayesian design for a linear model is constructed to maximize the posterior information about the model parameters, $\boldsymbol{\beta}$, which are conditional on prior information. In Bayesian design theory, the counterpart to D-optimality is Bayesian D-optimality. We refer the reader to Chaloner and Verdinelli (1995) and Atkinson et al. (2007, Ch. 18) for a detailed description and history of Bayesian design theory and Bayesian D-optimal

methods. In this paper, we focus on the Bayesian D-optimality criterion as presented in DuMouchel and Jones (1994).

Consider the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. Assume $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_n)$. Let the prior distribution of the parameters be $\boldsymbol{\beta}|\sigma^2 \sim N(\boldsymbol{\beta}_0, \sigma^2\mathbf{R}^{-1})$, where $\mathbf{R}$ is a prior covariance matrix, and the conditional distribution of $\mathbf{y}$ given $\boldsymbol{\beta}$ be $\mathbf{y}|\boldsymbol{\beta}, \sigma^2 \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$. The posterior distribution for $\boldsymbol{\beta}$ given $\mathbf{y}$ is then $\boldsymbol{\beta}|\mathbf{y} \sim N(\mathbf{b}, \sigma^2(\mathbf{X}'\mathbf{X} + \mathbf{R})^{-1})$, where $\mathbf{b} = (\mathbf{X}'\mathbf{X} + \mathbf{R})^{-1}(\mathbf{X}'\mathbf{y} + \mathbf{R}\boldsymbol{\beta}_0)$. As noted, D-optimal designs maximize $|\mathbf{X}'\mathbf{X}|$ to reduce the error variances of the parameter estimates. Similarly, Bayesian D-optimal designs aim to reduce the error variances of the parameter estimates, but the addition of prior information has changed the variance to $\mathrm{Var}(\mathbf{b}) = \sigma^2(\mathbf{X}'\mathbf{X}+\mathbf{R})^{-1}$. Therefore, Bayesian D-optimal designs are constructed to maximize $|\mathbf{X}'\mathbf{X} + \mathbf{R}|$.

The matrix $\mathbf{R}$ reflects the prior information assigned to each of the p terms in the model matrix, $\mathbf{X}$. DuMouchel and Jones (1994) incorporate prior information and model uncertainty into the regression parameters by splitting models terms into two sets: primary terms and potential terms. Primary terms are assumed to be active (i.e. a nonzero β_i), whereas potential terms may or may not be active. Using this information, the p_1 primary terms are given a diffuse prior distribution with an arbitrary prior mean and prior variance tending toward infinity. The arbitrary mean reflects no knowledge of the direction of the effect of the primary term, and the "infinite" variance implies the effect is likely to be much different than zero. The $p_2 = p - p_1$ potential terms, on the contrary, are not expected to have large effects and are given a prior mean zero and finite variance $\sigma^2\tau^2$, where τ represents the expected effect of a factor relative to residual standard error (DuMouchel and Jones, 1994). The matrix, $\mathbf{R}$, is consequently set to $\mathbf{R} = \mathbf{K}/\tau^2$, where

$$\mathbf{K} = \begin{pmatrix} \mathbf{0}_{p_1 \times p_1} & \mathbf{0}_{p_1 \times p_2} \\ \mathbf{0}_{p_2 \times p_1} & \mathbf{I}_{p_2 \times p_2} \end{pmatrix}.$$

The posterior distribution for $\boldsymbol{\beta}$ given $\mathbf{y}$ is then

$$\boldsymbol{\beta}|\mathbf{y} \sim N\left[\left(\mathbf{X}'\mathbf{X} + \frac{\mathbf{K}}{\tau^2}\right)^{-1}\left(\mathbf{X}'\mathbf{y} + \frac{\mathbf{K}}{\tau^2}\boldsymbol{\beta}_0\right), \sigma^2\left(\mathbf{X}'\mathbf{X} + \frac{\mathbf{K}}{\tau^2}\right)^{-1}\right]; \qquad (4.2)$$

and the Bayesian D-optimal design objective function becomes $|\mathbf{X}'\mathbf{X} + \mathbf{K}/\tau^2|$. The p_1 primary terms consist of those terms assumed to be in the true model, whereas higher-order effects make up the p_2 potential terms. The methodology allows the total number of model terms, $p = p_1 + p_2$, to be greater than the number of runs, n, because the addition of the prior information in $\mathbf{K}/\tau^2$ makes the information matrix invertible. Thus, the designs can estimate all p_1 primary terms while allowing the delectability of some of the p_2 potential terms.

This method was adapted to create SSDs in Jones et al. (2008). In an SSD(n, k), prior information is usually not available for any of the control factors, so all k are classified as potential terms; the intercept is the only primary term. If information is available to suggest $p_1 > 1$ factors are active, the Bayesian D-optimality criterion can create such a design, provided $p_1 < n$. Incorporating this prior information makes the technique more dynamic than a naive regularization of the information matrix.

Jones et al. set $\tau^2 = 5$ and used the Coordinate-Exchange Algorithm to create the designs. For two-level designs, the Coordinate-Exchange Algorithm can be summarized with the following steps: Generate a uniform random number from $[-1, 1]$ for each $x_{i,j}$ in $\mathbf{X}$. Then, iterate through each entry in $\mathbf{X}$, replacing the random number with the entry from $\{-1, +1\}$ that results in the largest value of the objective function. Because the resulting design is likely only locally optimal, the algorithm is repeated many times with different random starting values for the $x_{i,j}$ entries. After many

random starts, e.g. 100, the design with the largest determinant is approximately the Bayesian D-optimal design.

4.2.2 Augmenting Designs.

Augmenting a design with additional runs is the natural way to get more information about the system under study. One criterion used to add runs to traditional designs with $n < k$ is D-optimality. Suppose after an initial experiment, the investigator wishes to add specific terms to the assumed model matrix (e.g. two-factor interactions or quadratic effects). The model is specified *a priori* and runs are added to original model matrix to create a D-optimal design for the full, updated model. The overall goal is to maximize the information gained from the combined set. For a step-by-step example, see Goos and Jones (2011, p. 60-65).

Let $\mathbf{X}_1$ be a model matrix corresponding to the first n_1 runs of an experiment, and let $\mathbf{X}_2$ be the additional n_2 rows. To optimize the final design, we need to maximize $|\mathbf{X}'\mathbf{X}|$ of the final model matrix $\mathbf{X}$, where $\mathbf{X} = \begin{pmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{pmatrix}$.

To find $|\mathbf{X}'\mathbf{X}|$, first note that

$$\begin{aligned}\mathbf{X}'\mathbf{X} &= \begin{pmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{pmatrix}'\begin{pmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{pmatrix}\\ &= (\mathbf{X}_1'\mathbf{X}_2')\begin{pmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{pmatrix}\\ &= \mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2.\end{aligned} \tag{4.3}$$

The Coordinate-Exchange Algorithm can be used to construct the appropriate $\mathbf{X}_2$ matrix to maximize $|\mathbf{X}_1'\mathbf{X}_1+\mathbf{X}_2'\mathbf{X}_2|$ and create an augmented D-optimal design. Other algorithms and strategies for D-optimal augmentation can be found in Atkinson et al. (2007).

Follow-up runs to traditional designs can also be added with Bayesian techniques. Meyer et al. (1996) augmented designs with a Bayesian model-discrimination criterion to resolve ambiguities between many plausible models in the presence of observed data. Jones and Dumouchel (1996), in a discussion of Meyer et al.'s method, suggested an F-criterion based on Fisher's information matrix. Neff (1996) and Ruggoo and Vandebroek (2004) proposed a two-stage, sequential Bayesian D-D optimal method based on the Bayesian D-optimality criterion in DuMouchel and Jones. In the two-stage Bayesian D-D optimal method, a first stage design is constructing to support an assumed model with primary and potential terms. After the first stage, data is analyzed via Box and Meyer (1993)'s model-discrimination method of calculating posterior probabilities of possible models. A second stage design is then added to maximize a weighted D-optimality criterion to support and discriminate the many competing models.

In the next section, we extend the aforementioned work and develop the methodology to add runs to SSDs. It is important to mention several unique aspects to augmenting SSDs. First, we are typically not interested in adding interactions or quadratic effects to the assumed main-effect model; with the limited number of runs, detecting the active main effects is the top priority. Second, the large number of factors and small number of runs in SSDs means many models explain the data well. As such, it is difficult to pick which model or models to build upon in the follow-up runs. Therefore, instead of adding runs based on a model-discrimination criterion like in Ruggoo and Vandebroek (2004), we add runs based on a categorization of factors. A model-dependent augmentation strategy is computationally expensive. For example, it took 7 hours to search for all 6 factor models in a 124 factor, 24 run design (Edwards and Mee, 2011). Calculating larger models with a model-discrimination criterion would be impractical. Categorizing factors into groups is more efficient.

Further, categorization make the augmentation method dynamic because it is not tied to a specific analysis method; the experimenter can analyze the initial data with several methods to search for active factors.

4.3 Augmenting Supersaturated Designs with Bayesian D-optimality

Suppose an experimenter ran an SSD(n_1, k) and can afford to add n_2 more runs. Our objective is to create the best possible augmented design, SSD$(n_1 + n_2, k)$, given the information from the initial n_1 runs. To do this, we adopt the linear model assumptions used to create Bayesian D-optimal SSDs and adapt them to add n_2 runs to the design matrix. Let $\mathbf{X}_1$ be the original main-effect model matrix with response vector $\mathbf{y}_1$. Assume the prior distribution of $\boldsymbol{\beta}$ is $\boldsymbol{\beta}|\sigma^2 \sim N(\boldsymbol{\beta}_0, \sigma^2\mathbf{R}^{-1})$ for a prior covariance matrix, $\mathbf{R}$. Let the $n_2 \times 1$ vector of new observations, $\mathbf{y}_2$, have the conditional distribution $\mathbf{y}_2|\boldsymbol{\beta}, \sigma^2 \sim N(\mathbf{X}_2\boldsymbol{\beta}, \sigma^2\mathbf{I}_{n_2\times n_2})$, where $\mathbf{X}_2$ is the additional run matrix in model form. Then, as shown in Ruggoo and Vandebroek (2004), the posterior distribution for $\boldsymbol{\beta}$ given $\mathbf{y} = \begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix}$ is

$$\boldsymbol{\beta}|\mathbf{y} \sim N\left[\mathbf{b}, \sigma^2\left(\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2 + \mathbf{R}\right)^{-1}\right]; \tag{4.4}$$

where $\mathbf{b} = \left(\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2 + \mathbf{R}\right)^{-1}\left(\mathbf{X}_1'\mathbf{y}_1 + \mathbf{X}_2'\mathbf{y}_2 + \mathbf{R}\boldsymbol{\beta}_0\right)$. To create a Bayesian D-optimal augmented SSD, $\mathbf{X}_2$ is chosen to maximize $|\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2 + \mathbf{R}|$. Because runs are added to an existing design, the prior information for the n_2 follow-up runs comes from the analysis of the original SSD(n_1, k) with response vector $\mathbf{y}_1$. Like DuMouchel and Jones (1994), prior information is incorporated into the design process through the choice in $\mathbf{R}$ by classifying factors as primary or potential terms. We also introduce a category of *secondary* terms.

After the first n_1 runs, the experimenter can likely identify factors that appear to be the most active. For instance, some factor or factor set may be detected in

many different analysis methods. If evidence suggests the factor is in the true model, the experimenter can classify it as a primary term. If there is an indication the factor may be active, but it is not a predominant as the primary term, the factor can be classified as a secondary term. Any factor that does not appear active can be classified as a potential term (Section 4.3.1 expounds on classifying factors). Using this classification, the augmented design SSD$(n_1 + n_2, k)$ is constructed to reduce the error variances of the parameter estimates under the Bayesian paradigm.

Let p_1 denote the number of primary terms, p_2 denote the number of secondary terms, and p_3 be the number of potential terms, where $p_1 + p_2 + p_3 = k + 1 = p$. The p_1 primary terms are the most likely to be active, so their effects, denoted $\boldsymbol{\beta}_{\text{pri}}$, are given a diffuse prior. The p_2 secondary terms with effects $\boldsymbol{\beta}_{\text{sec}}$ are given a prior mean of zero and a finite variance $\sigma^2\gamma^2$, while the p_3 potential terms with effects $\boldsymbol{\beta}_{\text{pot}}$ are assigned a prior mean of zero and a finite variance $\sigma^2\tau^2$, where $\tau < \gamma$. Larger scaling factors for σ^2 represent stronger beliefs that certain factors are active (Ruggoo and Vandebroek, 2004). Using this information, $\mathbf{R} = \mathbf{J}/\gamma^2 + \mathbf{K}/\tau^2$, where

$$\mathbf{J} = \begin{pmatrix} 0 & & & \\ & j_{1,1} & & \mathbf{0} \\ & & j_{2,2} & \\ & \mathbf{0} & & \ddots \\ & & & & j_{k,k} \end{pmatrix} \text{ and } \mathbf{K} = \begin{pmatrix} 0 & & & \\ & k_{1,1} & & \mathbf{0} \\ & & k_{2,2} & \\ & \mathbf{0} & & \ddots \\ & & & & k_{k,k} \end{pmatrix}. \tag{4.5}$$

For each $i = 1, 2, \ldots, k$, we set $j_{i,i} = 1$ if $\mathbf{x}_i$ is a secondary term, 0 otherwise, and $\sum_{i=1}^{k} j_{i,i} = p_2$. Similarly, $k_{i,i} = 1$ if $\mathbf{x}_i$ is a potential term, 0 otherwise, and $\sum_{i=1}^{k} k_{i,i} = p_3$.

