3-22-2018

Experimental Designs, Meta-Modeling, and Meta-learning for Mixed-Factor Systems with Large Decision Spaces

Zachary C. Little

EXPERIMENTAL DESIGNS, META-MODELING, AND META-LEARNING FOR MIXED-FACTOR SYSTEMS WITH LARGE DECISION SPACES

DISSERTATION

Zachary C. Little

AFIT-ENS-DS-18-M-137

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, Department of Defense, or the United States Government.

AFIT-ENS-DS-18-M-137

EXPERIMENTAL DESIGNS, META-MODELING, AND META-LEARNING FOR MIXED-FACTOR SYSTEMS WITH LARGE DECISION SPACES

DISSERTATION

Presented to the Faculty

Department of Operational Sciences

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 (Operations Research)

Zachary C. Little, BS, MS

March 2018

AFIT-ENS-DS-18-M-137

EXPERIMENTAL DESIGNS, META-MODELING, AND META-LEARNING FOR MIXED-FACTOR SYSTEMS WITH LARGE DECISION SPACES

Zachary C. Little, BS, MS

Committee Membership:

Jeffery D. Weir, PhD
Chair

Raymond R. Hill, PhD
Member

Maj Brian B. Stone, PhD
Member

Maj Jason K. Freels, PhD
Member

ADEDEJI B. BADIRU, PhD
Dean, Graduate School of Engineering and Management

Abstract

Many Air Force studies require a design and analysis process that can accommodate for the computational challenges associated with complex systems, simulations, and real-world decisions. For systems with large decision spaces and a mixture of continuous, discrete, and categorical factors, nearly orthogonal-and-balanced (NOAB) designs can be used as efficient, representative subsets of all possible design points for system evaluation, where meta-models are then fitted to act as surrogates to system outputs. The mixed-integer linear programming (MILP) formulations used to construct first-order NOAB designs are extended to solve for low correlation between second-order model terms (i.e., two-way interactions and quadratics). The resulting second-order approaches are shown to improve design performance measures for second-order model parameter estimation and prediction variance as well as for protection from bias due to model misspecification with respect to second-order terms. Further extensions are developed to construct batch sequential NOAB designs, giving experimenters more flexibility by creating multiple stages of design points using different NOAB approaches, where simultaneous construction of stages is shown to outperform design augmentation overall. To reduce cost and add analytical rigor, meta-learning frameworks are developed for accurate and efficient selection of first-order NOAB designs as well as of meta-models that approximate mixed-factor systems.

Dedicated to my wife and family

Acknowledgments

I would like to thank my advisor, Dr. Weir, whose mentorship has been vital these past few years, and my research committee members, Dr. Hill, Dr. Stone, and Dr. Freels, for their guidance.

This research is in support of the Simulation and Analysis Facility (SIMAF), Air Force Life Cycle Management Center, Simulation and Analysis Division (AFLCMC/XZS).

Zachary C. Little

Table of Contents

Page

List of Figures

Page

List of Tables

Page

EXPERIMENTAL DESIGNS, META-MODELING, AND META-LEARNING FOR MIXED-FACTOR SYSTEMS WITH LARGE DECISION SPACES

I. Introduction

The complexities of real-world choices available to today's decision makers, as well as of simulations that aim to represent environments of ever-increasing fidelity and scope, make it necessary for simulators and analysts to have an experimental design and analysis process that accommodates the associated computational challenges. Simulations and systems with complex behavior often require more computation time and can have large design/decision spaces, making the exhaustive evaluation of all possible options infeasible. Individual decisions are not always quantitative and do not always have the same number of choices, so the ability to provide experimental designs that account for mixed factors (i.e., quantitative and qualitative factors with different numbers of levels for each) is needed.

Often times in studies of simulations having complex behavior, decision makers are interested in which assets to invest in as well as how to employ existing and future assets given expected budget constraints. These potential purchases, upgrades, and utilization decisions comprise portfolio selections in large decision spaces, requiring the use of efficient experimental designs of sufficient quality. The nearly orthogonal-and-balanced (NOAB) mixed design presented in [1] is an appropriate space-filling design for such simulation and decision support efforts due to the robustness of the design method with respect to different factor types. In the literature review, the balance feasibility test and construction method from [1] for NOAB designs with quantitative (discrete and

continuous) and qualitative (categorical) factors are implemented, which are then used to create designs for notional Intelligence, Surveillance, and Reconnaissance (ISR) [2], [3] asset decisions of interest. Input requirements and suggested design sizes (i.e., number of design points) for NOAB design construction are discussed, with examples of portfolio representation and associated NOAB designs presented. An exhaustive search process over balance-feasible design sizes is implemented, allowing for an examination of trade-offs between design size and quality. Techniques for evaluation and comparison of designs are outlined and design performance measures are presented, to include those for prediction accuracy, model coefficient estimation, and model misspecification.

Meta-models can act as surrogates to the actual simulation output in order to facilitate robust decision support processes. If the associated experimental design is constructed with forethought, meta-models can prevent the need for future costly simulation runs when new questions are posed by decision makers as well as prevent unnecessary costs in modeling and analysis. Different from simulation optimization, the interest is not to optimize a single simulation response, but to examine trade-offs between multiple responses for the various decisions of interest. While examining asset choices is a motivation of this work, note that there may be many other factors represented that are controllable within a simulation and uncontrollable in a real-world environment. Such factors can provide greater context when examining trade-offs.

Meta-learning, and the framework of the algorithm selection problem, will be used to efficiently determine which designs and meta-models are most appropriate based on design space and simulation output features, respectively. In order for the individual meta-

learning approaches to be conducted, two training sets are required: one for design spaces for which NOAB designs are to be constructed, and one for complex system responses to be meta-modeled. The review of literature concerning experimental designs, meta-models, and meta-learning approaches is provided in Chapter II.

Four main research questions will be addressed in four papers (Chapters III-VI):

1. *What benefits can second-order extensions to the mixed-integer linear programming (MILP) constructions of first-order NOAB designs provide with respect to design performance measures?* (Chapter III)

Measures of design performance are detailed in Chapter II. A subset of these measures are used for design evaluation and comparison, focusing on design size, prediction accuracy, model coefficient estimation, and protection from model misspecification for various implementations of the developed second-order MILP extensions (i.e., concerning two-way interactions and quadratic model terms). Two main design approaches for mixed-factor problems are developed in the form of *NOAB Resolution IV* designs for screening that protects from bias of second-order model terms and *NOAB Resolution V* designs that can provide better coefficient estimates for full second-order models.

2. *Can construction methods be developed for batch sequential NOAB designs?* (Chapter IV)

Sequential designs and their importance to simulation studies are discussed in Chapter IV. Different from single stage, or *one-shot*, experimental designs, batch sequential designs have multiple stages that allow for intermediate analysis as well as more flexibility in the choice of overall design size and of how later design points are selected.

Two techniques for construction of batch sequential NOAB designs are examined: simultaneous construction of stages and design augmentation. These batch sequential designs can use the different NOAB approaches from Chapter III at different stages to achieve certain design properties.

3. *How can meta-learning be implemented to develop a recommendation system for first-order NOAB design construction that also allows for design evaluation and comparison?* (Chapter V)

Meta-learning approaches from various fields of study are reviewed in Chapter II, which are presented in the context of the algorithm selection problem framework. The aim is to develop such a framework to provide insights regarding initial best practices for first-order NOAB design construction. In [1], NOAB designs are shown to be superior or as good as many other space-filling designs for continuous and discrete factors, and a basic guideline for lower and upper bounds on design size is given. Using an algorithm selection problem framework, a greater understanding of the resulting design performance measures for various design sizes and balance settings allows for the development of a recommendation system that efficiently selects designs to construct, with the potential for a meta-learning process that updates the recommendation system as new design spaces are examined.

4. *How do the newly developed mixed-factor designs compare with respect to resulting meta-model performance, and after a design is selected, how well do meta-model recommendation systems perform with respect to recommendation and ranking?* (Chapter VI)

Several candidate meta-models are reviewed from the literature in Chapter II. A simulation case study demonstrates the overall meta-modeling methodology as well as allows for comparison of the different NOAB design approaches developed in Chapter III.

Using an algorithm selection problem framework informed by [4], [5], a training set of complex system responses is examined for meta-learning for a mixed-factor design space.

II. Literature Review

2.1 Overview of Literature

In order to address the questions of interest, a review of literature is required covering a variety of research areas. The representation of large decision spaces and associated combinatorial challenges are discussed in the context of ISR portfolio selection problems. An overview of experimental designs, desired design properties, and associated measures of performance are presented, with emphasis on the nearly orthogonal-and-balanced (NOAB) mixed-factor designs and construction method. Additionally, techniques for design comparison and evaluation are considered. The potential for right-censored responses is detailed for the ISR portfolio example, with discussion of design and meta-model considerations for censored and survival data.

Candidate meta-modeling techniques are outlined, where an aim in this research is to sufficiently describe entire response surfaces from various systems or simulations with mixed factors. The algorithm selection problem and concept of meta-learning are presented, with a summary and update of the survey paper [6] that generalizes meta-learning approaches from various fields of study. Table 1 presents literature sources with the associated topics of interest.

Table 1. Overview of Literature Sources

Sources \ Research Areas	Decision Analysis / MOO	Computational Resources	Simulation / Black-box Systems	Military Application / Motivation	Experimental Design					Censored / Survival Data	Meta-modeling / Modeling	Meta-learning
					Standard / Other Designs	Optimal Designs	LH / NOLH	NOAB	Comparison and Evaluation			
Abbasi et al. 2012												●
AFDD 2-0 2012				●								
Ammeri et al. 2010			●									
Anderson-Cook et al. 2009a	●				●	●			●			
Anderson-Cook et al. 2009b	●								●			
Ang 2006							●					
Ankenman et al. 2010											●	
Barton 1992											●	
Ben-Tal 1980	●											
Bettonvil 1995					●							
Biganzoli et al. 1998										●	●	
Bischl et al. 2016												●
Box and Behnken 1960					●							
Box and Wilson 1951					●							
Breiman 2001											●	
Breiman et al. 1984											●	
Burke et al. 2013												●
Bursztyn and Steinberg 2006					●				●			
Chaloner and Larntz 1989						●				●		
Charnes and Cooper 1961	●											
Chatterjee et al. 2000					●							
Cioppa 2002			●	●			●					
Cioppa and Lucas 2007				●			●					
Clarke et al. 2005											●	
Cote 2010	●			●								
Cressie 1993											●	
Crombecq 2011					●	●	●		●			
Cui, Hu, et al. 2016							●				●	●
Cui, Wu, et al. 2016			●								●	●
de Souza et al. 2009												●
Derringer and Suich 1980	●											
Dorfman 1943					●							
Drucker et al. 1997											●	
Duan et al. 2017							●					
Duch and Grudzinksi 2001												●
Feurer et al. 2015												●
Florian 1992							●					
Flournoy 1993					●							
Fonseca et al. 2003			●								●	
Friedman 1991											●	
Garcia et al. 2016												●
Gareth et al. 2013											●	
Golchi and Loeppky 2015					●							
Goldberg and Kosorok 2012										●	●	
Gomes et al. 2012											●	●
Goos 2009									●			

Sources \ Research Areas	Decision Analysis / MOO	Computational Resources	Simulation / Black-box Systems	Military Application / Motivation	Experimental Design					Censored / Survival Data	Meta-modeling / Modeling	Meta-learning
					Standard / Other Designs	Optimal Designs	LH / NOLH	NOAB	Comparison and Evaluation			
Goos and Jones 2011									●			
Guo and Mettas 2010					●					●	●	
H. Fang et al. 2005			●								●	
Hardy 1971											●	
Hebb 1949											●	
Hedayat et al. 2012					●	●						
Hernandez et al. 2012							●					
Hickernell 1998									●			
Hill and Lewicki 2006									●			
Hoke 1974					●							
Hutter et al. 2014										●	●	
IBM Corporation 2014		●										
Iman and Conover 1982							●					
Jacoby and Harrison 1962			●	●	●							
Jin et al. 2002					●	●			●			
Johnson et al. 1990					●							
Johnson et al. 2011					●				●			
K.T. Fang 1980					●							
K.T. Fang et al. 2000					●							
Keeney 1996	●											
Kennedy 2013					●	●	●					
Kiefer and Wolfowitz 1959						●						
Kleijnen 2007			●		●	●					●	
Kleijnen 2009											●	
Kleijnen et al. 2003			●		●		●		●			
Kleijnen et al. 2005			●	●	●		●		●		●	
Konstantinou et al. 2014						●				●		
Köpf et al. 2000											●	●
Kotthoff 2014												●
Koul et al. 1981										●	●	
Kück et al. 2016												●
Kutner et al. 2004											●	
Law 2015			●		●						●	
Lawless 2011										●		
Lemke et al. 2015												●
Leyton-Brown et al. 2002												●
Liang et al. 2013			●									
Loeppky et al. 2010					●							
Loterman and Mues 2012											●	●
Lu et al. 2011						●			●			
Lu et al. 2012						●			●			
MacCalman 2013			●	●			●	●	●		●	
MacCalman et al. 2017							●					
Marler and Arora 2004	●											
Marlow et al. 2015			●	●				●				
Mason 2012		●										
Matheron 1963											●	
Matijaš et al. 2013												●

Sources \ Research Areas	Decision Analysis / MOO	Computational Resources	Simulation / Black-box Systems	Military Application / Motivation	Experimental Design					Censored / Survival Data	Meta-modeling / Modeling	Meta-learning
					Standard / Other Designs	Optimal Designs	LH / NOLH	NOAB	Comparison and Evaluation			
McCulloch and Pitts 1943											●	
McKay et al. 1979							●					
Mitchell 1974						●						
Montgomery 2013					●							
Morrice 1995					●							
Muñoz and Smith-Miles 2016			●									●
Muñoz et al. 2015			●								●	●
Myers et al. 2009			●		●	●	●				●	
Myers et al. 2016	●		●		●	●	●		●		●	
OpenSolver 2017		●										
Owen 1994							●					
Parker 2009									●			
Patel 1962					●							
Piepel 2009									●			
Poursoltan and Neumann 2016												●
Pronzato and Müller 2012					●							
Rao 1946					●							
Rao 1947					●							
Rennen 2010							●					
Rice 1976												●
Ripley and Ripley 2001										●	●	
Romero et al. 2013												●
Roquemore 1976					●							
Rosenblatt 1958											●	
Rossi et al. 2014											●	●
S. E. Burke 2016						●						
Sacks et al. 1989					●	●					●	
Saleh et al. 2016			●			●						
Sanchez and Sanchez 2005					●							
Sanchez et al. 2014			●				●	●			●	
Santner et al. 2003											●	
Satterthwaite 1959					●							
Schmee and Hahn 1979										●	●	
Schruben 1986					●							
SEED Center for Data Farming 2016		●		●	●		●	●				
Shamsuzzaman et al. 2015							●					
Shewry and Wynn 1987					●							
Smith-Miles 2008											●	●
Smith-Miles and Lopes 2012											●	●
Sobol' 1967					●							
Srivastava 1975					●							
Staum 2009											●	
Steinberg and Lin 2006							●					
Taguchi 1988					●							
Tang 1994							●					
Tsai and Hsu 2013												●
U.S. Air Force 2013				●								
Verdinelli and Chaloner 1995					●							

Sources \ Research Areas	Decision Analysis / MOO	Computational Resources	Simulation / Black-box Systems	Military Application / Motivation	Experimental Design					Censored / Survival Data	Meta-modeling / Modeling	Meta-learning
					Standard / Other Designs	Optimal Designs	LH / NOLH	NOAB	Comparison and Evaluation			
Viana 2013							●					
Vieira Jr. et al. 2013			●	●	●	●		●				
Wakeman 2012			●	●				●				
Wang et al. 2009												●
Wolpert and Macready 1997											●	●
Xiong et al. 2009							●					
Ye 1998							●					
Total	9	4	21	12	45	18	27	6	18	10	45	28

2.2 Decision Space Representations and Combinatorial Challenges

Even with a limited number of asset types to select from in a portfolio tradespace, the decision space can grow quickly. A relatively small portfolio space is provided in Table 2, similar to the representation in [7]. For this example, suppose two types of remotely piloted aircraft are of interest, *RPA1* and *RPA2*, both of which are available in two sets, denoted by *a* and *b*, of some specified quantity. Now with respect to system utilization, suppose that each set of RPA1 has two options for basing, two options for routing, and three options for sensor packages. Similarly, suppose that each set of RPA2 has three options for basing, three options for routing, and one option for sensor packages. Note that the experimental design research that follows is not dependent on the portfolio representation and types of options shown here, i.e., base, route, and sensor, as the aim here is to provide a sense of the number of possible options for a decision space of even limited scope.

Table 2. Small Portfolio Example (System Set View)

System Set	(B) Base Options	(R) Route Options	(S) Sensor Packages	$B \cdot R \cdot S$	Total Options
RPA1a	2	2	3	12	13
RPA1b	2	2	3	12	13
RPA2a	3	3	1	9	10
RPA2b	3	3	1	9	10

For this example, there are four sets of systems, *RPA1a, RPA1b, RPA2a,* and *RPA2b,* represented by the four rows in Table 2. For the two rows of RPA1a and RPA1b, there are the same number of choices for the number of base options, B, the number of route options, R, the number of possible sensor combinations, S, and thus the same number of total usage combinations, $B \cdot R \cdot S$, in addition to the option of not using a system set. From Table 2, there are 13·13·10·10 = 16,900 portfolio options, which is a large number of possible decisions given that only two system types are considered. If an additional sensor package option is permitted for the RPA2 sets, then $S = 2$ and $B \cdot R \cdot S = 18$, resulting in 13·13·19·19 = 61,009 total portfolio options, more than triple the number of possible decisions. Note that even though there are three choices for sensor combinations available to an RPA1 set, this does not imply that there are only three sensor types available. For example, suppose there are four possible sensors S1, S2, S3, and S4 available to RPA1, yet only three combinations, {S1,S2}, {S1,S3}, and {S1,S4}, are feasible due to physical constraints. The combinatorial challenges of such a decision space as well as the often significant amount of time required to produce simulation responses motivate an efficient, yet robust, approach to experimental design and meta-modeling.

In Table 3, a larger portfolio option space example is given that includes remotely piloted aircraft (RPA) as well as aircraft (AC) and satellites (SAT), where the decision space consists of 3,343,221,000 points, requiring over a century of computation time for design point evaluations costing only one second each. It is clear that a decision space of larger scope will require an efficient experimental design approach.

Table 3. Portfolio Space Example (Qualitative System Set View)

System Set	(B) Base Options	(R) Route Options	(S) Sensor Combos	B · R · S	Total Options
RPA1	1	3	3	9	10
RPA2a	1	6	3	18	19
RPA2b	1	6	3	18	19
RPA3a	2	3	1	6	7
RPA3b	2	3	1	6	7
RPA3c	2	3	1	6	7
AC1	2	2	2	8	9
AC2a	1	2	2	4	5
AC2b	1	2	2	4	5
SAT1	1	1	2	2	3
SAT2	1	1	1	1	2
SAT3	1	1	1	1	2

A final example is shown in Table 4 for a notional portfolio space examined in support of a real-world simulation effort, which happens to have more of a quantitative focus and consists of at most three-level factors. This portfolio example will be explored throughout the literature review to present various topics in greater detail. The portfolio representation is comprised partially of five two-level factors, to answer the question of which combination of systems will create a portfolio of greatest value:

- The baseline system A is either upgraded or not
- The baseline system C is either upgraded or not

- The baseline system D is either upgraded or not
- System E is either used or not
- System F is either used or not

No upgrade option is available to system B. Additionally, to address the question of how many A systems to employ, one four-level factor is introduced, representing quantities of 0, 1, 2, or 3 for a single route. Similarly, for system C, one three-level factor is used, representing quantities of 0, 1, or 2 for a single route. No more than a total of two B systems are to be flown on two dissimilar routes. To account for this constraint in quantity, two three-level categorical factors are created, one for each individual system B, where individual system B_a, and similarly system B_b, each have three options: use on route 1, use on route 2, and do not use.

Table 4. Portfolio Space Example (Mixed-Factor View)

Factor, x	Levels, l_x	Options	Description
A Type	2	{0,1}	0 - baseline, 1- upgrade
A Quantity	4	{0,1,2,3}	number of A to use
B_a	3	{1,2,3}	1 - route 1, 2 - route 2, 3 - do not use
B_b	3	{1,2,3}	1 - route 1, 2 - route 2, 3 - do not use
C Type	2	{0,1}	0 - baseline, 1- upgrade
C Quantity	3	{0,1,2}	number of C to use
D Type	2	{0,1}	0 - baseline, 1- upgrade
E	2	{0,1}	0 - do not use, 1 – use
F	2	{0,1}	0 - do not use, 1 - use

The non-encoded design matrix representation has nine factors and variable columns, while the encoded design for analysis has 11 variable columns, with *effect coding* used for categorical factors B_a and B_b. The design point [0,3,2,3,1,2,0,1,0], before encoding, represents the option that uses three baseline A systems, system B_a on Route 2, two upgraded C systems, and system E capability, which is in addition to the baseline

system D capability. This representation contains 3,456 total portfolio options. For a first order model with m variables,

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i \, ,$$

and effect coding for a categorical factor with j possible categories can be defined as follows for a design point or observation having category k:

$$x_i = \begin{cases} 1, & if\ k = i < j \\ -1, & if\ k = j \\ 0, & otherwise \end{cases}, \qquad for\ i = 1, 2, \dots, j$$

2.3 Desired Properties of Experimental Design

2.3.1 Overview

The aspects of experimental design important to this research are examined, including mixed factors, orthogonality, balance, efficiency, and space filling. As seen with the previous portfolio examples, a *mixed-factor design*, or *mixed design*, may be required, i.e., a design having some combination of continuous, discrete, and categorical factors in addition to possibly having different numbers of levels for each factor.

Note that there are many standard designs [8]–[11] that do not satisfy these design requirements, with disadvantages discussed for the various design properties. Beyond factorial and fractional factorial designs, some of the more standard designs include the following, as detailed in [12]:

- orthogonal array (OA) [13]–[15]
- central composite design (CCD) [16]
- face central composite design (FCCD)
- Box-Behnken design (BBH) [17]
- Hoke design [18]
- hybrid design [19]

- very large fractional factorials and CCDs [20].

Optimal designs are presented in [21]. Space-filling designs include Latin hypercube (LH) design [22]–[26], maximum entropy [27], sphere packing [28], and uniform [29], [30]. Improvements have been made to Latin hypercube designs, including the *orthogonal* Latin hypercube [31]–[33] as well as the *nearly orthogonal* Latin hypercube (NOLH) [34], [35]. Second-order NOLH designs have been created using a genetic algorithm [36]. A construction method for nearly orthogonal-and-balanced (NOAB) designs with mixed factors is developed in [1]. A single NOAB design with near orthogonality between second-order discrete factors was constructed using a genetic algorithm in [12], though the heuristic approach appears to have difficulty satisfying specific near balance requirements. Many space-filling designs have been created for deterministic simulation, requiring a sufficient number of replications for stochastic responses. In [37], there is an example that shows the use of a Latin hypercube design of 50 design points with 30 replications.

Other designs, as listed in [38], include:

- group screening [39], [40]
- random design [41]
- sequential bifurcation [42], [43]
- robust designs [44]
- Bayes designs [45], [46]
- search linear models [47], [48]
- frequency domain [49], [50].

2.3.2 Orthogonality

An orthogonal design allows each factor to be examined independently of other factors. Depending on the eventual meta-model used for each simulation response,

orthogonality can allow for examination of individual factors separately, which permits feature reduction. This property can be measured by the maximum absolute correlation of all possible pairs of encoded factor columns, denoted by ρ_{map}, where a design is considered orthogonal if $\rho_{map} = 0$, and nearly orthogonal if $\rho_{map} \leq 0.05$.

The rounding of design point values to achieve discrete levels from well-known continuous factor space-filling designs such as LH, uniform, and sphere-packing designs does not guarantee near orthogonality, and these designs do not address the need for categorical factors [1]. An example of rounding of NOLH designs and the associated loss of near orthogonality is provided in [12].

2.3.3 Balance

An experimental design is balanced when all factor levels occur for the same number of design points. A design is considered *nearly balanced* when the maximum imbalance for all factor columns, denoted by δ, is sufficiently close to zero. A nearly balanced design ensures that levels within each factor are represented nearly equally. Requiring $\delta < 1$ ensures that all factor levels occur in the design [1], with imbalance for a factor x defined as

$$\delta_x = \max_{i=1,\ldots,l_x} \left| \frac{w_{i,x} - (n/l_x)}{(n/l_x)} \right|$$

where l_x is the number of levels, $w_{i,x}$ is the number of times level i occurs, and n is the number of design points.

2.3.4 Space-filling

Experimental designs for meta-modeling of simulation output require good space-filling properties in order to efficiently model surfaces over regions that have a large number of input combinations. Crossing smaller standard designs to achieve a mixed-factor design can be inefficient. In [1], it is stated that orthogonal arrays (OAs) for experiments with many mixed factors are not readily available and likely inefficient as well. The modified L_2 discrepancy, or ML_2, [51] is a commonly-used space-filling measure, as discussed in [1], [12], [52].

2.4 Nearly Orthogonal-and-balanced (NOAB) Design from [1]

2.4.1 Overview

The NOAB design allows for mixed factors with different numbers of levels and has an existing construction method that aims to minimize correlations between pairs of design matrix columns (representing first-order model terms) while also satisfying near balance constraints. Though efficiency can be a subjective measure, NOAB designs have been shown to be consistently orders of magnitude smaller in size than other designs with similar design performance properties.

Inputs for the construction method are as follows:

- design size / number of design points (matrix rows) n, indexed by row $r = 1, 2, \dots, n$
- maximum allowed absolute pairwise correlation ρ_{map}
- maximum allowed imbalance δ
- factor types $C(x)$ for each factor x
- number of levels ℓ_x for each factor x, indexed by level $i = 1, 2, \dots, \ell_x$

where

$$C(x) = \begin{cases} 1, & if\ x\ is\ continuous \\ 2, & if\ x\ is\ discrete \\ 3, & if\ x\ is\ categorical \end{cases}$$

For a mixed design, general guidelines for bounds on the number of design points, n, are presented in [1]:

$$3\left(K - L + \sum_{x \in L}(l_x - 1)\right) \leq n \leq 10\left(K - L + \sum_{x \in L}(l_x - 1)\right)$$

where L is the number of categorical factors, and K is the total number of factors.

Pairwise correlation for columns $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$\rho(\boldsymbol{x}, \boldsymbol{y}) = 1/((n-1)\, s_{\boldsymbol{x}} s_{\boldsymbol{y}}) \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}})(y_r - \overline{\boldsymbol{y}})$$

with column elements x_r and y_r, means $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$, and standard deviations $s_{\boldsymbol{x}}$ and $s_{\boldsymbol{y}}$.

2.4.2 Balance Feasibility Test

The construction method is accompanied by a balance feasibility test (Appendix) that rules out design sizes based on a specified maximum imbalance parameter δ^*. This feasibility test is updated from [1], with differences highlighted in bold, to ensure that the maximum imbalance is calculated in each case. The original value for the imbalance δ was previously set to infinity (in practice, a sufficiently large number) and not zero, and the comparison to determine δ for each column is now shown as a maximization and not a minimization. For the majority of this research, the suggested bounds for first-order NOAB design size are used for the range of possible design sizes, which are then tested for balance feasibility.

2.4.3 Construction Algorithm

The NOAB design construction method (algorithm in Appendix) creates the NOAB design by sequentially appending columns for a single factor. First, the new factor columns are randomly generated to satisfy balance constraints, which serve as an initial solution to one of three mixed-integer linear programming (MILP) problems, dependent on factor type. The main goal for these first-order NOAB designs from the literature is to identify the most important factors to a response, so focus has previously been on near orthogonality and D-optimality as performance measures, which aims to maximize the determinant of the information matrix, $\boldsymbol{X}'\boldsymbol{X}$, for design matrix $\boldsymbol{X}$ [8]. Heuristic search parameters for the construction method include t_{min} and t_{max}, the minimum and maximum allowable time for MILP solution search, respectively, as well as h^*, the maximum number of iterations per design matrix column, and b^*, the maximum number of macro-iterations, i.e., full design construction attempts. Note that effect coding is used for categorical factor columns.

2.4.4 MILP Formulations

There are three MILP formulations, one for each factor type: continuous (Figure 1), discrete (Figure 2), and categorical (Figure 3). Notation for the formulations are as follows:

Table 5. Notation for NOAB Design Construction

j	number of previously constructed matrix columns, indexed by column $c = 1,\ 2,\ \ldots,\ j$
$\boldsymbol{M}$	previously constructed $n \times j$ design matrix (represents only first-order terms in the original method and both first- and second-order terms for the full second-order method)
$m_{r,c}$	element of $\boldsymbol{M}$ in row r and column c
$\boldsymbol{m}_{\cdot,c}$	column c of $\boldsymbol{M}$
C_1	subset of column indices $1,\ 2,\ \ldots,\ j$ for $\boldsymbol{M}$ that represent first-order terms only, indexed by c_1
$\boldsymbol{x}$	MILP decision variables ($n \times 1$ factor column), $\boldsymbol{x}^i$ is the i$^{\text{th}}$ column in the categorical case
x_r	element of $\boldsymbol{x}$ in row r
π_ℓ	encoded level value (with $\{\pi_1, \pi_2, \ldots, \pi_{\lambda(\boldsymbol{x})}\}$ being all possible values for column $\boldsymbol{x}$)
$\theta_{r,\ell}$	binary decision variable where $x_r = \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell \theta_{r,\ell}$ and $\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \theta_{r,\ell} = 1$ for row r and encoded level ℓ

For all three formulations, constraints (i) and (ii) ensure that the pairwise correlation between the new factor column(s) and all previously constructed columns are minimized, noting that the required near balance of each factor permits the removal of s_x. Constraint (iv) allows for the binary representation of the various factor column elements in $\boldsymbol{x}$. Constraint (iii), in addition to the binary constraint on the various θ values, ensures that each design point has exactly one level selected from $\{\pi_1, \pi_2, \ldots, \pi_n\}$ equally spaced values for a continuous factor (guaranteeing balance), $\{\pi_1, \pi_2, \ldots, \pi_{l_x}\}$ values in the discrete case, and $\{-1, 0, 1\}$ in the categorical case. The three options in the categorical case are

associated with the effect coding. Note that in the continuous case, constraint (iii) is required to ensure balance, $\sum_{r=1}^{n} \theta_{r\ell} = 1$, for $\ell = 1, \dots, n$.

Constraints (v) and (vi) for the discrete and categorical cases make sure that the specified imbalance is not violated. Additionally, for the categorical case, constraints $(vii) - (ix)$ are associated with the effect coding of $\ell_x - 1$ new columns.

Minimize	v	
Subject to		
(i)	$v \geq \frac{1}{s_c} \sum_{r=1}^{n} (x_r - \bar{\boldsymbol{x}}) \left(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}}\right)$	$c \in I(j)$
(ii)	$v \geq -\frac{1}{s_c} \sum_{r=1}^{n} (x_r - \bar{\boldsymbol{x}}) \left(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}}\right)$	$c \in I(j)$
(iii)	$\sum_{\ell=1}^{n} \theta_{r\ell} = 1$	$r \in I(n)$
(iv)	$x_r = \sum_{\ell=1}^{n} \pi_\ell \theta_{r\ell}$	$r \in I(n)$
(v)	$\theta_{r\ell} \in \{0,1\}$	$r \in I(n); \ell \in I(n)$

Figure 1. MILP Formulation for Continuous Factor [1]

For the literature review, the construction method and MILP formulations for the three factor types are implemented in MATLAB version2015a using CPLEX V12.6.1 [53] to obtain MILP solutions, with calculations performed on a HP Z420 Workstation with an Intel® Xeon® CPU E5-1620, 32 GB of RAM, and a 64-bit version of Windows 7.

Minimize	v	
Subject to		
(i)	$v \geq \frac{1}{s_c}\sum_{r=1}^{n}(x_r - \bar{\boldsymbol{x}})\left(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}}\right)$	$c \in I(j)$
(ii)	$v \geq -\frac{1}{s_c}\sum_{r=1}^{n}(x_r - \bar{\boldsymbol{x}})\left(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}}\right)$	$c \in I(j)$
(iii)	$\sum_{\ell=1}^{\ell_x}\theta_{r\ell} = 1$	$r \in I(n)$
(iv)	$x_r = \sum_{\ell=1}^{\ell_x}\pi_\ell\theta_{r\ell}$	$r \in I(n)$
(v)	$\sum_{r=1}^{n}\theta_{r\ell} \leq \left\lfloor(1+\delta)\frac{n}{\ell_x}\right\rfloor$	$l \in I(\ell_x)$
(vi)	$\sum_{r=1}^{n}\theta_{r\ell} \geq \left\lceil(1-\delta)\frac{n}{\ell_x}\right\rceil$	$l \in I(\ell_x)$
(vii)	$\theta_{r\ell} \in \{0,1\}$	$r \in I(n); l \in I(\ell_x)$

Figure 2. MILP Formulation for Discrete Factor [1]

2.4.5 Known Case Studies

When compared to many other designs, including NOLHs, NOAB designs have been shown to lie on the Pareto frontier with respect to near orthogonality, near balance, and space-filling properties, measured by ρ_{map}, δ, and ML_2, respectively. Figure 4 shows how a NOAB design performs when compared to 19 other designs, each with 25 design points and four discrete factors having three, four, five, and seven levels each [1]. In [1], the NOAB design is compared to the Faced Central Composite, BBH, D-optimal, I-optimal, sphere packing, uniform, Latin hypercube (LH), maximin LH, maximum entropy, and minimum potential designs as well as Sobol' and scrambled Sobol' sequences [54].