The posterior distribution for $\boldsymbol{\beta}$ in (4.4) can be rewritten as

$$\boldsymbol{\beta}|\mathbf{y} \sim N\left[\mathbf{b}, \sigma^2\left(\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2 + \frac{\mathbf{J}}{\gamma^2} + \frac{\mathbf{K}}{\tau^2}\right)^{-1}\right]; \tag{4.6}$$

where $\mathbf{b} = \left(\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2 + \mathbf{J}/\gamma^2 + \mathbf{K}/\tau^2\right)^{-1}\left(\mathbf{X}_1'\mathbf{y}_1 + \mathbf{X}_2'\mathbf{y}_2 + (\mathbf{J}/\gamma^2 + \mathbf{K}/\tau^2)\boldsymbol{\beta}_0\right)$. Therefore, a Bayesian D-optimal augmented SSD(n_1+n_2, k) with model matrix $\mathbf{X} = \begin{pmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{pmatrix}$ is constructed by choosing $\mathbf{X}_2$ to maximize

$$\left|\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_2'\mathbf{X}_2 + \frac{\mathbf{J}}{\gamma^2} + \frac{\mathbf{K}}{\tau^2}\right|. \tag{4.7}$$

Note that $p_1 < n_1+n_2$ is a necessary condition to make the determinant calculation in (4.7) nonzero. The Coordinate Exchange Algorithm is used to construct $\mathbf{X}_2$ to optimize the objective function in (4.7).

4.3.1 Classifying Factors.

Getting information from the original SSD(n_1, k) is not trivial, hence the motivation for additional runs. However, the objective function in (4.7) is dependent on the experimenter using *some* information from the initial runs in order to classify the k factors into groups. Different analysis techniques may identify conflicting sets of active factors, and this can make it a challenge to assign the k factors into the primary, secondary, or potential groups. Although a formal discussion about analyzing SSDs is beyond the scope of this paper, we provide some suggestions on prioritizing factors into the primary, secondary, or potential groups:

1. The intercept is always a primary term.

2. If an experimenter must add runs but is not comfortable classifying the factors, we suggest specifying all factors as potential terms to mimic the construction of Bayesian D-optimal SSDs.

3. If an analysis method (or many methods) highlight a group of less than $n_1 + n_2$ key factors, specify the terms as primary.

4. If the number of factors of interest is larger than $n_1 + n_2$ runs, specify the terms as secondary.

5. Terms with little evidence to suggest they are active should be classified as potential.

Secondary terms let the experimenter differentiate between terms when more than $n_1 + n_2$ factors are of interest. After running an SSD(n_1, k), an experimenter may identify a group of s key factors, where $s > n_1 + n_2$. Therefore, not all s factors can be classified as the p_1 primary terms, as $p_1 < n_1 + n_2$ is required. To differentiate between the s key factors and the remaining $k - s$, the experimenter can classify all s factors as the p_2 secondary terms. Secondary terms are given a larger prior variance to suggest they are likely more active than the $k - p_2$ potential terms. The augmentation criterion then selects runs to discriminate between the two groups. An example is given in Section 4.4.2.

4.3.2 Example Augmentation.

To visually compare how prior information influences the final SSD matrix, we created a Bayesian D-optimal SSD(25,100) with the JMP statistical software and added 25 runs to the original design. Using the Bayesian D-optimal augmentation strategy, we created two augmented designs. For the first design, SSD$(50,100)_1$, every factor was classified as a potential term prior to adding the 25 runs. For the

second design, SSD(50, 100)$_2$, factors $\mathbf{x}_1 - \mathbf{x}_{30}$ were listed as primary and all others potential. Figure 1 shows the grayscale maps of the correlations between the factors. All examples in this paper use $\gamma^2 = 100$ and $\tau^2 = 5$; see Jones et al. (2008).

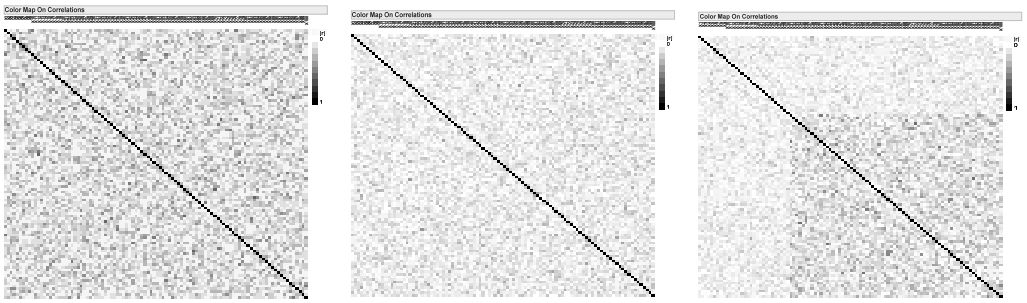

Figure 5. Correlation Grayscale Maps of Supersaturated Designs: SSD(25,100) (L), SSD$(50, 100)_1$ with all potential terms (M), and SSD$(50, 100)_2$ with 30 primary terms (R)

In the grayscale correlation maps, white represents a small correlation between factors (in absolute value), while black represents a perfect correlation. Maximizing the criterion in (4.7) has the byproduct of de-aliasing factors by reducing the correlations between factors. Comparing SSD(25,100) to SSD(50,100)$_1$, it is not surprising the color has lightened; the additional runs reduced the correlations between the 100 factors in the model, thereby increasing the likelihood an active factor will be identified. The difference between SSD(50,100)$_1$ (Figure 5 (M)) and SSD(50,100)$_2$ (5 (R)) shows how classifying factors in the primary group reduces the pairwise correlations between those factors. Analyzing the correlation values makes this relationship clearer.

The average absolution correlation between a group of factors is defined as

$$\overline{|r|} = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} |r_{i,j}|/(k(k-1)/2);$$

where $r_{i,j}$ is the correlation between factors $\mathbf{x}_i$ and $\mathbf{x}_j$. Smaller values $\overline{|r|}$ are preferred. Table 25 compares the designs' absolute average correlations between primary

terms, primary terms and potential terms, and potential terms, denoted by $\overline{|r_{\text{pri}\times\text{pri}}|}$, $\overline{|r_{\text{pri}\times\text{pot}}|}$, and $\overline{|r_{\text{pot}\times\text{pot}}|}$, respectively. First, note that only SSD(50,100)$_2$ differentiates between primary and potential terms, but Table 25 contains values for each group vs. group calculation to highlight how prior information reduces correlations between factors of interest.

Table 25. Average Correlations of Factors in Augmented SSD(50,100)

Correlations	$\overline{\lvert r_{\text{pri}\times\text{pri}}\rvert}$	$\overline{\lvert r_{\text{pri}\times\text{pot}}\rvert}$	$\overline{\lvert r_{\text{pot}\times\text{pot}}\rvert}$	$\overline{\lvert r\rvert}$
SSD(25, 100)	0.150	0.143	0.145	0.145
SSD(50, 100)$_1$	0.078	0.083	0.089	0.086
SSD(50, 100)$_2$	0.064	0.068	0.128	0.097

SSD(25,100) has the highest correlation in all groups because it has the least number of runs. Comparing $\overline{|r_{\text{pri}\times\text{pri}}|}$ and $\overline{|r_{\text{pri}\times\text{pot}}|}$ for SSD(50,100)$_1$ and SSD(50,100)$_2$ reveals that identifying factors as primary terms reduces the correlation between those factors. The average absolute correlations between factors $\mathbf{x}_1, \ldots, \mathbf{x}_{30}$ are lower in SSD(50,100)$_2$ (0.064) than in SSD(50,100)$_1$ (0.078) because we specified the terms *a priori* as primary factors of interest and the design criterion in (4.7) forces the additional runs to reduce the correlations between those factors. Note, however, that the reduced correlations of $\overline{|r_{\text{pri}\times\text{pri}}|}$ and $\overline{|r_{\text{pri}\times\text{pot}}|}$ for SSD(50,100)$_2$ were offset by a higher $\overline{|r_{\text{pot}\times\text{pot}}|}$.

To compare the designs further, define the maximum absolute correlation of factors in a group

$$|r|^{\max} = \max_{i \neq j} |r_{i,j}|.$$

Smaller values are also preferred here. Let $|r_{\text{pri}\times\text{pri}}|^{\max}$,$|r_{\text{pri}\times\text{pot}}|^{\max}$, and $|r_{\text{pot}\times\text{pot}}|^{\max}$ denote the maximum absolute correlations of factors in the primary, primary and potential, and potential groups, respectively. Table 26 shows the augmented designs have smaller values than the original SSD(25, 100), as expected. Further, classifying

factors as primary reduces the maximum absolute correlation between those factors in SSD(50,100)$_2$ compared to SSD(50,100)$_1$.

Table 26. Maximum Correlations of Factors in Augmented SSD(50,100)

Correlations	$\lvert r_{\text{pri}\times\text{pri}}\rvert^{\max}$	$\lvert r_{\text{pri}\times\text{pot}}\rvert^{\max}$	$\lvert r_{\text{pot}\times\text{pot}}\rvert^{\max}$	$\lvert r\rvert^{\max}$
SSD(25, 100)	0.603	0.603	0.603	0.603
SSD(50, 100)$_1$	0.281	0.414	0.361	0.414
SSD(50, 100)$_2$	0.250	0.327	0.560	0.560

4.4 Comparisons

In this section, we compare the performance of Bayesian D-optimal SSDs to the extended $E(s^2)$-optimal designs. Gupta et al. (2010) added runs to two $E(s^2)$-optimal designs, SSD(8,13) and SSD(7,15). For SSD(8,13), Gupta et al. listed the best $n_2 =$ 1, 2, 3, and 4 run(s) to add to the original design to minimize $E(s^2)$. For SSD(7,15), they listed the best $n_2 = 3$ additional runs. We highlight these examples because, to date, they are the only two-level augmented SSDs in the literature. The additional runs suggested in Gupta et al. are optimal with respect to $E(s^2)$, but the runs are independent of the initial data. Hence, for a given SSD(n_1, k) and number of new runs, n_2, the same additional runs are suggested. In contrast, the Bayesian D-optimal augmentation method uses information from the first n_1 runs to improve the selection of the additional n_2 runs.

We perform a side-by-side comparison of the proposed methods with the following methodology: First, we randomly created two main-effect models to study for both SSD(8,13) and SSD(7,15). Each model was randomly chosen to have 3 to 5 active factors with effect sizes drawn uniformly between -15 and 15. The location of the active factors was also random. All responses were generated from the models with $\sigma^2 = 1$. Next, we added the extended $E(s^2)$-optimal runs prescribed in Gupta et al.

and recorded the new response(s). For the Bayesian D-optimal approach, the initial design and response variables were analyzed. Then, we classified factors into their appropriate groups, added the required number of runs by maximizing the objective function in (4.7), and recorded the new responses. Finally, we analyzed the screening results of the final Bayesian D-optimal augmented SSDs and Gupta et al.'s final extended $E(s^2)$-optimal SSDs to see which strategy provides a better recovery of the underlying model.

The SSDs in this section are analyzed with basic regression methods and screening techniques: forward and all-subsets regression (for up to 5 factors) and Half Normal plots, which visually identify factors whose effects seem larger than random noise (Daniel, 1959). While traditional regression analysis methods do not always work well when used for the analysis of SSDs, the supposition is that if augmentation works well for the traditional methods, it will work well for more sophisticated analysis methods. All analysis results were calculated using the JMP software. For forward regression, terms were added based on a p-value to enter of 0.05.

4.4.1 Adding runs to an $E(s^2)$-optimal SSD(8,13).

Consider the $E(s^2)$-optimal SSD(8,13) in Table 27 (Runs 1-8) with responses generated from the equations

1. $\mathbf{y}_1 = 10\mathbf{x}_3 + 8\mathbf{x}_4 + 6\mathbf{x}_5 - 9\mathbf{x}_{11} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$; and

2. $\mathbf{y}_2 = -10\mathbf{x}_4 + 12\mathbf{x}_5 + 7\mathbf{x}_6 - 11\mathbf{x}_{10} - 6\mathbf{x}_{13} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_8)$.

Table 27 also contains the n_2=1, 2, 3, and 4 runs to add suggested by Gupta et al. to create SSD($8+n_2$,13) $E(s^2)$-optimal designs along with the appropriate responses. Again, we emphasize that extended $E(s^2)$-optimality recommends the same runs for each model, whereas the runs added via our Bayesian D-optimal approach will be different for each model.