Minimize	v	
Subject to		
(i)	$v \geq \frac{1}{s_c}\sum_{r=1}^{n}\left(x_r^i - \bar{\boldsymbol{x}}^{\boldsymbol{i}}\right)\left(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}}\right)$	$c \in I(j); i \in I(\ell_x - 1)$
(ii)	$v \geq -\frac{1}{s_c}\sum_{r=1}^{n}\left(x_r^i - \bar{\boldsymbol{x}}^{\boldsymbol{i}}\right)\left(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}}\right)$	$c \in I(j); i \in I(\ell_x - 1)$
(iii)	$\sum_{\ell=1}^{3}\theta_{r\ell}^i = 1$	$r \in I(n); i \in I(\ell_x - 1)$
(iv)	$x_r^i = \sum_{\ell=1}^{3}(\ell - 2)\theta_{r\ell}^i$	$r \in I(n); i \in I(\ell_x - 1)$
(v)	$\sum_{r=1}^{n}\theta_{r\ell}^i \leq \left\lfloor(1+\delta)\frac{n}{\ell_x}\right\rfloor$	$l = 1,3; i \in I(\ell_x - 1)$
(vi)	$\sum_{r=1}^{n}\theta_{r\ell}^i \geq \left\lceil(1-\delta)\frac{n}{\ell_x}\right\rceil$	$l = 1,3; i \in I(\ell_x - 1)$
(vii)	$\sum_{i=1}^{\ell_x - 1}\theta_{r3}^i \leq 1$	$r \in I(n)$
$(viii)$	$\sum_{i=1}^{\ell_x - 1}\theta_{r2}^i \leq \ell_x - 2$	$r \in I(n)$
(ix)	$\theta_{r1}^i - \theta_{r1}^1 = 0$	$r \in I(n);$ $i = 2, 3, \ldots, \ell_x - 1$
(x)	$\theta_{r\ell}^i \in \{0,1\}$	$r \in I(n); \ell \in I(3);$ $i \in I(\ell_x - 1)$

Figure 3. MILP Formulation for Categorical Factor [1]

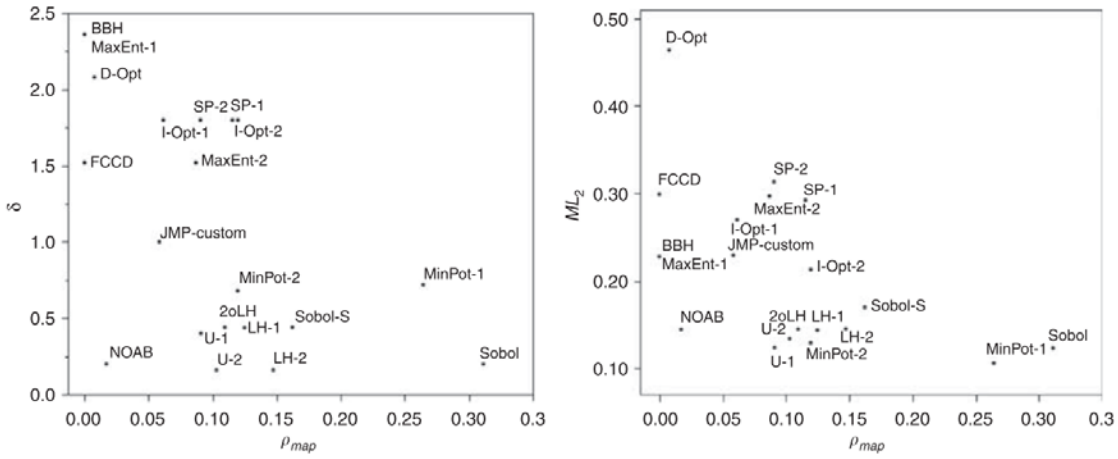


Figure 4. Design Comparisons for ρ_{map}, δ, and ML_2 [1]

A 300-factor, 512-point NOAB design is available at the SEED (Simulation Experiments & Efficient Designs) Center for Data Farming, Naval Postgraduate School website [55], comprised of 200 discrete factors and 100 continuous factors. The discrete factors are comprised of 10 sets of 20 factors having between two levels and 11 levels each.

Previous efforts that have used NOAB designs include the work by Wakeman [56], examining a discrete event simulation using 32 factors from the 512-point SEED Center design. In [57], a custom 19-factor, 1040-point NOAB design is constructed, comprised of 11 continuous, two discrete, and six categorical factors, where each point is replicated 50 times due to the stochastic nature of the fleet management simulation. Even when restricting the 11 continuous factors of the design to 10 levels each, the total number of points in the design space is 9.27E14, which would require over 3.5 million years of computation time [57].

2.4.6 Categorical Design Construction and Correlation Example

By the design size guideline, between $249 \leq n \leq 830$ design points are suggested for the ISR portfolio space presented in Table 3, where each system set is represented by a categorical factor. Note that each categorical factor again uses effect coding, and the maximum ρ_{map} and δ are set to 0.05 and 0.15, respectively. A 360-point design is constructed, requiring approximately 30 minutes of computation time. The design size was chosen ad-hoc by examining the least common multiples of values close to each of the number of levels, needing an additional 30 minutes for testing. Figure 5 shows the absolute correlation matrix in lower triangular form for the 12-factor, 360-point NOAB design, where there is low correlation between encoded columns not of the same factor. The full factorial design of 3,343,221,000 points would require approximately 106 years, assuming one second is required for each design point evaluation, whereas the 360-point NOAB design would require only six minutes of simulation run time. It is certainly true that a NOAB design with a larger number of points could be constructed to further improve design properties, while meeting the constraints for allowable simulation run time.

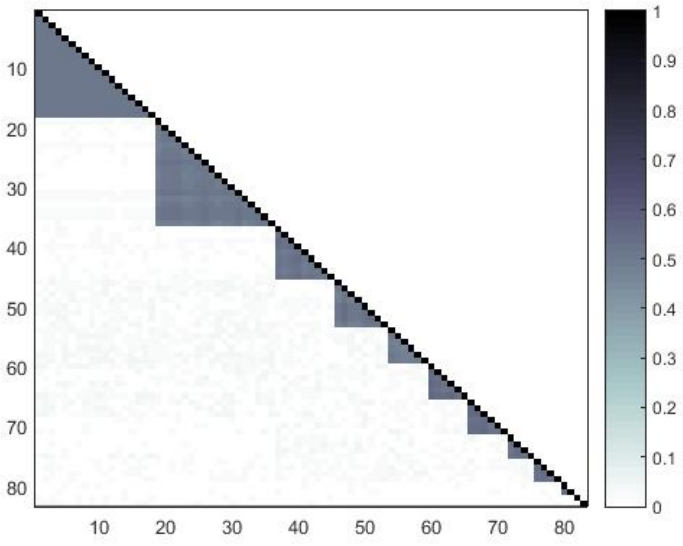


Figure 5. Absolute Correlation Heatmap for 12-factor, 360-point NOAB Design

2.5 Considerations for Censored or Survival Data

2.5.1 Potential for Right-censored Responses

For the initial ISR portfolio example in Table 2, there is an assumption that each of the system sets listed will be utilized for some combination of the three factors: base option, route option, and sensor package. Each individual system will be utilized in some way, which allows for performance measures to exist for all systems represented in the simulation. So given the existence of each individual system, measures regarding survivability and reliability may be of interest to a decision maker. Naturally, the time to failure/loss of a system (and the associated subsystems, components, or individual sensors) may be greater than the time window simulated, even in a long-duration study. This is when right-censored data would be injected for a portion of responses, since recording the

time to failure/loss of a system as the final timestamp of a simulation would not allow for an accurate representation of that reliability/survivability measure.

It is important to note that right-censoring can have the same censoring time for all individual systems or different, random censoring times due to the introduction of individual systems to the simulated environment at different times. For simulations capable of tracking such information for specific systems and components of interest, these time to failure/loss measures should be recorded and observed, where the right-censored data provides a lower bound on the times of interest.

Analysts would then be able to observe how the basing, routing, and sensor packages potentially impact the time to failure/loss for each system. It would be important to have consistent definitions for such measures, whether based on loss of a specific capability, total failure, or total loss. Value-focused thinking [58] could be used to map the system failure times of interest to the perceived value of a decision maker. It seems clear that a greater failure time would receive a larger value depending on the system, though there may be an acceptable time where the value sees diminishing returns for further marginal increases in time. These values associated with system survival time could then be aggregated into an overall value using relative weightings in a hierarchal structure, which may include other performance measures that are not censored. Such a hierarchy relies on mutual exclusivity of measures/metrics, so aggregating value or performance at a subsystem or component level would require more thought due to the potential for associated dependencies.

Another instance where right-censored data could be injected into such a simulation is if a system task has a finite-duration that may not see completion by the end of a simulation. In this case, the measure of completion time for a task may be impacted by the sensor package used in combination with environmental factors resulting from choices of basing and routing. Though a workaround may be to capture the percentage of completed tasks by the end of the simulation, the completion time itself may be of great importance to the study. The simulated time window could be associated with the maximum completion time that has zero or small value to a decision maker, thus lessening the importance of right-censored response estimation. However, if the simulation is stochastic rather than deterministic, as is often the case, the ability to provide meaningful average performance measures would seem to rely on imputation/estimation of the censored data.

2.5.2 Meta-model Considerations

The current design approach assumes no knowledge of the resulting responses, with the space-filling properties of the NOAB design allowing for the fitting of high-order meta-models, as the designs are "*amenable to trade-off analysis using non-parametric techniques*" [1, pp. 266]. However, it is important to consider which meta-models are appropriate for censored data. Space-filling designs are typically better for fitting semi-parametric and non-parametric models, as intended with the NOAB designs, though the performance of space-filling designs with respect to variance tends to be lesser near the boundaries and greater near the center of design regions [59].

Maximum likelihood estimation can be used for parametric models of various distributions, to include Weibull, exponential, and log-normal, with the likelihood ratio test

used to determine significant factors [60]. Other distributions include extreme value, log-normal, log-logistic, and gamma, while non-parametric methods include the Kapan-Meier estimate and the Cox proportional hazards model [61].

Though far from a comprehensive list, examples of models where censored data has been examined include regression [62], regression analysis with randomly right-censored data [63], random forests and approximate Gaussian processes to improve algorithm runtime prediction [64], neural networks for survival data [65], [66], and support vector machines [67].

Note that simply ignoring the censored observations and associated design points would most likely remove the (near) orthogonality and balance properties that NOAB designs are constructed to achieve, and assuming that the right-censored data are uncensored (by using each respective lower bound as the observed value) would bias any resulting meta-model as discussed in [64].

2.5.3 Potential Design Criteria and Design Alternatives

There are certain design criteria that would possibly be emphasized over others with the knowledge that say 30% of the observations in a fixed design region were censored, such as D-efficiency (maximize for good parameter estimates), I-efficiency (maximize for good prediction accuracy), and $tr(\boldsymbol{A'A})$ (minimize for protection from biased coefficient estimates), where A is an alias matrix for design matrix X [68, pp. 520-521].

Though there have been designs constructed specifically for censored data and associated assumed distributions, it is important to not tailor model construction too specifically to a single criteria (or type of response), since this research involves potentially

many response surfaces for each individual study. However, if distributions associated with survivability and reliability are of primary interest, then designs optimized for the associated criteria should be considered. Since many simulation studies require the use of non-standard designs, the focus in this section is on computer-generated optimal designs, specifically D-optimal designs.

2.5.4 Optimal Designs

Regarding censored data, the following research in optimal design construction is focused more on optimization for non-normal distributions and censored data. Bayesian D-optimality for nonlinear models, and logistic regression in particular, is presented in [69]. Efficient experimental designs have been constructed for generalized linear models where the goal is to maximize $|\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}|$ for weight matrix $\boldsymbol{W}$ [70]. Optimal designs for two-parameter nonlinear models have been examined using an example of exponential regression with the natural proportional hazards parameterization [71]. Optimal design for dual-responses systems has been examined for three cases (choosing two distinct responses from binary, normal, and Poisson distributions) using the measures of D-efficiency and Bayesian D-efficiency as appropriate in a multiplicative desirability function with a layered Pareto front algorithm [72].

2.6 Multiple Criteria for Design Selection

The nature of NOAB construction for mixed-factor experimental designs allows for many possible parameter settings and heuristic rules to be examined in order to determine how to create the “best” performing NOAB design for a specific study. There are several

performance criteria of possible interest, which are outlined in [68, pp. 520-521] and summarized in Table 6.

Table 6. Optimization Criteria for Multiple Facets of a Good Design [68]

Measure	Reason	Direction
D-efficiency	Parameter estimation	Max
I-efficiency	Average prediction variance	Max
G-efficiency	Worst-case prediction variance	Max
$tr(A'A)$	Protection from bias (model terms)	Min
$tr(R'R)$	Protection from bias (SSE)	Min
$tr(R'R)$	Estimates for lack of fit	Max
Number of replicates	Pure error estimation	Max
Number of design points	Experimental cost	Min

2.6.1 D-Efficiency and A-Efficiency for Good Model Parameter Estimation

For the number of design points n and the number of model parameters p in design matrix $\boldsymbol{X}$, the moment matrix is defined as $\boldsymbol{M} = (\boldsymbol{X}'\boldsymbol{X})/n$, with determinant $|\boldsymbol{M}| = |\boldsymbol{X}'\boldsymbol{X}|/n^p$. As stated in [68, pp. 468], "*under the assumption of independent normal model errors with constant variance, the determinant of* $\boldsymbol{X}'\boldsymbol{X}$ *is inversely proportional to the square of the volume of the confidence region on the regression coefficients*", thus the aim is to maximize $|\boldsymbol{M}|$ by choice of design ξ in order to improve estimation of model coefficients.

A D-optimal design is one where $|\boldsymbol{M}|$ is maximized. So the D-efficiency of a design ξ^* is defined as $D_{eff} = (|\boldsymbol{M}(\xi^*)|/\max_{\xi}|\boldsymbol{M}(\xi)|)^{1/p}$ from [68, Equation 9.12]. From [73, pp. 223], where D-efficiency is defined as $100\cdot|\boldsymbol{X}'\boldsymbol{X}|^{1/p}/n$, it is stated "*you should use this measure rather as a relative indicator of efficiency, to compare other designs of the same size, and constructed from the same design points candidate list*" as well as "*this measure*

can be interpreted as the relative number of runs (in percent) that would be required by an orthogonal design to achieve the same value of…" each respective alphabetical optimality. Note that Mitchell [74] states that this definition of D-efficiency can be interpreted as the "*relative number of runs (expressed as percent) required by a (possibly nonexistent) orthogonal design to achieve the same* $|\boldsymbol{X}'\boldsymbol{X}|$." This same definition of D-efficiency that is used in many software "*is only useful for comparing two designs that have the same scale or coding for the experimental factors as well as the same number of runs,*" from [75, Sec. 4.3.3]. However, for different design sizes, this depends on the D-criterion used. A D-criterion should not already be scaled by the design size when design size is one of the multiple criteria for design comparison. Thus, the D-criterion $|\boldsymbol{X}'\boldsymbol{X}|^{1/p}$ that uses the unscaled moment matrix $\boldsymbol{M} = \boldsymbol{X}'\boldsymbol{X}$, from [76, pp. 362], will be used when design size is also a criterion.

A-Optimality aims to improve estimation of model coefficients, as with D-Optimality, though covariances among coefficients are ignored, as only the diagonal elements of the moment matrix are used in its definition $\max_{\xi} tr[\boldsymbol{M}(\xi)]^{-1}$, from [68, pp. 472-473].

2.6.2 I-Efficiency and Use of Average Unscaled Prediction Variance

With scaled prediction variance, or SPV, written as the function $v(x) = nx^{(m)\prime}(\boldsymbol{X}'\boldsymbol{X})^{-1}x^{(m)}$, I-optimality is defined as $\min_{\xi} \frac{1}{K}\int_R v(x)dx = \min_{\xi} I(\xi)$, where R is the region of interest and $K = \int_R dx$. So I-optimal designs aim to minimize the

average SPV over a design region, where I-efficiency for design ξ^* is defined as $I_{eff} = \min_{\xi} I(\xi)\,/I(\xi^*)$ [68, pp. 473].

The unscaled prediction variance, $UPV = x^{(m)\prime}(\boldsymbol{X}'\boldsymbol{X})^{-1}x^{(m)}$, can be used instead of $SPV = v(x) = n \cdot UPV$, when design size n is also a criterion under consideration in order to have measures that are as mutually exclusive as possible and accurately examine trade-offs. The average UPV can be used as a design criterion and is estimated for continuous regions.

As stated in [68, pp. 407], UPV is an alternative measure to SPV when either design size n is not important or the marginal cost of design size is not described accurately by the simple penalty of n. UPV is a good measure of prediction accuracy and is often used over SPV [77, pp. 672].

The following arguments for use of UPV over SPV are summarized from the discussion papers from [77]. Parker states that in cases where a specified prediction quality is the focus, it is better to present the prediction variance in engineering units to a subject matter expert (SME) rather than an efficiency scaled by the number of design points [78]. Piepel gives several reasons for using UPV over SPV, in that different design sizes should always be examined and thus the trade-offs between UPV and experimental cost is better described when not tied to a single value, the trade-off is easier to make when a specific UPV property is desired, and graphical displays of UPV rather than SPV are easier to understand and present [59].

Goos suggests the use of UPV and that neither SPV nor G-efficiency are practical measures for ranking different design options [79], stating:

> "*This precision is directly related to the size of the experiment: larger experiments often lead to smaller prediction variances and thus to a better predictive precision. By looking at unscaled prediction variances, the researcher can evaluate the increase in precision obtained from using a larger experiment. Thus, unscaled prediction variances provide an experimenter with much more useful information than scaled ones.*" [79, pp. 658]

Goos explains that smaller experimental designs are typically favored when using SPV for evaluation and comparison, since larger experimental designs are penalized. Additionally, SPV potentially masks the poor prediction accuracy of much smaller designs, and design size may not always be an accurate measure for cost, such as in split-plot experiments, where some factors are more difficult to change than others, as well as in experiments with significant preparation time when compared to the time required for actual experimental runs. Goos also notes that the use of SPV goes against the idea of not relying solely on single-number criterion, which is explored in [77] through the use of graphical displays of prediction variance information.

In a rejoinder [80], it is stated that UPV gives the most direct way to examine the improvement in prediction variance as experimental cost increases, since the common choice of SPV makes a clear assumption regarding this relationship, though it is added that the choice to examine the true trade-off between UPV and n is subjective. In order to have a process of design comparison that is less case-specific, this research will emphasize the use of UPV as a measure of prediction accuracy with the knowledge that SPV can be used in later cases if desired.

2.6.3 G-Efficiency

A G-optimal design minimizes the maximum $v(x)$ over region of interest R, so a G-optimal design ξ is one that satisfies $\min_{\xi}\left[\max_{x\in R} v(x)\right]$ in order to protect against worst-case prediction accuracy [68, pp. 470].

2.6.4 Model Misspecification, Lack of Fit Estimates, and Other Criteria

When protection against model misspecification is important, $tr(\boldsymbol{A}'\boldsymbol{A})$ and $tr(\boldsymbol{R}'\boldsymbol{R})$ can be minimized to protect from bias for coefficient and variance estimates, respectively [81, pp. 208]. Here, $\boldsymbol{A} = (\boldsymbol{X_1}'\boldsymbol{X_1})^{-1}(\boldsymbol{X_1}'\boldsymbol{X_2})$ is the alias matrix, and $\boldsymbol{R} = \boldsymbol{X_1}\boldsymbol{A} - \boldsymbol{X_2}$, where $\boldsymbol{X_1}$ is the assumed linear model matrix and $\boldsymbol{X_2}$ includes additional linear terms. Maximizing $tr(\boldsymbol{R}'\boldsymbol{R})$ to provide estimates for lack of fit is also possible, so it is important for SMEs and analysts to understand which criterion are important to their specific application. The number of replicates can be used as a measure for estimating pure error with more degrees of freedom, and the total number of design points n often serves as a proxy measure for the experimental cost.

For Chapters III and IV, emphasis will be placed on small design size n, good model parameter estimation (D-criterion using the unscaled moment matrix), and good prediction accuracy (average or maximum UPV). Though the minimization of absolute pairwise correlations increases D-efficiency for the NOAB design, the space-filling properties are said to allow for high-order meta-models of the resulting response surface(s) [1], so prediction accuracy and protection from biased coefficient estimates are also of interest. Computation time for design construction will not be considered as a design

criteria in this research, due to the relatively small construction times when compared to simulation time of supported research efforts.

2.7 Design Comparison and Evaluation

2.7.1 Overview

In [77], the authors discuss how to examine trade-offs of competing criteria for several candidate experimental designs, including the various alphabetic optimality criteria, graphical methods for examining design properties, design robustness to model misspecification, and special cases of design comparison, including split-plots, mixture experiments, robust parameter designs, and generalized linear model designs. Their focus though is on response surface designs, mostly dealing with fitting first or second-order polynomials, which is in contrast to the space-filling NOAB designs of interest for use in simulation meta-modeling. The authors state that design considerations for fitting first-order models is easy when the experimental region is cuboidal or spherical, as first-order orthogonal designs possess many desirable characteristics, so there is more focus on designs for second-order models. It appears that the more complex mixed-factor design spaces with different numbers of levels are not considered in this assessment.

As suggested in [68, pp. 370], there are 11 characteristics that a good response surface design should satisfy as appropriate to each study, to include providing a good model fit (1), allowing for sequential model construction (2), blocking (3), and lack of fit tests (4), being cost-effective (5) as well as robust to outliers (6) and errors in control of design levels (7), providing good model parameter estimates (8), an estimate of pure experimental error (9), and a good distribution for prediction variance over a design region

(10), and finally, checking on the homogeneous variance assumption (11). It is noted that not all of these items are necessary, nor of equal importance in all cases.

Since I- and G-efficiency for integrated and maximum prediction variance, respectively, do not entirely capture the prediction variance properties for a design region of interest, graphical methods are suggested in [77]. Alternatives to using a single-number for comparison include the variance dispersion graph (VDG) and fraction of design space (FDS) plot. VDGs, developed by Giovannitti-Jensen and Myers, "*plot the minimum, average, and maximum SPVs against distances from the overall center of the design space*" [77, pp. 631], where multiple designs can be compared on the same plot. The FDS plot, developed by Zahran et al., displays the prediction variance by the fraction of design space with the prediction variance less than or equal to the current value. The authors note that there are instances of comparing designs where the same G-efficiency is obtained, yet different SPV values occur over the design region when examining an FDS plot, and that the use of SPV in such plots is "*relatively standard*" to incorporate the cost of the experiment (for completely randomized designs). For model robustness, work has been done on assessing design properties for nested models, using subjective weighting as well as FDS plots. In particular, FDS plots help to examine the bias-variance trade-off where each nested model curve is below the largest model. An important measure for examining the trade-off between prediction variance with bias is the mean squared error criterion, and a reminder is given in the authors' rejoinder that the type of model to protect against bias must be specified. From the discussion papers that follow [77], Khuri states that quantal

plots (QPs) of the prediction variance and the quantile dispersion graphs (QDGs) are also useful graphical tools.

With multiple criteria for design selection discussed in the previous section, the ISR portfolio example from Table 4 will serve as an illustrative example for the concepts and use of desirability functions, Pareto frontier, and synthesized efficiencies for the design comparison and evaluation process. Graphical approaches are used for both direct results of Pareto set, desirability, and synthesized efficiencies as well as for complementary results when examining UPV with FDS plots.

2.7.2 Desirability Functions

The measures of various objectives should have the same scale in order to be comparable, so one-sided *desirability functions* [82] are used for each of the criteria, with target T of lower and upper limits L and U, respectively [68, pp. 341]:

$$d = \begin{cases} 0, & y < L \\ \left(\dfrac{y - L}{T - L}\right)^r, & L \le y \le T \\ 1, & y > T \end{cases}$$

$$d = \begin{cases} 1, & y < T \\ \left(\dfrac{U - y}{U - T}\right)^r, & T \le y \le U \\ 0, & y > U \end{cases}$$

Two common approaches for forming an overall desirability function for m objectives are the additive function $D = \sum_{i=1}^{m} w_i d_i$ and the multiplicative function $D = \prod_{i=1}^{m} {d_i}^{w_i}$, where $\sum_{i=1}^{m} w_i = 1$. The additive desirability function allows for high scores in

one objective to make up for low scores from other objectives, while the multiplicative desirability function ensures that no single score is too low.

2.7.3 Pareto Front

Suppose there are multiple objective functions $f_1, f_2, \ldots, f_m$, where the goal is for each to be maximized. If designs ξ_1 and ξ_2 exist such that $f_i(\xi_1) \geq f_i(\xi_2)$ for all $i = 1,2,\ldots,m$, yet there is at least one j where $(\xi_1) > f_i(\xi_2)$, then ξ_1 is said to *Pareto dominate* ξ_2. A *Pareto set*, or *Pareto frontier*, of designs is comprised of all designs ξ not Pareto dominated by any other designs evaluated. Finding the Pareto front of experimental designs for a study potentially reduces the number of designs that require further evaluation and comparison, since the Pareto dominated designs would not perform as well for the measures of interest. In this case, each objective function is a single desirability function in order to have comparable design performance measures.

For the ISR example detailed in Table 4, five attempts are made to construct NOAB designs for each balance-feasible design size within suggested bounds of 33 and 110. Of the 298 NOAB designs found, there are 154 distinct designs with respect to the performance criteria, comprised of 74 designs in the Pareto set and 80 Pareto-dominated designs. The Pareto-dominated designs appear to perform similarly to the Pareto-set designs based on the scatter plots shown for the three measures of n, D-criterion, and average UPV (Figure 6).

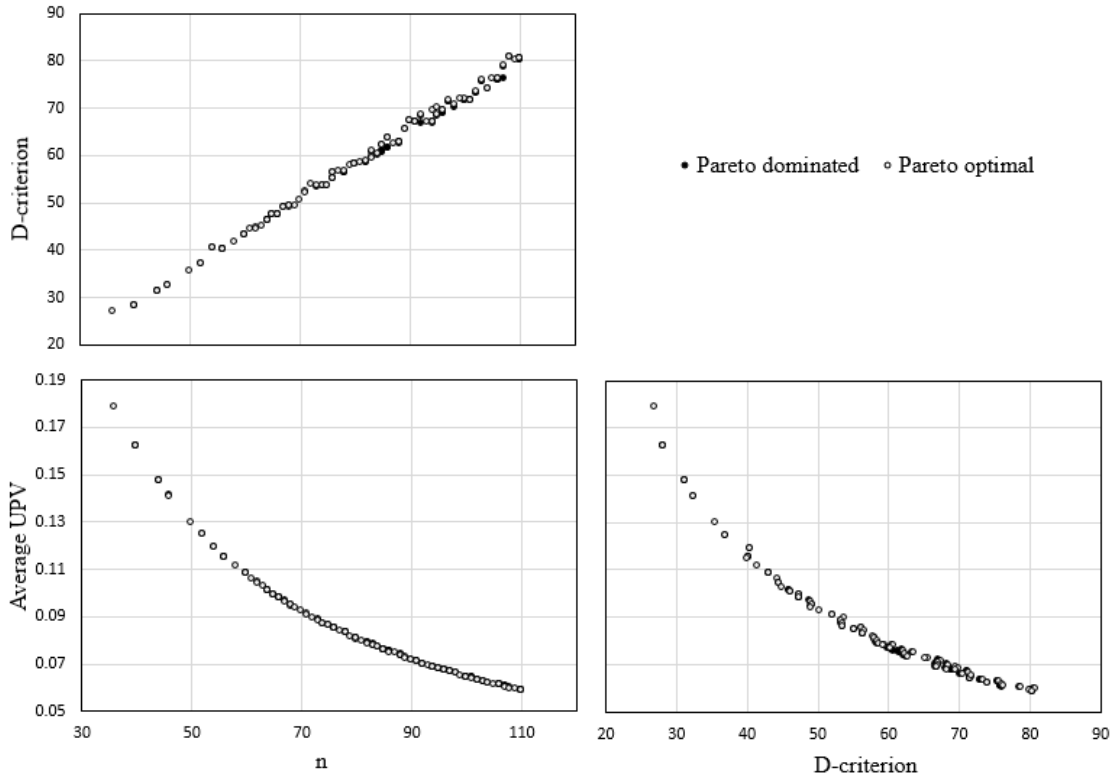
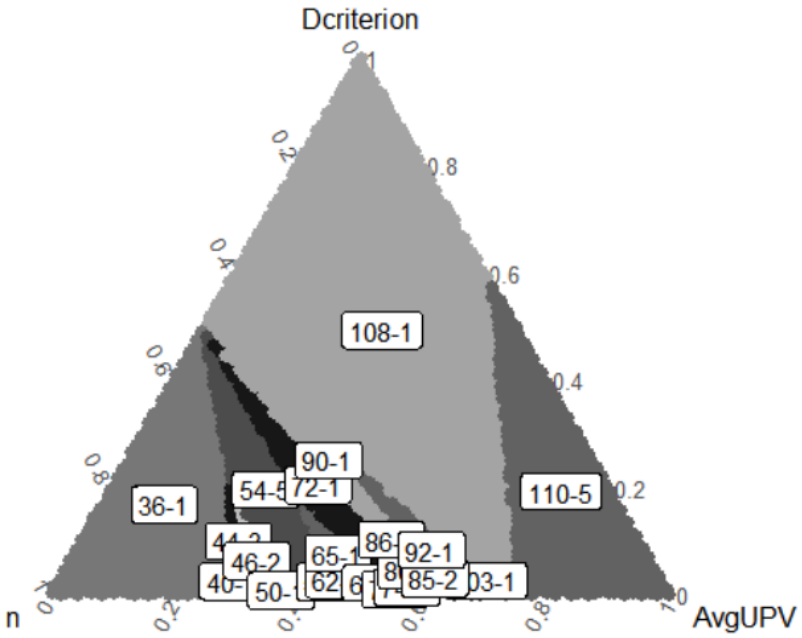


Figure 6. Performance Measures for 154 NOAB Designs

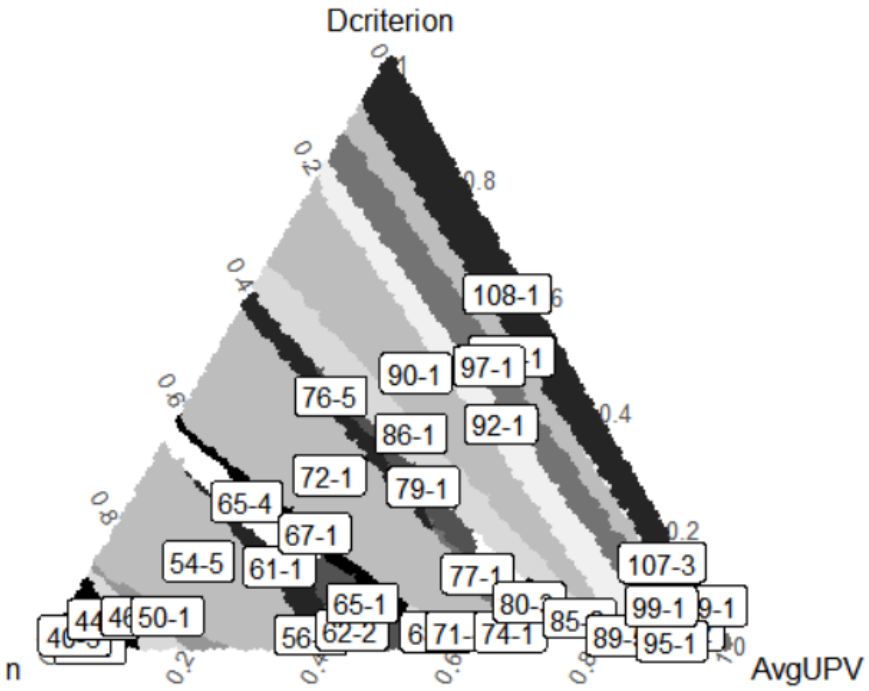
Let each individual desirability function be linear (i.e., $r = 1$). The NOAB designs having the best overall desirability for both the additive and multiplicative functions are determined for each of 5,000 different weighting combinations, constructed using a space-filling mixture design in JMP. Mixture plots are used to show the top performing design for various weighting combinations in the overall desirability function. Mixture plots for the additive desirability function (Figure 7) and the multiplicative desirability function (Figure 8) are shown, with a list of the top 10 performing designs based on estimated percentage of mixture area. The designs are labeled by design size and attempt number, so "110-5" would be the fifth attempt to construct a NOAB design of size 110.



Additive *D*	
Design	% Area
108-1	50.30%
36-1	15.14%
110-5	13.94%
54-5	6.54%
72-1	4.28%
90-1	2.50%
92-1	1.78%
65-1	1.30%
46-2	0.88%
44-3	0.78%

Figure 7. Best Designs for Weight Space with Additive Desirability

Promising designs include the top three performers for additive desirability (108-1, 36-1, 110-5) and the top five performers for multiplicative desirability (72-1, 90-1, 54-5, 108-1, 103-1). There are 21 designs found to be the top performer for some weighted combination in the additive desirability function as well as 33 designs for the multiplicative desirability function, comprising 35 distinct designs in total. The trade-off plot in Figure 9 shows the 35 top performing designs and their desirability score for each objective. As the design size n increases (lower desirability), there is a general increase in D-criterion and decrease in average UPV (higher desirability for both), so there is an apparent trade-off between experimental cost and design quality.