Table 27. $E(s^2)$**-optimal SSD(8,13) and additional 1, 2, 3, & 4 runs to create extended** $E(s^2)$**-optimal designs, as presented in Gupta et al. (2010)**

Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{y}_1$	$\mathbf{y}_2$
1	1	1	1	1	1	1	1	1	1	1	1	1	1	15.320	-6.433
2	1	1	1	-1	-1	1	-1	-1	-1	1	-1	-1	1	3.588	-11.122
3	1	-1	-1	-1	1	-1	1	-1	1	-1	-1	1	1	-3.159	19.684
4	1	-1	1	1	1	-1	-1	-1	-1	-1	1	-1	-1	14.380	12.237
5	-1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1	1	1.696	-22.798
6	-1	1	-1	-1	1	-1	-1	1	-1	1	1	1	-1	-20.391	8.646
7	-1	-1	1	-1	-1	1	1	1	1	-1	1	-1	-1	-12.956	21.218
8	-1	-1	-1	1	-1	1	1	-1	-1	1	-1	1	-1	0.306	-20.313
9	1	1	1	1	-1	-1	1	1	-1	-1	-1	1	-1	20.707	-12.700
9	1	1	1	1	-1	-1	1	1	-1	-1	-1	1	-1	18.236	-11.398
10	-1	-1	-1	-1	1	1	-1	-1	1	1	1	-1	1	-19.953	12.007
9	-1	-1	-1	1	1	1	-1	-1	1	1	1	-1	1	-4.712	-9.609
10	1	1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1	-3.304	47.024
11	1	-1	1	-1	-1	-1	-1	1	1	1	-1	1	-1	6.918	-13.521
9	1	1	1	1	-1	-1	1	1	-1	-1	-1	1	-1	21.600	-11.596
10	1	-1	-1	1	1	1	-1	1	1	1	-1	-1	-1	14.120	4.539
11	-1	1	-1	-1	1	1	1	-1	-1	-1	1	-1	1	-20.515	33.206
12	-1	-1	1	-1	-1	-1	-1	-1	1	1	1	1	1	-13.339	-25.110

Table 28. Analysis of initial SSD(8,13) data and classification of factors. Active factors are identified with a •

Analysis Results		$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$
Model 1	Half Normal													
	Forward	•			•	•						•		
	All Subsets	•		•	•	•								
	Classification	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$
Model 2	Half Normal													
	Forward		•		•	•					•	•		
	All Subsets				•	•	•				•			•
	Classification	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$

Using Half Normal Plots, forward regression, and all-subsets regression, we analyzed the response variables from SSD(8,13) and classified factors as primary, secondary, or potential. Table 28 summarizes the initial analysis results. For example, the Half Normal Plot failed to indicate any factor to be significantly greater than experimental noise for either model. However, applying forward regression on $\mathbf{y}_1$ selected factors $\mathbf{x}_4, \mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_{11}$ as the top four "active" factors. Further analysis on the first 8 runs with all-subsets regression indicated factors $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4$, and $\mathbf{x}_5$ are of particular interest, as the top models contain only those four factors. Coupled with the results from forward regression, five factors are likely to be active: $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$, and $\mathbf{x}_{11}$. If the analysis stopped here, all true active factors would be identified - $\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$, and $\mathbf{x}_{11}$ - but a false effect would remain, $\mathbf{x}_1$. Augmenting the design with additional runs may help resolve this issue. Based on the initial results, these five factors of interest were classified as primary terms, as indicated by $\boldsymbol{\beta}_{\text{pri}}$ in Table 28. All other terms were classified as potential because there was no indication any other factor was truly active.

A similar approach was carried out to analyze $\mathbf{y}_2$. Forward regression identified $\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_{10}$, and $\mathbf{x}_{11}$ as potentially active, whereas the best five-term model selected with all-subsets regression contained $\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_{10}$, and $\mathbf{x}_{13}$. The union of terms were placed in the primary group; all others were classified as potential. Next, runs were added to SSD(8,13) to get more information out of the respective models. The suggested $n_2 = 1$, 2, 3, or 4 Bayesian D-optimal runs to add for each model are listed in Table 29.

Table 30 compares the final analysis results of $\mathbf{y}_1$ on SSD(9,13), SSD(10,13), SSD(11,13), and SSD(12,13). The true underlying model contained the active factors $\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$, and $\mathbf{x}_{11}$. These factors, and only these factors, were identified by at least one analysis method in each of the Bayesian D-optimal designs. In all extended

Table 29. Additional 1, 2, 3, & 4 Bayesian D-optimal runs for SSD(8,13). Runs 9_i^*-$(9+j)_i^*$ are Bayesian D-optimal for responses $\mathbf{y}_i$

	Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{y}_1$	$\mathbf{y}_2$
1 Run	9_1*	-1	1	1	1	1	1	1	-1	-1	-1	-1	-1	1	32.764	
	9_2*	1	1	1	1	1	1	-1	-1	1	-1	-1	-1	-1		25.358
2 Runs	9_1*	1	-1	-1	1	-1	-1	1	1	-1	1	1	-1	1	-17.549	
	10_1*	-1	-1	1	1	1	-1	1	1	-1	1	-1	-1	1	34.161	
	9_2*	1	1	-1	-1	-1	-1	1	-1	-1	-1	1	1	1		-5.498
	10_2*	1	1	-1	-1	1	1	1	-1	-1	-1	-1	1	-1		46.855
3 Runs	9_1*	-1	-1	1	1	1	1	-1	1	-1	-1	-1	1	1	30.908	
	10_1*	-1	-1	-1	-1	1	1	-1	-1	1	1	1	-1	1	-20.592	
	11_1*	1	-1	-1	1	-1	1	-1	1	-1	-1	1	1	1	-17.837	
	9_2*	1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1		11.525
	10_2*	1	-1	1	-1	-1	-1	1	1	1	1	1	-1	1		-23.607
	11_2*	-1	-1	1	1	1	1	-1	1	-1	1	-1	-1	1		-7.140
4 Runs	9_1*	-1	-1	1	-1	-1	-1	-1	-1	1	1	1	1	1	-13.432	
	10_1*	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	1	-6.887	
	11_1*	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-34.060	
	12_1*	-1	-1	1	-1	1	1	-1	1	-1	-1	-1	1	1	16.459	
	9_2*	1	1	1	-1	-1	-1	1	1	1	1	-1	1	-1		-14.887
	10_2*	-1	1	1	-1	1	1	1	-1	-1	-1	-1	-1	-1		45.149
	11_2*	-1	-1	1	-1	1	-1	1	1	-1	1	-1	-1	1		1.230
	12_2*	-1	1	1	-1	-1	-1	1	-1	-1	-1	1	1	1		-3.211

Table 30. Final analysis of $\mathbf{y}_1$ on SSD(8+n_2,13): Comparing extended $E(s^2)$-optimal SSDs and augmented Bayesian D-optimal SSDs. Active factors are identified with a •

Factors		$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	Correct?
	True Model			•	•	•						•			
9 Run $E(s^2)$	Half Normal	•			•	•						•			
	Forward	•			•	•			•			•			
	All Subsets	•		•	•	•									
9 Run Bayes D	Half Normal														
	Forward			•	•	•			•			•			
	All Subsets			•	•	•						•			•
10 Run $E(s^2)$	Half Normal	•		•	•										
	Forward	•		•	•	•								•	
	All Subsets	•		•	•	•									
10 Run Bayes D	Half Normal			•	•	•			•			•			
	Forward			•	•	•			•			•			
	All Subsets			•	•	•						•			•
11 Run $E(s^2)$	Half Normal	•		•	•	•									
	Forward	•			•	•						•			
	All Subsets	•		•	•	•									
11 Run Bayes D	Half Normal			•	•	•						•			•
	Forward			•	•	•						•			•
	All Subsets			•	•	•						•			•
12 Run $E(s^2)$	Half Normal	•	•		•	•						•			
	Forward	•			•	•			•			•			
	All Subsets	•			•	•						•			
12 Run Bayes D	Half Normal			•	•	•						•			•
	Forward			•	•	•						•			•
	All Subsets			•	•	•						•			•

$E(s^2)$-optimal designs, $\mathbf{x}_1$ was incorrectly selected as an active factor, a Type I error. Moreover, all three analysis methods correctly identified all active factors for the 11-run and 12-run Bayesian D-optimal designs. The results suggest using information from the initial design can improve the selection of additional runs and ultimately improve screening results. This example also highlights that having a false effect, $\mathbf{x}_1$, labeled as a primary factor after the first 8 runs is helpful because the new runs will test to see if it is truly active. Table 31 compares the final analysis results of the data generated from the second model. Analysis of $\mathbf{y}_2$ is more consistent between the $E(s^2)$ and Bayesian D-optimal SSDs than for $\mathbf{y}_1$, but note for the 9 and 10-run designs, the Bayesian design performed better with respect to forward regression.

4.4.2 Adding runs to an $E(s^2)$-optimal SSD(7,15).

Consider the $E(s^2)$-optimal SSD(7,15) in Table 32 (Runs 1-7) with responses generated from the equations

1. $\mathbf{y}_1 = -8\mathbf{x}_5 - 3\mathbf{x}_{10} + 11\mathbf{x}_{14} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_7)$.

2. $\mathbf{y}_2 = -10\mathbf{x}_2 + 6\mathbf{x}_4 + 3\mathbf{x}_7 + 11\mathbf{x}_9 + 5\mathbf{x}_{13} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_7)$.

In this example, we can afford to add three more runs to the design. Table 32 contains the three runs suggested by Gupta et al. (Runs 8-10), as well as the three runs created with the Bayesian D-optimal method. Note again that the new runs under the Bayesian approach are different for each model. The runs were added based on the classification presented in Table 33.

An initial analysis of $\mathbf{y}_1$ on SSD(7,15) correctly identified the true active factors. To confirm the results, $\mathbf{x}_5, \mathbf{x}_{10}$, and $\mathbf{x}_{14}$ were placed in the primary group while all others were classified as potential terms. For $\mathbf{y}_2$, the analysis of SSD(7,15) was more challenging. A Half Normal Plot did not indicate any factors were substantially larger than experimental noise. Forward regression, on the other hand, selected

Table 31. Final analysis of $\mathbf{y}_2$ on SSD(8+n_2,13): Comparing extended $E(s^2)$-optimal SSDs and augmented Bayesian D-optimal SSDs

Factors		$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	Correct?
	True Model				•	•	•				•			•	
9 Run $E(s^2)$	Half Normal														
	Forward				•				•		•	•	•		
	All Subsets				•	•	•				•			•	•
9 Run Bayes D	Half Normal														
	Forward				•	•	•				•			•	•
	All Subsets				•	•	•				•			•	•
10 Run $E(s^2)$	Half Normal			•											
	Forward				•						•	•			
	All Subsets				•	•	•				•			•	•
10 Run Bayes D	Half Normal														
	Forward				•	•	•				•			•	•
	All Subsets				•	•	•				•			•	•
11 Run $E(s^2)$	Half Normal														
	Forward				•	•	•				•			•	•
	All Subsets				•	•	•				•			•	•
11 Run Bayes D	Half Normal				•	•					•				
	Forward				•	•	•				•			•	•
	All Subsets				•	•	•				•			•	•
12 Run $E(s^2)$	Half Normal														
	Forward				•	•	•				•			•	•
	All Subsets				•	•	•				•			•	•
12 Run Bayes D	Half Normal				•	•	•				•			•	•
	Forward				•	•	•				•			•	•
	All Subsets				•	•	•				•			•	•

Table 32. Adding 3 runs to an $E(s^2)$-optimal SSD(7,15): Runs 8-10 are extended $E(s^2)$-optimal from Gupta et al.; Runs 8_i^*-10_i^* are Bayesian D-optimal for responses $\mathbf{y}_i$

Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{y}_1$	$\mathbf{y}_2$
1	-1	1	-1	-1	-1	-1	1	1	1	-1	-1	1	1	-1	-1	-0.700	3.381
2	-1	-1	-1	-1	1	-1	-1	1	-1	-1	-1	-1	-1	1	1	4.847	-16.135
3	-1	1	1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	0.429	-25.445
4	1	-1	-1	1	-1	-1	-1	-1	-1	1	-1	1	1	1	-1	16.358	8.879
5	1	-1	-1	-1	1	1	1	-1	1	1	1	1	-1	-1	1	-20.882	13.271
6	-1	1	1	1	-1	-1	-1	-1	1	1	1	-1	-1	-1	1	-6.671	-0.594
7	1	-1	1	1	-1	1	1	1	1	-1	1	-1	1	1	1	21.237	33.838
8	1	1	1	-1	1	-1	1	-1	1	1	-1	-1	1	1	1	0.547	3.185
9	1	-1	1	1	1	-1	-1	1	1	-1	1	1	-1	-1	-1	-16.836	18.931
10	-1	1	-1	1	-1	1	1	-1	-1	-1	1	1	-1	1	1	22.934	-13.361
8_1*	1	1	1	1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	2.962	
9_1*	1	1	-1	1	1	1	-1	-1	-1	-1	1	1	1	-1	-1	-15.616	
10_1*	1	-1	1	1	1	-1	1	1	-1	1	-1	-1	1	-1	-1	-21.181	
8_2*	1	1	-1	1	1	1	-1	1	1	-1	-1	1	1	-1	-1		8.842
9_2*	1	1	1	-1	-1	1	-1	-1	-1	-1	1	-1	-1	-1	1		-34.834
10_2*	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1		-18.872

Table 33. Analysis of initial SSD(7,15) data and classification of factors

Analysis Results		$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$
Model 1	Half Normal					•					•				•	
	Forward					•					•				•	
	All Subsets					•					•				•	
	Classification	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pri}}$	$\boldsymbol{\beta}_{\text{pot}}$
Model 2	Half Normal	•														
	Forward	•			•	•				•	•					
	All Subsets		•	•	•			•	•		•		•	•		
	Classification	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{sec}}$	$\boldsymbol{\beta}_{\text{pot}}$	$\boldsymbol{\beta}_{\text{pot}}$

$\mathbf{x}_1, \mathbf{x}_9, \mathbf{x}_5, \mathbf{x}_{10}, \mathbf{x}_4$ as the five most important factors. All-subsets regression presented conflicting results because the best five-factor model only contained one factor in the best four-factor model. Factors $\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_{10}, \mathbf{x}_{12}$, and $\mathbf{x}_{13}$ were all flagged in either the best four-factor or five-factor model in all-subsets regression. Coupled with the factors from forward regression, this creates 11 factors of interest.

Because 11 factors are of interest and the final design will only have 10 runs, all factors cannot be listed as primary terms. Moreover, there is not substantial evidence to suggest some of the 11 factors are likely more active than the others, but evidence does suggest these 11 factors are more important than the four factors not detected by any analysis method. Therefore, we classified these 11 factors as secondary and classified the remaining four as potential terms.

The final analysis of both models is presented in Table 34. For the first model, both the $E(s^2)$ and Bayesian D designs performed well. For the second model, however, the Bayesian design performed better. In the Half Normal Plot, factors $\mathbf{x}_1, \mathbf{x}_9, \mathbf{x}_5$, and $\mathbf{x}_2$ were deemed active using the method proposed by Gupta et al., but only factors $\mathbf{x}_2$ and $\mathbf{x}_9$ are truly active. Further, factors $\mathbf{x}_4, \mathbf{x}_7$, and $\mathbf{x}_{13}$ were not detected, even though they are active. In contrast, the Half Normal Plot for the Bayesian D-optimal SSD(10,15) correctly identified only the five important factors. Forward regression and all-subsets regression also indicate the Bayesian D-optimal method is favorable, as forward regression on the extended $E(s^2)$-optimal SSD(10,15) detected $\mathbf{x}_1, \mathbf{x}_9, \mathbf{x}_5, \mathbf{x}_2, \mathbf{x}_8$ as important, whereas forward regression on the extended Bayesian D-optimal SSD(10,15) marked the true active factors as important: $\mathbf{x}_9, \mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_{13}, \mathbf{x}_7$.

4.5 Discussion and Conclusions

We adapted Bayesian D-optimality to add runs to existing supersaturated designs by using information from the initial experiment. After running and analyzing

Table 34. Final analysis of $\mathbf{y}_1$ and $\mathbf{y}_2$ on SSD(10,15): Comparing extended $E(s^2)$-optimal SSDs and augmented Bayesian D-optimal SSDs

Factors		$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	Correct?
Model 1	True Model					•					•				•		
10 Run $E(s^2)$	Half Normal					•					•	•			•		
	Forward					•					•				•		•
	All Subsets					•					•				•		•
10 Run Bayes D	Half Normal					•					•				•		•
	Forward					•					•				•		•
	All Subsets					•					•				•		•
Model 2	True Model		•		•			•		•				•			
10 Run $E(s^2)$	Half Normal	•	•			•				•							
	Forward	•	•			•			•	•							
	All Subsets	•				•			•						•	•	
10 Run Bayes D	Half Normal		•		•			•		•				•			•
	Forward		•		•			•		•				•			•
	All Subsets		•		•			•		•				•			•

an SSD(n_1, k), an experimenter can classify factors as primary, secondary, or potential depending on how active they appear to be. Using this prior information, n_2 runs are added to form a Bayesian D-optimal augmented SSD$(n_1 + n_2, k)$. The comparison study in Section 4.4 indicates the augmentation strategy can perform well against previous methods where designs are augmented to maintain $E(s^2)$-optimality independently of the data.

Our goal with this paper is to introduce the method, but several points deserve explanation. Additional runs are chosen to maximize the Bayesian D-optimality criterion, which is dependent on a classification of factors. The initial classification can play a role in the reliability of the method, but misclassification is not always troubling. In Section 4.4.1, an inactive factor, $\mathbf{x}_1$, was listed as a primary term because the complicated confounding pattered in the SSD(8,13) inflated its initial parameter estimate. The additional runs reduced the bias from the true active factors, so in the final design, the parameter estimate for $\mathbf{x}_1$ was no longer artificially inflated. The misclassification was not detrimental to the screening process. We have seen some models where an incorrect initial classification led to more Type I or Type II errors than the extended $E(s^2)$-optimal designs, but this is not a surprising result. Regardless of the optimality criterion used to add runs, both the initial design and augmented design are still supersaturated with complicated aliasing structures. As such, there is always a risk of not finding the true active factors. Our methodology, however, is more general than the extended $E(s^2)$-optimality approach, as it can augment any SSD with any number of designed runs, whereas extended $E(s^2)$-optimal designs are only known for certain combinations of n_1, n_2, and k. Moreover, our technique can easily extend to SSDs with more than two levels, and while we employed the Coordinate Exchange Algorithm, different design algorithms could be applied if desired.

Another important issue, suggested by one referee, is the determination of n_2 if the decision maker asked for a recommendation. In other words, given n_1, what will be an ideal n_2? This is a sensible issue; we hope that we will be able to report some findings in the near future. In a perfect world, n_2 would be as large as possible while keeping within the screening budget. The results in Section 4.4.1 provide evidence to this because the simulation results improved as more runs were added. Of course, all SSDs take place in a constrained environment. If the budget was highly constrained, an experimenter is already taking on a certain amount of risk. Some research suggests SSDs work best when k is no more than $2n$ (Marley and Woods, 2010). Thus, an initial suggestion to a decision maker on n_2 may be to add at least n_2 runs to make $n_1 + n_2 > .5k$. With that said, the presented method can still augment an existing SSD with any number of runs.

V. Constructing Supersaturated Designs with High Resolution-Rank

This article addresses concerns with the $E(s^2)$-optimality criterion for balanced, two-level supersaturated designs and introduces a catalogue of new designs with high resolution-rank, a criterion that directly assesses a supersaturated design's ability to detect active factors. Several of the designs presented are provably optimal. The search for supersaturated designs with high resolution-rank is aided by binary integer programming and design isomorphism properties.

5.1 Introduction

Supersaturated designs are fractional factorial designs with k factors and $n < k+1$ runs, denoted SSD(n, k). They are used in screening experiments when a great number of input factors must be efficiently separated into (i) the small number of factors that actually produce an effect on the output, and (ii) the larger pool of irrelevant factors. Call these factors *active* and *inactive*, respectively. The difficulty with such a screening experiment is that we do not know which or how many of the k factors are active. To put another way, assume the input-output relationship of the system can be mathematically expressed with the linear main-effects model

$$\mathbf{y} = \beta_0\mathbf{1} + \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k + \boldsymbol{\epsilon} \tag{5.1}$$

where $\mathbf{y}$ is the response vector, $\beta_0, \ldots, \beta_k$ are the $p = k+1$ unknown model parameters, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_n)$ is the error term. The goal of SSDs is to identify the nonzero β_i's. It is, of course, impossible to simultaneously calculate the ordinary least squares (OLS) estimates for all main effects because $n < k+1 = p$ (i.e the system is underdetermined). But, if the assumption of *effect sparsity* holds true and $g < n$

factors are indeed active (Box and Meyer, 1986), it would be possible to calculate the OLS estimates of a g-term model, provided the g factor columns formed a linearly independent set. Therefore, it seems reasonable to construct designs capable of estimating the largest g-term main-effect model possible. This article describes construction algorithms for such designs.

Some notation and background discussion are necessary. This paper focuses on two-level SSDs with an even number of runs, n. Let SSD(n, k) be an $n \times k$ design matrix of $+1$'s and -1's, which represent each factor's high-level and low-level setting, respectively. Each row of the matrix contains the factor-level settings for all k factors in an experimental run. We assume that each column $\mathbf{x}_i$ has the same number of $+1$'s and -1's; such a design is called *balanced* and ensures each factor's orthogonality to the grand mean. Further, no two columns are completely aliased. Thus, if $i \neq j$, $\mathbf{x}_i \neq \mathbf{x}_j$ and $\mathbf{x}_i \neq -\mathbf{x}_j$ for all $i, j \in \{1, 2, \ldots, k\}$. Consequently, a balanced two-level SSD(n, k) can have at most $\binom{n}{n/2}/2 = \binom{n-1}{n/2-1}$ factors. The number of factors, k, is therefore bounded by

$$n - 1 < k \leq \binom{n-1}{n/2-1}. \tag{5.2}$$

The most prevalent criterion to measure the quality or goodness of a balanced two-level SSD is $E(s^2)$. For a matrix $\mathbf{X} = (\mathbf{x}_1 | \ldots | \mathbf{x}_k)$, $E(s^2) = \sum_{i<j} s_{ij}^2/(k(k-1)/2)$, where s_{ij} is the (i, j)th element of $\mathbf{X}'\mathbf{X}$. Reducing a design's $E(s^2)$ effectively reduces the average correlation between all possible pairs of factors. $E(s^2)$ has a long history in the study of SSDs; it was proposed by Booth and Cox (1962) and later studied by Lin (1993), Wu (1993), Nguyen (1996), Tang and Wu (1997), Bulutoglu and Cheng (2004), Suen and Das (2010), and many others. Designs that are optimal with respect to the $E(s^2)$ criterion, however, may not adequately be able to detect all active factors. Consider the SSD(8,10) in Table 35.

Table 35. $E(s^2)$-optimal SSD(8,10)

Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$
1	−	−	−	−	−	+	−	+	+	−
2	+	−	−	+	+	+	−	−	−	−
3	−	−	+	−	+	−	−	+	−	+
4	+	+	+	+	−	+	+	+	−	+
5	−	+	+	+	−	−	−	−	+	+
6	+	−	−	+	+	−	+	+	+	+
7	+	+	−	−	−	−	+	−	−	−
8	−	+	+	−	+	+	+	−	+	−

The design is $E(s^2)$-optimal with $E(s^2) = 4.2667$. As noted earlier, another important quality of a design is its ability to estimate the largest g-term main-effect model possible. The $E(s^2)$-optimal design in Table 35 can technically estimate a model with $n - 1 = 7$ factors (or $n = 8$ total effects with the intercept term) because rank($\mathbf{X}$)=7. However, the design cannot estimate *all* seven-factor models because not all sets of seven columns in $\mathbf{X}$ are linearly independent. For example, the five-column submatrix $\mathbf{X}^* = (\mathbf{x}_1|\mathbf{x}_3|\mathbf{x}_4|\mathbf{x}_7|\mathbf{x}_9)$ has rank($\mathbf{X}^*$) = 4. A main-effect model with these active factors would not be estimable because $\mathbf{b}^* = (\mathbf{X}^{*\prime}\mathbf{X}^*)^{-1}\mathbf{X}^{*\prime}\mathbf{y}$ cannot be calculated. The design is only guaranteed to estimate all four-factor models because all $\binom{10}{4} = 210$ four-column sets of Table 35 are linearly independent, and there exists at least one set of five columns, $\mathbf{X}^*$, that is linearly dependent. A preferable SSD(8,10) is presented in Table 36. The design is also $E(s^2)$-optimal but it has the capability of estimating all six-factor models because all $\binom{10}{6} = 210$ six-column sets are linearly independent. We will prove in later sections that this is the best possible SSD(8,10) with respect to model estimation.

Table 36. Alternative $E(s^2)$-optimal SSD(8,10) with Resolution-Rank=6

Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$
1	+	+	+	+	+	+	+	+	+	+
2	+	+	+	+	−	−	−	−	−	−
3	+	+	−	−	+	+	+	−	−	−
4	+	−	−	−	+	−	−	+	+	+
5	−	−	+	−	−	+	−	+	−	−
6	−	−	+	+	+	−	+	−	+	−
7	−	+	−	−	−	+	−	−	+	+
8	−	−	−	+	−	−	+	+	−	+

There are several criteria based on a design's ability to project into the largest possible estimable model (see §5.2). Here, we focus on the *resolution-rank* criterion presented in Deng et al. (1996, 1999) because it was the first model estimation criterion proposed to study SSDs. Two formal definitions are as follows:

Definition (Deng et al., 1996) Let $\mathbf{X} = (\mathbf{x}_1|\ldots|\mathbf{x}_k)$ be an $n \times k$ matrix. The *resolution-rank* of $\mathbf{X}$ (r-rank for short) is defined as $\max\{g$: for any $(\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ig})$ of $\mathbf{X}$, the set $\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ig}$ is linearly independent$\}$.

Definition (Lin, 2003) Let $\mathbf{X}$ be an equal occurrence (aka balanced) design matrix. The r-rank of $\mathbf{X}$ is defined as $g = d - 1$, where d is the minimum number of subset columns that are linearly dependent.

The design in Table 35 has r-rank = 4 while the design in Table 36 has r-rank = 6. While these designs have equivalent $E(s^2)$ values, it is often the case that $\mathbf{X}_1$ = SSD$_1(n, k)$ and $\mathbf{X}_2$ = SSD$_2(n, k)$ have different $E(s^2)$ values if r-rank$(\mathbf{X}_1) \neq r$-rank$(\mathbf{X}_2)$. It is also possible for two designs to have equivalent r-ranks and different $E(s^2)$ values. In this paper, we search for balanced two-level SSDs with high r-rank, and if more than one design for a given r-rank is discovered, we use $E(s^2)$ as a secondary criterion and report the design with the lowest $E(s^2)$ value.

In Section 5.2, we review similar criteria to r-rank and discuss past results. In Section 5.3, we introduce new techniques to search for designs with high r-rank. We seek the largest k such that SSD(n, k) has r-rank = g by formulating the search as a set-covering problem. Then, we present a search algorithm based on design equivalence. Results and new designs are presented in Section 5.4 and a general summary concludes the paper in Section 5.5.

5.2 Design Criteria

Several design criteria measure the ability of an SSD to estimate all g-term models, where g is as large as possible. The r-rank of a design is such a criterion. Others include a design's *resolving power*, *estimation capacity* (EC_g), and *MDS-abberation*. Each is described in this section.

5.2.1 Resolving Power of Search Designs.

To detect the few non-negligible factors in an experiment from the many negligible, Srivastava (1975) introduced *search linear models* (SLM), which are represented as

$$E(\mathbf{y}) = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2, \tag{5.3}$$

where $\mathbf{y}$ is the response vector with $\mathrm{Var}(\mathbf{y}) = \sigma^2\mathbf{I}$, $\mathbf{X}_1(n \times k_1)$ and $\mathbf{X}_2(n \times k_2)$ are known model matrices, $\boldsymbol{\beta}_1$ is a vector of k_1 fixed, but unknown, effects of primary interest, and $\boldsymbol{\beta}_2$ is a vector k_2 effects in which it's assumed at most g elements of $\boldsymbol{\beta}_2$ are nonzero. The identity of the g elements are unknown. If the nonzero elements of $\boldsymbol{\beta}_2$ are not correctly identified, they will be a source of bias for $\boldsymbol{\beta}_1$. Srivastava accordingly proposed a *search* of the nonzero terms in $\boldsymbol{\beta}_2$. In the case of SSDs, the only fixed effect, $\mathbf{X}_1$ in (5.3), is the intercept term, $\mathbf{1}(n \times 1)$. All other factors are unknown. Therefore, for an SSD design matrix $\mathbf{X}$=SSD(n, k), Equation (5.3) can be rewritten as

$$E(\mathbf{y}) = \mathbf{1}\boldsymbol{\beta}_0 + \mathbf{X}\boldsymbol{\beta}, \tag{5.4}$$

with $\mathrm{Var}(\mathbf{y}) = \sigma^2\mathbf{I}$. Here, $\boldsymbol{\beta}$ is the vector of the k effects in which it's assume at most g elements are nonzero. Srivastava gave the following theorem:

Theorem 5.2.1 *In the noiseless linear search model* $\mathbf{y} = \mathbf{1}\boldsymbol{\beta}_0 + \mathbf{X}\boldsymbol{\beta}$*, a necessary and sufficient condition to find* f *active factors is that for every* $(n \times 2f)$ *submatrix* $\mathbf{X}^*$

of $\mathbf{X}$*,*

$$\text{rank}(\mathbf{1}|\mathbf{X}^*) = 1 + 2f.$$

The condition is equivalent to: $\text{rank}(\mathbf{X}^*) = 2f$ for every submatrix $\mathbf{X}^*$ of $\mathbf{X}$. This highlights a difference between a design's ability to *estimate* a model and its ability to *differentiate* between models. A design that can *estimate* all g-term models can *differentiate* between all competing $f = g/2$-term models. Two models, each with f-terms, will not be able to produce the same output since the combined set of factors has at most $2f = g$ columns, which are linearly independent and can therefore only produce one output. The quality of every $(n \times 2f)$ submatrix being linear independent is called the P_{2f} *property* or *resolving power*. For more on linear search designs, see Ghosh and Avila (1985), Morgan et al. (2012), and references therein.