Multiplicative *D*	
Design	% Area
72-1	16.62%
90-1	12.44%
54-5	10.28%
108-1	8.86%
103-1	8.30%
92-1	7.68%
86-1	7.14%
97-1	5.92%
65-4	2.38%
65-1	2.30%

Figure 8. Best Designs for Weight Space with Multiplicative Desirability

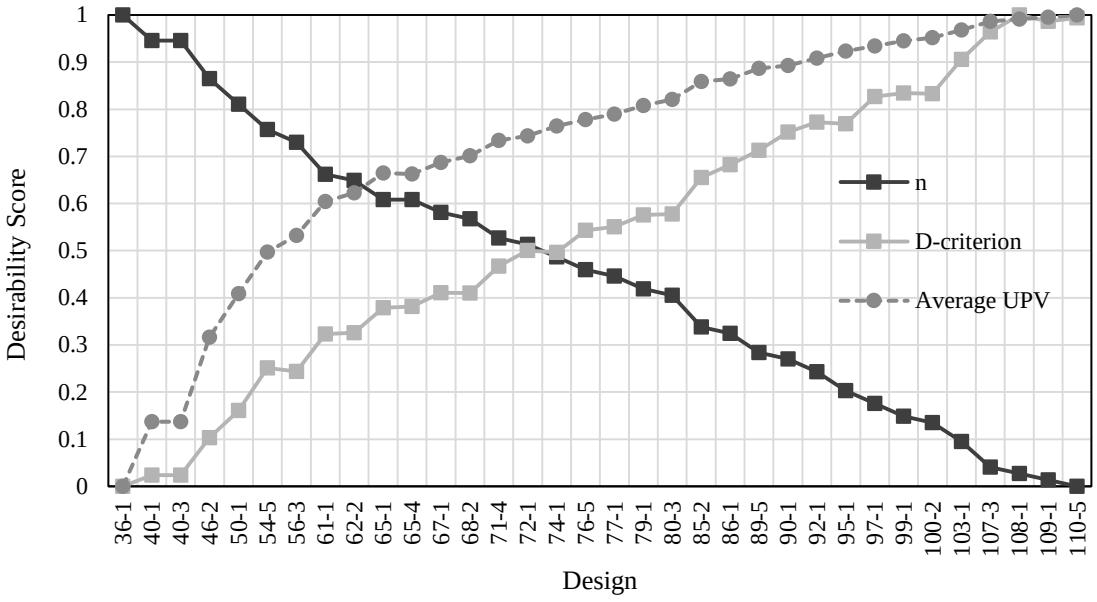


Figure 9. Trade-offs of Top Performing Designs

2.7.4 Synthesized Efficiency

Synthesized efficiency is defined as

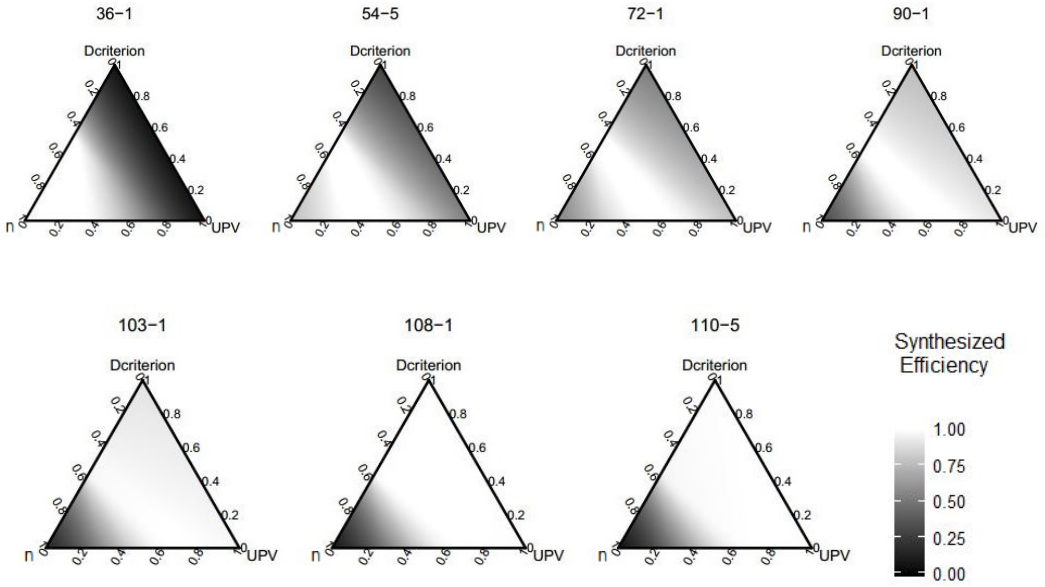
$$\frac{D(\xi, w_1, \ldots, w_m)}{\max_{\xi^*} D(\xi^*, w_1, \ldots, w_m)}$$

and can be used to examine how a single design ξ compares to the top performing design for various weighting combinations $(w_1, \ldots, w_m)$ of overall desirability [68, pp. 330]. The seven designs of interest for the ISR example are examined further using mixture plots of synthesized efficiency for each design (Figure 10). From these mixture plots, it appears that designs 36-1 and 110-5 are poor designs as they allow individual desirability scores to become too low, as seen with the multiplicative desirability, and have regions for additive desirability where synthesized efficiency is also low. Designs 72-1 and 90-1 appear to be most promising due to high synthesized efficiency values for much of the additive and multiplicative mixture areas.

2.7.5 Graphical Approaches

Additionally, a fraction of weight space (FWS) plot is used to compare multiple designs of interest, displaying synthesized efficiency by the fraction of weighted combinations with efficiency above. Figure 11 shows an FWS plot for five designs of interest (now excluding 36-1 and 110-1) and the multiplicative desirability function. Design 72-1 appears to be the most promising design, since when the design is compared to design 90-1, the synthesized efficiency is higher for FWS of greater than approximately 0.8 and only marginally lower near FWS of 0.5.

Additive Desirability



Multiplicative Desirability

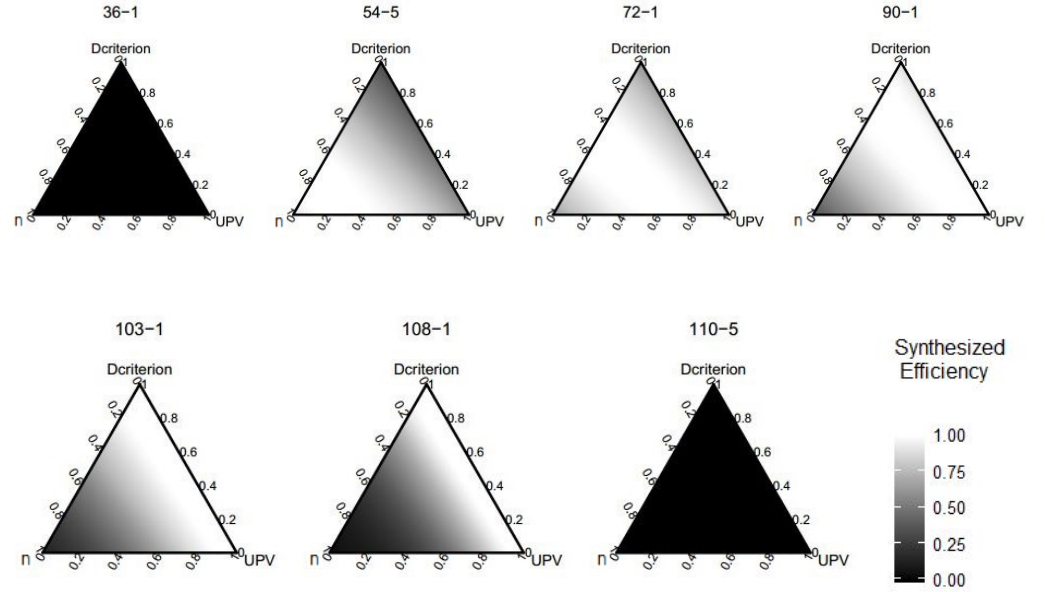


Figure 10. Synthesized Efficiency of Designs over Weight Space

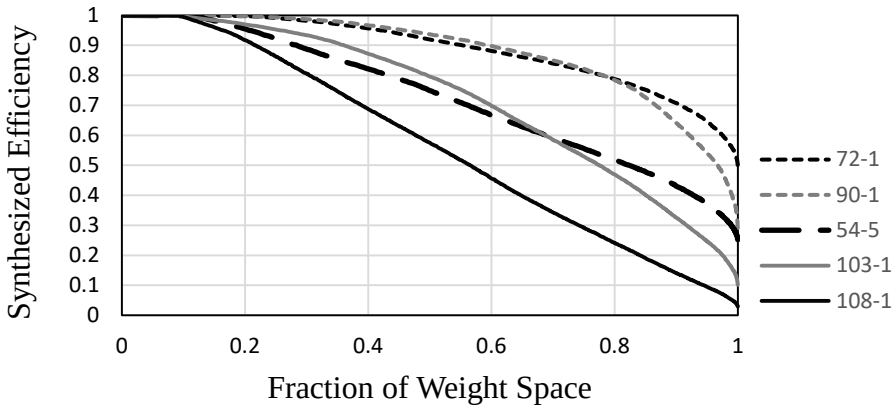


Figure 11. Fraction of Weight Space (FWS) Plot with Synthesized Efficiency Above

Additional graphical methods can be used, specifically for the evaluation of UPV. To ensure that the UPV is low for a large fraction of the design space, a fraction of design space (FDS) plot (Figure 12) can be examined to show the fraction of design space at or below a specific UPV [80].

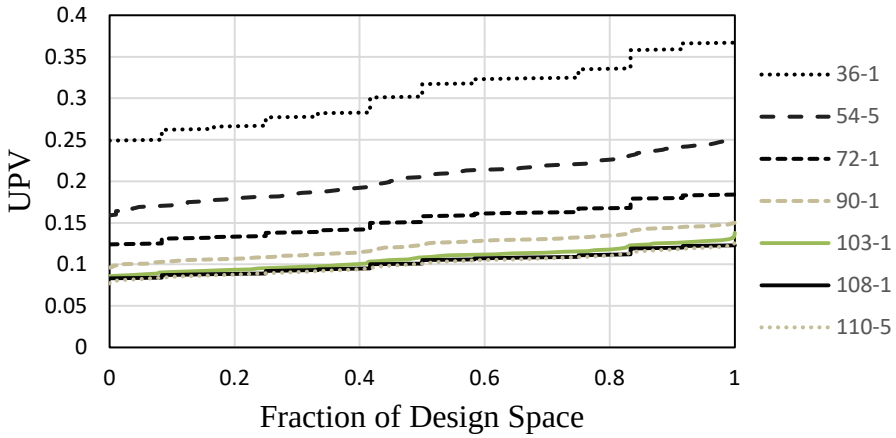


Figure 12. Fraction of Design Space (FDS) Plot for UPV

Though design 90-1 sees consistently lower UPV in the FDS plot, design 72-1 appears to be small enough in size to achieve a higher minimum synthesized efficiency over the entirety of the weight space, based on the mixture plots and FWS plot.

Upon calculating all UPV values for each design of interest and all 3,456 possible points in the ISR example design space to create the FDS plot, it is apparent that the JMP-reported average UPV values are not consistent with those calculated using the full factorial design. From Table 7, the ratio of the two approaches for average UPV values are consistent, with the calculated values approximately 71% higher than on average than the average UPVs reported in JMP. Thus, it appears that the relative magnitudes for the JMP-reported average UPVs are correct and the difference in average UPVs is possibly due to either JMP not recognizing the encoding of categorical factors with more than two levels or additional evaluations of UPV at intermediate points in an assumed continuous design space.

Table 7. Average UPV Comparison

Average UPV	36-1	54-5	72-1	90-1	103-1	108-1	110-1
Actual	0.3056	0.2039	0.1528	0.1222	0.1069	0.1019	0.1001
JMP	0.1787	0.1190	0.0894	0.0714	0.0623	0.0596	0.0585
(Actual / JMP)	1.7098	1.7135	1.7098	1.7117	1.7145	1.7098	1.7103

2.7.6 Discussion of Model Misspecification Criteria and Example

The NOAB design approach assumes no knowledge of the resulting responses and is stated to be well suited for highly nonlinear response surfaces, so it is reasonable to examine model misspecification criteria for higher-order parametric models. For the ISR portfolio example, the design matrix $\boldsymbol{X}$ has all first-order terms, while the alias matrix $\boldsymbol{A}$ accounts for all second-order terms (quadratic terms for quantitative factors with more than three levels and all two-way interactions). Of particular interest is the $tr(\boldsymbol{A}'\boldsymbol{A})$ criterion for

protection against biased coefficient estimates. The average UPVs as reported by JMP are also used here to provide consistency with the previous illustration of design comparison and evaluation.

Of the 298 NOAB designs constructed, there are 246 distinct designs with respect to the four performance measures of interest (n, D-efficiency, average UPV, and $tr(\boldsymbol{A}'\boldsymbol{A})$), comprised of 106 designs in the Pareto set and 140 Pareto dominated designs. Scatter plots for $tr(\boldsymbol{A}'\boldsymbol{A})$ and the three previous measures are provided in Figure 13, with the plots between other measures similar to Figure 6. This measure for protection against biased coefficient estimates appears to not have as direct of a relationship with the choice of n for the NOAB design construction, also seen in the trade-off plot for the four criteria (Figure 14).

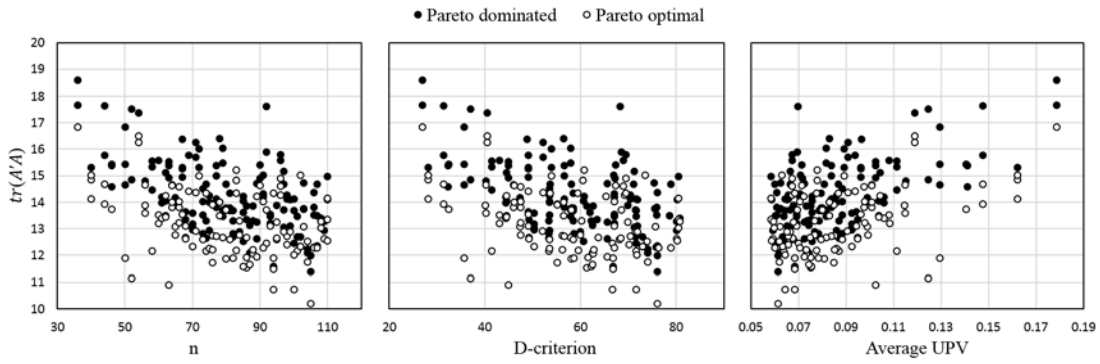


Figure 13. Design Performance of 246 NOAB designs (tr($\boldsymbol{A}'\boldsymbol{A}$) Included)

A 5,000-point mixture design was created in JMP for the four criteria. There are 32 designs found to be the top performer for some weighted combination in the additive desirability function as well as 44 designs for the multiplicative desirability function,

totaling 48 distinct designs. The top ten performing designs are listed in Table 8, based on percentage of weighted combinations where a design has the greatest overall desirability.

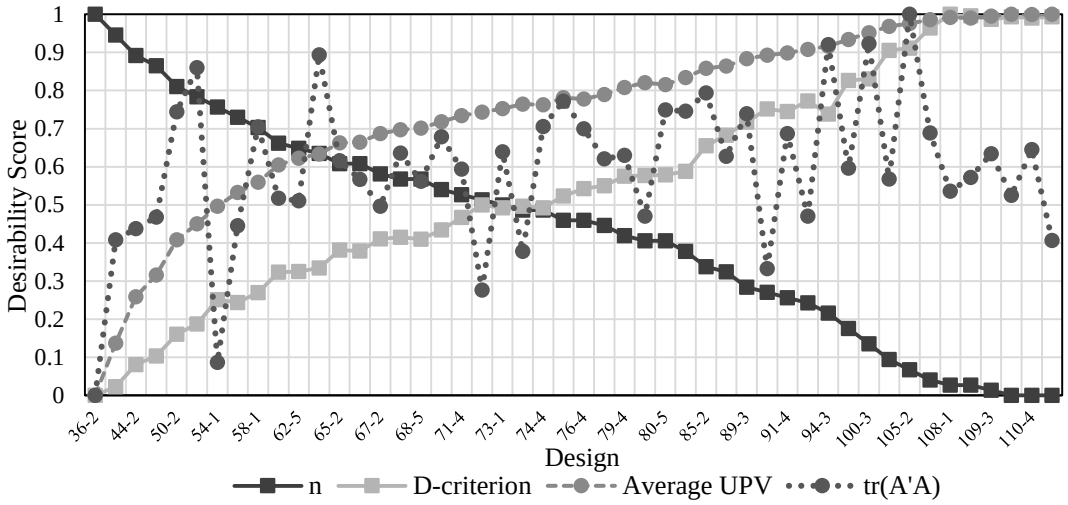


Figure 14. Design Performance Trade-offs (tr($\boldsymbol{A'A}$) Included)

Table 8. Top Performing Designs (% of Weight Space)

Additive *D*		Multiplicative *D*	
Design	% Mixture	Design	% Mixture
105-2	58.78%	63-3	22.72%
52-2	13.48%	105-2	20.72%
108-1	7.70%	94-3	15.16%
63-3	6.64%	85-2	9.42%
110-4	5.16%	52-2	7.52%
36-2	2.32%	76-3	4.92%
40-4	2.06%	100-3	3.90%
110-2	0.62%	76-4	3.08%
85-2	0.60%	89-3	1.94%
108-4	0.60%	97-3	1.00%

The top five performers for multiplicative desirability are promising (63-3, 105-2, 94-3, 85-2, 52-2), which include the top two performers for additive desirability (105-2, 52-2).

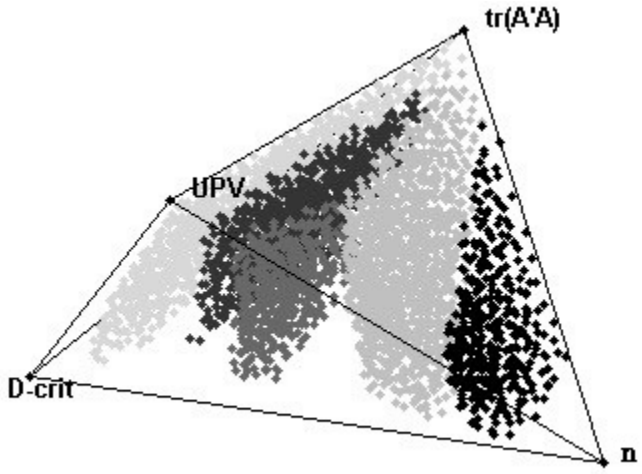


Figure 15. Top Five Performing Designs in Weight Space (tr($\boldsymbol{A'A}$) Included)

Figure 15 is a mixture plot showing the weighted combinations for the top five performing designs with respect to multiplicative desirability. The driving weight appears to be the design size (with each cluster of points representing 105-2, 94-3, 85-2, 63-3, 52-2 from left to right in Figure 15). However, the weighting of $tr(\boldsymbol{A'A})$ also appears be important as well, given that $tr(\boldsymbol{A'A})$ appears to not have as strong of a pattern. It is clear that visualization for more than three criteria becomes more challenging to construct and describe to a decision maker. A visual that is easier to convey performance information with is the FWS plot for synthesized efficiency (Figure 16), which shows design 85-2 as the most promising due to the slower decrease in synthesized efficiency over a large fraction of the weight space. Though the additional criteria of $tr(\boldsymbol{A'A})$ did change the

Pareto set of designs, it is clear that the trade-off between design quality and design size still exists to an extent (with designs having size near the guideline bounds not as robust as others to the weighting combinations).

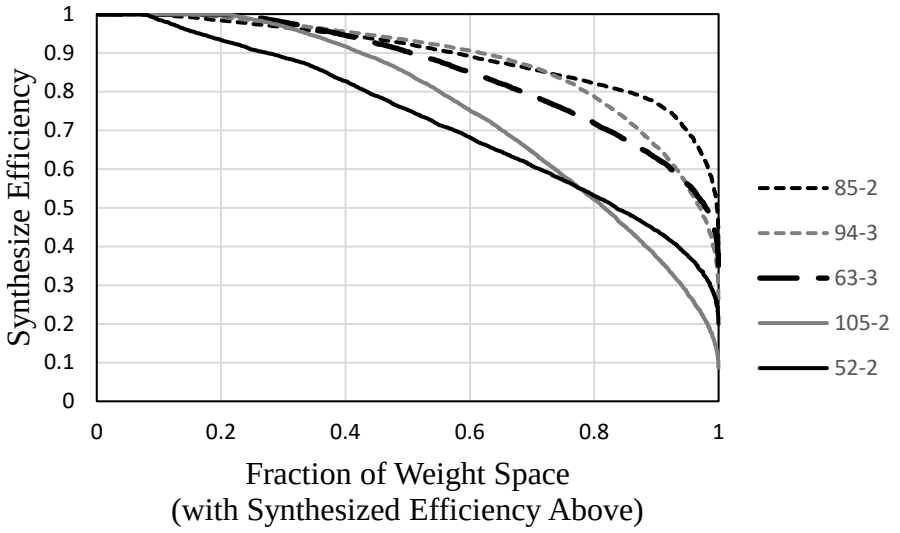


Figure 16. FWS Plot for Synthesized Efficiency (tr($\boldsymbol{A'A}$) Included)

2.8 Meta-modeling

With the first-order NOAB design construction method as implemented, the focus changes to meta-modeling for mixed factors. Different from traditional simulation optimization, the eventual aim is to perform trade-off analyses rather than simply find optimal, or sub-optimal, decisions. For global optimization of simulation, heuristics such as tabu search, evolutionary algorithms, and simulated annealing are commonly used. However, these methods are not as robust with respect to answering decision maker questions that focus on trade-offs over an entire decision space or even specific regions of a decision space, in particular when such questions can change over time. See [83] for an

overview of simulation optimization techniques. Potential meta-model constructions, partition trees, and other analytical and graphical methods will be explored in this research as appropriate. The following techniques are under initial consideration:

- artificial neural network (ANN)
- kriging (i.e., Gaussian process)
- polynomial regression
- multivariate adaptive regression splines (MARS)
- classification and regression trees (CART)
- radial basis function (RBF)
- support vector regression (SVR)

With the exception of CART, each of meta-models listed are used in [4]. JMP 12 allows for ANN, CART, kriging, and polynomial regression. MATLAB has toolboxes for ANN and general optimization, functions for CART, polynomial regression, RBF, and SVR, and open-source options for kriging and MARS. R software and Python both have open-source library packages and scripts available for each of these modeling techniques.

Though the NOAB design allows for the identification and elimination of insignificant factors without fear of losing information, some non-parametric meta-models may provide betters fits to simulation output, allowing for more accurate prediction and optimization, yet may not be as interpretable as polynomial regression.

Artificial neural networks [84]–[86] are models that are known to capably represent highly nonlinear surfaces. Often comprised of three layers - input, hidden, and output - containing nodes that somewhat represent how neurons are connected in a biological nervous system. It is possible to increase the number of hidden layers and associated neurons in the network, and much work has been done in examining parameter tuning of neural networks, as noted in [4]. In [87], general guidelines are provided for developing

ANN meta-models for simulation, in addition to a case study for job shop sequencing simulation.

Kriging [88], also known as a Gaussian process model, is a non-parametric method of interpolation that assumes the data are modeled by a Gaussian process, with the model comprised of a global polynomial model, $f(x)$, and a Gaussian random process, $Z(x)$, with zero mean and stationary covariance, as follows:

$$y(x) = f(x)\beta + Z(x)$$

where the correlation function in the covariance is often defined as the Gaussian correlation function [4]. Kriging has been found to perform well for highly nonlinear surfaces when compared to other commonly used models. In [37], general references for kriging are listed for both deterministic [89]–[91] and stochastic simulations [92]–[95].

Polynomial regression is a special case of linear regression [96], with polynomial terms up to some nth degree. As noted in [4], polynomial regression can be unstable for highly nonlinear surfaces [97].

Multivariate adaptive regression splines [98] is a non-parametric regression technique that has been show to work well with high dimensional, nonlinear data. The model is a weighted sum of a set of basis functions $B_i(x)$, as follows:

$$f(x) = a_0 + \sum_{i=1}^{m} a_i B_i(x)$$

where the basis functions are of three types: constant (i.e., the intercept term), a hinge function, and a product of hinge functions. A MARS model is constructed with basis functions using a forward and backward pass with generalized cross validation.

A classification and regression tree, or CART [99], is a recursive partitioning technique that constructs a binary decision tree for potentially both qualitative and quantitative data. Much like polynomial regression, a simple CART model is easier to understand than the non-parametric models discussed, yet often does not provide as good of prediction accuracy. Improvements to the single CART model with respect to prediction include the use of bagging, boosting, and random forests [100].

The radial basis function [101] is a linear combination of radial functions, $\phi(x)$, that interpolates some data set $\{x_1, x_2, \dots, x_n\}$, defined as follows:

$$f(x) = \sum_{i=1}^{n} w_i \phi(\|x - x_i\|)$$

where the coefficients w_i are found using the least-squares method. Vehicle crash simulations are studied in [102], where RBF is shown to perform well for highly nonlinear data and the most common basis functions are presented, with $r = \|x - x_i\|$ and $0 < c \leq 1$:

- thin-plate spline: $\phi(r) = r^2 \log(cr^2)$
- Gaussian: $\phi(r) = e^{-cr^2}$
- multiquadric: $\phi(r) = \sqrt{r^2 + c^2}$
- inverse multiquadric: $\phi(r) = 1/(r^2 + c^2)$

Support vector regression [103], or SVR, is a model of the following form that aims to have ε precision from each of m sample points while also aiming for flatness, resulting from a quadratic programming problem using Lagrangian theory:

$$f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) k(x_i, x) + b$$

where α_i and α_i^* are dual variables, and $k(x_i, x)$ is the kernel function, with common choices being linear, polynomial, Gaussian, sigmoid, and inhomogeneous polynomial

[104]. In an examination of 26 approximated functions, SVR was found to outperform kriging, MARS, RBF, and RSM with respect to overall accuracy as well as robustness across different sample sets [104].

2.9 Multiple Response Optimization

Once satisfactory models are obtained that approximate simulation outputs sufficiently, the overall value of the various portfolio options can be determined, dependent on decision maker questions and preferences. The use of desirability functions, a Pareto front, synthesized efficiency, and the various graphical approaches in design comparison and evaluation can be implemented, now with multiple simulation responses in place of design performance measures. Other simulation output mapping approaches can be considered in addition to desirability functions, such as value-focused thinking [58], lexicographic [105], or goal programming [106]. A survey of multi-objective optimization (MOO) methods is presented in [107].

2.10 The Algorithm Selection Problem and Meta-learning

2.10.1 Summary of [6]

Meta-learning was developed to understand learning algorithm performance for classification problems. In [6], Smith-Miles generalizes the developments in meta-learning from fields such as machine learning, artificial intelligence, computer science, statistics, and operations research, which are presented in a unified framework that considers the algorithm selection problem as a learning problem, generalizing tasks such as regression, time-series forecasting, sorting, constraint satisfaction, and optimization.

Researchers aim to understand algorithm performance for various problem types with the goal of learning which easy-to-obtain problem features are related to algorithm performance, with the abstraction of the algorithm selection problem provided in [108]. The No Free Lunch (NFL) theorems of [109] present an understanding that no single algorithm will perform best for a large set of problem types. The machine learning community saw the algorithm selection problem as a learning problem and applied such algorithms to classification problems. Smith-Miles notes a separation in the literature from this initial research in machine learning, which potentially slowed the progress of using meta-learning concepts in a broader range of problems. There are four common meta-learning prerequisites for the algorithm selection problem from the various fields: 1) a large number of diverse problem instances, 2) a large number of diverse algorithms, 3) measures of algorithm performance, and 4) problem instance features.

Rice formalizes the algorithm selection problem where the abstract model, displayed in Figure 17, is comprised of the problem space P, feature space F, algorithm space A, and performance space Y, with the algorithm selection problem stated as follows:

> "*For a given problem instance $x \in P$, with features $f(x) \in F$, find the selection mapping $S(f(x))$ into algorithm space A, such that the selected algorithm $\alpha \in A$ maximizes the performance mapping $y(\alpha(x)) \in Y$.*"
>
> [6, pp. 3]

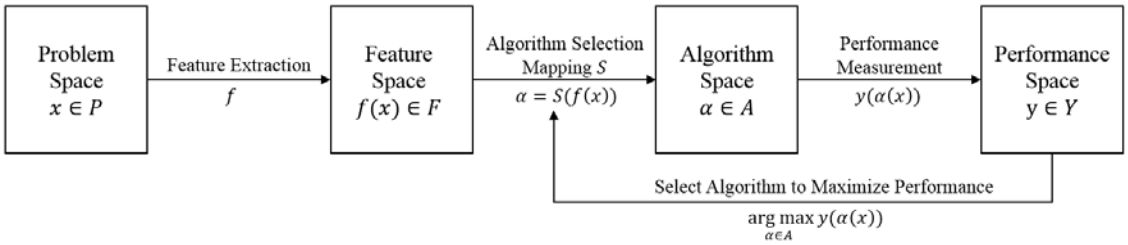


Figure 17. Diagram of Rice's model [4], [6], [108]

It is often difficult to capture the feature space F of a problem, due to the inherent complexities of many problems, as well as the selection of a mapping function S, which is itself an algorithm selection problem as noted by Smith-Miles. The training and test instances are also important for determining the mapping function S. The meta-learning process can lead to automated learning, the ranking of algorithms, algorithm combination, and self-adaptive algorithms.

Smith-Miles proceeds to examine the history of meta-learning for classification problems using the framework set by Rice's model. Foundational work in the machine learning community did not reference Rice's work, though feature spaces were constructed using the size and concentration of the problem classes for classification problems, with extensions to rule-based learning algorithms, where nearest-neighbor classifiers, set covering rule learners, and decision trees were the algorithms used in A as well as to learn the mapping S. Dynamic search also helped to develop rules to recognize algorithm performance and update algorithm selection accordingly.

From 1991 to 1994, the European Strategic Program on Research in Information Technology (ESPRIT) project StatLog (Comparative Testing of Statistical and Logical Learning) assessed the performance of machine learning, statistical, and neural methods

on classification problems. The feature space F was updated to include not only size of the data, but statistical measures and information theory measures as well. Algorithms used to learn the mapping S initially included decision trees, and later regression. Researchers continued to use the StatLog meta-data, with a notable advancement being the idea of using simple algorithms that are more efficient than calculating some meta-features, leading to the concept of landmarking.

A second ESPRIT project from 1998-2001 called METAL (*A Meta-learning Assistant for Providing User Support in Machine Learning Mining)* examined both classification and regression problems with goals of developing approaches for model selection and combination. Learning algorithms were updated to include neural networks, naïve Bayes and linear discriminant approaches, with performances measures of accuracy and time base on ten-fold cross-validation. K-nearest neighbor was used as an instance-based learning to predict algorithm rankings. Other recent developments include the use of an unsupervised approach, with self-organizing feature maps used to cluster classification datasets to identify common features and examine performance of each cluster.

Smith-Miles then discusses work in meta-learning for classification beyond algorithm selection, such as how to select optimal parameter settings (e.g., which kernel to use within support-vector machines (SVMs) for classification), which is argued to essentially be algorithm selection. A similar approach has been used for selecting the width of a radial basis function (RBF) kernel. In 2001, a framework is presented for using meta-learning to optimize parameter selection [110].

Smith-Miles then generalizes algorithm selection in other domains, to include regression, time-series forecasting, sorting algorithms, constraint programming, and optimization. Regression was examined in the METAL project. The suitability of meta-learning for regression problems was examined using mostly StatLog features, algorithms of neural networks as well as linear and quadratic discriminant analysis, and error rate performance measured by mean absolute deviation, mean square error, normalized mean absolute deviation, and normalized error/residual mean square [111].

Time-series forecasting work in the 1990's did not appear to reference the work of StatLog or Rice, though were similar in structure, using various forecasting methods as potential algorithms and average standard error as the performance measure. A two-stage neural network approach was developed to determine which group of algorithms is most appropriate for the specific time-series, whereupon a second neural network selects which algorithm in the selected group will give the smallest forecasting error. Algorithm ranking, clustering, and unsupervised learning approaches were also developed as of 2006. Sorting algorithms have been examined with notable classifiers including naïve Bayes and a Bayesian network learner, in addition to the use of dynamic algorithm selection for recursive sorting algorithms.

Constraint programming and artificial intelligence (AI) problems are also discussed, where the AI community in particular has focused on features associated with problem hardness as well as predicting and controlling problem computation time. Leyton-Brown et al. notably examine a constrained optimization for a combinatorial auction problem using a single algorithm in CPLEX, and use up to second-order regression and

spline models as learners for the mapping S [112]. Landmarking has also been used in constraint satisfaction problems, with simpler algorithm performances captured and used as problem features, in addition to dynamic algorithm selection.

Smith-Miles explains the two broad approaches to solving constraint satisfaction problems: the exact approach that may be restricted by computational complexity and available memory, and the heuristic approach that aims to find near-optimal solutions quickly. Here the performance space Y can be defined by computation time or solution quality. Meta-heuristics such as simulated annealing, tabu search, ant colony optimization, and evolutionary algorithms have been the focus of the operations research community, in addition to exact branch-and-bound algorithms. Efforts to learn relationships between such algorithm performance and problem features are ongoing, with similarities to landmarking and dynamic algorithm selection approaches.

Smith-Miles reiterates the generalized concepts of landmarking, dynamic algorithm selection, real-time analysis of algorithms, and algorithm design rather than selection. Any algorithm selection problem from various fields of study can be generalized when the four spaces of Rice's model are available. Additionally, the author proposed a three-phase framework for automated algorithm selection where the first phase involves the generation of meta-data for some training set of problem instances, the second phase learns from the meta-data to develop the mapping from instance features to performance measures and provide rules or rankings for the available algorithms, and the third phase examines the results from a theoretical view and for algorithm refinement. Domains for extensions include financial trading, help-desk automation, data compression algorithms,

bioinformatics (sequence alignment, gene prediction, protein identification, and pattern matching), cryptography, clustering, and matrix inversion algorithms.

In conclusion, the author generalizes algorithm selection problems from several different problem domains in order to show the common and distinct threads in research advancement as well as bridge the gap in vocabulary from the various literature.

2.10.2 Update of [6]

As an update to Smith-Miles' survey of meta-learning and generalization of algorithm selection problems from other domains, the aim here is to provide references to advancements in classification, regression, time-series forecasting, constraint satisfaction, optimization, and meta- modeling problems as well as generalize applications of interest from other fields to the language of Rice's model.

Bischl et al. have created a standard format for algorithm selection problems in the artificial intelligence community as well as the ASlib (Algorithm Selection Library) repository for data sets from the literature [113]. An R package is also available that provides benchmark machine learning models with problem scenarios, including the propositional satisfiability problem (SAT), maximum satisfiability (MAXSAT), and constraint satisfaction problem (CSP), among others. In the ASlib paper, Bischl et al. examine feature subset selection as well as three approaches to algorithm selection: classification to predict to best performer, regression to predict each algorithm's performance, and clustering to assign new instances to known problem instance clusters in the feature space with an associated algorithm.

Lemke et al. provide a survey on the new directions meta-learning has taken, including problems beyond algorithm selection/recommendation, and note that many of the frameworks or repositories regarding algorithm selection in particular, such as the METAL project, have not been maintained [114, pp. 122].

One could argue that the discussion of (meta-) model and parameter selection in the associated section in [6] could be combined with the discussion of regression problems or placed in a new section for meta-modeling. In line with Smith-Miles' discussion of SVMs, Gomes et al. combine search algorithms (particle swarm optimization and tabu search) with meta-learning for parameter selection [115].

Loterman and Mues use meta-learning for "comprehensible" regression models, including ordinary least squares (OLS), multivariate adaptive regression splines (MARS), classification and regression trees (CART), linear trees (CART with OLS leaves), and spline trees (CART with MARS at the leaves) [116]. Acknowledging Rice's framework, meta-features are binned by independent variable (size, dimensionality, and composition), dependent variable (symmetry and dispersion), and relationships (linear correlation, spline correlation, discriminatory power, and nonlinear correlation), with a performance measure of root mean square error (RMSE) for the validation set.

Rossi et al. present a meta-learning based method for algorithm selection called MetaStream that maps statistical meta-features from historical and incoming data to six algorithms (random forest (RF), SVM, CART, projection pursuit regression (PPR), and MARS), or a combination of these algorithms, for regression, using standard parameter settings in R for these meta-models [117]. Performance is measured using normalized mean

square error (NMSE) and classification error rate, while meta-learners (i.e., mapping functions) include RF, k-nearest neighbor, and Naive Bayes (NB).

Cui et al., also referencing Rice's framework, create a meta-modeling recommendation system with the four components [4]: the problems space *P* is comprised of 44 benchmark functions, the algorithm space *A* includes six meta-models (polynomial regression, kriging, SVR, RBF, MARS, and artificial neural networks), the performance space *Y* uses NRMSE (normalized root mean square error) for ranking with measures of Spearman's rank correlation and hit ratio, and the feature space *F* includes 15 meta-features describing the response values:

- the mean, median, standard deviation and maximum of the gradient of response surface point,
- mean, standard deviation, skewness, and kurtosis of response values,
- 25%, 50%, and 75% quartile of response values,
- outlier ratio,
- ratio of local minima and maxima, and
- averaged local biggest difference of response values.

Two meta-learning algorithms (mapping functions) are used: the instance-based k-nearest neighbor ranking approach and the model-based ANN. Cui et al. also compare singular value decomposition, stepwise regression, and ReliefF for meta-feature selection.

Wang et al. examine rule induction for selection of forecasting models using characteristics of univariate time series, including trend, seasonality, periodicity, serial correlation, skewness, kurtosis, non-linearity, self-similarity, and chaos [118]. The forecasting models examined include exponential smoothing (ES), auto-regressive integrated moving average (ARIMA), random walk (RW) and neural networks. Mapping

functions included self-organizing map clustering and characteristic-based meta decision trees (CMDT) using the C4.5 algorithm.

Matijaš et al. use a multi-variate learning system for load forecasting where the problem space *P* consists of 65 load forecasting tasks, the algorithm space *A* is comprised of RW, autoregressive moving average, similar days, layer recurrent neural network, multilayer perceptron, v-SVR, and robust LS-SVM, the feature space *F* introduces new meta-features to load forecasting, and the performance space *Y* uses mean absolute scaled error (MASE) for one year of testing cycles [119]. The mapping functions include SVM, CART, and Gaussian processes among several others.

Kück et al. study forecasting model selection for different feature sets, including error-based features as landmarking and statistical tests [120]. The authors reference Rice's and Smith-Miles's definitions of the algorithm selection problem for the model selection problem, where the problem space *P* is comprised of 111 time series from industry data, the feature space *F* uses global characteristics of time series, statistical and complexity measures, and error-based meta-features, and the algorithm space *A* are the four forecasting models of single, seasonal, seasonal-trended, and trended exponential smoothing. Additionally, the performance space *Y* is the averaged rolling-origin symmetric mean absolute percentage error (RO-sMAPE) on the hold out set of later time data. Neural network was used as a meta-learner (mapping function *S*), benchmarked by the use of aggregate model selection.

Burke et al. provide a survey of the state of the art in *hyper-heuristics*, examining both heuristic selection and heuristic generation and discussing problems such as vehicle

routing, bin packing, educational timetabling, satisfiability, the traveling salesman problem, workforce scheduling, and production scheduling, in addition to constraint satisfaction [121]. Referencing Rice's model, feature-based algorithm selection is developed for constrained continuous optimization among variants of differential evolution, particle swarm optimization, and evolution strategies [122].

Smith-Miles and Lopes provide a survey of combinatorial optimization problems to include assignment, traveling salesman, knapsack, bin-packing, graph, timetabling, and constraint satisfaction with a focus on problem features related to algorithm performance, where some features are independent of the problem to (fitness landscape and landmarking) and other features are problem specific [123]. A greater knowledge of these meta-features informs the meta-learning process for combinatorial optimization problems.

Feurer et al. use meta-learning for the hyper-parameter search problem to initialize sequential model-based Bayesian optimization (SMBO) for global optimization of black-box functions that are costly to evaluate [124]. Meta-features used are binned in five groups: principal component analysis (PCA), information theory, statistical, landmarking, and simple dataset features.

Muñoz et al. survey algorithm selection for black-box continuous optimization problems, starting from Rice's model, classifying various landscape analysis methods, and detailing the various algorithms and performance measures used [125]. Muñoz and Smith-Miles examine the use of *footprints* in the problem space, the regions where an algorithm is expected to perform well, for continuous black-box optimization [126].

A meta-learning approach to gene expression data classification has been developed using statistical, information theory, and basic dataset meta-features with meta-learners of nearest neighbor and SVM used to rank various algorithms [127]. Algorithm performance is measured by mean ranking accuracy (using Spearman's rank correlation coefficient) as well as weighted rank correlation.

Garcia et al. use meta-learning to predict performance of various noise filtering techniques for the identification of noisy data [128]. The performance of the filters is measured by the *F-score*, or F-measure (the harmonic mean of precision and recall, two measures common to pattern recognition and information retrieval systems). The mapping functions considered are k-nearest neighbor with Gaussian kernel, RF, and SVM.

Romero et al. use meta-learning to recommend a subset of 19 classification algorithms, all rule-based or tree-based, for datasets from an open-source learning platform called Moodle [129]. Meta-features include statistical, complexity, and domain (source of the dataset, such as report, quiz, or forum) features. Nearest-neighbor (1-NN) was used to recommend the classification algorithms, with a hold-one-out approach used to examine the performance of the various combinations of meta-features, measured by F-measure [129, pp. 4].

Meta-learning has also been used for bankruptcy prediction [130] and detecting financial fraud [131]. Cui et al. apply their meta-learning recommendation system to short-term building energy models [5].

Though progress has been made in connecting the various domains using meta-learning for algorithm selection and other problems, there still appears to be duplication in

effort across these domains as well as a lack of awareness of the benefits of meta-learning in fields such as computer science and software development, as noted in [132].

III. Second-order Extensions to Nearly Orthogonal-and-balanced (NOAB) Mixed-factor Experimental Designs

3.1 Abstract

When simulation studies involve many quantitative and qualitative factors with different numbers of choices for each, meta-models of simulation responses can benefit from the use of mixed-factor space-filling designs. The first-order nearly orthogonal-and-balanced (NOAB) design is a popular approach in these situations. This research develops second-order extensions for an existing construction method of NOAB designs, estimating the pairwise correlations between possible first-order and second-order terms. These extensions permit additional linear constraints in the mixed-integer linear programming (MILP) formulations previously developed for first-order NOAB designs. A case study is presented for NOAB designs of different sizes and construction approaches. The second-order MILP extensions show improvements in performance measures for parameter estimation and prediction variance for an assumed second-order model as well as for model misspecification with respect to second-order terms for an assumed first-order model.

Keywords: mixed-integer linear programming; pairwise correlation; categorical factor; model misspecification; meta-model

3.2 Introduction

Simulations and studies of black-box systems can involve a large number of both quantitative and qualitative factors of interest, and exhaustively simulating decision spaces can become infeasible due to computational requirements associated with problem scope

and fidelity. An efficient experimental design that can accommodate these computational challenges is desired so that meta-models can be constructed to estimate the resulting simulation outputs for an entire decision space. If the experimental design is created with forethought, meta-models can help facilitate robust decision support processes by preventing the need for future costly simulation runs when new questions are asked by decision makers.

Nearly orthogonal-and-balanced (NOAB) mixed-factor designs from [1] have been shown to have good space-filling and parameter estimation properties for large decision spaces. Space-filling designs allow for the estimation of models of greater than linear order under conditions when the order of the true model being estimated is unknown. NOAB designs provide *near orthogonality* between factors to better examine them independently of each other as well as *near balance* so that the levels of each factor are represented nearly equally. A *mixed-factor* design has some combination of continuous, discrete, and categorical factors in addition to possibly different numbers of levels for factors. The first-order NOAB design construction method from [1] aims to provide near orthogonality for linear order terms and uses a balance feasibility test to determine if a design size, n, can feasibly satisfy a specified maximum allowed imbalance, given design space properties. The method constructs design matrix columns for a single factor at a time, iterating until all factors are represented. The columns for the first factor can be randomly generated to satisfy (near) balance. The column structure of the remaining factors is then determined iteratively, one factor at a time, using one of three mixed-integer linear programming (MILP) problems based on factor type. The common objective of the sequence of MILPs

is to minimize the maximum absolute pairwise correlation between the factor columns currently under consideration and all previously constructed columns, while ensuring (near) balance with various linear constraints.

This paper introduces extensions to the original MILPs that allow for mixed-factor designs with near orthogonality between all first-order and second-order terms. Near orthogonality for second-order terms permits independent estimates of two-way interactions for both qualitative and quantitative factors as well as quadratic effects for quantitative factors. Consider a design space that represents a generic portfolio tradespace within a simulation study, where the factors can be both qualitative (which system to use) and quantitative (how many of a system to use). In addition to capturing the possible improvement associated with each individual system, these independent second-order estimates also identify any added benefit of using two different systems in combination as well as possible diminishing or increasing returns from increases in a system quantity.

Background material is next presented relating to the first-order NOAB, or *NOAB resolution III*, design construction method from [1], with design performance measures of interest also discussed. Then, the first-order MILP formulations are extended for the construction of second-order NOAB, or *NOAB resolution V* designs, with additional design approaches based on which pairs of first- and second-order terms are considered when minimizing the maximum absolute pairwise correlation for a design matrix. Four approaches are examined:

- *NOAB resolution III* (NOAB-III) – minimizes correlation between all first-order terms, from [1]

- *Quadratic only* (NOAB-Q) – minimizes correlation between all first-order and quadratic terms
- *NOAB resolution IV* (NOAB-IV) – ignores correlation between pairs of second-order terms
- *NOAB resolution V* (NOAB-V) – minimizes correlation between all first- and second-order terms

The NOAB-Q design approach is used to examine possible improvements to design performance when selecting a small subset of second-order terms. The NOAB-IV designs are considered with the intention of constructing efficient screening designs that protect against bias from second-order terms. A case study is presented where NOAB designs are constructed using the four different approaches for various design sizes.

3.3 Material and Methods

3.3.1 Experimental Designs

There are many standard designs [9]–[11] that do not simultaneously allow for mixed factors, a relatively low number of design points, and good parameter estimation, with strong space-filling properties. With respect to space-filling designs, improvements have been made to the Latin hypercube design, including the orthogonal Latin hypercube [31]–[33] as well as the nearly orthogonal Latin hypercube (NOLH) [34], [35]. However, these standard designs do not account for categorical factors, and the various techniques' use of rounding of design point values from continuous to discrete does not guarantee near orthogonality. An example of this rounding for NOLH designs is provided in [12]. Second-order NOLH designs have been created using a genetic algorithm approach for continuous factors [36]. The first-order NOAB mixed-factor designs presented in [1] perform well with respect to measures for good parameter estimation (D-efficiency), near orthogonality

(ρ_{map}), near balance (δ), and space-filling properties when compared to other designs, including orthogonal arrays, computer-generated optimal designs, and various space-filling designs (Latin hypercube, maximum entropy, sphere packing, and uniform).

3.3.2 First-order NOAB Designs: Notation and General Formulation

Inputs for the NOAB design construction method include:

- number of design points (matrix rows) n, indexed by row $r = 1,\ 2, \ldots,\ n$
- maximum allowed absolute pairwise correlation ρ_{map}
- maximum allowed imbalance δ
- factor types (continuous, discrete, or categorical) for each factor x
- number of levels ℓ_x for each factor x, indexed by level $i = 1, 2, \ldots, \ell_x$

Orthogonality permits independent factor effect estimates and, depending on the eventual meta-model used for each simulation response, clearer model interpretation. Perfect independence among columns is difficult to obtain in designs capable of estimating higher order models. Pairwise correlation for columns $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as $\rho(\boldsymbol{x}, \boldsymbol{y}) = 1/((n-1)\, s_{\boldsymbol{x}} s_{\boldsymbol{y}}) \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}})(y_r - \overline{\boldsymbol{y}})$, with column elements x_r and y_r, means $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$, and standard deviations $s_{\boldsymbol{x}}$ and $s_{\boldsymbol{y}}$. Orthogonality can be measured by the maximum absolute correlation of all appropriate pairs of factor columns, denoted by $\rho_{map} = \max_{\boldsymbol{x} \neq \boldsymbol{y}} |\rho(\boldsymbol{x}, \boldsymbol{y})|$, where a design is considered orthogonal if $\rho_{map} = 0$, and nearly orthogonal if $\rho_{map} \leq 0.05$. The imbalance for a factor x is defined as $\delta_x = \max_{i=1,\ldots,\ell_x} |(w_{i,x} - (n/\ell_x))/(n/\ell_x)|$, where $w_{i,x}$ is the number of times level i occurs for factor x [1]. A design is considered nearly balanced when the maximum imbalance, $\delta = \max_x \delta_x$, is close to zero.

Table 9 provides a summary of notation used to describe the original first-order NOAB design construction approach as well as the second-order extensions derived in this

work, with matrices and vectors in bold. The factor type informs which of the three MILP formulations in the original method is used for column construction. A single column is created for both the continuous and discrete factor cases, while $\ell_x - 1$ columns are created for each categorical factor x to account for {-1, 0, 1} effect coding. For simplicity in indexing, the three MILP formulations and second-order extensions are generalized to show a single factor column $\boldsymbol{x}$. Continuous factor columns are permitted exactly $\lambda(\boldsymbol{x}) = n$ evenly spaced design point values, which ensures balance. For discrete factor columns, the number of possible values is equal to the number of desired levels, so $\lambda(\boldsymbol{x}) = \ell_x$. The {-1, 0, 1} effect coding for categorical factors gives $\lambda(\boldsymbol{x}) = 3$ possible values in each column.

Table 9. Notation for Second-order NOAB Design Construction

j	number of previously constructed matrix columns, indexed by column $c = 1, 2, \ldots, j$
$\boldsymbol{M}$	previously constructed $n \times j$ design matrix (represents only first-order terms in the original method and both first- and second-order terms for the full second-order method)
$m_{r,c}$	element of $\boldsymbol{M}$ in row r and column c
$\boldsymbol{m}_{\cdot,c}$	column c of $\boldsymbol{M}$
C_1	subset of column indices $1, 2, \ldots, j$ for $\boldsymbol{M}$ that represent first-order terms only, indexed by c_1
$\boldsymbol{x}$	MILP decision variables ($n \times 1$ factor column)
x_r	element of $\boldsymbol{x}$ in row r
$\boldsymbol{x}_0$	initial randomly-generated MILP solution ($n \times 1$ column)
$\boldsymbol{z}$	centered MILP decision variable ($n \times 1$ column), with $z_r = x_r - \overline{\boldsymbol{x}} = x_r - (1/n)\sum_{k=1}^{n} x_k$
$\lambda(\boldsymbol{x})$	number of encoded levels for column $\boldsymbol{x}$, indexed by encoded level ℓ = 1, 2, …,$\lambda(\boldsymbol{x})$

π_ℓ	encoded level value (with $\{\pi_1, \pi_2, \ldots, \pi_{\lambda(\boldsymbol{x})}\}$ being all possible values for column $\boldsymbol{x}$)
$\theta_{r,\ell}$	binary decision variable where $x_r = \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell \theta_{r,\ell}$ and $\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \theta_{r,\ell} = 1$ for row r and encoded level ℓ

In the original first-order method, each pairwise correlation between factor column $\boldsymbol{x}$ and the previously constructed columns in matrix $\boldsymbol{M}$ representing only first-order terms (i.e., $\boldsymbol{m}_{\cdot,c}, c = 1,\ 2,\ \ldots,\ j$) is estimated by

$$\rho^*(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c}) = \rho(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c}) s_{\boldsymbol{x}} = \ 1/((n-1)\ s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}}) \left(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}}\right)$$

in order to have linear constraints for the MILP (Figure 18) when constructing each column $\boldsymbol{x}$. These estimates are considered accurate enough for the MILP, since changes to a nearly balanced $\boldsymbol{x}$ will result in relatively small changes to $s_{\boldsymbol{x}}$. The general MILP formulation for the first-order method is:

Minimize		v	
Subject to	(i)	$v \geq \rho^*(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c})$	$c = 1,\ 2, \ldots,\ j$
	(ii)	$v \geq -\rho^*(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c})$	$c = 1,\ 2, \ldots,\ j$
		$\boldsymbol{x} \in \Omega$	

Figure 18. General MILP Formulation for First-order Method

Constraints (i) and (ii) ensure that the maximum absolute value of $\tilde{\rho}(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c})$ for all $c = 1,\ 2,\ \ldots,\ j$ is minimized. For additional constraints that ensure $\boldsymbol{x}$ is balance-feasible, i.e., $\boldsymbol{x} \in \Omega$, based on factor types of continuous, discrete, and categorical, see [1]. A general guideline for design size of first-order NOAB designs, which will inform the case study that follows, is $3J \leq n \leq 10J$, where *J* is the number of encoded columns that correspond to first-order terms [1].

3.3.3 Design Performance Measures

For design performance measures, emphasis is placed on low experimental cost (as measured by design size, n) as well as on good model parameter estimation and prediction accuracy, while also accounting for model misspecification when appropriate. To distinguish among similar designs sizes, the average unscaled prediction variance (UPV $=$ $\boldsymbol{x}^{(q)\prime}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{x}^{(q)}$ for design matrix $\boldsymbol{X}$) over all possible design points q is examined in place of the average scaled prediction variance (SPV $= n \cdot$ UPV) as discussed in [59], [77]–[79]. UPV can also be examined in fraction of design space (FDS) plots to serve as a complementary look at UPV that does not rely solely on the single-valued, average UPV [77]. For good parameter estimation, the D-criterion $|\boldsymbol{X}'\boldsymbol{X}|^{1/p}$ for p model parameters uses the unscaled moment matrix as in [76]. If $\boldsymbol{X}_1$ is the assumed linear model matrix and $\boldsymbol{X}_2$ includes additional linear terms excluded from the defined model, then the alias matrix $\boldsymbol{A} = (\boldsymbol{X}_1'\boldsymbol{X}_1)^{-1}(\boldsymbol{X}_1'\boldsymbol{X}_2)$ gives the degree of biasing of each linear model term in $\boldsymbol{X}_1$ due to each term in $\boldsymbol{X}_2$. A common model misspecification measure is $tr(\boldsymbol{A}'\boldsymbol{A})$ [133]. For this research, $\boldsymbol{X}_2$ contains all second-order terms, i.e., quadratic terms for continuous and discrete factors having more than two levels as well as two-way interactions. The aim is to minimize n, average UPV, and $tr(\boldsymbol{A}'\boldsymbol{A})$, while also maximizing the D-criterion.

3.4 Theory

To account for second-order terms with respect to near orthogonality, extensions to the MILP formulations from [1] in Figure 18 are made through additional (i) and (ii) correlation constraints for five new cases of v, denoted by $v_1, v_2, \ldots, v_5$. The Hadamard

(element-wise) product, denoted by the ∘ operator, is used to account for second-order terms in column (vector) form. In the first-order method, only correlations between current column $\boldsymbol{x}$ and the columns in $\boldsymbol{M}$ are considered, where $\boldsymbol{M}$ contains only columns associated with first-order terms. The second-order method requires matrix $\boldsymbol{M}$ to now include columns for the desired two-way interactions and quadratics associated with the previously constructed first-order columns. Additionally, first-order columns for quantitative factors should be centered to have low correlation between pairs of first-order and second-order terms, so the extended formulation needs to account for the centered column $\boldsymbol{z}$, two-way interactions with all previous first-order columns ($\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, c_1 \in C_1$), and the quadratic column ($\boldsymbol{z} \circ \boldsymbol{z}$). This results in the five new cases for pairwise correlations between:

1. two-way interaction columns $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, c_1 \in C_1$ and columns $\boldsymbol{m}_{\cdot,c}$, $c = 1, 2, \ldots, j$
2. quadratic column $\boldsymbol{z} \circ \boldsymbol{z}$ and columns $\boldsymbol{m}_{\cdot,c}, c = 1, 2, \ldots, j$
3. first-order column $\boldsymbol{z}$ and two-way interaction columns $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, c_1 \in C_1$
4. first-order column $\boldsymbol{z}$ and quadratic column $\boldsymbol{z} \circ \boldsymbol{z}$
5. quadratic column $\boldsymbol{z} \circ \boldsymbol{z}$ and two-way interaction columns $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, c_1 \in C_1$

Correlation estimates for these five extensions are derived so that they are linear with respect to x_r and $\theta_{r,\ell}$ decision variables, which permits the mathematical programming formulations to remain linear. An additional extension for correlations between two-way interaction columns $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, c_1 \in C_1$ and $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_2}, c_2 \in C_1$ such that $c_1 \neq c_2$ is not required, since low absolute correlations for terms from the five extensions consistently results in low absolute correlations between the associated interaction terms. The quadratic terms for categorical factor columns are not of interest, so the second-order formulation for categorical factors requires only extensions 1 and 3, using $\boldsymbol{x}$ rather than

centered $\boldsymbol{z}$ to preserve effect coding. The general MILP formulation from the original method is updated in Figure 19 to show the five (centered) extensions, where the set of balance-feasible decision variables $\boldsymbol{x} \in \Omega$ are defined as in the original first-order method based on factor type.

Minimize	$v + v_1 + v_2 + v_3 + v_4 + v_5$	
Subject to		
(i)	$v \geq \tilde{\rho}(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c})$	$c = 1,\ 2,\ \ldots,\ j$
(ii)	$v \geq -\tilde{\rho}(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c})$	$c = 1,\ 2,\ \ldots,\ j$
$(i$-1)	$v_1 \geq \tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c})$	$c_1 \in C_1; c = 1,\ 2,\ \ldots,\ j$
$(ii$-1)	$v_1 \geq -\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c})$	$c_1 \in C_1; c = 1,\ 2,\ \ldots,\ j$
$(i$-2)	$v_2 \geq \tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{m}_{\cdot,c})$	$c = 1,\ 2,\ \ldots,\ j$
$(ii$-2)	$v_2 \geq -\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{m}_{\cdot,c})$	$c = 1,\ 2,\ \ldots,\ j$
$(i$-3)	$v_3 \geq \tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1$
$(ii$-3)	$v_3 \geq -\tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1$
$(i$-4)	$v_4 \geq \tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{z})$	
$(ii$-4)	$v_4 \geq -\tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{z})$	
$(i$-5)	$v_5 \geq \tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1$
$(ii$-5)	$v_5 \geq -\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1$
where	$\boldsymbol{z} = \boldsymbol{x} - \overline{\boldsymbol{x}},$	$\boldsymbol{x} \in \Omega$

Figure 19. General MILP Formulation with (Centered) Extensions 1 through 5

The pairwise correlation from the original method is now estimated by $\tilde{\rho}(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c}) = 1/((n-1)\, s_{\boldsymbol{x_0}} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}})(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}})$. The linearity of these correlation estimates with respect to the decision variables is, in part, made possible by the use of a randomly-generated (nearly) balanced initial solution $\boldsymbol{x}_0$, rather than decision variables $\boldsymbol{x}$, to estimate various means and standard deviations. The derivations also use the two propositions that follow.

Proposition 1. An important property used to simplify the correlation estimates for second-order terms is that for any constant k (with respect to r) and for any column $\boldsymbol{x}$, $\sum_{r=1}^{n} k(x_r - \overline{\boldsymbol{x}}) = 0$.

Proof: By definition, $\overline{\boldsymbol{x}} = (1/n)\sum_{r=1}^{n} x_r$, so $\sum_{r=1}^{n} \overline{\boldsymbol{x}} = n\overline{\boldsymbol{x}} = \sum_{r=1}^{n} x_r$.
Thus, $\sum_{r=1}^{n} k(x_r - \overline{\boldsymbol{x}}) = k\sum_{r=1}^{n}(x_r - \overline{\boldsymbol{x}}) = k(\sum_{r=1}^{n} x_r - \sum_{r=1}^{n} \overline{\boldsymbol{x}}) = k(0) = 0$. ■

Proposition 2. This property concerns the binary representation, $x_r = \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell \theta_{r,\ell}$, where $\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \theta_{r,\ell} = 1$ and $\theta_{r,\ell} \in \{0,1\}$. For any $p \in \mathbb{N}$, the following holds: $x_r^p = (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell \theta_{r,\ell})^p = \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell^p \theta_{r,\ell}$.

Proof: Fix row r. By induction, the $p = 1$ case is true by definition.
Suppose $x_r^p = \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell^p \theta_{r,\ell}$. Since $\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \theta_{r,\ell} = 1$ and $\theta_{r,\ell} \in \{0,1\}$, it is necessary that $\theta_{r,\ell} = 1$ for exactly one $\ell \in \{1,2,\ldots,\lambda(\boldsymbol{x})\}$.
Thus, $x_r^{p+1} = x_r^p(x_r) = (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell^p \theta_{r,\ell})(\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell \theta_{r,\ell})$
$= \sum_{\ell_1=1}^{\lambda(\boldsymbol{x})} \sum_{\ell_2=1}^{\lambda(\boldsymbol{x})} (\pi_{\ell_1}^p \theta_{r,\ell_1})(\pi_{\ell_2} \theta_{r,\ell_2})$
$= \sum_{\ell_1=1,\ell_2\neq\ell_1}^{\lambda(\boldsymbol{x})} (\pi_{\ell_1}^p \theta_{r,\ell_1})(\pi_{\ell_2} \theta_{r,\ell_2}) + \sum_{\ell_1=1,\ell_2=\ell_1}^{\lambda(\boldsymbol{x})} (\pi_{\ell_1}^p \theta_{r,\ell_1})(\pi_{\ell_2} \theta_{r,\ell_2})$
$= 0 + \sum_{\ell_1=1,\ell_1=\ell_2}^{\lambda(\boldsymbol{x})} (\pi_{\ell_1}^p \theta_{r,\ell_1})(\pi_{\ell_2} \theta_{r,\ell_2})$
$= \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell^{p+1} \theta_{r,\ell}^2$
$= \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell^{p+1} \theta_{r,\ell}$. ■

The second-order MILP extensions are now derived, including the non-centered versions of extensions 1 and 3 for categorical factors.

Extension 1. Correlation estimates between interactions $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}$ and previous columns $\boldsymbol{m}_{\cdot,c}$, for all $c_1 \in C_1$ and $c = 1, 2, \ldots, j$:

$$\tilde{\rho}\left(\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c}\right) = 1/((n-1)s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} m_{r,c_1}(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})(x_r - \overline{\boldsymbol{x}}).$$

Derivation: From the definition of pairwise correlation and using centered initial solution $\boldsymbol{z}_0 = \boldsymbol{x}_0 - \overline{\boldsymbol{x}_0}$ to estimate the standard deviation of the interaction terms,

$\rho(\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c})(n-1)s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}} s_{\boldsymbol{m}_{\cdot,c}} \approx \sum_{r=1}^{n}(z_r m_{r,c_1} - \overline{\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}})(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$

$= \sum_{r=1}^{n}(z_r m_{r,c_1})(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$ by proposition 1 for constant $-\overline{\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}}$ with respect to r

$= \sum_{r=1}^{n}(x_r - \overline{\boldsymbol{x}})m_{r,c_1}(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$ for centered $\boldsymbol{z}$.

Extension 1 (non-centered). Similarly, correlation estimates between interactions $\boldsymbol{x} \circ \boldsymbol{m}_{\cdot,c_1}$ and previous columns $\boldsymbol{m}_{\cdot,c}$, for all $c_1 \in C_1$ and $c = 1, 2, \ldots, j$:

$$\tilde{\rho}\left(\boldsymbol{x} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c}\right) = 1/((n-1)s_{\boldsymbol{x}_0 \circ \boldsymbol{m}_{\cdot,c_1}} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} m_{r,c_1}(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})x_r.$$

Extension 2. Correlation estimates between quadratic term $\boldsymbol{z} \circ \boldsymbol{z}$ and previous columns $\boldsymbol{m}_{\cdot,c}$, for all $c = 1, 2, \ldots, j$:

$$\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{m}_{\cdot,c}) = 1/((n-1)s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n}(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^2 - 2\overline{\boldsymbol{x}_0}\pi_\ell)\theta_{r,\ell}).$$

Derivation: By definition and using $\boldsymbol{z}_0$ to estimate the standard deviation of the quadratic term, $\rho(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{m}_{\cdot,c})(n-1)s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0} s_{\boldsymbol{m}_{\cdot,c}} \approx \sum_{r=1}^{n}(z_r^2 - \overline{\boldsymbol{z} \circ \boldsymbol{z}})(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$

$= \sum_{r=1}^{n} z_r^2(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$ by proposition 1 for constant $-\overline{\boldsymbol{z} \circ \boldsymbol{z}}$ with respect to r

$= \sum_{r=1}^{n}(x_r - \overline{\boldsymbol{x}})^2(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$ for centered $\boldsymbol{z}$

$= \sum_{r=1}^{n}(x_r^2 - 2\overline{\boldsymbol{x}}x_r + \overline{\boldsymbol{x}}^2)(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$

$= \sum_{r=1}^{n}(x_r^2 - 2\overline{\boldsymbol{x}}x_r)(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$ by proposition 1 for constant $\overline{\boldsymbol{x}}^2$ with respect to r

$= \sum_{r=1}^{n}(\sum_{\ell=1}^{\lambda(x)} \pi_\ell^2 \theta_{r,\ell} - 2\overline{\boldsymbol{x}} \sum_{\ell=1}^{\lambda(x)} \pi_\ell \theta_{r,\ell})(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$ by proposition 2 (binary representation)

$= \sum_{r=1}^{n}(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^2 - 2\overline{\boldsymbol{x}}\pi_\ell)\theta_{r,\ell})(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$

$\approx \sum_{r=1}^{n}(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^2 - 2\overline{\boldsymbol{x}_0}\pi_\ell)\theta_{r,\ell})(m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})$ by estimation of $\overline{\boldsymbol{x}}$ with $\overline{\boldsymbol{x}_0}$.

Extension 3. Correlation estimates between $\boldsymbol{z}$ and interactions $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}$, for all $c_1 \in C_1$:

$$\tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}) = 1/((n-1)s_{\boldsymbol{z}_0} s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}}) \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^2 - 2\overline{\boldsymbol{x}_0}\pi_\ell)\theta_{r,\ell} + \overline{\boldsymbol{x}_0}^2).$$

Derivation: By definition and using $\boldsymbol{z}_0$ to estimate the standard deviations,

$\rho(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})(n-1)s_{\boldsymbol{z}_0} s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}} \approx \sum_{r=1}^{n} (z_r - \bar{\boldsymbol{z}})(z_r m_{r,c_1} - \overline{\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}})$

$= \sum_{r=1}^{n} (z_r - \bar{\boldsymbol{z}}) z_r m_{r,c_1}$ by proposition 1 for constant $-\overline{\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}}$ with respect to r

$= \sum_{r=1}^{n} m_{r,c_1} (x_r - \overline{\boldsymbol{x}})^2$ for centered $\boldsymbol{z}$ where $\bar{\boldsymbol{z}} = 0$

$= \sum_{r=1}^{n} m_{r,c_1} (x_r^2 - 2\overline{\boldsymbol{x}} x_r + \overline{\boldsymbol{x}}^2)$

$= \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell^2 \theta_{r,\ell} - 2\overline{\boldsymbol{x}} \sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell \theta_{r,\ell}) + \overline{\boldsymbol{x}}^2)$ by proposition 2

$\approx \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^2 - 2\overline{\boldsymbol{x}_0}\pi_\ell)\theta_{r,\ell} + \overline{\boldsymbol{x}_0}^2)$ by estimation of $\overline{\boldsymbol{x}}$ with $\overline{\boldsymbol{x}_0}$.