5.2.2 Model Robust Supersaturated Designs and $g_{\max}$.

Jones et al. (2009) extended the work of Li and Nachtsheim (2000)'s model robust factorial designs into *model-robust supersaturated* (MRSS) designs. They state "a key characteristic of any supersaturated design is its estimation capacity EC_g," where

$$EC_g = \frac{\text{number of estimable } g\text{-term main-effect models}}{\text{total number of } g\text{-term main-effect models}}. \tag{5.5}$$

There are $\binom{k}{g}$ g-term main-effect models and an $EC_g = 1.0$ implies all are estimable. For a given design, the criterion of interest, $g_{\max}$, is defined as

$$g_{\max} := \text{the maximum value of } g \text{ for which } EC_g \text{ is 1.0.} \tag{5.6}$$

A design's $g_{\max}$ is equivalent to its r-rank because a design is only estimable if its factor columns are linearly independent. Jones et al. used a computer search to create MRSS designs (and consequently designs with high r-rank) that addressed the

following questions: (1) For a given SSD(n, k), what is the maximum number of active factors, g, that can be accommodated with $EC_g = 1.0$? (2) For a given number of factors, k, and an upper bound g on the number of active factors, how small can the sample size be, retaining $EC_g = 1.0$? And, (3) For a given sample size n and an upper bound g on the number of active factors, how many factors k can an MRSS design incorporate, while maintaining $EC_g = 1.0$? The authors searched for designs to answer the above questions using heuristic computer searchers. We extend their work and pay particular attention to question (3) because answering it effectively answers question (1) and (2).

5.2.3 MDS-resolution.

Two recent papers by Miller and Tang (2012, 2013) considered *minimal dependent sets* (MDSs) to evaluate SSDs. For a given matrix, an MDS is a set of column vectors that are linearly dependent, but if you remove any one column, the remaining subset becomes linearly independent. The size of an MDS is the number of columns in the set. For a matrix, $\mathbf{X}$, let A_j be the number of MDSs of size j. This defines an MDS sequence $(A_1, A_2, \ldots, A_k)$. The MDS-resolution is defined as the smallest j such that $A_j \neq 0$. More succinctly, MDS-resolution $:= \min_{A_j \neq 0} j$. The MDS resolution is the size of the smallest dependent set of columns. Thus, MDS-resolution = r-rank + 1.

Another useful criterion is MDS-aberration. When comparing the MDS sequences of two designs, find the smallest j such that the A_j's are not equal. The matrix with the smaller A_j is said to have less MDS-abberation. Miller and Tang created some MDS-optimal SSDs using computer searchers for designs with a few more columns than rows. Specifically, they searched for SSD(n, k) designs where $k = n, n+1, n+2, n+3$, and $n+4$. These designs can be said to be "oversaturated" since they have only a few columns beyond saturation (Deng et al., 1996).

5.2.4 Summary.

Note that r-rank, $g_{\max}$, and the P_{2f} property, despite subtle nuances, are equivalent. All provide the answer to the question: given a matrix, $\mathbf{X}$, what is the maximum number g such that all g columns are linearly independent? This number is exactly one less than $\mathbf{X}$'s MDS-resolution. If one column is removed from the smallest set linearly dependent columns, the remaining g will be linearly independent. The motivation for each criterion can be summarized as:

1. When a design matrix is projected into *any* submatrix of g or fewer factors, we want the main effects of the design to be estimable. This implies there is a least one $g + 1$ main effect model that is not estimable.

2. We also want designs with the ability to differentiate or discriminate between competing models.

We now describe the methodology to find the find the largest number of columns k such that SSD(n, k) has a given r-rank $= g$. Note that calculating the r-rank can be a time-consuming task. To test if r-rank$(\mathbf{X}) = g$, the rank of all $\binom{k}{g}$ sets of columns must be calculated. As k grows, the calculations become infeasible. In fact, the problem can be shown to be NP-hard (Khachiyan, 1995). Given this constraint, most research on model discrimination for SSDs has focused on designs where n and k are not exceedingly large. We focus on the r-rank of designs where n=6, 8, 10, and 12, and prove optimality in cases $n = 6$ and $n = 8$.

5.3 Methodology

To explore the r-rank of designs, it is helpful to establish lower and upper bounds. Let $\mathbf{X} = \text{SSD}(n, k)$ and $\mathbf{1}$ be an $n \times 1$ column of 1's. Since $\mathbf{X}$ is balanced, $\mathbf{1}'\mathbf{X} = \mathbf{0}'$. Thus, the left null space of $\mathbf{X}$ is nonempty and rank$(\mathbf{X}) \leq n - 1$. And, by definition,

it is clear that r-rank($\mathbf{X}$) $\leq$ rank($\mathbf{X}$), so r-rank($\mathbf{X}$) $\leq n-1$. The lower bound for r-rank($\mathbf{X}$) was shown to be three (Miller and Tang, 2012). The bounds for $g = r$-rank($\mathbf{X}$) are then

$$3 \leq g \leq n-1. \tag{5.7}$$

From Eq. 5.2, an SSD with n runs can have at most $k = \binom{n-1}{n/2-1}$ factors. Combined with Eq. 5.7, we know that for any n, SSD$(n, \binom{n-1}{n/2-1})$ has r-rank = 3. Intuitively, it makes sense that designs with large k have a low r-rank because the presence of many columns means there is a greater chance for a set to be linearly dependent. As the number of columns in a design decreases, the r-rank can only increase. The difficult part is deciding which columns to remove from the full design to create a new design with a larger r-rank.

5.3.1 Formulation as a Set Covering Problem.

Let $\mathbf{X}$ be the full SSD(n, k), where $k = \binom{n-1}{n/2-1}$. Any design with n rows and r-rank > 3 contains a subset of columns from the full design, $\mathbf{X}$. The goal is to minimize the number of columns deleted from $\mathbf{X}$ such that the remaining design has r-rank = 4. That is to say, no four columns of the resulting matrix can form a dependent set. For $i = 1, \ldots, k$, let $c_i = 1$ if column $\mathbf{x}_i$ is deleted from $\mathbf{X}$, 0 otherwise. Denote the remaining columns of $\mathbf{X}$ as $\mathbf{X}^*$. Thus, the objective is

$$\min \sum_{i=1}^{k} c_i, \text{ subject to } r\text{-rank}(\mathbf{X}^*) = 4. \tag{5.8}$$

To accomplish this, enumerate all 4-tuples of $\mathbf{X}$ that are linearly dependent. Say there are m of these, denoted $A_1, A_2, \ldots, A_m$. At least one entry from each A_i must be deleted, i.e. $\sum_{j \in A_i} c_j \geq 1$ for $i = 1, \ldots, m$. This creates a Set Covering Problem because the columns deleted from $\mathbf{X}$ must "cover" the dependent 4-tuples. Written

another way, let $\mathbf{A}$ be an $m \times k$ binary matrix where row i of $\mathbf{A}$ contains 1's in the column entries corresponding to A_i, 0 otherwise. Eq. 5.8 can then be written as a binary integer program:

$$\min \sum_{i=1}^{k} c_i \text{ such that } \begin{cases} \mathbf{Ac} \geq \mathbf{1} \\ \\ c_i \in \{0, 1\} \end{cases}, \tag{5.9}$$

where $\mathbf{c}' = (c_1, \ldots, c_k)$. As an example, consider the full SSD for $n = 6$, SSD$(6, 10)$ in Table 37.

Table 37. Full SSD(6,10)

Run	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$
1	+	+	+	+	+	+	+	+	+	+
2	+	+	+	+	−	−	−	−	−	−
3	+	−	−	−	+	+	+	−	−	−
4	−	+	−	−	+	−	−	+	+	−
5	−	−	+	−	−	+	−	+	−	+
6	−	−	−	+	−	−	+	−	+	+

All $\binom{10}{4} = 210$ sets of four columns were tested for linearly dependencies. If the set of columns formed a dependent set (i.e. had rank 3), the indices of the columns were marked in a row of the binary matrix, $\mathbf{A}_6$. 15 problem cases were detected, resulting in

$$\mathbf{A}_6 = \begin{array}{c} \begin{array}{cccccccccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 & \mathbf{x}_8 & \mathbf{x}_9 & \mathbf{x}_{10} \end{array} \\ \left(\begin{array}{cccccccccc} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array} \right) \end{array} \tag{5.10}$$

The second row of $\mathbf{A}$, for instance, reveals that $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_7, \mathbf{x}_9\}$ of SSD$(6, 10)$ form a dependent set. Formulating the problem as described in Eq. 5.9 allows the use of the `bintprog` function in MATLAB. One solution to the problem is given by

$$\mathbf{A}_6 = \begin{array}{c} \begin{array}{cccccccccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 & \mathbf{x}_8 & \mathbf{x}_9 & \mathbf{x}_{10} \end{array} \\ \left(\begin{array}{cccccccccc} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array} \right) \end{array} \tag{5.11}$$

At least one entry from each row is "covered" by the gray columns in Eq. 5.11. Deleting the corresponding rows from Table 37 results in the design shown in Table

38. Because at least one entry from all dependent sets has been removed, the design has r-rank ≥ 4. In this case, the design actually has r-rank $= 5$; it is therefore optimal.

Table 38. SSD(6,6) with r-rank = 5

1	2	3	4	5	6
+	+	+	+	+	+
+	+	−	−	−	−
−	−	+	+	−	−
−	−	+	−	+	−
+	−	−	−	−	+
−	+	−	+	+	+

We also applied the methodology to the full SSD with $n = 8$, SSD$(8, 35)$. The binary matrix $\mathbf{A}_8$ contained 630 row (i.e. constraints) and consequently took many hours to solve. The `bintprog` program identified 21 columns to delete from the full matrix, resulting in the SSD$(8, 14)$ design in Table 39 with r-rank $= 4$.

Table 39. SSD(8,14) with r-rank = 4

1	2	3	4	5	6	7	8	9	10	11	12	13	14
+	+	+	+	+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	−	−	−	−	−	−	−	−
+	+	−	−	−	−	+	+	+	+	−	−	−	−
+	−	+	−	−	−	+	−	−	−	+	+	+	−
−	−	+	+	−	−	−	+	+	−	+	−	−	+
−	−	−	+	+	−	+	−	−	+	−	+	−	+
−	+	−	−	−	+	−	+	−	−	−	+	+	+
−	−	−	−	+	+	−	−	+	+	+	−	+	−

The set cover formulation of the problem quickly reaches computational infeasibility. For $n = 10$, the full SSD(10,126) contains 20475 dependent 4-tuples. The process of finding and generating $\mathbf{A}_{10}$ is itself a computationally expensive task; finding the minimum set cover of such a large binary matrix is impractical. The full SSD(12,462) contains 623700 constraints, and our code failed to even find all dependent 4-tuples of SSD(14,1716), as this would require the search and rank calculations for $\binom{1716}{4} = 3.6003 \times 10^{11}$ column sets. To further complicate the issue, the constraints are not redundant; in other words, there is no known way to reduce the number of

rows in $\mathbf{A}_n$. This realization highlights how difficult it is to search for r-rank optimal SSDs. Nevertheless, the set cover formulation can be applied to a general SSD, as opposed to the full design, to see how many columns can be removed to increase the r-rank. For example, the SSD(8,10) in Table 36 can be found by searching for all dependent 6-tuples in the optimal SSD(8,14). We note, however, that this does not guarantee the optimal number of columns for a given r-rank because the column search is limited to a subset of the full design.

5.3.2 Design Equivalence Extension Algorithm.

The construction of SSDs with high r-rank can also be accomplished with an extensive computer search of non-equivalent designs. Two designs, $\mathbf{X}_1$ and $\mathbf{X}_2$, are said to be *equivalent* if one can be obtained from the other through a series of row permutations, column permutations, multiplying row i by -1 ($1 \leq i \leq n$), or multiplying column j by -1 ($1 \leq j \leq k$) (McKay, 1979). Thus, designs that are equivalent to each other maintain the same statistical properties; for example, rank. As an extension, two equivalent designs must also have the same r-rank. The motivation behind working with design equivalence is that it reduces the number of distinct matrices in the search set. For instance, SSD(8,35) has $\binom{35}{4} = 52360$ four-column sets, of which 51730 are linearly independent (the other 630 make up the rows of the aforementioned $\mathbf{A}_8$). The 51730 designs can be reduced to just 12 non-equivalent designs, a substantial decrease. All design equivalence reductions in this paper were performed with the program `nauty` (**n**o **aut**omorphisms, **y**es?) from McKay and Piperno (2013), a free program in C.