Extension 3 (non-centered). Similarly, correlation estimates between $\boldsymbol{x}$ and interactions $\boldsymbol{x} \circ \boldsymbol{m}_{\cdot,c_1}$, for all $c_1 \in C_1$:

$$\tilde{\rho}(\boldsymbol{x}, \boldsymbol{x} \circ \boldsymbol{m}_{\cdot,c_1}) = 1/((n-1)s_{\boldsymbol{x}_0} s_{\boldsymbol{x}_0 \circ \boldsymbol{m}_{\cdot,c_1}}) \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^2 - \overline{\boldsymbol{x}_0}\pi_\ell)\theta_{r,\ell}).$$

Derivation: By definition and using $\boldsymbol{x}_0$ to estimate the standard deviations,

$\rho(\boldsymbol{x}, \boldsymbol{x} \circ \boldsymbol{m}_{\cdot,c_1})(n-1)s_{\boldsymbol{x}_0} s_{\boldsymbol{x}_0 \circ \boldsymbol{m}_{\cdot,c_1}} \approx \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}})(x_r m_{r,c_1} - \overline{\boldsymbol{x} \circ \boldsymbol{m}_{\cdot,c_1}})$

$= \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}}) x_r m_{r,c_1}$ by proposition 1 for constant $-\overline{\boldsymbol{x} \circ \boldsymbol{m}_{\cdot,c_1}}$ with respect to r

$= \sum_{r=1}^{n} m_{r,c_1} (x_r^2 - \overline{\boldsymbol{x}} x_r)$

$= \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell^2 \theta_{r,\ell} - \overline{\boldsymbol{x}} \sum_{\ell=1}^{\lambda(\boldsymbol{x})} \pi_\ell \theta_{r,\ell})$ by proposition 2

$\approx \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^2 - \overline{\boldsymbol{x}_0}\pi_\ell)\, \theta_{r,\ell})$ by estimation of $\overline{\boldsymbol{x}}$ with $\overline{\boldsymbol{x}_0}$.

Extension 4. Correlation estimates between $\boldsymbol{z}$ and quadratic $\boldsymbol{z} \circ \boldsymbol{z}$:

$$\tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{z}) = 1/((n-1)s_{\boldsymbol{z}_0} s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0}) \sum_{r=1}^{n} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^3 - 3\pi_\ell^2 \overline{\boldsymbol{x}_0} + 3\overline{\boldsymbol{x}_0}^2 \pi_\ell)\theta_{r,\ell} - \overline{\boldsymbol{x}_0}^3).$$

Derivation: By definition and using $\boldsymbol{z}_0$ to estimate the standard deviations,

$\rho(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{z})(n-1)s_{\boldsymbol{z}_0} s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0} \approx \sum_{r=1}^{n} (z_r - \bar{\boldsymbol{z}})(z_r^2 - \overline{\boldsymbol{z} \circ \boldsymbol{z}})$

$= \sum_{r=1}^{n}(z_r - \bar{\boldsymbol{z}})z_r^2$ by proposition 1 for constant $-\overline{\boldsymbol{z} \circ \boldsymbol{z}}$ with respect to r

$= \sum_{r=1}^{n}(x_r - \bar{\boldsymbol{x}})^3$ for centered $\boldsymbol{z}$ where $\bar{\boldsymbol{z}} = 0$

$= \sum_{r=1}^{n}(x_r^3 - 3\bar{\boldsymbol{x}}x_r^2 + 3\bar{\boldsymbol{x}}^2x_r - \bar{\boldsymbol{x}}^3)$

$= \sum_{r=1}^{n}(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^3 - 3\bar{\boldsymbol{x}}\pi_\ell^2 + 3\bar{\boldsymbol{x}}^2\pi_\ell)\theta_{r,\ell} - \bar{\boldsymbol{x}}^3)$ by proposition 2

$\approx \sum_{r=1}^{n}(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^3 - 3\overline{\boldsymbol{x}_0}\pi_\ell^2 + 3\overline{\boldsymbol{x}_0}^2\pi_\ell)\theta_{r,\ell} - \overline{\boldsymbol{x}_0}^3)$ by estimation of $\bar{\boldsymbol{x}}$ with $\overline{\boldsymbol{x}_0}$.

Extension 5. Correlation estimates between quadratic $\boldsymbol{z} \circ \boldsymbol{z}$ and interactions $\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}$, for all $c_1 \in C_1$:

$$\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}) = 1/((n-1)s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0}s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}})$$
$$\cdot \sum_{r=1}^{n} m_{r,c}(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^3 - 3\overline{\boldsymbol{x}_0}\pi_\ell^2 + \left(4\overline{\boldsymbol{x}_0}^2 - \overline{\boldsymbol{x}_0 \circ \boldsymbol{x}_0}\right)\pi_\ell)\theta_{r,\ell} - 2\overline{\boldsymbol{x}_0}^3 + \overline{\boldsymbol{x}_0}\,\overline{\boldsymbol{x}_0 \circ \boldsymbol{x}_0})$$

Derivation: By definition and using $\boldsymbol{z}_0$ to estimate the standard deviations,

$\rho(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})\,(n-1)s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0}s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}} \approx \sum_{r=1}^{n}(z_r^2 - \overline{\boldsymbol{z} \circ \boldsymbol{z}})(z_r m_{r,c_1} - \overline{\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}})$

$= \sum_{r=1}^{n}(z_r^2 - \overline{\boldsymbol{z} \circ \boldsymbol{z}})z_r m_{r,c_1}$ by proposition 1 for constant $-\overline{\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}}$ with respect to r

$= \sum_{r=1}^{n} m_{r,c_1}((x_r - \bar{\boldsymbol{x}})^3 - (x_r - \bar{\boldsymbol{x}})(1/n)\sum_{k=1}^{n}(x_k - \bar{\boldsymbol{x}})^2)$ for centered $\boldsymbol{z}$

$= \sum_{r=1}^{n} m_{r,c_1}((x_r^3 - 3\bar{\boldsymbol{x}}x_r^2 + 3\bar{\boldsymbol{x}}^2x_r - \bar{\boldsymbol{x}}^3) - (x_r - \bar{\boldsymbol{x}})(1/n)\sum_{k=1}^{n}({x_k}^2 - 2\bar{\boldsymbol{x}}x_k + \bar{\boldsymbol{x}}^2))$

$= \sum_{r=1}^{n} m_{r,c_1}((x_r^3 - 3\bar{\boldsymbol{x}}x_r^2 + 3\bar{\boldsymbol{x}}^2x_r - \bar{\boldsymbol{x}}^3) - (x_r - \bar{\boldsymbol{x}})(\overline{\boldsymbol{x} \circ \boldsymbol{x}} - \bar{\boldsymbol{x}}^2))$

$= \sum_{r=1}^{n} m_{r,c_1}(x_r^3 - 3\bar{\boldsymbol{x}}x_r^2 + (4\bar{\boldsymbol{x}}^2 - \overline{\boldsymbol{x} \circ \boldsymbol{x}})x_r - 2\bar{\boldsymbol{x}}^3 + \bar{\boldsymbol{x}}\,\overline{\boldsymbol{x} \circ \boldsymbol{x}})$

$= \sum_{r=1}^{n} m_{r,c_1}(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^3 - 3\bar{\boldsymbol{x}}\pi_\ell^2 + (4\bar{\boldsymbol{x}}^2 - \overline{\boldsymbol{x} \circ \boldsymbol{x}})\pi_\ell)\theta_{r,\ell} - 2\bar{\boldsymbol{x}}^3 + \bar{\boldsymbol{x}}\,\overline{\boldsymbol{x} \circ \boldsymbol{x}})$ by proposition 2

$\approx \sum_{r=1}^{n} m_{r,c_1}(\sum_{\ell=1}^{\lambda(x)}(\pi_\ell^3 - 3\overline{\boldsymbol{x}_0}\pi_\ell^2 + (4\overline{\boldsymbol{x}_0}^2 - \overline{\boldsymbol{x}_0 \circ \boldsymbol{x}_0})\pi_\ell)\theta_{r,\ell} - 2\overline{\boldsymbol{x}_0}^3 + \overline{\boldsymbol{x}_0}\,\overline{\boldsymbol{x}_0 \circ \boldsymbol{x}_0})$ by estimation of $\bar{\boldsymbol{x}}$ and $\overline{\boldsymbol{x} \circ \boldsymbol{x}}$ with $\overline{\boldsymbol{x}_0}$ and $\overline{\boldsymbol{x}_0 \circ \boldsymbol{x}_0}$, respectively.

While approach NOAB-V requires all five MILP extensions where matrix $\boldsymbol{M}$ represents all first- and second-order terms for previously constructed factors, approach NOAB-IV uses extensions 1 through 4 where $\boldsymbol{M}$ represents only first-order terms for the extensions and includes second-order terms for the original constraints, and NOAB-Q uses extensions 2 and 4 where $\boldsymbol{M}$ represents first-order terms and associated quadratics.

3.5 Case study

3.5.1 Design Space and Parameter Settings

The mixed-factor decision space of interest is comprised of two discrete factors (four- and three-level) and seven categorical factors (two three-level and five two-level), where a full factorial design requires 3,456 points. The two discrete factors have levels of {1, 2, 3, 4} and {1, 2, 3}. With $J = 11$ first-order columns required for the encoded design matrix, the suggested lower and upper bounds for first-order NOAB design size n are $3J = 33$ and $10J = 110$, respectively. For the second-order design approaches, it is expected that larger design sizes are needed to achieve near orthogonality ($\rho_{map} \leq 0.05$), so the upper bound is increased to 504. Typically, fewer design sizes within the specified bounds are found to be balance-feasible when the maximum allowed imbalance (δ^*) is decreased for the balance feasibility test. In this case study, an imbalance restriction in this test allows for faster traversal of design size ranges while still obtaining a sufficient number of designs to examine. While each NOAB design construction uses $\delta^* = 0.05$ for the MILPs, the balance feasibility test for n uses $\delta^* = 0.05$ for approaches NOAB-III and NOAB-Q when $33 \leq n \leq 110$, and the restricted $\delta^* = 0$, otherwise. The MILP solver is allowed two

attempts to achieve near orthogonality for each factor construction with a time limit of 60 seconds for NOAB-III, NOAB-Q, and NOAB-IV and 300 seconds for NOAB-V. When a solution $\boldsymbol{x}$ gives $\rho_{map} > 0.05$ after the first attempt, $\boldsymbol{x}$ is used as the initial solution $\boldsymbol{x}_0$ for the second attempt. The construction method is implemented in MATLAB R2015a using CPLEX V12.6.1 to obtain MILP solutions, with 222 NOAB designs resulting from the four approaches and various balance-feasible design sizes examined.

3.5.2 First-order Model Results

Let $\boldsymbol{X}$ be the design matrix for the full first-order model. Figure 20 shows the performance of the constructed NOAB designs for the four design approaches and various balance-feasible design sizes. The smallest design sizes are n = 36 for approaches NOAB-III and NOAB-Q, n = 96 for NOAB-IV, and n = 264 for NOAB-V, i.e., smaller designs using each approach do not satisfy near orthogonality for the intended model terms. Approaches NOAB-IV and NOAB-V result in clear improvements for the model misspecification measure $tr(\boldsymbol{A}'\boldsymbol{A})$. The design approaches do not appear to change D-criterion or average UPV when assuming a first-order model.

3.5.3 Second-order Model Results

Let $\boldsymbol{X}$ be the design matrix for a full second-order model with centered first-order columns for quantitative factors, so the performance measures now exclude $tr(\boldsymbol{A}'\boldsymbol{A})$. Due to numerical instability of average UPV calculations for approaches NOAB-III and NOAB-Q, only designs having average UPV $\leq$ 100 (and UPV $\leq$ 2) are displayed in Figure 20. For fixed n, the second-order extensions tend to improve both the D-criterion and average UPV, with greater improvement seen when requiring near orthogonality for more pairs of

model terms. In this study, average UPV is generally indicative of the relative quality of UPV over fractions of the design space (FDS) for the different design approaches. However, approach NOAB-V does see smaller increases in UPV over large FDS when compared to NOAB-IV (Figure 21).

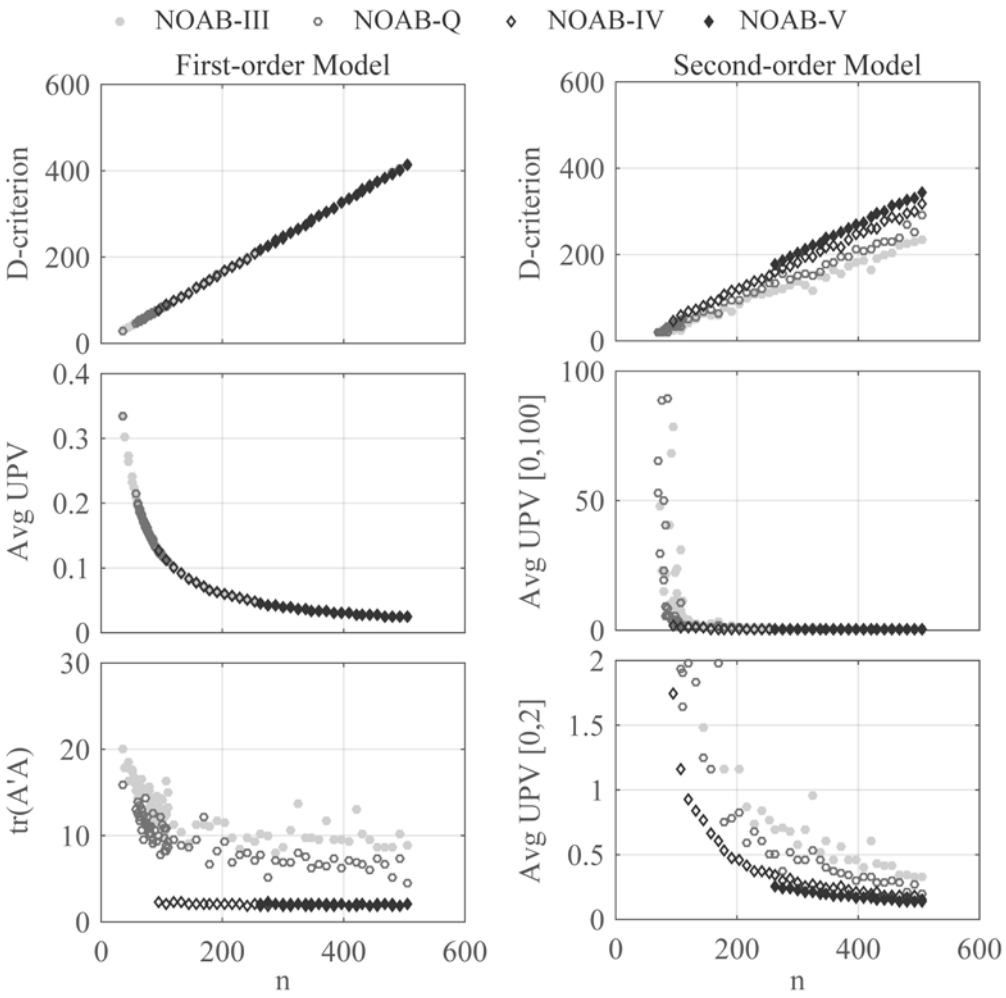


Figure 20. NOAB Design Performance by Approach and Design Size

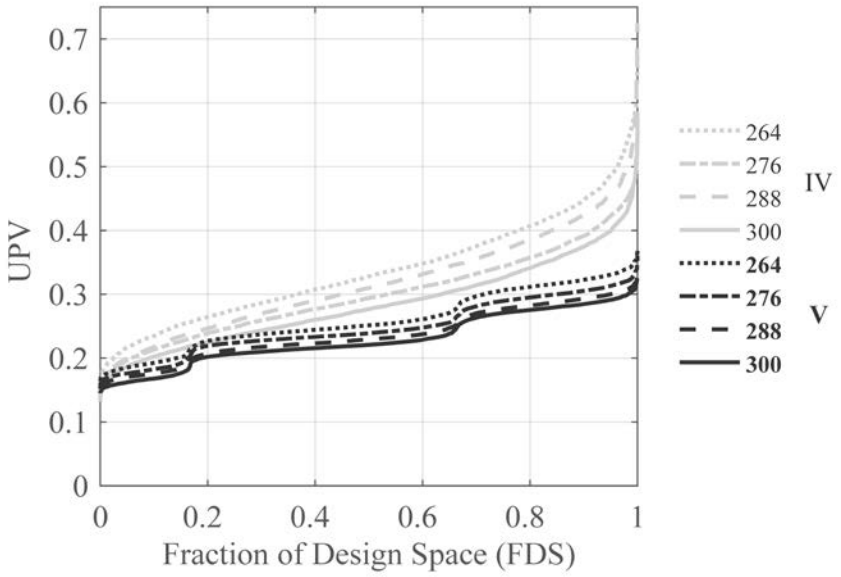


Figure 21. UPV by FDS for NOAB-IV and NOAB-V Designs (Second-order Model)

3.5.4 Comparison of Absolute Correlations

The second-order extensions are shown to improve different performance measures depending on the assumed model. However, it is important to also examine the absolute pairwise correlations for resulting second-order design matrices. Figure 22 gives the absolute correlation matrices for four 264-point designs, each using one of the four approaches, with model terms ordered and partitioned by first-order, interactions, and quadratics. As the different second-order extensions are used to achieve near orthogonality, it is clear that the absolute correlations are decreasing for the appropriate partitions. Though it appears that NOAB-V dominates the other approaches for the 264-point designs with respect to the absolute correlations in Figure 22, using an approach that minimizes correlations for too many model terms with too few design points can result in unsatisfactory correlation values. See the Appendix for an example of absolute correlations

for a 36-point design using the NOAB-V approach. There is a small number of off-diagonal elements with consistently high absolute correlations, which are most apparent for the matrix associated with the NOAB-V design. These correlations are associated with multiple columns in the design matrix that represent the same categorical factor, meaning they should be highly correlated, and thus, are ignored in the second-order method.

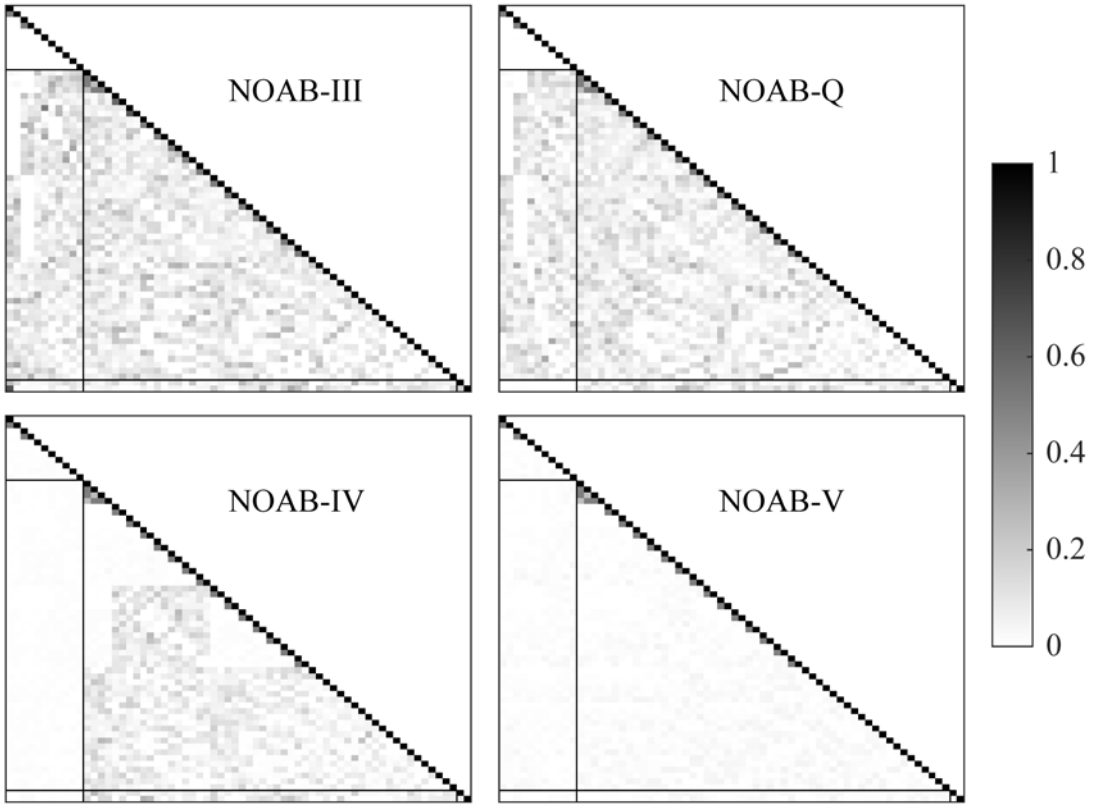


Figure 22. Absolute Correlation Matrices for NOAB Designs

3.5.5 Further Design Evaluation and Comparison

In this case study, the assumption is that an analyst would have an initial preference for one of the design approaches based on the context of their problem, i.e., near orthogonality is desired for some specified set of model terms. When examining different

sizes and parameter settings in the construction method for a specific design approach, further design evaluation and comparison is warranted. Various performance measures of interest to an analyst can be given the same scale by using desirability functions [82], with an overall desirability using additive or multiplicative weights. Synthesized efficiency can then be examined, as in [68], which is the overall desirability of a design relative to the most desirable design for a specific weighting combination. Graphical approaches such as trade-off plots and mixture plots can also be used for design comparison [68], [77].

3.6 Conclusions and Further Research

MILP extensions to the NOAB design construction method allow for near orthogonality between first- and second-order terms, improving performance measures associated with good parameter estimation and prediction variance for an assumed second-order model. When assuming a first-order model, the extensions allow for construction of designs that protect against model misspecification with respect to second-order terms. Even if small design size is of great importance, using the second-order extensions for a subset of second-order terms may still improve other performance measures of interest. Many studies may see value in a process that uses a first-order (*NOAB resolution III*) design or *NOAB resolution IV* design for initial screening of a large number of factors, followed by a second-order (*NOAB resolution V*) design for significant factors and associated second-order effects.

Future research includes the development of a meta-learning framework [6], [108] for NOAB design construction parameter selection, based on meta-features extracted from the design space. Additionally, a Microsoft Excel tool for first- and second-order NOAB

design construction has been created, utilizing the open-source add-in OpenSolver [134], [135] to ensure availability of the original method as well as the second-order extensions. This tool will be provided online by the Air Force Institute of Technology.

IV. Batch Sequential NOAB Designs by Way of Simultaneous Construction and Augmentation

4.1 Abstract

Space-filling designs help experimenters to represent simulation outputs efficiently when entire input spaces cannot be exhaustively explored. Batch sequential designs allow for intermediate analyses to occur as later batches of experimental design points are being tested, and give the ability to change later design points based on the outputs observed as well as stop the experiment when the current observations are deemed sufficient in order to reduce experimental cost. Nearly orthogonal-and-balanced (NOAB) designs have been shown to have good space-filling properties and can accommodate design spaces with continuous, discrete, and categorical factors. In this paper, mixed-integer linear programming (MILP) formulations used to solve for NOAB resolution III, IV, and V designs are extended to construct batch sequential NOAB designs, where design stages can use different NOAB approaches. A case study is presented where a simultaneous construction approach results in overall more desirable designs than when using design augmentation, yet requires a predefined number of points for each design stage.

Keywords: design of experiments; mixed factor; space filling; nearly orthogonal-and-balanced; mixed-integer linear program; meta-model

4.2 Introduction

Many studies of systems and simulations with complex behavior aim to understand relationships between a large number of inputs and outputs, where exhaustively testing all

input combinations of interest can quickly become infeasible due to a lack of time and resources. Space-filling designs can help experimenters represent simulation outputs for an entire input space efficiently by fitting meta-models, or surrogate models, to a relatively small set of design points and using each meta-model to predict the behavior of each system output. Furthermore, system inputs may be represented by mixed factors, i.e., a mixture of continuous, discrete, and categorical factors with potentially different numbers of factor levels for each. A popular approach for mixed-factor designs with good space-filling properties is the nearly orthogonal-and-balanced (NOAB) design. The original, first-order NOAB design approach presented in [1] uses a mixed-integer linear programming (MILP) formulation to ensure near balance of factor levels, i.e., for each factor, the individual levels are represented nearly equally in the design, while solving for near orthogonality with respect to terms in the first-order model:

$$y = \beta_0 + \sum \beta_i x_i + \varepsilon$$

with response y, input factors x_i, coefficients β_i, and error ε. Near orthogonality is defined as when the maximum absolute correlation among pairs of design columns, ρ_{map}, is less than 0.05. Low correlation between design matrix columns representing first-order terms allows for separate examination of individual factors. Second-order extensions to the original MILP are developed in Chapter III, which can minimize ρ_{map} for design matrix columns representing a full, second-order model that includes two-way interactions and quadratic terms:

$$y = \beta_0 + \sum \beta_i x_i + \sum_i \sum_{j>i} \beta_{ij} x_i x_j + \sum \beta_{ii} x_i^2 + \varepsilon$$

The first-order NOAB and second-order extensions result in three main approaches that aim to have near orthogonality for different sets of model terms:

- *NOAB resolution III* (NOAB-III) – minimizes correlation between all first-order terms, from [1]
- *NOAB resolution IV* (NOAB-IV) – ignores correlation between pairs of second-order terms
- *NOAB resolution V* (NOAB-V) – minimizes correlation between all first- and second-order terms

Currently, the NOAB designs can be thought of as *one-shot*, or single-stage, approaches. To give greater flexibility to experimenters, batch sequential designs have multiple stages to allow for intermediate analyses that occur alongside later batches of experimental runs as well as to permit early termination of runs when sufficient information has been collected. Intermediate analyses can inform the choice of later design points for design augmentation, help to determine insignificant factors that can be eliminated from further evaluation, and highlight subsets of important factors and specific regions of the design space that may be of greater interest. Such advantages, and disadvantages, of sequential designs are presented in [42].

Designs with good space-filling properties are thought to be preferable for meta-modeling, as discussed in [136]. Space-filling designs are reviewed in [137], where model-free methods of geometric criteria, Latin hypercube designs, and other approaches are discussed in addition to model-based design methods for Kriging (or Gaussian-process modeling) and combinations of space-filling and estimation designs. Some distance performance metrics for space-filling designs are reviewed in [138], where a distance correlation-based metric is proposed for Latin hypercube designs. An empirical study of

prediction performance of space-filling designs in detailed in [139], where the authors state that the best approach for improving prediction accuracy is to add design points and suggest that efficient augmentation of space-filling designs is an important area of research. Sequential sampling is listed as an open research topic for Latin hypercube designs in [140], which references the nested Latin hypercube designs from [141] as well as designs that are augmented based on information from surrogate models such as Kriging from [137], [142], [143]. Quasi-Latin hypercube design sampling is detailed in [143], which provides an overview of sequential sampling and notes that objective-oriented sequential sampling is suited for design optimization, while space-filling sequential sampling concerns the global accuracy of a meta-model. Sequential space-filling designs are reviewed in [142], where sequential nested Latin hypercube, global Monte Carlo, and optimization-based methods are presented. Space-filling designs for constrained domains are developed in [144] using a sequential Monte Carlo based algorithm and distance-based design criteria. An overview of criteria for sequential sampling is given in [145], which presents extended orthogonal array-based Latin hypercube sampling while introducing two distance-based metrics for batch sequential sampling. Sliced full factorial-based Latin hypercube designs (sFFLHD) are developed in [146], which are batch sequential and do not require a predefined number of total design points.

Commonly-used design performance measures and single-stage NOAB designs are presented (Section 4.3) as background material for the construction methods of the batch sequential NOAB designs (Section 4.4), where each design stage can be constructed using either of the NOAB-III, NOAB-IV, or NOAB-V approaches. Two techniques are

developed for construction of the multiple design stages: design augmentation and simultaneous construction of each stage. Section 4.5 provides a study of design properties for the two batch sequential techniques where the design stages use the same NOAB approach as well as where the NOAB approaches are different.

4.3 Background Material

4.3.1 NOAB Design Notation and Background

A balance feasibility test is developed in [1] to determine if a design size n can feasibly satisfy a specified maximum allowed imbalance δ, given possibly different numbers of factor levels. Using a balance-feasible n, the NOAB design construction methods create design matrix columns for a single factor at a time, iterating until all factors are represented. The first column is randomly generated to have imbalance no greater than δ. The column structure of the remaining factors is then determined iteratively, one factor at a time, using one of three mixed-integer linear programming (MILP) problems based on factor type (i.e., continuous, discrete, and categorical). In this paper, the continuous factor and discrete factor cases are considered to be the same formulation, since the continuous factor case in previous methods assumes n equally-spaced factor levels. Any required fidelity for representing a continuous factor can be met by assuming a large enough number of equally-spaced factor levels in the discrete factor MILP formulation. Removing the need for discretizing all continuous factors by exactly n levels also allows for greater flexibility when developing batch sequential techniques. Table 10 provides the notation used for NOAB design construction.

Table 10. Notation for Batch Sequential NOAB Design Construction

n	number of design points (matrix rows), i.e., design size, indexed by row $r = 1, 2, \ldots, n$
j	number of previously constructed matrix columns, indexed by column $c = 1,\ 2,\ \ldots,\ j$ (comprising the set C)
$\boldsymbol{M}$	previously constructed $n \times j$ design matrix (represents only first-order terms in the original method and both first- and second-order terms for the extended method)
$m_{r,c}$	element of $\boldsymbol{M}$ in row r and column c
$\boldsymbol{m}_{\cdot,c}$	column c of $\boldsymbol{M}$
C_1	subset of column indices $1,\ 2,\ \ldots,\ j$ for $\boldsymbol{M}$ that represent first-order terms only, indexed by c_1
$\boldsymbol{x}_{\cdot,i}$	MILP decision variables ($n \times 1$ column), indexed by the number of factor columns $i = 1, \ldots, I$ (I = 1 for discrete, and $I = \ell_x$-1 for categorical with ℓ_x levels)
$x_{r,i}$	element of column $\boldsymbol{x}_{\cdot,i}$ in row r
$\boldsymbol{z}_{\cdot,i}$	centered MILP decision variable ($n \times 1$ column), with $z_{r,i} = x_{r,i} - \overline{\boldsymbol{x}_{\cdot,i}} = x_{r,i} - (1/n)\ \sum_{k=1}^{n} x_{k,i}$
$\boldsymbol{x}_0$	initial randomly-generated MILP solution ($n \times I$)
$\rho(\boldsymbol{x}, \boldsymbol{y})$	pairwise correlation for columns $\boldsymbol{x}$ and $\boldsymbol{y}$ is $\rho(\boldsymbol{x}, \boldsymbol{y}) = 1/((n - 1)\ s_{\boldsymbol{x}} s_{\boldsymbol{y}}) \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}})(y_r - \overline{\boldsymbol{y}})$, with column elements x_r and y_r, means $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$, and standard deviations $s_{\boldsymbol{x}}$ and $s_{\boldsymbol{y}}$
ρ_{map}	maximum allowed absolute pairwise correlation, $\rho_{map} = \max_{\boldsymbol{x} \neq \boldsymbol{y}} \lvert\rho(\boldsymbol{x}, \boldsymbol{y})\rvert$
δ	maximum allowed imbalance, $\delta = \max_{x} \delta_x$, imbalance for factor x is defined as $\delta_x = \max_{i=1,\ldots,\ell_x} \lvert(w_{i,x} - (n/\ell_x))/(n/\ell_x)\rvert$, $w_{i,x}$ is the number of times level i occurs for factor x with ℓ_x possible levels
$\lambda(\boldsymbol{x})$	number of encoded levels for column $\boldsymbol{x}$, indexed by encoded level ℓ = 1, 2, …, $\lambda(\boldsymbol{x})$
π_ℓ	encoded level value (with $\pi_1 < \pi_2 < \cdots < \pi_{\lambda(\boldsymbol{x})}$ being all possible values for column $\boldsymbol{x}$)

$\theta_{r,\ell}^{i}$	binary decision variable where $x_{r,i} = \sum_{\ell=1}^{\lambda(x)} \pi_\ell \theta_{r,\ell}^{i}$ and $\sum_{\ell=1}^{\lambda(x)} \theta_{r,\ell}^{i} = 1$ for row r, encoded level ℓ, and factor column i

4.3.2 Design Performance Measures

As n increases, designs are expected to improve model coefficient estimation and prediction accuracy. The D-criterion $|\boldsymbol{X}'\boldsymbol{X}|^{1/p}$ for p model parameters [76] is used as a measure for good model coefficient estimation, with larger values more desirable. The average and maximum unscaled prediction variance (UPV $= \boldsymbol{x}^{(q)\prime}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{x}^{(q)}$ for design matrix $\boldsymbol{X}$) over all possible design points q are examined as well [77]. When assuming a second-order model, the D-criterion and UPV measures have been shown to improve for designs of the same size as more second-order terms are considered when minimizing ρ_{map}, i.e., the NOAB-V outperforms the NOAB-IV and NOAB-IV outperforms the NOAB-III (Chapter III).

If $\boldsymbol{X}_1$ is an assumed linear model matrix of first-order terms and $\boldsymbol{X}_2$ includes additional linear terms excluded from the defined model, then the alias matrix $\boldsymbol{A} = (\boldsymbol{X}_1'\boldsymbol{X}_1)^{-1}(\boldsymbol{X}_1'\boldsymbol{X}_2)$ gives the degree of biasing of each first-order terms represented in the linear model matrix $\boldsymbol{X}_1$ due to each second-order term in $\boldsymbol{X}_2$ [133]. NOAB-V and NOAB-IV designs have been shown to improve the common model misspecification measure $tr(\boldsymbol{A}'\boldsymbol{A})$ with respect to second-order terms (assuming a first order model) when compared to NOAB-III designs. This paper will primarily examine the impact of the different batch sequential techniques, in combination with the NOAB design approaches, on the maximum

absolute correlation ρ_{map}, while highlighting any notable differences in the other design performance measures.

4.4 Construction Methods for Batch Sequential NOAB Designs

Two main approaches are developed for creating the batch sequential NOAB designs: simultaneous construction (Section 4.4.2) and design augmentation (Section 4.4.3). However, for design spaces where there are many factors with low numbers of levels for each, repeated design points can commonly occur in NOAB designs. In a batch sequential NOAB design, the assumption is that the system of interest is deterministic, where the design points would need to be repeated to understand the randomness in a stochastic system. Section 4.4.1 describes new constraints in the MILP formulations that limit repeated design points for NOAB designs.

4.4.1 Limiting Repeated Points in NOAB Designs

For a *discrete factor* column $\boldsymbol{x}$ where the element in rows $r = 1, 2, \dots, n$ is denoted x_r, let $\{\pi_1, \pi_2, \dots, \pi_{\ell_x}\}$ be the ℓ_x possible levels. The following constraint limits repeated design points:

$$x_{r'} - x_r \geq \min_{i_1 \neq i_2} |\pi_{i_1} - \pi_{i_2}|, \text{ for ordered pairs of rows } (r, r'),\ r < r'$$

where ordered pairs of rows (r, r') are determined by examining which rows from the previously constructed columns are currently repeated and by ensuring that the corresponding values in the new columns are different. Such row pairs, (r, r'), $r < r'$ comprise the set RP, which are determined in the initial generation of each factor column (i.e., initial solution) for use in the MILP formulations (Figure 23).

IF discrete factor THEN $drawArray = [\pi_1, \pi_2, \ldots, \pi_{\ell_x}]$
IF categorical factor THEN $drawArray = [\ \ell_x, 1, 2, \ldots, \ell_x - 1]$ (levels before encoding)
IF any repeated points/rows exist in the previously constructed design matrix
    FOR each set of repeated rows that match
        WHILE |set of repeated rows unassigned| > 1
            Assign distinct $drawArray$ values to initial solution for up to ℓ_x rows at a time and record ordered row pairs (r', r) in RP, where $r < r'$ and x_r occurs before $x_{r'}$ in the *drawArray* for these assignments
        end WHILE
    end FOR
end IF statement
Randomly assign $drawArray$ values to remaining non-repeated rows, while satisfying balance constraints

Figure 23. Initial Solution Generation for MILP

For the set of columns $\boldsymbol{x}_{\cdot,1}, \boldsymbol{x}_{\cdot,2}, \ldots, \boldsymbol{x}_{\cdot,(\ell_x-1)}$ for a single *categorical factor* using {-1, 0, 1} effect coding, let

$$x_{r,i} = \begin{cases} 1, & if\, x_r = i < \ell_x \\ -1, & if\, x_r = \ell_x \\ 0, & otherwise. \end{cases}$$

The following constraint helps to limit repeated design points in the categorical factor case:

$\sum_{i=1}^{\ell_x-1} i\, x_{r',i} - \sum_{i=1}^{\ell_x-1} i\, x_{r,i} \geq 1$, for ordered pairs of rows (r, r'), $r < r'$

Note the difference in the ordering of levels in the *drawArray* ($[\ell, 1, 2, \ldots, \ell - 1]$) from the discrete factor case. An example of how this constraint ensures the correct ordering of categorical levels is provided in Table 11 for a categorical factor with four levels. The ordering of the x_r values matches that of the *drawArray*, with the summation used within the RP constraint maintaining the same ordering.

Table 11. Example Categorical Factor Level Order for RP Constraints

r	x_r	$x_{r,1}$	$x_{r,2}$	$x_{r,3}$	$\sum_{i=1}^{\ell_x-1} i x_{r,i}$
1	4	-1	-1	-1	-6
2	1	1	0	0	1
3	2	0	1	0	2
4	3	0	0	1	3

The use of these repeated point (RP) constraints encourages diversity of design points for each column construction, where it is desired to have the set RP decrease in size for each new column construction and eventually have $RP = \emptyset$ (i.e., there are no repeated design rows), though this is not guaranteed. The use of the RP constraints is important when assuming a deterministic simulation, since any repeated points in the design can needlessly use valuable experimental resources. As an example, designs using the one-shot NOAB-V approach are constructed with and without the RP constraints for different design sizes, where the design space contains two discrete factors (four-level and three-level) and seven categorical factors (two three-level and five two-level), with 3,456 possible design points in total. The number of repeated points increases as n increases when no RP constraints are used, while the use of RP constraints prevents repeated points for each of the design sizes (Figure 24). Similar patterns in the number of repeated points occur for the other NOAB approaches.

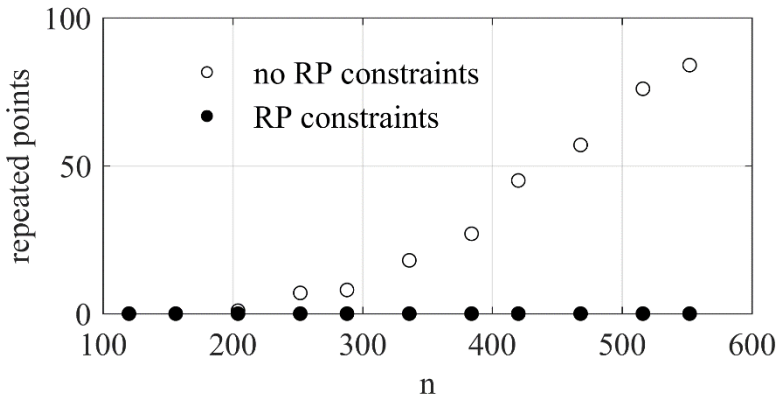


Figure 24. Limiting Repeated Points in One-shot NOAB-V Designs

4.4.2 Simultaneous Construction

In this section, MILP formulations are presented for simultaneous construction of batch sequential NOAB designs, with one formulation for the discrete factor case and one for the categorical factor case. Just as in the previous methods for NOAB design construction, the columns of the design matrix are solved iteratively, where an approximation of ρ_{map} is minimized. To extend the batch sequential NOAB design, each of the multiple stages have constraints requiring near balanced, while the aim is to minimize correlations using one of the three main NOAB design approaches for each stage. Let $N = N_{III} \cup N_{IV} \cup N_V$ be the set of all design stage sizes of interest, where N_{III}, N_{IV}, and N_V are the sets of design sizes using either the NOAB-III, NOAB-IV, or NOAB-V approaches, respectively. In Figure 25, the column structure for a simultaneous construction is presented for a design with stage sizes n_1 and n_2, previously constructed matrix $\boldsymbol{M}$, and new factor column $\boldsymbol{x}$.

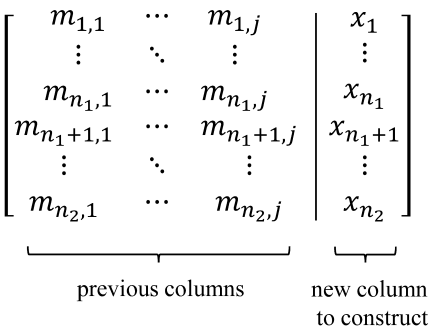


Figure 25. Simultaneous Construction for Two Stages

The MILP formulation for simultaneous construction of the multiple stages for a discrete factor is presented in Figure 26. Let $n_{max} = \max_{n \in N} n$ be the total number of design points for the batch sequential design. Constraints (*i*) and (*ii*) ensure that decision variable v is the maximum of absolute correlation estimates between column $\boldsymbol{x}$ and all previously constructed first-order columns $\boldsymbol{m}_{\cdot,c}, c \in C_1$ for each design size $n \in N$. The function $\tilde{\rho}_n$ is an estimate of the pairwise correlation ρ for only rows $1, 2, \ldots, n$. Constraints (*i*-1) and (*ii*-1) through (*i*-5) and (*ii*-5) similarly help to represent v_1 through v_5, respectively, the maximum absolute pairwise correlation estimates for the various cases involving second-order model terms (i.e., $n \in N_{IV}$ or $n \in N_V$). The function $\tilde{\rho}_n$ for the various correlation cases is defined in the Appendix. The objective is to minimize the sum of these cases, which can provide greater control over these values than when minimizing the objective function $v = v_1 = \cdots = v_5$. Constraints (*iii*) and (*iv*) require that exactly one level is assigned to each row in column $\boldsymbol{x}$ for the entire design. Constraints (*v*) and (*vi*) ensure that the maximum allowed imbalance δ is satisfied for each design stage size $n \in N$. The RP constraint (*vii*) limits the number of repeated points in the design size. Constraint (*viii*)

requires the decision variable $\theta_{r,\ell}$ to be binary (i.e., either a level ℓ is assigned to row r or not). Note that having only a single n in one of the sets N_{III}, N_{IV}, and N_V results in a single-stage NOAB design for that respective design approach.

Quadratic model terms are not examined in the categorical factor formulation (Figure 27), so the correlation estimates for v_2, v_4, and v_5 are not included. Otherwise, constraints (*i*) through (*vi*) for the categorical case have the same purpose as in the discrete case, except now there are possibly multiple factor columns $\boldsymbol{x}_{\cdot,i}$, $i = 1, 2, \ldots, \ell_x - 1$ due to factor encoding. Constraints (*vii*) through (*ix*) make sure that the factor columns use effect coding, as in the original method from [1]. The RP constraint (*x*) limits repeated points as discussed previously, and constraint (*xi*) ensures a binary $\theta_{r,\ell}$ as in the discrete factor case.

4.4.3 Design Augmentation

In contrast to the simultaneous construction technique, design augmentation is used to construct the full batch sequential NOAB design by creating a one-shot NOAB design for the smallest design stage and repeatedly augmenting the design using the MILP formulations and desired NOAB approaches to achieve later design stages. In other words, for a set of design stage sizes $n_1 < n_2 < \ldots < n_k = n_{max}$ for $k \geq 2$ stages, once a NOAB design of size n_i is constructed, a design of size n_{i+1} is then created by fixing the first n_i rows of the design matrix and letting $N = \{n_{i+1}\}$ be the only design size considered in the MILP formulations. The MILP decision variables for the factor column then concern only rows $r = n_i + 1, n_i + 2, \ldots, n_{i+1}$. The column structure of the design matrix for the

augmentation technique is given in Figure 28, where points for an n_2-point design are created by augmenting an n_1-point design.

Minimize	$v + v_1 + \ v_2 + v_3 + v_4 + v_5$	
Subject to		
(*i*)	$v \geq \tilde{\rho}_n(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c})$	$(c,n) \in (C_1 \times N_{III}) \cup (C \times (N_{IV} \cup N_V))$
(*ii*)	$v \geq -\tilde{\rho}_n(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c})$	$(c,n) \in (C_1 \times N_{III}) \cup (C \times (N_{IV} \cup N_V))$
(*i*-1)	$v_1 \geq \tilde{\rho}_n(\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c})$	$c_1 \in C_1;\ (c,n) \in (C_1 \times N_{IV}) \cup (C \times N_V)$
(*ii*-1)	$v_1 \geq -\tilde{\rho}_n(\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c})$	$c_1 \in C_1;\ (c,n) \in (C_1 \times N_{IV}) \cup (C \times N_V)$
(*i*-2)	$v_2 \geq \tilde{\rho}_n(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{m}_{\cdot,c})$	$(c,n) \in (C_1 \times N_{IV}) \cup (C \times N_V)$
(*ii*-2)	$v_2 \geq -\tilde{\rho}_n(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{m}_{\cdot,c})$	$(c,n) \in (C_1 \times N_{IV}) \cup (C \times N_V)$
(*i*-3)	$v_3 \geq \tilde{\rho}_n(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1;\ n \in N_{IV} \cup N_V$
(*ii*-3)	$v_3 \geq -\tilde{\rho}_n(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1;\ n \in N_{IV} \cup N_V$
(*i*-4)	$v_4 \geq \tilde{\rho}_n(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{z})$	$n \in N_{IV} \cup N_V$
(*ii*-4)	$v_4 \geq -\tilde{\rho}_n(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{z})$	$n \in N_{IV} \cup N_V$
(*i*-5)	$v_5 \geq \tilde{\rho}_n(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1;\ n \in N_V$
(*ii*-5)	$v_5 \geq -\tilde{\rho}_n(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1;\ n \in N_V$
(*iii*)	$\sum_{\ell=1}^{\ell_x} \theta_{r,\ell} = 1$	$r = 1, 2, \ldots, n_{max}$
(*iv*)	$x_r = \sum_{\ell=1}^{\ell_x} \pi_\ell \theta_{r,\ell}$	$r = 1, 2, \ldots, n_{max}$
(*v*)	$\sum_{r=1}^{n} \theta_{r,\ell} \leq \left\lfloor (1+\delta)\frac{n}{\ell_x} \right\rfloor$	$\ell = 1, 2, \ldots, \ell_x;\ n \in N$
(*vi*)	$\sum_{r=1}^{n} \theta_{r,\ell} \geq \left\lceil (1-\delta)\frac{n}{\ell_x} \right\rceil$	$\ell = 1, 2, \ldots, \ell_x;\ n \in N$
(*vii*)	$x_{r'} - x_r \geq \min_{i_1 \neq i_2} \lvert \pi_{i_1} - \pi_{i_2} \rvert$	$(r, r') \in RP$
(*viii*)	$\theta_{r,\ell} \in \{0,1\}$	$r = 1, 2, \ldots, n_{max};\ \ell = 1, 2, \ldots, \ell_x$

Figure 26. MILP Formulation for Simultaneous Construction - Discrete Factor

Minimize	$v + v_1 + v_3$	
Subject to		
(*i*)	$v \geq \tilde{\rho}_n(\boldsymbol{x}_{\cdot,i}, \boldsymbol{m}_{\cdot,c})$	$(c,n) \in (C_1 \times N_{III}) \cup (C \times (N_{IV} \cup N_V))$; $i = 1, 2, \ldots, \ell_x - 1$
(*ii*)	$v \geq -\tilde{\rho}_n(\boldsymbol{x}_{\cdot,i}, \boldsymbol{m}_{\cdot,c})$	$(c,n) \in (C_1 \times N_{III}) \cup (C \times (N_{IV} \cup N_V))$; $i = 1, 2, \ldots, \ell_x - 1$
(*i*-1)	$v_1 \geq \tilde{\rho}_n(\boldsymbol{x}_{\cdot,i} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c})$	$c_1 \in C_1$; $(c,n) \in (C_1 \times N_{IV}) \cup (C \times N_V)$; $i = 1, 2, \ldots, \ell_x - 1$
(*ii*-1)	$v_1 \geq -\tilde{\rho}_n(\boldsymbol{x}_{\cdot,i} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c})$	$c_1 \in C_1$; $(c,n) \in (C_1 \times N_{IV}) \cup (C \times N_V)$; $i = 1, 2, \ldots, \ell_x - 1$
(*i*-3)	$v_3 \geq \tilde{\rho}_n(\boldsymbol{x}_{\cdot,i}, \boldsymbol{x}_{\cdot,i} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1$; $n \in N_{IV} \cup N_V$; $i = 1, 2, \ldots, \ell_x - 1$
(*ii*-3)	$v_3 \geq -\tilde{\rho}_n(\boldsymbol{x}_{\cdot,i}, \boldsymbol{x}_{\cdot,i} \circ \boldsymbol{m}_{\cdot,c_1})$	$c_1 \in C_1$; $n \in N_{IV} \cup N_V$; $i = 1, 2, \ldots, \ell_x - 1$
(*iii*)	$\sum_{\ell=1}^{3} \theta_{r,\ell}^{i} = 1$	$r = 1, 2, \ldots, n_{max}$; $i = 1, 2, \ldots, \ell_x - 1$
(*iv*)	$x_{r,i} = \sum_{\ell=1}^{3} (\ell - 2)\theta_{r,\ell}^{i}$	$r = 1, 2, \ldots, n_{max}$; $i = 1, 2, \ldots, \ell_x - 1$
(*v*)	$\sum_{r=1}^{n} \theta_{r,\ell}^{i} \leq \left\lfloor (1+\delta)\frac{n}{\ell_x} \right\rfloor$	$\ell = 1, 3$; $n \in N$; $i = 1, 2, \ldots, \ell_x - 1$
(*vi*)	$\sum_{r=1}^{n} \theta_{r,\ell}^{i} \geq \left\lceil (1-\delta)\frac{n}{\ell_x} \right\rceil$	$\ell = 1, 3$; $n \in N$; $i = 1, 2, \ldots, \ell_x - 1$
(*vii*)	$\sum_{i=1}^{\ell_x - 1} \theta_{r,3}^{i} \leq 1$	$r = 1, 2, \ldots, n_{max}$
(*viii*)	$\sum_{i=1}^{\ell_x - 1} \theta_{r,2}^{i} \leq \ell_x - 2$	$r = 1, 2, \ldots, n_{max}$
(*ix*)	$\theta_{r,1}^{i} - \theta_{r,1}^{1} = 0$	$r = 1, 2, \ldots, n_{max}$; $i = 2, 3, \ldots, \ell_x - 1$
(*x*)	$\sum_{i=1}^{\ell_x - 1} i\, x_{r',i} - \sum_{i=1}^{\ell_x - 1} i\, x_{r,i} \geq 1$	$(r, r') \in RP$
(*xi*)	$\theta_{r,\ell}^{i} \in \{0,1\}$	$r = 1, 2, \ldots, n_{max}$; $\ell = 1, 2, 3$; $i = 1, 2, \ldots, \ell_x - 1$

Figure 27. MILP Formulation for Simultaneous Construction - Categorical Factor

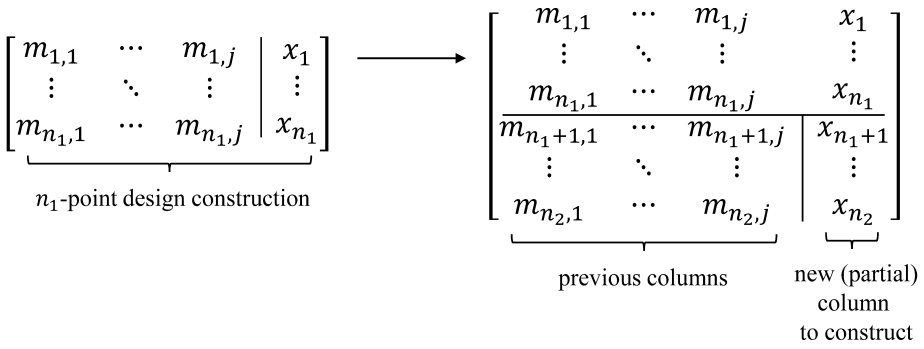


Figure 28. Design Augmentation for Two Stages

4.5 Case Study

A relatively small design space is used in this paper to examine the design properties resulting from batch sequential NOAB techniques, using the different NOAB approaches of NOAB-III, NOAB-IV, and NOAB-V. The design space of interest includes four two-level categorical factors and four discrete factors (two six-level, one four-level, and one three-level), resulting in a total of 6,912 possible design points. The low number of factors is amenable to achieving near orthogonality with respect to second-order models for smaller design sizes, with an assumption that some screening of a larger number of factors may have already occurred. The low numbers of levels for each factor gives a design space that is representative of similar real-world problems where space-filling has been desired. The maximum allowed imbalance is set to δ^*= 0.05 throughout the case study, which permits more accurate estimations of standard deviations, and thus, pairwise correlations, in the MILP formulations. Each discrete factor MILP is given 60 seconds and each categorical factor MILP is given 300 seconds of solve time, per design stage, using MATLAB R2016a with CPLEX v12.6.1. An additional solver attempt is made for each

factor if the first column solution does not achieve near orthogonality. The allotted solver times make certain that the two techniques, and the individual designs acting as a baseline, are given an equivalent amount of time to construct batch sequential NOAB designs. Let ρ_{map}^{III}, ρ_{map}^{IV}, and ρ_{map}^{V} denote the maximum absolute pairwise correlations for models terms considered in the NOAB-III, NOAB-IV, and NOAB-V design approaches, respectively, to better examine if each approach performs as intended.

4.5.1 Comparison of Augmentation and Simultaneous Construction

For the *NOAB-III* approach, six individual designs with $n = 24$, 36, 48, 60, 72, and 84 are compared to multiple-stage designs with the same sizes, using either the simultaneous construction ($N_{III} = \{24, 36, 48, 60, 72, 84\}$) or augmentation techniques ($N_{III} = \{24\}$, augment with $N_{III} = \{36\}$, and so on). No repeated points were observed in the designs resulting from the NOAB-III approach. The ρ_{map}^{III} values with respect to the first-order model are given in Table 12, where the individual designs have lower ρ_{map}^{III} than for the design stages resulting from the batch sequential techniques. However, it appears that the augmentation technique sees improving ρ_{map}^{III} for each new stage constructed, while both techniques provide sufficient correlation values for the first-order model assumption. Augmentation and simultaneous techniques for the three stages of $n = 24$, 48, and 84 result in ρ_{map}^{III} near zero, possibly implying that too many stages, or too small of batches, may constrain the MILP formulations so much that the ρ_{map}^{III} suffers.

Table 12. ρ_{map}^{III} for NOAB-III Designs

n	Individual	Augmentation	Simultaneous	Augmentation (Fewer Stages)	Simultaneous (Fewer Stages)
24	0.0000	0.0000	0.0488	0.0000	0.0000
36	0.0000	0.0497	0.0497	--	--
48	0.0000	0.0417	0.0417	0.0000	0.0000
60	0.0000	0.0334	0.0409	--	--
72	0.0000	0.0278	0.0373	--	--
84	0.0000	0.0250	0.0426	0.0000	0.0000

The NOAB-IV approach is examined for designs with $n = 60$, 72, 84, 96, 108, 120, and 132 for individual designs as well as the two batch sequential techniques. As with the NOAB-III designs, augmentation and simultaneous construction are used for designs with fewer stages as well (n =60, 96, and 120). The ρ_{map}^{IV} values, ignoring correlations between pairs of second-order terms, are shown in Table 13. The performance of the augmentation technique appears to suffer for the larger number of stages, yet is comparable to the simultaneous construction technique for the designs with fewer stages. The MILP formulation considers only the maximum absolute correlations for each case of v, $v_1, \ldots, v_5$, so the objective functions currently do not account for solutions that could improve in later stages and may be constrained by the worst-case ρ_{map}^{IV} for the $n = 60$ stage. This issue may be resolved by choosing a larger n for the first stage with respect to each NOAB approach used. Restructuring the MILP formulation to account for ρ_{map}^{IV} of the different stages in addition to the different correlation cases may also provide benefit. Both remedies will be examined later in this case study. The $tr(\boldsymbol{A}'\boldsymbol{A})$ measure for protection against bias from second-order terms, when assuming a first-order model,

follows a pattern similar to that of ρ_{map}^{IV}, where augmentation with more stages creates designs that suffer in quality. None of the other performance measures considered show clear differences between techniques, and no repeated points were found in the NOAB-IV designs.

Table 13. ρ_{map}^{IV} for NOAB-IV Designs

n	Individual	Augmentation	Simultaneous	Augmentation (Fewer Stages)	Simultaneous (Fewer Stages)
60	0.0656	0.0656	0.0707	0.0656	0.0656
72	0.0341	0.0802	0.0681	--	--
84	0.0337	0.1118	0.0628	--	--
96	0.0244	0.1018	0.0634	0.0511	0.0617
108	0.0262	0.0919	0.0651	--	--
120	0.0214	0.0928	0.0613	--	--
132	0.0207	0.0940	0.0665	0.0441	0.0612

When examining the construction of batch sequential NOAB-V designs, an improvement step is introduced to determine if there is benefit in restructuring the MILP formulations, as hypothesized. For the discrete factor case, after determining a factor column $\boldsymbol{x}'$ from the simultaneous construction technique, an additional MILP is solved with objective function

$$\sum_{n\in N} v_n + \sum_{n\in N_{IV}\cup N_V}(v_{1,n} + v_{2,n} + v_{3,n} + v_{4,n}) + \sum_{n\in N_V} v_{5,n}$$

where updates to the correlation constraints are made so that each estimate $\tilde{\rho}_n$ corresponds to the appropriate v_n, such as for the example constraints:

$$(i) \qquad v_n \geq \tilde{\rho}_n(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c}) \qquad (c,n) \in (C_1 \times N_{III}) \cup (C \times (N_{IV} \cup N_V))$$

$$(ii) \qquad v_n \geq -\tilde{\rho}_n(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c}) \qquad (c,n) \in (C_1 \times N_{III}) \cup (C \times (N_{IV} \cup N_V))$$

An additional constraint is added to provide an upper bound on each v_n using $\boldsymbol{x}'$, for all $n \in N$:

$$v_n \leq \max\{\left|\tilde{\rho}_{n'}(\boldsymbol{x}', \boldsymbol{m}_{\cdot,c})\right|, (c,n) \in (C_1 \times N_{III}) \cup (C \times (N_{IV} \cup N_V)) \text{ and } n' = n\}.$$

Similar updates and additional constraints are used for $v_{1,n}, \ldots, v_{5,n}$, $n \in N$, and the categorical factor MILP is updated as appropriate to comprise the complete improvement step. Additional solver time in the improvement step (90 seconds per discrete factor, 300 seconds per categorical factor) safeguards against a solution $\boldsymbol{x}'$ of poor quality from the simultaneous construction technique.

The NOAB-V approach is used to construct designs with $n = 144$ through 240 in increments of 12. The augmentation technique sees larger ρ_{map}^V than the simultaneous construction for full nine-stage designs as well as three-stage designs where $n = 144$, 192, and 240 (Table 14). The batch sequential techniques see smaller ρ_{map}^V overall when fewer stages are required. To lower ρ_{map}^V further, the simultaneous construction is attempted for only later stage sizes n =192 and 240 in addition to the improvement step on nine-stage and three-stage simultaneous constructions. The restructuring of the MILP decreases ρ_{map}^V for the nine-stage design, yet has inconsistent results for the three-stage design. Allowing the simultaneous construction to consider only stages n =192 and 240, ρ_{map}^V decreases even further, suggesting that a large enough n should be selected for the first stage with respect to each NOAB design approach. A small number of repeated points were found for the nine-stage designs using the simultaneous construction technique (three design points)

and the improvement step (two points) as well as the two-stage design using simultaneous construction (one point).

Table 14. ρ^{V}_{map} for NOAB-V Designs

n	Ind	Aug	Sim	Imp	Aug (Fewer)	Sim (Fewer)	Imp (Fewer)	Sim (Later)
144	0.0734	0.0734	0.0853	0.0772	0.0734	0.0616	0.0779	--
156	0.0489	0.1137	0.0859	0.0779	--	--	--	--
168	0.0532	0.1085	0.0850	0.0772	--	--	--	--
180	0.0541	0.1341	0.0953	0.0812	--	--	--	--
192	0.0434	0.1156	0.0885	0.0858	0.0881	0.0688	0.0746	0.0550
204	0.0393	0.1147	0.0725	0.0727	--	--	--	--
216	0.0372	0.1191	0.0926	0.0800	--	--	--	--
228	0.0311	0.1111	0.0923	0.0766	--	--	--	--
240	0.0358	0.1051	0.0887	0.0753	0.0870	0.0672	0.0632	0.0560

4.5.2 Batch Sequential NOAB Designs with Different Stage Approaches

In contrast to the constructions of batch sequential NOAB designs in the previous section, the aim now is to create designs that use different NOAB approaches for different stages of the overall design. In other words, the first stage design may use a NOAB-III approach, an intermediate stage may use NOAB-IV, and the last stage constructed may use NOAB-V. This gives even greater flexibility to an experimenter with respect to design choice by allowing for later stages to incorporate NOAB approaches that may not have been as appropriate in earlier stages. For the simultaneous construction, let $N_{III} = \{36\}$, $N_{IV} = \{72, 120\}$, and $N_V = \{168, 240\}$. Correlations for the five-stage designs are also compared to three-stage designs that no longer include the 72-point and 168-point stages. The simultaneous construction technique outperforms design augmentation for both

the five-stage and three-stage designs (Table 15). The improvement step improves the respective correlations for each stage of the five-stage design, with inconsistent results for the three-stage design. Repeated points occur for the five-stage designs using augmentation (three design points) and the improvement step (three points) as well as the three-stage designs using augmentation (one point), simultaneous construction (three points), and the improvement step (three points), which typically appear in the later design stages. Such small numbers of repeated points can quickly be removed for designs associated with deterministic systems, potentially using design augmentation to replace such points. Heatmaps of the absolute correlations matrices for each stage of the three-stage design, constructed simultaneously and using the improvement step, are provided in Figure 29. The matrix rows are partitioned by first-order model terms, then two-way interactions, and then quadratics to show exactly which pairs of model terms have low correlations for the different stages.

Table 15. ρ_{map} for Stages using Different NOAB Approaches

Correlation	n	Ind	Aug	Sim	Imp	Aug (Fewer)	Sim (Fewer)	Imp (Fewer)
ρ_{map}^{III}	36	0.0000	0.0000	0.0680	0.0680	0.0000	0.0325	0.0325
ρ_{map}^{IV}	72	0.0341	0.1147	0.0625	0.0564	--	--	--
	120	0.0214	0.1089	0.0740	0.0733	0.0463	0.0375	0.0397
ρ_{map}^{V}	168	0.0532	0.2288	0.0822	0.0731	--	--	--
	240	0.0358	0.1500	0.0760	0.0699	0.1267	0.0450	0.0412

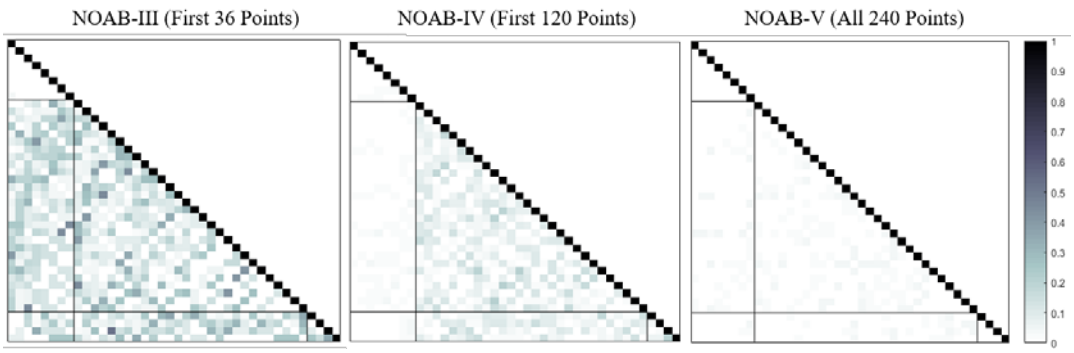


Figure 29. Absolute Correlations for Three-stage Design with Improvement Step

4.6 Conclusions and Further Research

Simultaneous construction of batch sequential NOAB designs appears to be the preferred technique overall, though design augmentation works well when fewer stages are required, or when batches contain more design points. Experimenters have greater flexibility when using NOAB designs by implementing a batch sequential process, which allows for different design stages to use different NOAB approaches, based on which first- and second-order model terms are of most interest. Regardless of the NOAB approach used for each stage, it appears that the simultaneous construction technique works best when the higher resolution NOAB approaches are used at later stages and when each stage has enough new design points to achieve near orthogonality. Except for design augmentation when there are many stages, with small batches of design points, the batch sequential techniques perform relatively well in achieving low correlation values, even if the strict definition of near orthogonality is not always met. The developed improvement step for simultaneous construction was shown to lower correlations for designs with several stages, yet the improvement to correlations was inconsistent for designs with fewer stages. The

MILP constraints for limiting the number of repeated points also works as intended, though future research may find techniques that result in greater reductions of repeated points.

Further research may include updates to the NOAB design augmentation technique that could incorporate design points based on meta-model exploration or existing sequential sampling techniques, followed by later stages of augmentation based on one of the three NOAB approaches. In other words, the simultaneous construction remains a model-free space-filling design approach, while an updated augmentation technique could incorporate points from model-based sampling approaches that account for mixed-factors. For research concerning the combination of space-filling and optimal design, see [147].

V. A Recommendation System for First-order NOAB Designs with Multiple Performance Measures

5.1 Abstract

The construction of nearly orthogonal-and-balanced (NOAB) designs is examined for full first-order models in a framework that is informed by the algorithm selection problem for multiple design performance measures and various design size and imbalance settings. Based on a randomly-generated set of large decision spaces, the choice of design size drives the changes in other design performance measures, with decision space features found to impact the measures as well. In this multi-objective setting, prediction of design performance within the framework consistently results in the recommendation of designs that perform well over an entire weight space in addition to designs for specific weights.