For a given n and r-rank $= g$, we seek the largest k such that SSD(n, k) satisfies r-rank $= g$. Our algorithm is as follows:

1. Enumerate and store all non-equivalent, linearly independent g-factor, n-run balanced designs.

2. Extend each stored design with all possible balanced columns.

3. Reduce the entire set of extended designs to a set of non-equivalent designs.

4. Enumerate and store all non-equivalent extended designs with r-rank = g. Delete all others.

5. Repeat Steps 2-4 until the number of viable extensions goes to 0.

A similar algorithm was performed by Miller and Tang (2013), though they did not use `nauty` and only searched for designs with up to $k = n+4$ columns. In comparison, the final iteration of our algorithm will contain the set all maximum non-equivalent SSDs with n rows and r-rank = g.

We carried out the algorithm on $n = 6$ and $n = 8$ for $g = 4, \ldots, n-1$. For $n = 6$, one non-equivalent SSD(6,6) was discovered for both $g = 4$ and $g = 6$. The design is therefore equivalent to the design in Table 38 found via set covering. For $n = 8$, the results are more interesting. As mentioned, there are 12 non-equivalent 8×4 balanced designs with rank 4. Each of the 12 designs in this "seed" was extended with the columns from the full SSD(8,35) to create a class of 8×5 balanced designs. The set was reduced to a set of mutually exclusive, non-equivalent designs using `nauty`. All designs with r-rank = 4 were stored, leaving 29 unique 8×5 balanced designs with r-rank 4. The process continued adding columns until the algorithm could no longer find a viable extension. The final design was an SSD(8,14) with r-rank = 4, proving that 14 is the maximum number of columns an 8-run SSD can have to maintain r-rank = 4. Note the same design, shown previously in Table 39, was also found via set covering.

Table 40 shows the number of $8 \times k$ non-equivalent designs with r-rank $= g$ as k starts at g and increases until no extensions are plausible. The final designs SSD(8,14) with r-rank $= 4$; SSD(8,12) with r-rank $= 5$; and SSD(8,10) with r-rank $= 6$ are all unique up to equivalence and contain the largest number of columns for the given r-rank. For $n = 8$ and $g = 7$, 38 non-equivalent designs with $k = 8$ were created. Thus, SSD(8,8) is the largest design with r-rank $= 7$, but there are 38 such designs. One design, for instance, has $E(s^2) = 10.8571$ while another has $E(s^2) = 4.00$. The design with $E(s^2) = 4.00$ is shown in Section 5.4.1 along with SSD(8,10) and SSD(8,12).

Table 40. Number of $8 \times k$ non-equivalent designs with r-rank $= g$

k	g			
	4	5	6	7
4	12	.	.	.
5	29	28	.	.
6	80	77	73	.
7	185	171	144	135
8	314	253	101	38
9	345	202	6	.
10	205	61	1	.
11	61	8	.	.
12	11	1	.	.
13	3	.	.	.
14	1	.	.	.

5.3.3 Heuristic Search.

Like the set cover formulation, the design equivalence extension algorithm quickly becomes computationally intractable. For $n = 10$, there are 28 10×4 non-equivalent balanced designs with rank 4, 168 10×5 designs with r-rank 4, 1668 10×6 designs, and 21445 10×7 designs. The computer froze at the point; extending each of the 21445 designs with every column from SSD(10, 126) caused a memory error. To combat this, we modified the above algorithm; rather than a seed set of all non-

equivalent $n \times g$ designs, we started the extension algorithm on a random set of non-equivalent $n \times (n-1)$ designs with rank $n-1$. The motivation for this was based on our observation that SSDs with high r-rank have rank $n-1$. Further, a full rank $n \times (n-1)$ design by definition satisfies the property that all $\binom{n-1}{g}$ set of columns are linearly independent for $g = 2, \ldots, n-1$. In other words, it represents a starting point for any design with any r-rank. We then add columns to this design, testing if each extension maintains r-rank $= g$. As with every heuristic algorithm, optimality is not guaranteed, but we can still get new and improved results. The heuristic algorithm is as follows:

1. Create a random set of $n \times (n-1)$ balanced designs and store all designs with rank $n-1$.
2. Reduce the designs to create a set of non-equivalent designs with rank $n-1$.
3. Extend each stored design with a random set of balanced columns.
4. Reduce the entire set of extended designs to a set of non-equivalent designs.
5. Enumerate and store all non-equivalent extended designs with r-rank $= g$. Delete all others.
6. Repeat Steps 3-5 until the number of viable extensions goes to 0.

We applied the heuristic algorithm to search for SSDs with $n = 10$ and $n = 12$ with $g = 4, \ldots, n-1$. For $n = 10$, we discovered an SSD(10,13) with r-rank $= 8$, improving the SSD(8,12) with r-rank $= 8$ in Miller and Tang (2013). We also found new several new designs for $n = 12$. All designs are shown in Section 5.4.2. Unfortunately, this heuristic algorithm eventually blows up, computationally speaking. The bottleneck occurs because calculating the r-rank is NP-hard; hence, calculating the r-rank of

thousands of extended designs quickly becomes impractical. Nevertheless, we were able to apply the algorithm and find new SSDs with high r-rank.

5.4 Designs with High Resolution Rank

5.4.1 Provable Optimal Designs.

The set covering algorithm and design equivalence extension algorithm found the provably largest k such that SSD(n, k) has a given r-rank where $n = 6$ and $n = 8$. This section includes designs not already shown in the paper.

Table 41. SSD(8,12) with r-rank = 5

1	2	3	4	5	6	7	8	9	10	11	12
+	+	+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	−	−	+	−	−	−	−
+	+	+	−	−	+	−	−	+	−	−	+
+	−	−	+	−	−	+	−	−	−	+	−
−	+	−	−	−	+	+	+	−	+	+	−
−	−	+	−	−	−	+	−	+	+	−	−
−	−	−	+	+	+	−	−	−	+	−	+
−	−	−	−	+	−	−	+	+	−	+	+

Table 42. SSD(8,10) with r-rank = 6

1	2	3	4	5	6	7	8	9	10
+	+	+	+	+	+	+	+	+	+
+	+	+	+	−	−	−	−	−	−
+	+	−	−	+	+	+	−	−	−
+	−	−	−	+	−	−	+	+	+
−	−	+	−	−	+	−	+	−	−
−	−	+	+	+	−	+	−	+	−
−	+	−	−	−	+	−	−	+	+
−	−	−	+	−	−	+	+	−	+

Table 43. SSD(8,8) with r-rank = 7

1	2	3	4	5	6	7	8
+	+	+	+	+	+	+	+
+	+	+	+	+	−	−	−
+	+	−	−	−	+	−	+
+	−	+	−	−	−	+	−
−	+	−	+	−	−	+	+
−	−	+	−	+	+	−	+
−	−	−	+	−	+	−	−
−	−	−	−	+	−	+	−

5.4.2 Improved or New Designs.

The designs in this section were created with the heuristic design equivalence extension algorithm. Only designs that represent new and improved designs are shown. The SSD(10,15) with r-rank = 6 is included because the SSD(10,15) in Jones et al. (2009) actually had r-rank = 5.

Table 44. SSD(10,13) with r-rank = 8

1	2	3	4	5	6	7	8	9	10	11	12	13
+	+	+	+	+	+	+	+	+	+	+	+	+
−	−	−	−	+	+	+	−	−	+	+	+	+
+	+	+	−	−	+	+	+	−	−	−	−	+
−	−	−	+	+	−	−	+	−	−	+	−	+
−	−	−	+	−	−	+	+	+	+	−	−	−
−	+	−	+	−	+	−	−	+	+	+	−	+
+	−	+	+	−	−	−	−	−	−	−	+	−
+	+	−	−	+	−	−	−	−	+	−	+	−
+	−	+	−	−	−	+	−	+	−	+	−	−
−	+	+	−	+	+	−	+	+	−	−	+	−

Table 45. SSD(10,15) with r-rank = 6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
−	+	+	−	−	+	−	−	−	−	+	−	−	−	+
−	+	−	−	−	−	−	−	+	+	−	+	−	+	+
+	−	−	−	+	+	−	+	+	−	−	−	−	+	−
−	−	+	+	+	+	+	−	−	−	−	+	−	+	−
+	+	+	+	+	−	−	−	−	+	−	−	+	−	−
−	−	−	−	−	+	−	+	−	+	+	+	+	−	−
+	+	−	+	−	−	+	−	+	−	+	+	+	−	−
−	−	+	+	−	−	+	+	+	−	−	−	+	−	+
+	−	−	−	+	−	+	+	−	+	+	−	−	+	+

Table 46. SSD(12,17) with r-rank = 9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+
−	−	−	−	−	+	+	+	+	+	+	+	+	+	+	+	−
−	−	−	+	+	−	−	−	−	−	−	+	+	+	+	+	−
−	−	+	+	+	−	+	+	+	+	+	−	−	−	−	+	−
−	+	+	−	+	+	−	−	+	+	+	−	−	+	+	−	+
−	+	+	+	−	+	+	+	−	−	+	−	+	−	+	−	+
+	−	+	−	+	+	−	+	−	−	−	+	−	−	+	−	−
+	−	+	+	−	+	+	−	−	+	−	−	+	+	−	+	+
+	+	−	−	+	−	+	−	−	+	+	+	+	−	−	+	−
+	+	−	+	−	−	+	−	+	+	−	+	−	+	+	−	+
+	+	−	+	+	+	−	+	+	−	−	−	−	−	−	+	+
+	+	+	−	−	−	−	+	+	−	+	+	+	+	−	−	−

Table 47. SSD(12,18) with r-rank = 8

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
+	−	−	−	+	−	−	−	+	+	−	−	+	−	+	+	−	−
−	+	+	−	−	+	+	+	−	−	+	−	+	−	+	+	−	−
−	+	−	−	+	−	+	−	−	+	+	+	+	+	−	−	−	−
+	−	−	+	+	+	−	+	+	−	+	−	+	−	−	−	+	+
−	+	+	−	+	−	+	−	+	−	−	+	−	−	−	−	−	+
+	−	−	+	−	−	+	−	−	+	−	−	+	−	−	−	+	+
−	−	+	+	−	+	−	−	−	+	−	+	−	+	+	−	−	+
−	−	−	−	−	−	+	+	−	+	+	+	−	−	+	+	+	+
+	+	+	−	+	+	−	−	−	−	−	−	−	+	+	−	+	−
−	−	+	+	−	−	−	+	+	−	+	−	−	+	−	+	+	−
+	+	−	+	−	+	−	+	+	−	−	+	−	+	−	+	−	−

Table 48. SSD(12,22) with r-rank = 7

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
+	−	−	−	+	−	−	−	+	+	−	−	+	−	+	+	−	−	+	−	+	−
−	+	+	−	−	+	+	+	−	−	+	−	+	−	+	+	−	−	+	+	−	+
−	+	−	−	+	−	+	−	−	+	+	+	+	+	−	−	−	−	+	−	−	−
+	−	−	+	+	+	−	+	+	−	+	−	+	−	−	−	+	+	+	−	−	+
−	+	+	−	+	−	+	−	+	−	−	+	−	−	−	−	−	−	−	−	+	+
+	−	−	+	−	−	+	−	−	+	−	−	+	−	−	−	+	−	−	+	+	−
−	−	+	+	−	+	−	−	−	+	−	+	−	+	+	−	−	+	−	+	−	+
−	−	−	−	−	−	+	+	−	+	+	+	−	−	+	+	+	+	−	−	+	+
+	+	+	−	+	+	−	−	−	−	−	−	−	+	+	−	+	+	−	+	+	−
−	−	+	+	−	−	−	+	+	−	+	−	−	+	−	+	+	−	+	−	−	−
+	+	−	+	−	+	−	+	+	−	−	+	−	+	−	+	−	+	−	+	−	−

Table 49. SSD(12,24) with r-rank = 6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
+	−	−	−	+	−	−	−	+	+	−	−	+	−	+	+	−	+	+	−	−	−	−	−
−	+	+	−	−	+	+	+	−	−	+	−	+	−	+	+	−	+	+	+	−	−	−	+
−	+	−	−	+	−	+	−	−	+	+	+	+	+	−	−	−	−	−	−	+	−	+	−
+	−	−	+	+	+	−	+	+	−	+	−	+	−	−	−	+	+	−	+	−	−	+	−
−	+	+	−	+	−	+	−	+	−	−	+	−	−	−	−	+	+	−	+	+	+	−	+
+	−	−	+	−	−	+	−	−	+	−	−	+	−	−	−	−	−	−	−	+	+	−	+
−	−	+	+	−	+	−	−	−	+	−	+	−	+	+	−	−	−	−	+	−	+	+	+
−	−	−	−	−	−	+	+	−	+	+	+	−	−	+	+	+	−	+	−	−	+	+	−
+	+	+	−	+	+	−	−	−	−	−	−	−	+	+	−	+	−	+	−	+	+	−	−
−	−	+	+	−	−	−	+	+	−	+	−	−	+	−	+	+	+	+	−	−	−	+	+
+	+	−	+	−	+	−	+	+	−	−	+	−	+	−	+	−	−	−	+	+	−	−	−

Table 50. SSD(12,41) with r-rank = 5

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
+	−	−	−	+	−	−	−	+	+	−	−	+	−	+	+	−	+	+	−	−
−	+	+	−	−	+	+	+	−	−	+	−	+	−	+	+	−	+	+	+	−
−	+	−	−	+	−	+	−	−	+	+	+	+	+	−	−	−	−	−	−	+
+	−	−	+	+	+	−	+	+	−	+	−	+	−	−	−	+	+	−	+	−
−	+	+	−	+	−	+	−	+	−	−	+	−	−	−	−	+	+	−	+	+
+	−	−	+	−	−	+	−	−	+	−	−	+	−	−	−	−	−	−	−	+
−	−	+	+	−	+	−	−	−	+	−	+	−	+	+	−	−	−	−	+	−
−	−	−	−	−	−	+	+	−	+	+	+	−	−	+	+	+	−	+	−	−
+	+	+	−	+	+	−	−	−	−	−	−	−	+	−	+	−	+	−	+	−
−	−	+	+	−	−	−	+	+	−	+	−	−	+	−	+	+	+	+	−	−
+	+	−	+	−	+	−	+	+	−	−	+	−	+	−	+	−	−	−	+	+

22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
−	−	+	+	−	−	−	+	−	−	−	−	+	+	+	+	−	+	−	−
−	−	−	−	+	+	−	−	+	+	+	+	−	−	−	+	−	−	+	−
−	−	−	−	−	−	−	−	−	+	+	−	−	+	−	−	+	−	+	+
−	+	−	−	−	+	−	+	−	+	+	−	+	+	+	−	−	+	−	+
+	+	−	+	+	+	+	+	+	−	−	−	+	−	−	+	−	−	+	+
+	−	−	−	+	−	+	−	−	−	+	+	+	−	−	+	+	+	+	−
+	−	+	−	−	−	+	−	−	−	−	−	+	+	+	−	+	+	+	+
+	+	+	−	+	+	−	+	+	−	−	+	−	−	−	−	−	+	−	+
+	+	−	+	−	−	−	−	+	+	−	+	−	−	+	−	+	−	−	−
−	−	+	+	+	+	+	+	−	+	−	−	−	+	+	−	+	−	−	−
−	+	+	+	−	−	+	−	+	−	+	+	−	−	−	+	−	−	−	−

5.4.3 Summary of Known Designs.

All results for the r-rank of SSDs with $n = 6, 8, 10$, or 12 are summarized in Table 51. Some of the designs were also discovered independently by Jones et al. (2009) and Miller and Tang (2013). New designs presented here are marked in bold, and designs proven to have the maximum number of columns for a given r-rank are indicated with "*". The designs with $n = 10$ are thought to be near optimal because we performed an extensive, although incomplete, search of the design space.