Keywords: space-filling design, meta-model, desirability function, synthesized efficiency

5.2 Introduction

Large decision spaces for complex, black-box systems often cannot be exhaustively explored, requiring space-filling experimental designs with possibly mixed factors (i.e., quantitative and qualitative with different numbers of levels). Such designs allow for the construction of meta-models to efficiently represent system responses, and the nearly orthogonal-and-balanced (NOAB) mixed-factor designs are a popular approach for these situations. Orthogonality allows for examination of individual factors separately and can be measured by the maximum absolute pairwise correlation of design matrix columns, denoted by ρ_{map}. An orthogonal design has $\rho_{map} = 0$, while a *nearly orthogonal* design

has $\rho_{map} \leq 0.05$. The first-order NOAB designs are created to ensure near orthogonality between first-order model terms (i.e., main effects). A design is considered *nearly balanced* when the maximum imbalance for all factor columns, δ, is close to zero, which ensures that all levels for a factor are represented nearly equally. A construction method is developed for first-order NOAB designs in [1], though beyond a suggested range for the number of design points there exists a need for greater knowledge of design performance for different design sizes and other construction parameter settings. With design matrix columns constructed sequentially by solving various mixed-integer linear programs (MILP), there are many possible parameter settings that could be examined to determine how to create the "best" performing design for a specific study. The framework of an algorithm selection problem can aid in such understanding by examining different parameter settings in the design construction method for a number of different decision space problems. The aim is to accurately predict design performance to allow for efficient design selection and construction. This knowledge will also allow for the development of a recommendation system that accounts for multiple design performance measures of possible interest to an analyst.

Meta-learning was developed to understand learning algorithm performance for classification problems, and developments in meta-learning from many different fields have been generalized and presented in a unified framework in [6] that considers the algorithm selection problem as a learning problem. Rice formalized the algorithm selection problem in [108], where the abstract model (Figure 30) is comprised of a problem space P, feature space F, algorithm space A, and performance space Y, with the algorithm selection problem stated as follows:

> "*For a given problem instance $x \in P$, with features $f(x) \in F$, find the selection mapping $S(f(x))$ into algorithm space A, such that the selected algorithm $\alpha \in A$ maximizes the performance mapping $y(\alpha(x)) \in Y$.*" [6]

The selection of a mapping function S is also an algorithm selection problem. Though the algorithm space A of interest will be a set of parameter settings for design construction, previous work in meta-learning for meta-model selection and other selection problems from [4], [116], [117], [125] can inform a model-based S that accurately predicts design performance measures based on meta-features from the problem space (i.e., set of decision spaces). The process permits the ranking of algorithms (i.e., parameter settings) and can lead to automated learning.

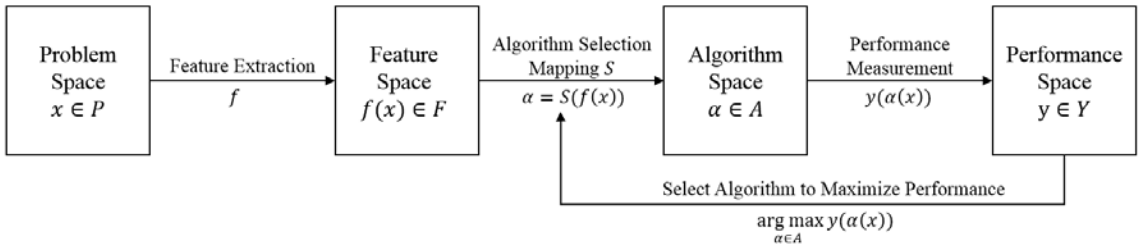


Figure 30. Diagram of Rice's model [4], [6], [108]

Design evaluation and comparison for when multiple performance measures are of interest are discussed, which will lead to how the performance space Y is defined. The algorithm selection problem for first-order NOAB design construction is then presented, with computational results for design performance as well as prediction performance of the resulting recommendation system provided.

5.3 Methodology

5.3.1 Experimental Design Evaluation and Comparison

With respect to design performance measures, focus is placed on low experimental cost (the number of design points, n, for a design matrix X) as well as good model parameter estimation and prediction accuracy. The average and maximum *unscaled prediction variance*, UPV $= x^{(m)\prime}(X'X)^{-1}x^{(m)}$, over all possible design points m are examined, as in [77]. When it is infeasible to compute the exact average or maximum UPV over a large decision space, an estimate is calculated using a Monte Carlo approach for up to ten million points from the design/decision space. In order to consistently estimate UPV values, all constructed designs for the same decision space problem are compared using the same sampling of points. For good parameter estimation, the *D-criterion*, $|X'X|^{1/p}$, from [76] is used. Due to finding similar overall trends for the average and maximum UPV measures, only maximum UPV is used as a design performance measure in the framework due to the greater variability seen over design choices. In this multi-objective setting, the aim is to minimize n and maximum UPV, while also maximizing the D-criterion.

The measures of various objectives should have the same scale in order to be comparable, so linear, one-sided *desirability functions* [82] are used for each of the criteria, with lower and upper limits set relative to the available designs [68]. A common approach for forming an overall desirability function for m objectives is the multiplicative function $D = \prod_{i=1}^{m} {d_i}^{w_i}$, for individual desirability scores d_i and weights w_i, where $\sum_{i=1}^{m} w_i = 1$. The multiplicative function ensures that no individual objective scores too low. *Synthesized efficiency* (SEff), defined as $D(X, w_1, \dots, w_m) / \max_{X^*} D(X^*, w_1, \dots, w_m)$ for design X, is

used to examine how X compares to the top performing design for various weightings $(w_1, \ldots, w_m)$ of overall desirability [68]. These techniques for design evaluation and comparison are used to obtain aggregate measures for the performance space.

5.3.2 Algorithm Selection Problem

The *problem space* consists of 30 randomly-generated decision spaces (Figure 31) with between eight and 20 factors overall, where categorical factors have between three and seven levels and discrete factors have between two and 12 levels. Previous work in decision support efforts for portfolio selection inform the decision spaces having multiple factors of the same type with the same number of levels. Note that continuous factors in NOAB designs are a special case of discrete factors with n levels equally spaced between zero and one.

The *algorithm space* is comprised of combinations of $m = 2, 3, \ldots, 10$ and maximum allowed imbalance $\delta^* = 0.05, 0.10, 0.15, 0.20, 0.25$. The smallest balance-feasible design size $n = m_{bf} \cdot s \geq m \cdot s$ is attempted for each choice of m where s is the number of design matrix columns, so it is possible that multiple (m, δ^*) combinations result in a single combination of (m_{bf}, δ^*). Larger δ^* values allow for greater imbalance and typically smaller values of balance-feasible n. Each MILP considers the set of design matrix columns for a single factor and is permitted up to two attempts of 30 seconds each to satisfy near orthogonality ($\rho_{map} \leq 0.05$). However, resulting designs with $\rho_{map} > 0.05$ are also recorded for better prediction of design performance. It is possible that some smaller designs may not be able to achieve near orthogonality, yet may have acceptable

ρ_{map} depending on the particular study. Larger designs may require more run time in the MILP solver to achieve near orthogonality due to greater computational requirements.

Problem	Factor (number of levels)																			
21	**6**	**6**	**6**	**6**	**4**	**4**	**4**	**4**	12	12	12	8	8	6	4	4	4	4	4	2
13	**6**	**6**	**6**	**6**	**6**	**3**	**3**	11	10	10	10	10	10	6	6	6	6	6	2	2
12	**7**	**7**	**6**	**3**	**3**	12	12	12	12	11	11	11	11	11	5	5	5	5	2	2
15	**6**	**5**	**5**	**5**	**5**	12	11	11	10	10	10	8	7	6	6	6	5	3	3	3
19	**6**	**4**	**4**	**4**	**4**	12	11	11	10	10	9	9	9	9	7	4	4	4	4	4
20	**7**	**7**	**7**	**7**	**7**	**6**	**6**	11	11	11	11	6	6	6	6	2	2	2	2	
10	**7**	**7**	**7**	**6**	**6**	**6**	**6**	**6**	**3**	12	10	10	9	7	7	7	7	7		
25	**3**	**3**	**3**	11	11	10	10	10	7	7	5	5	5	5	5	3	3	2		
1	**7**	**7**	**7**	**7**	**5**	**5**	11	11	11	11	11	10	7	7	6	6				
29	**7**	**6**	**6**	**6**	**6**	**6**	12	9	9	7	5	5	5	5	2	2				
22	**6**	**6**	9	9	9	9	6	6	6	6	6	5	2	2	2	2				
18	**7**	**7**	9	9	9	9	9	7	7	6	6	6	6	6						
17	**6**	**6**	**6**	**4**	**4**	**4**	**4**	**4**	6	6	6	6	5							
4	**7**	**6**	**6**	**6**	**6**	10	10	10	10	9	5	5	5							
23	**5**	**5**	**5**	**3**	9	9	9	9	5	5	5	5	5							
26	**4**	**4**	**4**	10	10	10	10	10	3	3	2	2	2							
11	**6**	**6**	**6**	**6**	**6**	**4**	**4**	6	6	6	6	6								
8	**6**	**6**	**6**	**3**	**3**	11	11	11	9	4	4	3								
3	**5**	**5**	**5**	8	8	3	3	3	3	3	2	2								
6	**6**	**6**	**6**	10	9	9	7	4	4	4	4	4								
2	**6**	**6**	10	10	10	10	10	5	5	5	5	5								
24	**4**	**4**	8	8	8	7	7	6	6	6	6	6								
30	**4**	**4**	11	11	11	11	6	6	5	5	5	2								
27	**6**	**6**	7	7	7	7	7	6	6	6	2									
7	**3**	9	9	9	9	7	4	4	4	3	3									
28	10	10	9	5	3	3	3	3	3	2	2									
14	**5**	**5**	**5**	**4**	**4**	11	11	11	11	2										
16	**7**	**7**	**3**	**3**	10	10	8	5	5											
9	**4**	**3**	12	8	8	8	8	8	4											
5	**7**	**4**	**4**	8	8	6	6	6												

Categorical

Discrete

Figure 31. Generated Problem Set of Decision Spaces

The *feature space* includes 24 meta-features with the goal of sufficiently describing each decision space problem: the number of factors for each factor type (discrete and categorical) as well as statistical measures of the number of levels for each factor type, to include minimum, mean, maximum, Q1, median, Q3, sum, standard deviation, skewness, and kurtosis. The product of all numbers of levels (i.e., full factorial design size) and least common multiple of all numbers of levels are also included as meta-features.

The *performance space* is multi-objective where the aim is to minimize design size n and estimated maximum UPV, while maximizing the parameter estimation measure D-criterion. As previously discussed, linear desirability functions of the three measures form an overall multiplicative desirability, with weights given to each individual desirability. The entire weight space $\{(w_1, w_2, w_3) | \sum_{i=1}^{3} w_i = 1\}$ is sampled using a 5,000-point space-filling mixture design. While multiplicative desirability for a specific set of weights can be informative, designs that are robust to weightings can also be found by examining average and minimum synthesized efficiencies (SEffs) over the weight space. With respect to overall desirability, average SEff, and minimum SEff over the weight space, the relative performance of the top five predicted designs is compared with that of the actual top performing design and Spearman's rank correlation coefficient is used to compare the actual and predicted rankings.

A model-based approach examines a set of possible mappings S from the parameter settings and meta-features to each of the performance measures, where the meta-model providing the smallest root mean square error (RMSE) for each measure is selected. The meta-models considered include artificial neural networks (ANN) [84]–[86], classification and regression trees (CART) [99], multivariate adaptive regression splines (MARS) [98], Gaussian processes (GP) with linear, polynomial, and radial kernels [88], [90], [91], random forests (RF) [148], and support vector machines (SVM) with linear, polynomial and radial kernels [103], [104]. Each meta-model uses the standard parameter grid search settings from the R package *caret*. The training and test instances are important for determining the meta-model S, so all observations for the problem to be predicted are held out from the training data. In order to reduce bias in design performance predictions, 10-

fold cross-validation is used where the training data is randomly partitioned so that all designs for the same problem instance will exist in either the training or validation set for each of the folds.

5.4 Computational Results

5.4.1 First-order NOAB Design Performance

Design construction is implemented in MATLAB R2015a using CPLEX V12.6.1 to obtain MILP solutions. Over the 30 decision space problems, there are 1,304 constructed designs in total, resulting from distinct combinations of m_{bf} and δ^* parameters. In Figure 32, there are clear trends in D-criterion as well as average and maximum UPV estimates over the true design size n and relative size m_{bf}. For designs of the same size, those requiring fewer columns tend to be more desirable for each design performance measure. The relative design size m_{bf} appears to have a strong relationship with average UPV, while designs with fewer columns tend to have higher maximum UPV for designs of the same relative size. It is clear that the choice of relative design size m is important as well as the number of columns s in the design matrix. The number of columns is comprised of defined meta-features, since each discrete factor is represented by a single column and each categorical factor with ℓ levels is represented by $\ell - 1$ columns when using effect coding.

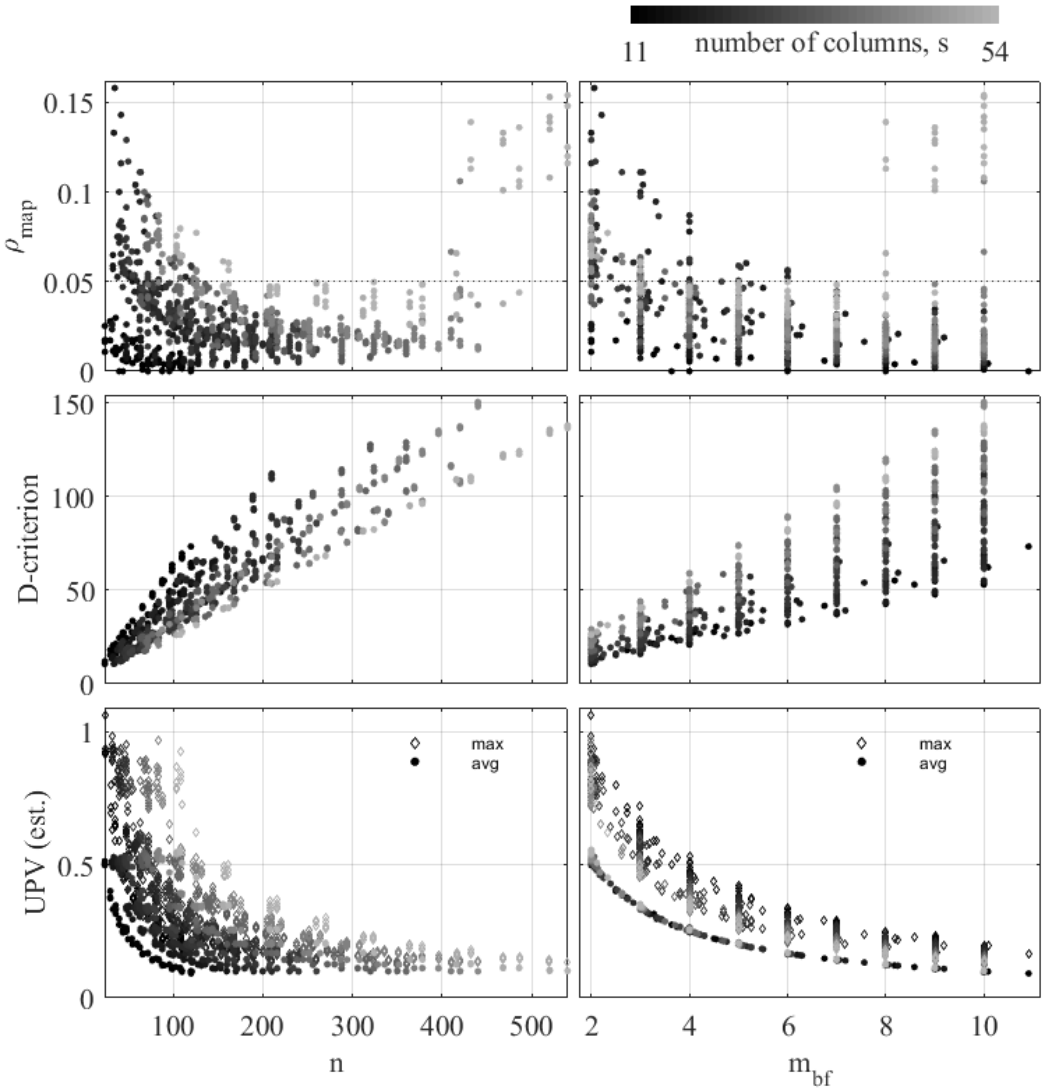


Figure 32. First-order NOAB Design Performance

Only 181 of 1,304 constructed designs are found to not be nearly orthogonal (ρ_{map} > 0.05), yet 26 larger designs (with $m_{bf} \geq 8$) can be constructed with near orthogonality when the MILP solver is permitted 60 seconds rather than 30 seconds per attempt (not shown in Figure 32). This is consistent with the overall trend for ρ_{map} as well as the idea that larger designs have greater computational requirements. When provided enough time in construction, it appears that larger designs will generally result in sufficient ρ_{map}. The

remaining 155 smaller designs that do not satisfy near orthogonality suggest that if small n is of the greatest concern to an analyst, even for problems requiring a small number of design matrix columns, they should examine whether the resulting ρ_{map} is sufficient for their particular problem.

5.4.2 Prediction Performance of Recommendation System

Design size n is predetermined by each choice of m and δ^* (and thus, m_{bf}) using the balance-feasibility test from [1]. For prediction of D-criterion, SVM with a polynomial kernel results in the smallest RMSE over all 30 training sets, with no other meta-model providing similarly small RMSE. For maximum UPV, RF provides the smallest RMSE for all 30 training sets with an average RMSE of 0.0225, while MARS provided the second best average of 0.0260. Figure 33 shows the actual versus predicted values of D-criterion and maximum UPV as well as their respective desirability scores for all 1,304 designs. The desirability scores for D-criterion are scaled relative to the designs found for each problem, which appear to resolve some of the bias that exists for a small number of problems.

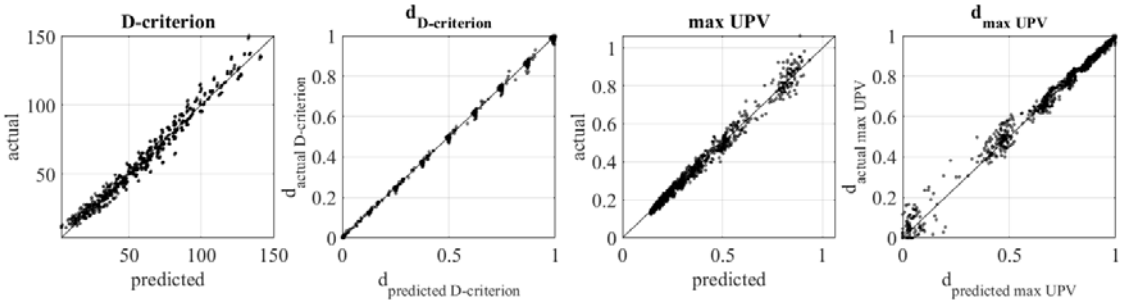


Figure 33. Actual by Predicted Design Performance

Table 16. Top-k Relative Performance and Spearman's Correlation Coefficient

			Robust Selections		Multiplicative Desirability Over Weight Space	
Over Problem Space			avg SEff	min SEff	min	avg
Top-k Relative Performance	1	min	0.9817	0.9809	0.8143	0.9823
	2		0.9906	0.9868	0.8813	0.9888
	3		0.9913	0.9913	0.9601	0.9920
	4		0.9914	0.9922	0.9601	0.9943
	5		0.9914	1.0000	0.9601	0.9974
	1	avg	0.9959	0.9966	0.9758	0.9965
	2		0.9978	0.9976	0.9916	0.9983
	3		0.9982	0.9989	0.9926	0.9991
	4		0.9985	0.9994	0.9940	0.9995
	5		0.9987	1.0000	0.9964	0.9998
Spearman's correlation coefficient		min	0.9613	0.9469	0.8475	0.9681
		avg	0.9764	0.9732	0.9652	0.9872

The larger residuals for high maximum UPV (low desirability) occur when design size n is small (high desirability), causing a small region of the weight space to have lower top-k relative performance and Spearman's correlation coefficient when examining the multiplicative overall desirability (Table 16). Otherwise, the top-k relative performance and Spearman's correlation coefficient are satisfactory for both robust design recommendations using average SEff and minimum SEff as well as multiplicative desirability for specific weights. For example, if we examine the top-1 relative performance for multiplicative desirability, the worst case (minimum) over both the weight space and problem space gives 0.8143, while the worst-case average over the 30 problems is 0.9753 and the worst-case average over the weight space is 0.9823. Though parameters associated with the most desirable designs will change over the weight space, common selections for

m_{bf} across all problems are 6 and 7 for high average SEff (often near 0.89) and 6 for high minimum SEff (often near 0.5). Increasing δ^* generally relaxes balance constraints to achieve smaller n, and thus, m_{bf}.

For a single decision space in this set of problems, the best and worst case for computation time required to construct designs for all (m_{bf}, δ^*) combinations are approximately two and 14 hours, respectively. For the recommendation system, building meta-models for D-criterion and maximum UPV on existing design data requires roughly 30 seconds when using the respective mappings of SVM with polynomial kernel and RF. Constructing a single, recommended design within this problem space needs only between three and 19 minutes. It is clear that the developed framework and resulting recommendation system allow for efficient selection and construction of first-order NOAB designs.

5.5 Conclusions and Further Research

This work shows it is possible to accurately predict first-order NOAB design performance measures for various design sizes and maximum allowed imbalance settings. These predictions permit a recommendation system that can provide both robust selections in the form of designs that have high average and maximum SEff over the weight space as well as designs that perform well for specific weights. For the 30 decision space problems considered, larger designs are generally more desirable with respect to good model parameter estimation as well as low prediction variance. Decision spaces with more design matrix columns tend to need more design points to achieve performance similar to other, smaller decision spaces.

We have derived extensions to the original first-order construction method to allow for the creation of second-order NOAB designs (i.e., near orthogonality includes two-way interactions and quadratic effects) (Chapter III), which may be examined in a similar framework. The second-order extensions also allow for an examination of *NOAB resolution IV* screening designs, in contrast to the first-order NOAB, or *NOAB resolution III* designs, that are the focus of this work. Additionally, future work could examine the computational requirements of these approaches based on the decision space of interest, whether by changing the allowed run time or implementing other stopping criteria for the MILP solver. A comparison with computer-generated optimal designs is also warranted for a large number of decision spaces with multiple performance measures of interest.

VI. Comparison of Mixed-factor Space-filling Designs for Meta-model Recommendation Systems

6.1 Abstract

Systems often have complex behavior and can be computationally expensive to evaluate. When there are many system input variables of interest in a study, exhaustive evaluation can be infeasible. Space-filling experimental designs are used to efficiently represent an entire design space for such variables, where the system output observed from the design are used to fit meta-models that can approximate each output variable for an entire input space. Space-filling designs that account for categorical, discrete, and continuous input variables (i.e., mixed factors) are compared in a case study with respect to the resulting meta-model performance. Beyond the question of which experimental design to use, it is not always clear which meta-modeling technique provides the best fit for an output variable, and fitting and comparing many meta-models for a large number of outputs can be costly and subjective. After selecting a second-order nearly orthogonal-and-balanced design (NOAB-V) as an appropriate mixed-factor experimental design, a meta-model recommendation system, based on the features of each output variable, is developed for a notional, complex system. The selected recommendation system suggests meta-models for 30 system outputs, with an average relative performance of 96.52% when compared to the true best and worst meta-models.

Keywords: operations research; computer simulation; experimental design; algorithm selection problem; nearly orthogonal-and-balanced

6.2 Introduction

Systems often have unknown, complex behavior and can be computationally expensive to evaluate. In this context, complex behavior may be nonlinear and difficult to model due to underlying subsystem or component interactions. When a system has many input variables of interest in a study, exhaustive evaluation can be infeasible. Experimental designs can accommodate these challenges by evaluating an efficient and representative subset of all possible input combinations of interest. Such designs are said to have good *space-filling* properties. The resulting experimental observations for each system output are then fitted to a *meta-model*, or surrogate model, that approximates the output for the entire input space. This meta-modeling approach allows engineers and analysts an efficient way of gaining insights from a complex system, whether it be a computer simulation or even a physical black box. For many systems in general, input variables can be categorical (e.g., should a new subsystem/feature be added or not? or which system mode should be used?), discrete (e.g., how many subsystems of a certain type are needed?), or continuous (e.g., how to set parameters/dials of system components?). When these different types of variables (i.e., factors) occur for the same system and different numbers of input values (i.e., factor levels) are possible, the system is said to have *mixed factors*.

Section 6.3 details a notional, complex system with mixed factors of interest used in the study. Section 6.4 describes a common approach for mixed-factor space-filling designs, the nearly orthogonal-and-balanced (NOAB) design, and compares recent extensions from Chapter III to the original NOAB design method in [1] for different design sizes (i.e., number of design points, or rows in the design matrix). Beyond the question of

which experimental design to use, it is not always clear which modeling technique will provide the best fit for each output, and fitting and comparing many meta-models for a large number of outputs can be costly and subjective. After choosing an appropriate experimental design, a recommendation system is developed in Section 6.5 with the aim of being able to efficiently recommend a single meta-model to use for new system outputs, based on features extracted from each output. The algorithm selection problem [108], as presented in [4], [6], provides a framework to develop this meta-model recommendation system.

6.3 Complex System with Mixed Factors

The notional system has seven input variables and 30 output variables (Figure 34). The aim is to use observations of the true system that result from a space-filling experimental design to approximate system output over the entire input space of interest.

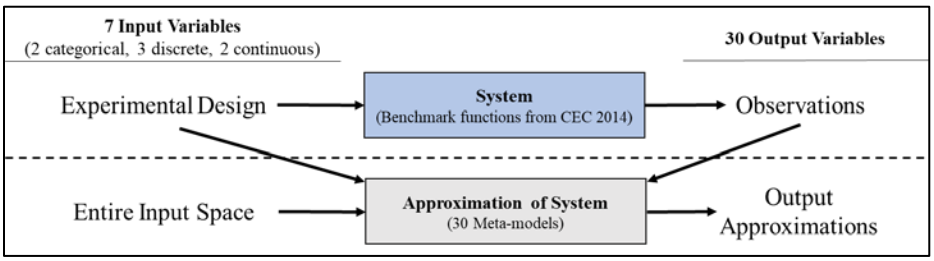


Figure 34. Overview of Experimental Design and Meta-modeling for Case Study

This complex system with mixed factors is constructed from 30 continuous benchmarks functions from the IEEE Congress on Evolutionary Computation (CEC) 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization [149]. Each of the benchmark functions act as an individual output for the system, consisting of three unimodal functions, 13 simple multimodal functions, six hybrid

functions, and eight composition functions, originally treated as black-box optimization problems in continuous space. Each function has 10 input variables $X_1, X_2, \ldots, X_{10}$ with domains of [-100,100], which are adapted to represent a system with seven mixed factors as follows:

- five-level categorical factor, where three of the original continuous inputs are confounded with randomly-selected choices (X_1, X_2, X_3) = (-46, -4, -31), (62, 71, -31), (-45, 10, -45), (-63, -30, -76), or (-9, 48, 62) to represent the five levels
- three-level categorical factor, where two of the original continuous inputs are confounded with randomly-selected choices (X_4, X_5) = (-37, 41), (-38, 79), or (25, -16) to represent the three levels
- three discrete factors with 12, nine, and four evenly-spaced levels over the domain [-100,100], respectively, for X_6, X_7, X_8 (e.g., the four-level discrete factor can have values of approximately -100, -33.33, 33.33, and 100 for input variable X_8)
- two continuous factors with 41 evenly-spaced levels over the domain [-100,100], respectively, for X_9, X_{10}.

The system is assumed to be deterministic, i.e., the same output is observed whenever inputs are repeated. Otherwise, the experimental designs would require repeated points to examine system randomness. No other assumptions are made with respect to the behavior of the system other than that 41 levels will provide sufficient fidelity for the two continuous factors.

6.4 Mixed-factor Space-filling Designs

6.4.1 Design Approaches

By comparing how various experimental designs perform with respect to how well resulting meta-models fit the data, this case study is intended as a proof-of-concept for other studies that may benefit from the use of mixed-factor space-filling designs. A common design for complex systems with mixed factors is the nearly orthogonal-and-balanced (NOAB) design from [1]. Near orthogonality allows for separation of factors (i.e.,

input variables) when examining relationships with outputs and means that there is sufficiently low correlation between columns of the experimental design matrix representing each factor (typically, the absolute value of these pairwise correlations are less than 0.05). Near balance means that the possible factor levels (i.e., input values) are represented a nearly equal number of times for each factor in the design.

The three main design approaches are NOAB-III, NOAB-IV, and NOAB-V. The NOAB-III from [1] constructs a design so that there is low correlation between design columns representing first-order effects. The NOAB-IV and NOAB-V are extensions of the NOAB-III that are derived in Chapter III, which are constructed to have low correlation between columns representing first- and second-order effects. The NOAB-V approach solves for low correlation for all possible pairs of first- and second-order effects, while the NOAB-IV approach ignores correlations between second-order effects. These three approaches are used to construct designs of size 164, 246, 328, 410, 508, and 600, sizes that allow for a maximum imbalance for all factors of 0.05. Imbalance for a factor x is defined as $\delta_x = \max_{i=1,\ldots,\ell_x} |(w_{i,x} - (n/\ell_x))/(n/\ell_x)|$, where $w_{i,x}$ is the number of times level i occurs for factor x with ℓ_x possible levels for design size n. The six design sizes and three approaches result in 18 design combinations, each of which are used to construct eight different designs to examine possible variation in resulting meta-model performance.

6.4.2 Design Comparison: Resulting Meta-model Performance

While the different design sizes and NOAB approaches have previously been compared with respect to traditional design properties, where larger design sizes and the NOAB-V designs have been shown to outperform smaller designs and other approaches,

respectively, we aim to show a more practical comparison of such designs by examining how well the resulting meta-models fit the complex behavior of a system. We consider a collection of 10 modeling approaches from the R software package *caret*, consisting of artificial neural networks (ANN), classification and regression trees (CART), multivariate adaptive regression splines (MARS), Gaussian processes (GP) with linear, polynomial, and radial kernels, random forests (RF), and support vector machines (SVM) with linear, polynomial, and radial kernels. Each of the 10 meta-model types was fitted to each of the 30 system outputs using standard parameter tuning in R and 10-fold cross validation.

To examine the performance of a single design approach in practice, the meta-model resulting in the smallest normalized root mean square error (NRMSE) is selected for each of the system outputs and that smallest NRMSE is then averaged for the 30 system outputs (i.e., 30 selected meta-models). NRMSE is often normalized using the largest observed difference or the mean of a system output with respect to the exact design being used. In order to better compare the different design approaches, we calculate the root mean square error using 100,000 randomly sampled design points, which is then normalized by the actual largest observed difference in system output over the entire design space. This normalization gives a more accurate sense of how the design approaches compare with respect to the resulting meta-model fits, and is made possible due to the notional system having significantly faster evaluation times than one would typically expect when meta-modeling a complex system.

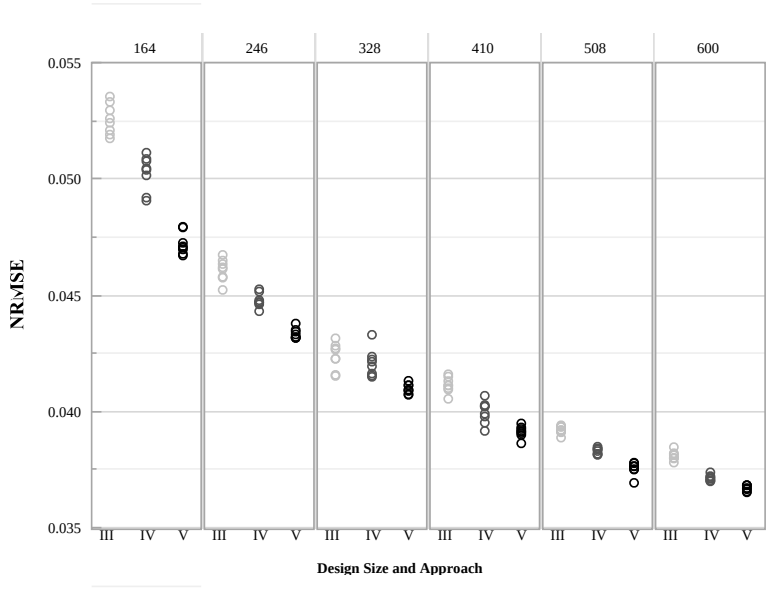


Figure 35. Average NRMSE for Selected Meta-models by Design Size and Approach

Figure 35 shows that while larger design sizes appear to generally result in lower NRMSE for each selected meta-model, with some variability within the combinations of design size and approach, the NOAB-V approach typically outperforms NOAB-IV, and NOAB-IV often outperforms NOAB-III. Many of the NOAB-V designs appear to perform nearly as well as NOAB-III designs that have approximately 80 more design points. Thus, the already efficient NOAB-III is further improved upon by using NOAB-IV and -V approaches, with smaller improvements seen for larger design sizes. We will now examine how a recommendation system for meta-models performs using a mixed-factor NOAB-V design with 600 points, randomly selected from the eight constructed designs of this approach and size.

6.5 Meta-model Recommendation System

6.5.1 Framework

The algorithm selection problem framework in Figure 36 shows how a meta-model recommendation system can be built. For the 30 system outputs to be meta-modeled (which comprise the *problem space*), features are extracted from each system output to be mapped to which of the 10 meta-models (the *algorithm space*) with the aim of having the best meta-model fit (the *performance space*), measured again by normalized root mean square error (NRMSE).