Table 51. Maximum k such that SSD(n, k) has r-rank $= g$. Numbers with * are optimal, and numbers in bold signify new designs.

g	n			
	6	8	10	12
3	10*	35*	126*	462*
4	6*	14*	24	48
5	6*	12*	23	**41**
6	.	10*	15	**24**
7	.	8*	14	**22**
8	.	.	**13**	**18**
9	.	.	11	**17**
10	.	.	.	15
11	.	.	.	14

5.5 Discussion

This paper explored the creation of balanced, two-level supersaturated designs with high resolution-rank. Specifically, we searched for the largest number of columns, k, such that SSD(n, k) has r-rank $= g$. New designs and a summary of all known SSDs with $n = 6, 8, 10,$, and 12 were reported. Moreover, we discussed the overall difficulty of finding SSDs with high r-rank. The search quickly becomes intractable, even for a moderate number of runs, n. In future work, we hope to run the algorithms presented here on a more powerful computer.

VI. Summary and Conclusions

Supersaturated designs can be used in large screening experiments when the number of factors exceeds the number of available runs. This dissertation explored the construction, analysis, and data-driven augmentation of such designs. Chapter II gave an introduction to large screening experiments and discussed the basics of supersaturated designs. It also presented a brief summary of construction and analysis techniques, as well as a novel approach to add runs to the designs to discriminate competing models. More in-depth research was done in Chapters III, IV, and V, which form the heart of this research. Each chapter addressed a specific challenge in the field and proposed an original research contribution to address the challenge.

First, the difficulty when using a supersaturated design stems from the inability to simultaneously estimate all main-effects in a linear model. Numerous analysis methods for supersaturated designs have been proposed, but no method can be guaranteed to find the true underlying model. Chapter III provided detailed explanations and examples of why this occurs. Further, it proposed two analysis methods to mitigate Type I errors and presented a comprehensive (and corrected) review of past simulation studies on the Williams (1968)'s data set. Moreover, the chapter serves as a good introduction to supersaturated designs for those not familiar with the topic.

In any experiment, the only way to get more information is to collect more data. Supersaturated designs, with their limited run-size, would certainly provide more definitive results if the experimenter could perform more experimental runs. Chapter IV developed the methodology to do this. The specific research question was presented as: "Suppose after running an SSD(n_1, k), the experimenter can afford n_2 more runs to resolve ambiguities. What is the best way to augment the original design to reduce uncertainty and get the most information out of the final SSD$(n_1 + n_2, k)$?" A Bayesian D-optimality augmentation technique was described which uses information

gained from the initial experiment to strategically plan the best follow-up runs. In a simulation study, the proposed method outperformed the augmentation strategy presented in Gupta et al. (2010), which added runs based solely on reducing the design's $E(s^2)$.

Chapter V explored the construction of balanced, two-level supersaturated designs with high resolution-rank. The resolution-rank of a design $\mathbf{X}$ is the maximum number g such that all subsets of g columns in $\mathbf{X}$ are linearly independent. It directly measures the ability of a design to estimate models. As such, we searched for the largest designs for a given run-size, n, and resolution-rank, g. Using binary integer programming and design isomorphism properties, we created new designs with high resolution-rank and summarized all results available in the literature.

6.1 Recommendations for Future Research

This work focused on two-level supersaturated designs with continuous factors and a single response variable. It would be interesting to study the proposed methods on mixed-level designs with categorical factors and multiple responses. The formulation of the Bayesian D-optimal augmentation process on mixed-level designs would be particulary interesting, as the study of mixed-level designs seems to be gaining momentum. See, for example, Liu and Liu (2011) and Sun et al. (2011). As these designs become more prevalent, it will be important to find ways to add follow-up runs.

Another interesting area for future work is the exploration of r-rank on nonbalanced designs (i.e. there exists a column, $\mathbf{x}_i$ of $\mathbf{X}$ =SSD(n, k) such that $\mathbf{1}'\mathbf{x}_i \neq 0$. This is certainly the case if n is odd. But, if n is even, it would be interesting to see how r-rank is affected if balance is abandoned. Related work has been done in the field of compressed sensing. See, for example, Candes and Tao (2007).

6.1.1 Compressed Sensing.

While the applied statistics and industrial engineering communities studied and proposed new results for supersaturated designs, the applied mathematics community developed theory for the increasingly popular field compressed sensing. The amount of signal data collected today is huge, and it is impossible to store it all. The idea of compressed sensing is to store a small, linear collection of measurements from the data, so long as it's possible to recover the actual signal. Compressed sensing relies on the assumption of spare signals, just like supersaturated designs rely on the assumption of sparse main-effects. Matrices in compressed sensing are typically much wider than design matrices (e.g. thousands of columns) and have a tremendous amount of columns compared to rows ($k \gg n$). A key performance metric of these matrices is called *Spark*. Before a formal definition, let's review some notation.

6.1.1.1 Notation.

Let $\mathbf{v} = (v_1, v_2, \ldots, v_k)'$ be an $k \times 1$ vector in $\mathbb{R}^k$.

- The *support* of $\mathbf{v}$, denoted $supp(\mathbf{v})$, is defined as the set $supp(\mathbf{v}) = \{i | v_i \neq 0\}$.
- The *zero norm* or ℓ_0*-norm* of $\mathbf{v}$, denoted $\|\mathbf{v}\|_0$, is the number of nonzero elements in $\mathbf{v}$, i.e. $\|\mathbf{v}\|_0 = |supp(\mathbf{v})|$. While this is called a norm, it is not a true norm of a vector space because it does not scale: $\|\alpha\mathbf{v}\|_0 \neq |\alpha|\|\mathbf{v}\|_0$ if $\alpha \neq 0$.

6.1.1.2 Spark.

Definition The Spark of a matrix $\mathbf{X}$ is the size of the smallest subset of columns that are linearly dependent. i.e.

$$\text{Spark}(\mathbf{X}) = \min_{\mathbf{v} \neq 0} \|\mathbf{v}\|_0 \text{ such that } \mathbf{X}\mathbf{v} = 0. \tag{6.1}$$

Notice the Spark of a matrix is identical its MDS-resolution. Thus, Spark($\mathbf{X}$) = r-rank($\mathbf{X}$) - 1.

6.1.2 Random Matrices.

One of the key breakthroughs in compressed sensing was the realization that random matrices are full spark. In other words, if a matrix $\mathbf{M}$ was generated randomly, typically with independent Gaussian or binary ± 1 entries, the matrix would have many desirable properties. What's interesting, perhaps fortuitous, is that supersaturated designs were first proposed as random designs. Satterthwaite (1959) and Budne (1959) suggested designs with a random balance ± 1 entries in each column would be ideal to screen a large number of factors with little runs, assuming effect sparsity. In fact, the second ever issue of the journal *Technometrics* was dedicated to "random balance" designs. What's even more interesting is that the majority of the issue was dedicated to some of the top statisticians of time (e.g. G.E.P. Box, J.W. Tukey, & J.S. Hunter) criticizing the idea and utility of random balance designs (Youden et al., 1959). Here are some amusing quotes from the discussion on the papers of Satterthwaite and Budne:

- "Whenever I have heard presentations of random balance I have always reached the conclusion: 'This is nonsense because there simply aren't enough degrees of freedom to go around'." (O. Kempthorne)

- "I believe Dr. Satterthwaite and Mr. Budne do not have sufficient fear of the difficulty of interpreting random balanced designs." (O. Kempthorne)

- "Dr. Satterthwaite lists 7 circumstances in which random balance is likely to be a good procedure. I found that I could see possibility of agreement with him on none..." (O. Kempthorne)

- "I think this presentation of random balance raises many questions which are important to statistics, even if random balance should fade away." (O. Kempthorne)

- "Random balance is, in part, directed toward the needs of the statistically untrained." (J. W. Tukey)

- "There are those who say that random balance is the 'wave of the future', that most experimentation will come to use random assignment of levels or versions of each factor to trials. They are wrong. There are those who say that random balance is worthless, inefficient, and dangerous. They, too, are wrong, though not quite as wrong." (J. W. Tukey).

- "Let me therefore begin by saying that I believe the only thing wrong with random balance is random balance." (G. E. P. Box)

To be fair, many of the criticism stemmed from claims that random balance designs are easy and suitable replacements for traditional designs ($n > k$). Work in D-optimal designs, non-aliasing designs, etc., show that systematic and computational construction of designs is preferred. On the issue of supersaturated designs, however, reviewers with slightly less critical.

- "I think it is perfectly natural and wise to do some supersaturated experiments. ...[I]t seems to me..., random balance is going to be used completely or indefinitely in the supersaturated region." (J. W. Tukey)

- (On supersaturated designs) "Satterthwaite and his colleagues have done a considerable service to statistics in pointing out the importance of this situation, although I do not believe that even here random balance is the answer." (G. E. P. Box)

Box went on to say supersaturated designs should be constructed systematically, suggesting an efficient design would have the column vectors at maximum angles with one another. Indeed, the first systematic SSDs by Booth and Cox (1962) were introduced a few years later using the $E(s^2)$ criterion, and researchers haven't looked back. Perhaps, however, mathematical theory has caught up to random designs.

For example, in Chapter V, it was shown that an 8-run balanced SSD with r-rank = 4 can have at most 14 columns. If we removed the balanced requirement and construct a design randomly, we can quickly generate an SSD(8,16) with r-rank = 4. Balance is typically a concern with smaller matrices because it ensures no main effect is correlated with the grand mean. But, as n increases, a random column will be "nearly balanced" and the large n will keep the correlation to a minimum. Thus, it may be the case that large two-level supersaturated designs should be constructed randomly to optimize the r-rank. This will be explored further, as it could aid in the construction and application of supersaturated designs.

Bibliography

Abraham, B., Chipman, H., and Vijayan, K. (1999). Some risks in the construction and analysis of supersaturated designs. *Technometrics*, 41(2):135–141.

Atkinson, A. C., Donev, A. N., and Tobias, R. D. (2007). *Optimum experimental designs, with SAS. (Vol. 34)*. Oxford University Press, Oxford.

Beattie, S. D., Fong, D. K. H., and Lin, D. K. J. (2002). A two-stage bayesian model selection strategy for supersaturated designs. *Technometrics*, 44(1):55–63.

Booth, K. and Cox, D. R. (1962). Some systematic supersaturated designs. *Technometrics*, 4(4):489–495.

Box, G. E. P. (1992). George's column. *Quality Engineering*, 5(2):321–330.

Box, G. E. P. and Draper, N. R. (1987). *Empirical Model-Building and Response Surfaces.* John Wiley & Sons, New York, NY.

Box, G. E. P., Hunter, J. S., and Hunter, W. G. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. John Wiley & Sons, Hoboken, NJ.

Box, G. E. P. and Meyer, R. D. (1986). An analysis for unreplicated fractional factorials. *Technometrics*, 28(1):11–18.

Box, G. E. P. and Meyer, R. D. (1993). Finding the active factors in fractionated screening experiments. *Journal of Quality Technology*, 25(2):94–105.

Budne, T. (1959). The application of random balance designs. *Technometrics*, 1(2):139–155.

Bulutoglu, D. A. and Cheng, C. S. (2004). Construction of $E(s^2)$-optimal supersaturated designs. *The Annals of Statistics*, 32(4):1662–1678.

Candes, E. and Tao, T. (2007). The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304.

Daniel, C. (1959). Use of half-normal plots in interpreting factorial two-level experiments. *Technometrics*, 1(4):311–341.

Das, A., Dey, A., Chan, L. Y., and Chatterjee, K. (2008). On $E(s^2)$-optimal supersaturated designs. *Journal of Statistical Planning and Inference*, 138(12):3749–3757.

Deng, L. Y., Lin, D. K. J., and Wang, J. (1996). Marginally oversaturated designs. *Communications in Statistics–Theory and Methods*, 25(11):2557–2573.

Deng, L. Y., Lin, D. K. J., and Wang, J. (1999). A resolution rank criterion for supersaturated designs. *Statistica Sinica*, 9:605–610.

DuMouchel, W. and Jones, B. (1994). A simple bayesian modification of D-optimal designs to reduce dependence on an assumed model. *Technometrics*, 36(1):37–47.

Edwards, D. J. and Mee, R. W. (2011). Supersaturated designs: Are our results significant? *Computational Statistics & Data Analysis*, 55(9):2652–2664.

Georgiou, S. (2012). Supersaturated designs: A review of their construction and analysis. *Journal of Statistical Planning and Inference*, page http://dx.doi.org/10.1016/j.bbr.2011.03.031.

Georgiou, S. D. (2008). Modelling by supersaturated designs. *Computational Statistics & Data Analysis*, 53(2):428–435.

Ghosh, S. and Avila, D. (1985). Some new factor screening designs using the search linear model. *Journal of Statistical Planning and Inference*, 11(2):259–266.

Gilmour, S. G. (2006). Factor screening via supersaturated designs. In Dean, A. and Lewis, S. (Eds.), *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics* (pp 169-190). Springer, New York, NY.

Goos, P. and Jones, B. (2011). *Optimal Design of Experiments*. John Wiley & Sons, New York, NY.

Gupta, S. and Kohli, P. (2008). Analysis of supersaturated designs: A review. *Journal of the Indian Society of Agricultural Statistics*, 62(2):156–168.

Gupta, V. K., Chatterjee, K., Das, A., and Kole, B. (2012). Addition of runs to an s-level supersaturated design. *Journal of Statistical Planning and Inference*, 142(8):2402–2408.

Gupta, V. K., Singh, P., Kole, B., and Parsad, R. (2010). Addition of runs to a two-level supersaturated design. *Journal of Statistical Planning and Inference*, 140(9):2531–2535.

Gutman, A. J., White, E. D., and Hill, R. R. (2013a). Large screening experiments: An overview of supersaturated designs for practitioners. In Krishnamurthy, A. and Chan, W. K. V., editors, *Proceedings of the 2013 Industrial and Systems Engineering Research Conference*: 3328 – 3337.

Gutman, A. J., White, E. D., and Hill, R. R. (2013b). Supersaturated designs for the practitioner: Analytical challenges and new analysis methods. *Submitted to the Journal of Quality Technology*.

Gutman, A. J., White, E. D., Lin, D. K. J., and Hill, R. R. (2013c). Augmenting supersaturated designs with Bayesian D-optimality. *Working Paper*.

Holcomb, D. R. and Carlyle, W. M. (2002). Some notes on the construction and evaluation of supersaturated designs. *Quality and Reliability Engineering International*, 18(4):299–304.

Holcomb, D. R., Montgomery, D. C., and Carlyle, W. M. (2003). Analysis of supersaturated designs. *Journal of Quality Technology*, 35(1):13–27.

Holcomb, D. R., Montgomery, D. C., and Carlyle, W. M. (2007). The use of supersaturated experiments in turbine engine development. *Quality Engineering*, 19(1):17–27.

Jones, B. (2011). A class of three-level screening designs for definitive screening in the presence of second-order effects: Presented at: *Design and Analysis of Experiments*. September 2011, Issace Newton Institute for Mathematical Sciences, Cambridge, UK. Available Online: http://www.newton.ac.uk/programmes/DAE/seminars/090214301.html.

Jones, B. and Dumouchel, W. (1996). Follow-up designs to resolve confounding in multifactor experiments: Discussion. *Technometrics*, 38(4):323–326.

Jones, B., Li, W., Nachtsheim, C., and Ye, K. (2007). Model discrimination: another perspective on model-robust designs. *Journal of statistical planning and inference*, 137(5):1576–1583.

Jones, B., Lin, D. K. J., and Nachtsheim, C. J. (2008). Bayesian D-optimal supersaturated designs. *Journal of Statistical Planning and Inference*, 138(1):86–92.

Jones, B. and Nachtsheim, C. J. (2011). A class of three-level designs for definitive screening in the presence of second-order effects. *Journal of Quality Technology*, 43(1):1–15.

Jones, B. A., Li, W., Nachtsheim, C. J., and Ye, K. Q. (2009). Model-robust supersaturated and partially supersaturated designs. *Journal of Statistical Planning and Inference*, 139(1):45–53.

Khachiyan, L. (1995). On the complexity of approximating extremal determinants in matrices. *Journal of Complexity*, 11(1):138–153.

Kleijnen, J. P. C. (1987). Simulation with too many factors: Review of random and group-screening designs. *European journal of operational research*, 31(1):31–36.

Kleijnen, J. P. C., Bettonvil, B., and Persson, F. (2006). Screening for important factors in large discrete-event simulation models: Sequential bifurcation and its applications. In Dean, A. and Lewis, S. (Eds.), *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics* (pp 287-307). Springer, New York, NY.

Li, P., Zhao, S., and Zhang, R. (2010). A cluster analysis selection strategy for supersaturated designs. *Computational Statistics & Data Analysis*, 54(6):1605–1612.

Li, R. and Lin, D. K. J. (2002). Data analysis in supersaturated designs. *Statistics & Probability Letters*, 59(2):135–144.

Li, R. and Lin, D. K. J. (2003). Analysis methods for supersaturated design: Some comparisons. *Journal of Data Science*, 1(3):249–260.

Li, W. and Nachtsheim, C. (2000). Model-robust factorial designs. *Technometrics*, 42(4):345–352.

Lin, D. K. J. (1993). A new class of supersaturated designs. *Technometrics*, 35(1):28–31.

Lin, D. K. J. (1995a). Generating systematic supersaturated designs. *Technometrics*, 37(2):213–225.

Lin, D. K. J. (1995b). Letters to the editor: Response to "Comments on Lin (1993)". *Technometrics*, 37(3):358–359.

Lin, D. K. J. (2003). Industrial Experimentation for Screening. In Rao, C.R. and Khattree, R. (Eds.), *Handbook of Statistics, Vol. 22, Ch. 2* (33-73). North Holland, New York.

Liu, Y. and Liu, M. Q. (2011). Construction of optimal supersaturated design with large number of levels. *Journal of Statistical Planning and Inference*, 141(6):2035–2043.

Lu, X. and Wu, X. (2004). A strategy of searching active factors in supersaturated screening experiments. *Journal of Quality Technology*, 36(4):392–399.

Marley, C. J. and Woods, D. C. (2010). A comparison of design and model selection methods for supersaturated experiments. *Computational Statistics & Data Analysis*, 54(12):3158–3167.

Matsuura, S., Suzuki, H., Iida, T., Kure, H., and Mori, H. (2010). Robust parameter design using a supersaturated design for a response surface model. *Quality and Reliability Engineering International*, 27(4):541–554.

McKay, B. D. (1979). Hadamard equivalence via graph isomorphism. *Discrete Mathematics*, 27:213–214.

McKay, B. D. and Piperno, A. (2013). *Nauty and Traces User's Guide (Version 2.5)*. Computer Science Department, Australian National University, Canberra, Australia.

Mee, R. (2009). *A Comprehensive Guide to Factorial Two-Level Experimentation*. Springer, New York, NY.

Meyer, R. D., Steinberg, D. M., and Box, G. (1996). Follow-up designs to resolve confounding in multifactor experiments. *Technometrics*, 38(4):303–313.

Meyer, R. K. and Nachtsheim, C. J. (1995). The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, 37(1):60–69.

Miller, A. and Tang, B. (2012). Minimal dependent sets for evaluating supersaturated designs. *Statistica Sinica*.

Miller, A. and Tang, B. (2013). Finding mds-optimal supersaturated designs using computer searches. *Journal of Statistical Theory and Practice*, (just-accepted).

Montgomery, D. C. (2009). *Design and Analysis of Experiments*. John Wiley & Sons, New York, NY.

Morgan, J., Ghosh, S., and Dean, A. (2012). Jn srivastava and experimental design. *Journal of Statistical Planning and Inference*.

Morris, M. C. (2006). An overview of group factor screening. In Dean, A. and Lewis, S. (Eds.), *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics* (pp 191-206). Springer, New York, NY.

Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. (2009). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons, New York, NY.

Neff, A. R. (1996). *Baysian two stage design under model uncertainty*. PhD Dissertation, Virginia Polytechnic Institute and State University.

Nguyen, N. K. (1996). An algorithmic approach to constructing supersaturated designs. *Technometrics*, 38(1):69–73.

Phoa, F. K. H., Pan, Y. H., and Xu, H. (2009). Analysis of supersaturated designs via the Dantzig selector. *Journal of Statistical Planning and Inference*, 139(7):2362–2372.

Plackett, R. L. and Burman, J. P. (1946). The design of optimum multifactorial experiments. *Biometrika*, 33(4):305–325.

Rais, F., Kamoun, A., Chaabouni, M., Claeys-Bruno, M., Phan-Tan-Luu, R., and Sergent, M. (2009). Supersaturated design for screening factors influencing the preparation of sulfated amides of olive pomace oil fatty acids. *Chemometrics and Intelligent Laboratory Systems*, 99(1):71–78.

Ruggoo, A. and Vandebroek, M. (2004). Bayesian sequential DD-optimal model-robust designs. *Computational Statistics & Data Analysis*, 47(4):655–673.

Ryan, K. J. and Bulutoglu, D. A. (2007). $E(s^2)$-optimal supersaturated designs with good minimax properties. *Journal of Statistical Planning and Inference*, 137(7):2250–2262.

Satterthwaite, F. E. (1959). Random balanced experimentation (with discussion). *Technometrics*, 1(2):111–137.

Scinto, P. R., Wilkinson, R. G., and Lin, D. K. J. (2011). Screening for fuel economy: A case study of supersaturated designs in practice. *Quality Engineering*, 23(1):15–25.

Srivastava, J. (1975). Designs for searching non-negligible effects. *A survey of statistical design and linear models*, pages 507–519.

Suen, C. and Das, A. (2010). $E(s^2)$-optimal supersaturated designs with odd number of runs. *Journal of Statistical Planning and Inference*, 140(6):1398–1409.

Sun, F., Lin, D. K. J., and Liu, M. Q. (2011). On construction of optimal mixed-level supersaturated designs. *The Annals of Statistics*, 39(2):1310–1333.

Sundberg, R. (2008). A classical dataset from williams, and its role in the study of supersaturated designs. *Journal of Chemometrics*, 22(7):436–440.

Tang, B. and Wu, C. F. J. (1997). A method for constructing supersaturated designs and its $E(s^2)$ optimality. *Canadian Journal of Statistics*, 25(2):191–201.

Vine, A. E., Lewis, S. M., Dean, A. M., and Brunson, D. (2008). A critical assessment of two-stage group screening through industrial experimentation. *Technometrics*, 50(1):15–25.

Watson, G. S. (1961). A study of the group screening method. *Technometrics*, 3:371–388.

Westfall, P. H., Young, S. S., and Lin, D. K. J. (1998). Forward selection error control in the analysis of supersaturated designs. *Statistica Sinica*, 8:101–118.

Williams, K. R. (1968). Designed experiments. *Rubber Age*, 100:65–71.

Wu, C. F. J. (1993). Construction of supersaturated designs through partially aliased interactions. *Biometrika*, 80(3):661–669.

Wu, C. F. J. and Hamada, M. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*. Wiley, New York, NY.

Yamada, S. (2004). Selection of active factors by stepwise regression in the data analysis of supersaturated design. *Quality Engineering*, 16(4):501–513.

Youden, W., Kempthorne, O., Tukey, J. W., Box, G. E. P., and Hunter, J. (1959). Discussion of the papers of Messrs. Satterthwaite and Budne. *Technometrics*, 1(2):157–184.

Zhang, Q. Z., Zhang, R. C., and Liu, M. Q. (2006). A method for screening active effects in supersaturated designs. *Journal of Statistical Planning and Inference*, 137(6):2068–2079.

Vita

Mr. Alex J. Gutman is a 2003 graduate from Botkins High School in Botkins, Ohio. He attended Wright State University in Dayton, Ohio and graduated Summa Cum Laude in 2007 with a Bachelor of Science degree in Mathematics. He continued at Wright State to pursue a Masters of Science degree in Pure Mathematics, and in the summer of 2008, he worked as an Applied Research Mathematician for the National Security Agency's Graduate Mathematics Program (GMP). Prior to completing his M.S. degree in June 2009, he accepted a job with Riverside Research, where he was placed at the Air Force Institute of Technology's (AFIT's) Center for Operational Analysis (COA). As a Research Associate for the COA, Mr. Gutman has consulted numerous Air Force and Department of Defense organizations on experimental design and statistical analysis techniques. In 2012, he was awarded the Riverside Research Mastrandrea Memorial Award for significant achievement and practical devotion to the company over the past year.

Mr. Gutman began his doctoral studies at AFIT in 2010 with the assistance of Riverside Research's education benefit program. He conducted his dissertation research in the area of experimental design under the guidance of Dr. Edward D. White. Upon graduation, he will continuing working at AFIT.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
13–09–2013	PhD Dissertation	Jan 2010 – Sept 2013

4. TITLE AND SUBTITLE

Construction, Analysis, and Data-Driven Augmentation of Supersaturated Designs

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Gutman, Alex J.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENC-DS-13-S-02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally left blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Screening designs are used in the early stages of industrial and computer experiments to find the most important input factors affecting a system's output. They provide an economical way to remove unimportant factors from further, potentially costly, experimentation. However, when an experiment has a large number of control factors and limited number of available runs, it is infeasible to run a traditional screening design. In these situations, experimenters can use supersaturated designs. A supersaturated design is a fractional factorial design that can screen a set of k factors in n runs, where $k > n - 1$. Unfortunately, they do not always provide definitive results. Improper and incomplete analysis of supersaturated designs can cause an experimenter to misclassify active factors and waste resources in subsequent experiments. In light of these concerns, this research investigates how to construct efficient and effective supersaturated designs, how to analyze such designs, and how to strategically plan follow-up runs to designs.

15. SUBJECT TERMS

Bayesian D-Optimality, Design of Experiments, Optimal Designs, Supersaturated Designs

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Edward D. White
U	U	U	UU	140	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636 ext 4540; edward.white@afit.edu