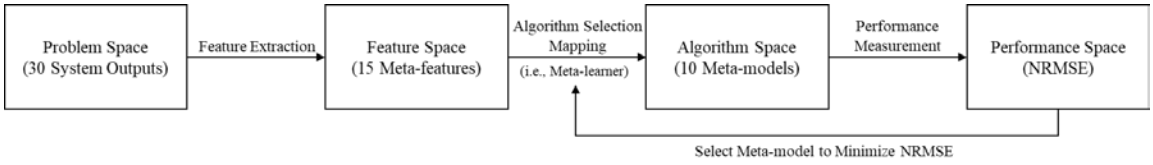


Figure 36. Diagram of Algorithm Selection Problem Framework [4], [6], [108]

The *feature space* consists of 15 meta-features from [4]:

- mean, median, standard deviation, and maximum of the gradient of the simulation output (1-4),
- mean, standard deviation, skewness, kurtosis, Q1 (first quartile), median, and Q3 (third quartile) of the simulation output (5-11),
- ratio of outliers in the simulation output (by repeatedly using Grubbs test to iteratively remove outliers) (12),
- ratio of local minima and maxima within a neighborhood (13-14), and
- average local biggest difference in simulation output (15).

The gradient for each point is defined as the difference between the output values for that point and the nearest neighbor. The local neighborhood is defined as the five nearest neighbors for features 13 through 15. Effect coding with values of -1, 0, and 1 is used for

the categorical factors when fitting meta-models as well as when extracting features from the system output that rely on a sense of distance between the input variables.

A meta-model based *meta-learner* (or algorithm selection mapping) is used to map the 15 meta-features to predict NRMSE for each of the 10 meta-models. Determining a meta-learner that works well is itself an algorithm selection problem. We compare 11 meta-learners, 10 of which are the meta-models in question for the recommendation system as well as an additional meta-learner that uses k-nearest neighbor. The meta-learner parameter settings are tuned using the default settings in the R *caret* package. The recommendation system framework from [4] suggests the use of singular value decomposition (SVD) to reduce the feature space, which allows us to reduce the dimensionality from 15 meta-features to a rank five approximation. In all, 22 recommendation systems are developed for combinations of 11 meta-learners with and without feature reduction using SVD.

6.5.2 Recommendation Performance

While the performance space of the framework is focused on NRMSE for meta-model performance, we must also measure the accuracy of the recommendation system itself with respect to meta-model selection. Three measures are used to examine meta-model recommendation performance for the 30 system outputs: *average relative performance* of the recommended meta-model when compared to the true best- and worst-performing meta-models in Figure 37 (i.e., the NRMSE from each recommended meta-model is scaled by the largest and smallest NRMSE over all meta-models fitted to the same system output so that the meta-models with smallest and largest NRMSE have relative performance of 1 and 0, respectively), average difference in NRMSE between the selected

meta-model and the true best (Figure 38), and the number of times the true best-performing meta-model is recommended (Figure 39). The k-NN meta-learner with and without SVD as well as GP with SVD appear to provide the largest average relative performance values among the recommendation systems (0.9652, 0.9568, and 0.9557, respectively, in Figure 37). Feature reduction using SVD does not appear to consistently improve or worsen the average relative performance of the 11 meta-learners.

The recommendation system with k-NN meta-learner using all 15 features provides the smallest average difference in NRMSE between the recommended meta-models and true best meta-models (0.00199 in Figure 38). The use of SVD improves the average differences in NRMSE for eight of the 11 meta-learners, suggesting that SVD can be useful to reduce the dimensionality of the feature space and possibly remove noise in the data, yet may worsen the average relative performance of a recommendation system due to improving the fit of the true worst meta-model. The two top-performing recommendation systems based on the number of times a true best meta-model is selected use the SVM with polynomial kernel and k-NN meta-learners, both with all features included (16 and 15, respectively, in Figure 39). While these top performers do not select the true best meta-models in every case, the average relative performance and average difference in NRMSE measures would indicate that both recommendation systems perform well for this case study.

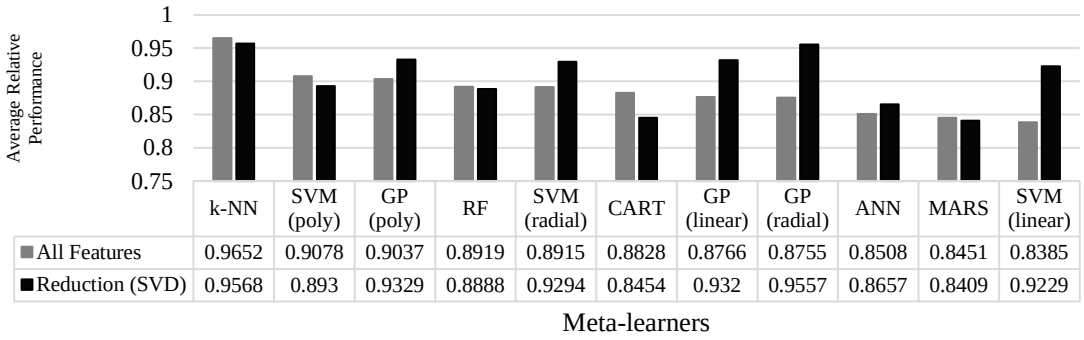


Figure 37. Average Relative Performance over 30 System Outputs by Meta-Learner

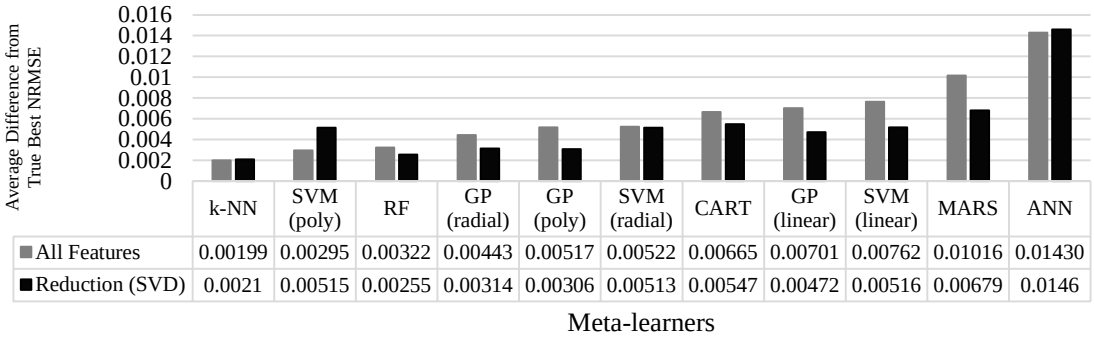


Figure 38. Average Difference from True Best NRSME over 30 System Outputs by Meta-learner

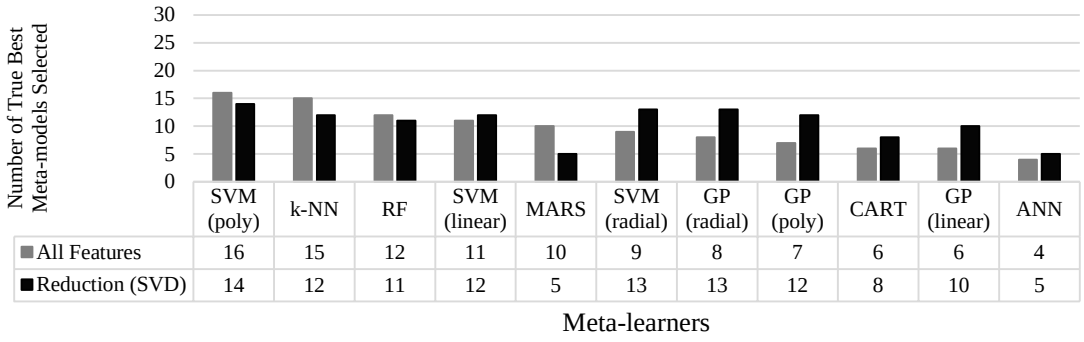


Figure 39. Number of True Best Meta-models Recommended for 30 System Outputs by Meta-learner

Figure 40 shows the NRMSE for 30 selected meta-models using the recommendation system with k-NN meta-learner and all 15 meta-features. When compared with the true best and true worst meta-models, the recommended meta-models perform relatively well overall. While the choice of meta-modeling approach greatly changes the NRMSE for most of the system outputs, there are several outputs that do not have a large difference in NRMSE for the 10 different meta-models and tend to have the largest NRMSE. The 30 true best meta-models include eight GP (poly), 11 RF, eight SVM (poly), one MARS, one CART, and one SVM (radial). The 30 recommended meta-models include 13 GP (poly), 12 RF, and five SVM (poly). It is clear that while the chosen recommendation system selects only three different meta-model approaches, relatively good meta-models are selected for each system output.

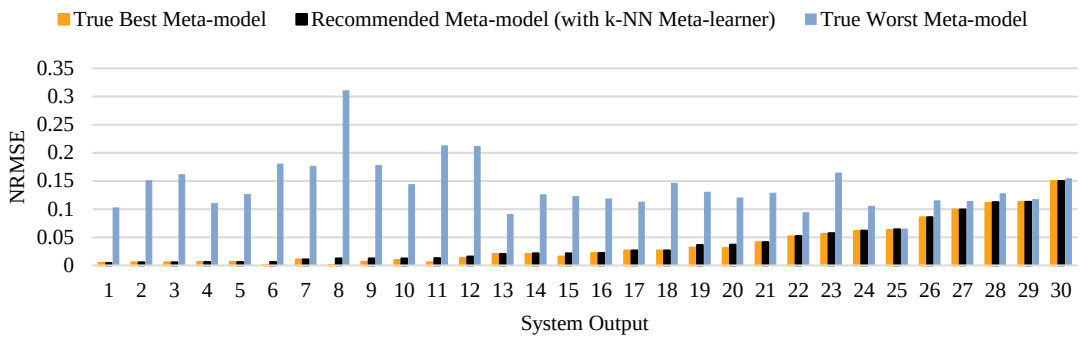


Figure 40. NRMSE by System Output

6.6 Conclusions

In this case study of a notional, complex system with mixed factors, the NOAB-V designs outperform NOAB-IV and -III designs when examining the best-fitting meta-models of system outputs. The resulting meta-model recommendation system, built from observations using a 600-point NOAB-V design and using a k-NN meta-learner, shows the importance of meta-model selection and suggests meta-models that provide relatively good fits when compared to the true best- and worst-performing meta-models. However, the true best meta-model was typically not recommended for more than half of the system outputs. This study uses a feature space that does not incorporate statistical features of input types (categorical, discrete, and continuous) or numbers of levels. The inclusion of such meta-features may not only result in better recommendations, but also allow for the recommendation system to extend to multiple mixed-factor systems with different input spaces. A construction tool for these NOAB design approaches will be available on the Air Force Institute of Technology website.

VII. Recommendations for Future Research

Future areas of research may include updating the objective functions of the MILP formulations to achieve certain design properties. Implementing weights for the maximum absolute correlation estimates $v, v_1, \ldots, v_5$ may benefit both the individual designs in Chapter III as well as the multiple stages of the batch sequential NOAB designs in Chapter IV. Techniques such as priority weighting may be useful when specific model terms are of greater interest, while low correlations overall are desired. The batch sequential formulations could also be updated to account for different maximum allowed imbalances depending on the design stage, so that requirements for smaller stages could be relaxed in order to have smaller balance-feasible design sizes available. Using the techniques developed in this research, deriving similar pairwise correlation estimates for third-order or higher model terms with respect to the current factor column(s) may prove difficult, though higher-order terms for the set of previously constructed columns (i.e., associated first-order model terms) could be incorporated in each MILP formulation.

Chapters III and IV placed little emphasis on setting low MILP solver times due to the significant amount of time required for planning and simulation in the real-world efforts that this research supports. The Appendix provides a batch sequential NOAB design that permits only 10 seconds of solver time per factor and stage, showing that significant reductions in time requirements may not practically hinder design performance. An in-depth examination of the time requirements for the construction methods is warranted if such techniques were to be used in statistical software, where the commercial expectation is typically to receive good designs quickly. However, there may be benefit in allowing

longer solver times for later factor columns, since there are more correlations to consider in the MILP formulation.

In this research, there was an initial attempt at developing a heuristic approach to construct the various NOAB designs. However, determining a good move set that ensures near balance, while aiming to minimize the various correlations, proved challenging.

The MILP formulations lend themselves to partitioning of the design matrix not only by consecutive rows as in the batch sequential approach, but also for arbitrary subsets of rows, allowing for an improvement scheme where most rows are fixed and a subset of rows are resolved in the MILP. As with design augmentation, this approach may alleviate issues with computer memory when the number of design points, and the number of associated decision variables, becomes too large. Chapter IV also discusses potential improvement to the batch sequential NOAB designs by incorporating augmentation of model-based or optimal points.

There is also potential benefit in updating the MILP formulations to account for LH sampling with respect to continuous factors, so that factor levels are bound by intervals and are not just evenly-spaced within the entire interval. The MILP formulations allow for the user to define specific factor level values, so random values could be assigned within the appropriate intervals, with reassignments for each additional solver attempt (i.e., if a sufficient factor column was not found). Another option would be to keep the evenly-spaced level values, but add a continuous decision variable that shifts each level assignment, though there is no immediately clear way to linearize the constraints for such an approach.

In the recommendation system for first-order NOAB designs (Chapter V), the desirability functions are bound by the worst- and best-performing choices available. An examination of how recommendations may change when using user-defined bounds rather than relative performance could be beneficial. Though the imbalance parameter was found to mostly impact how small a NOAB design could be, due to balance-feasibility, an analysis of designs with larger allowed imbalances may be of interest in order to observe any practical differences in the first-order NOAB design performance. Note that the second-order NOAB designs rely on small imbalances for accurate correlation estimates used in the construction method. The recommendation system for NOAB designs could be updated to examine additional settings such as computation time and the number of MILP solver iterations per design and factor. A similar algorithm selection problem framework could be used to develop recommendation systems for second-order NOAB designs and batch sequential NOAB designs to better understand and accurately predict performance of these new designs.

With respect to the meta-model recommendation system for mixed-factor systems (Chapter VI), future work may include an examination of the important meta-features found across the various system outputs. There is also opportunity to incorporate different design spaces for a more robust recommendation system. The Appendix provides a supplementary look at the correlations of the designs constructed in Chapter VI.

VIII. Conclusions

While the original, first-order NOAB designs can accommodate the computational challenges associated with complex systems, simulations, and real-world decisions, the second-order NOAB designs developed in this research are shown to provide practical improvement when fitting meta-models to system outputs (Chapter VI), while also improving design performance measures associated with second-order model parameter estimation and prediction variance (Chapter III). When assuming a first-order model, the second-order extensions allow for designs that protect against model misspecification with respect to second-order terms. The indexing within the MILP formulations can also be updated to focus on specific first- and second-order model terms of interest. Many studies may see value in a process that uses a NOAB-III or NOAB-IV design approach for initial screening of a large number of factors, followed by the second-order NOAB-V approach for significant factors and their associated second-order effects.

Two techniques were developed for construction of batch sequential NOAB designs, with simultaneous construction outperforming design augmentation overall, though each stage requires a predefined number of design points. Design augmentation was found to work well when there was a sufficiently large number of design points between each stage. The batch sequential NOAB designs give greater flexibility in how an experiment is conducted by providing mixed-factor designs that can be implemented in multiple stages, have been shown to have good space-filling properties, and can result in meta-models having better prediction accuracy. A natural path for future research is to

examine how model-based or optimal points can be augmented to these design for even better performance.

The algorithm selection problem framework was used to develop an accurate recommendation system for selection and construction of designs using the NOAB-III approach. In a multi-objective setting with a focus on design size, prediction variance, and good model parameter estimation, the prediction of design performance measures within the framework consistently results in design recommendations that are robust to changes in performance weights. The choice of design size was found to be the largest driver of changes in performance measures, with relaxed imbalance settings permitting smaller balance-feasible design sizes. Design spaces requiring more design matrix columns tend to need more design points to achieve performance similar to other smaller design spaces.

The meta-model recommendation system, built from observations using a 600-point NOAB-V design and a k-nearest neighbor meta-learner, suggests meta-models that provide relatively good fits when compared to the true best- and worst-performing meta-models. The poor performance of some meta-models, even when using a good experimental design, highlights the importance of selecting meta-modeling techniques that fit each system output well and of not relying on a single type of meta-model. This research not only contributes to the ever-advancing stream of research on experimental designs for complex systems, but also provides further examples of how the algorithm selection problem framework can be used to gain insight on challenging problems, whether those algorithms are construction methods for first-order NOAB designs or meta-models for approximation of complex system behavior.

Appendix

Supplementary Background Material from [1]

$\delta \leftarrow \mathbf{0}$
for $j = 1, j < K$ do
    if $C(x_j) \in \{2,3\}$ do
        $\delta_{x_j} \leftarrow \left(\frac{\ell_{x_j}}{n}\right) \max\left(\left(\left\lceil \frac{n}{\ell_{x_j}} \right\rceil - \frac{n}{\ell_{x_j}}\right), \left(\frac{n}{\ell_{x_j}} - \left\lfloor \frac{n}{\ell_{x_j}} \right\rfloor\right)\right)$
        $\delta \leftarrow \mathbf{max}\left(\delta, \delta_{x_j}\right)$
    end
end
if $\delta > \delta^*$
    RETURN "No feasible solution exists with current balance constraints. Increase n until the feasibility check is passed, or set $\delta^* = \delta$"
else RETURN "Initial balance feasibility check passed"

Figure 41. Balance Feasibility Test – Original Notation [1] (Updates in Bold)

$b \leftarrow 0$

if $\{b < b^*\}$

$M_0 \leftarrow \emptyset, \widetilde{M}_0 \leftarrow \emptyset$
$j \leftarrow 0$
if $\{j < K\}\ do$
    $solution_{j+1} \leftarrow$ "FALSE"
    $h \leftarrow 1$
    if $\{h < h^*$ AND $solution_{j+1} =$ "FALSE" $\}\ do$
        $t \leftarrow t_{min}$
        $x \leftarrow$ an $n \times 1$ vector, randomly generated from $x \in \widetilde{\mathcal{B}}(n, c_{j+1})$
        if $\{t < t_{max}\ AND\ solution_{j+1} =$ "FALSE" $\}\ do$
            call MILP using $\widetilde{M}_j, \delta^*, t, x, \ell_x,$ and $C(x)$
            $v^* \leftarrow$ MILP objective function value
            $x^* \leftarrow$ MILP modified column vector
            $s_{x^*} \leftarrow$ standard deviation of x^*
            if $\{v^* \leq \alpha^* s_{x^*}\}\ do$
                $solution_{j+1} \leftarrow$ "TRUE"
            else if $\{v^* > \alpha^* s_{x^*}$ AND t < t_{max} $\}$ do
                $t \leftarrow t + t_{min}$
            else do
                $h \leftarrow h + 1$
                $t \leftarrow t_{min}$
            end
        end
    end
    if $\{solution_{j+1} =$ "TRUE"$\}$ do
        if $\{C(x) = 3$ (i.e., x is categorical)$\}$ do
            $x^{*i} \leftarrow$ i^{th} indicator vector associated with x^* $(i = 1, 2, \dots, \ell_x - 1)$
            $\widetilde{M}_{j+1} \leftarrow [\widetilde{M}_j\ x^{*1}\ x^{*2}\ \cdots\ x^{*(\ell_x - 1)}]$
        else do
            $\widetilde{M}_{j+1} \leftarrow [\widetilde{M}_j\ x^*]$
        end
        $M_{j+1} \leftarrow [M_j\ x^*]$
        $j \leftarrow j + 1$
    end
end
if $\{solution_K =$ "TRUE"$\}$ RETURN M_K
else $b \leftarrow b + 1$
end
RETURN "No solution found that meets near-orthogonality criteria"

Figure 42. First-order NOAB Construction Method – Original Notation [1]

Summary of Pairwise Correlation Estimates, Derived in Chapter III

Update to First-order Correlation Estimates from [1]

$\tilde{\rho}(\boldsymbol{x}, \boldsymbol{m}_{\cdot,c}) = 1/((n-1)\, s_{\boldsymbol{x_0}} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} (x_r - \overline{\boldsymbol{x}})(m_{r,\boldsymbol{c}} - \overline{\boldsymbol{m}_{\cdot,c}})$

Discrete Factor Case

Extension 1. $\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c}) = 1/((n-1) s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} m_{r,c_1} (m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}})(x_r - \overline{\boldsymbol{x}})$

Extension 2. $\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{m}_{\cdot,c}) = 1/((n-1) s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} (m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}}) (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^2 - 2\overline{\boldsymbol{x}_0} \pi_\ell) \theta_{r,\ell})$

Extension 3. $\tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}) = 1/((n-1) s_{\boldsymbol{z}_0} s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}}) \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^2 - 2\overline{\boldsymbol{x}_0} \pi_\ell) \theta_{r,\ell} + \overline{\boldsymbol{x}_0}^2)$

Extension 4. $\tilde{\rho}(\boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{z}) = 1/((n-1) s_{\boldsymbol{z}_0} s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0}) \sum_{r=1}^{n} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^3 - 3\pi_\ell^2 \overline{\boldsymbol{x}_0} + 3\overline{\boldsymbol{x}_0}^2 \pi_\ell) \theta_{r,\ell} - \overline{\boldsymbol{x}_0}^3)$

Extension 5. $\tilde{\rho}(\boldsymbol{z} \circ \boldsymbol{z}, \boldsymbol{z} \circ \boldsymbol{m}_{\cdot,c_1}) = 1/((n-1) s_{\boldsymbol{z}_0 \circ \boldsymbol{z}_0} s_{\boldsymbol{z}_0 \circ \boldsymbol{m}_{\cdot,c_1}}) \sum_{r=1}^{n} m_{r,c} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^3 - 3\overline{\boldsymbol{x}_0} \pi_\ell^2 +$

$\left(4\overline{\boldsymbol{x}_0}^2 - \overline{\boldsymbol{x}_0 \circ \boldsymbol{x}_0}\right) \pi_\ell) \theta_{r,\ell} - 2\overline{\boldsymbol{x}_0}^3 + \overline{\boldsymbol{x}_0}\, \overline{\boldsymbol{x}_0 \circ \boldsymbol{x}_0})$

Categorical Factor Case

Extension 1. $\tilde{\rho}(\boldsymbol{x}_{\cdot,i} \circ \boldsymbol{m}_{\cdot,c_1}, \boldsymbol{m}_{\cdot,c}) = 1/((n-1) s_{\boldsymbol{x}_{0,i} \circ \boldsymbol{m}_{\cdot,c_1}} s_{\boldsymbol{m}_{\cdot,c}}) \sum_{r=1}^{n} m_{r,c_1} (m_{r,c} - \overline{\boldsymbol{m}_{\cdot,c}}) x_{r,i}$

Extension 3. $\tilde{\rho}(\boldsymbol{x}_{\cdot,i}, \boldsymbol{x}_{\cdot,i} \circ \boldsymbol{m}_{\cdot,c_1}) = 1/((n-1) s_{\boldsymbol{x}_{0,i}} s_{\boldsymbol{x}_{0,i} \circ \boldsymbol{m}_{\cdot,c_1}}) \sum_{r=1}^{n} m_{r,c_1} (\sum_{\ell=1}^{\lambda(\boldsymbol{x})} (\pi_\ell^2 - \overline{\boldsymbol{x}_{0,i}} \pi_\ell) \theta_{r,\ell}^i)$

Supplementary Results for Chapter III

Using the design space from the Chapter III case study, the NOAB-V approach is used to construct a 36-point design. Figure 43 shows the heatmap of absolute correlations for this design, where pairwise correlations between only first-order model terms (with $\rho_{map}^{III} = 0.2222$) are much higher than for the 36-point design using the NOAB-III approach (satisfying near orthogonality with respect to first-order model terms in Chapter III).

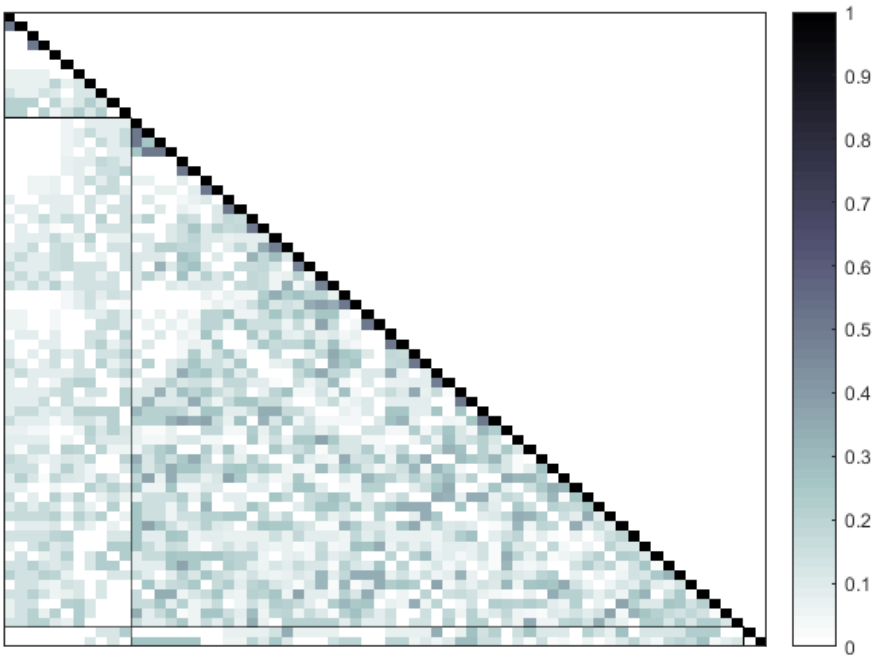


Figure 43. Absolute Correlation Heatmap for 36-point NOAB-V Design

Supplementary Results for Chapter VI

Maximum absolute correlations for different model terms (Table 17) as well as absolute correlation heatmaps (Figure 43) are provided for a sample of the NOAB designs constructed to study a mixed-factor system with 10,892,880 possible design points.

Table 17. Maximum Absolute Correlations for Some Chapter VI designs

Approach	n	ρ_{map}^{III}	ρ_{map}^{IV}	ρ_{map}^{V}
NOAB-III	164	0.0120	0.3774	0.4578
NOAB-III	246	0.0079	0.4198	0.4198
NOAB-III	328	0.0059	0.2899	0.2899
NOAB-III	410	0.0012	0.4082	0.4082
NOAB-III	508	0.0038	0.1845	0.1922
NOAB-III	600	0.0002	0.4428	0.4428
NOAB-IV	164	0.0195	0.0231	0.2983
NOAB-IV	246	0.0160	0.0294	0.2472
NOAB-IV	328	0.0149	0.0326	0.2095
NOAB-IV	410	0.0166	0.0227	0.1764
NOAB-IV	508	0.0100	0.0307	0.1614
NOAB-IV	600	0.0085	0.0445	0.1150
NOAB-V	164	0.0507	0.0801	0.1176
NOAB-V	246	0.0339	0.0709	0.0878
NOAB-V	328	0.0368	0.0560	0.0854
NOAB-V	410	0.0244	0.0456	0.0636
NOAB-V	508	0.0198	0.0261	0.0420
NOAB-V	600	0.0163	0.0333	0.0374

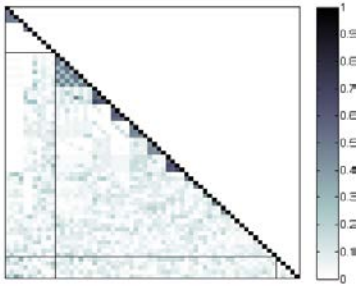
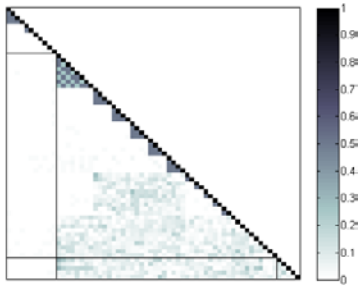
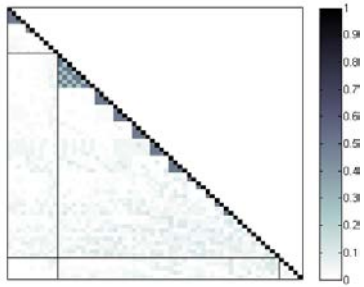
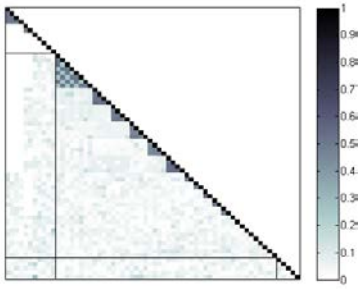
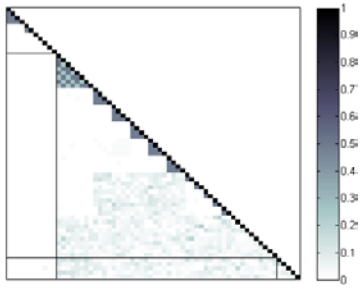
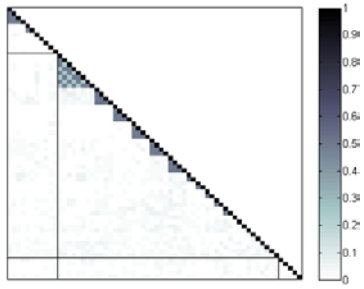
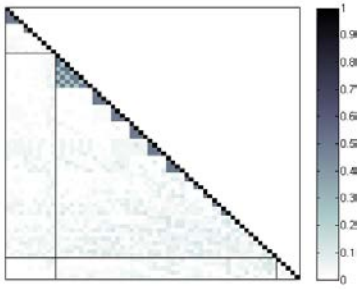
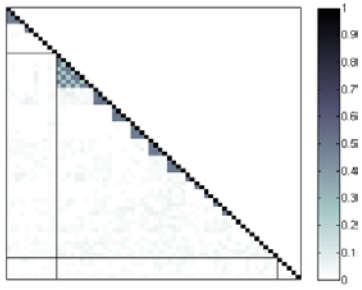
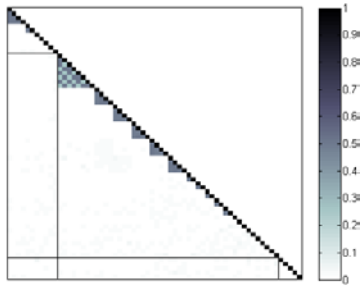
n	NOAB-III Approach	NOAB-IV Approach	NOAB-V Approach
164			
328			
600			

Figure 44. Absolute Correlation Heatmaps for Different NOAB Approaches and Sizes

Fast Computation of Batch Sequential NOAB Design

The design space from Chapter IV is updated to include one three-level categorical factor, four two-level categorical factors, two six-level discrete factors, one five-level discrete factor, and one three-level discrete factor, totaling 25,920 possible design points. Let $N_{III} = \{30\}, N_{IV} = \{72, 120\}$, and $N_V = \{168, 240\}$. To examine an initial reduction in computation time for the simultaneous construction approach, we implement a baseline case where each factor is given 180 seconds per NOAB-III and NOAB-IV stage and 600 seconds per NOAB-V stage in the MILP solver. Since there are five stages with two of the later stages using the NOAB-V approach, the total number of seconds allowed per factor construction is 1,740 seconds. The "fast" approach is then given 10 seconds per stage regardless of the NOAB approach used, totaling 50 seconds per factor. We provide maximum absolute correlations for the different stages (Table 18) as well as absolute correlation plots (Figure 45 through Figure 49) to show how similar the correlations are for the intermediate designs, with pairs of model terms on the x-axis ordered by first-order pairs, then between first-order and second-order pairs, and finally, second-order pairs. The design constructed using the baseline approach required approximately 24,404 seconds, while the fast design required only about 720 seconds, a 97% decrease in time. Further analysis of computational requirements is suggested as a future area of research.

Table 18. Maximum Absolute Correlations for Batch Sequential NOAB Designs (Time Comparison)

		ρ_{map}^{III}		ρ_{map}^{IV}		ρ_{map}^{V}	
	n	Baseline	Fast	Baseline	Fast	Baseline	Fast
	30	0.0816	0.0976	0.4245	0.4910	0.6547	0.7184
	72	0.0890	0.0896	0.0946	0.1295	0.3934	0.5115
Stage	120	0.0621	0.0506	0.0621	0.0845	0.2544	0.3219
	168	0.0583	0.0657	0.0583	0.0713	0.1415	0.1666
	240	0.0343	0.0351	0.0444	0.0544	0.1005	0.1104

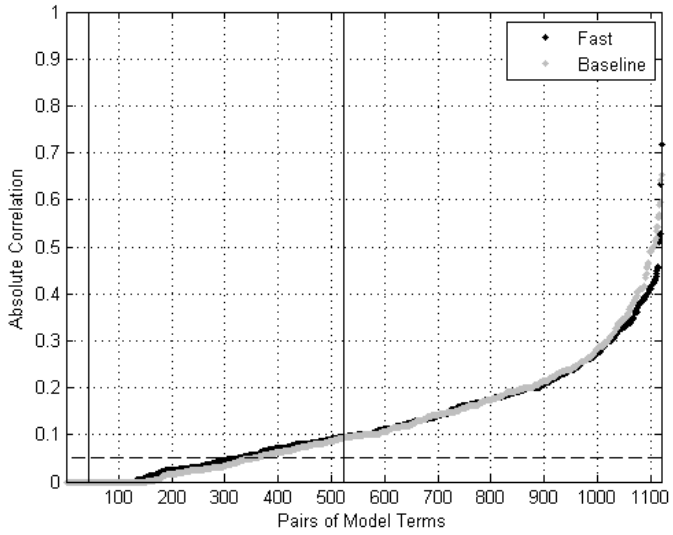


Figure 45. Absolute Correlations for 30-point Stage NOAB-III

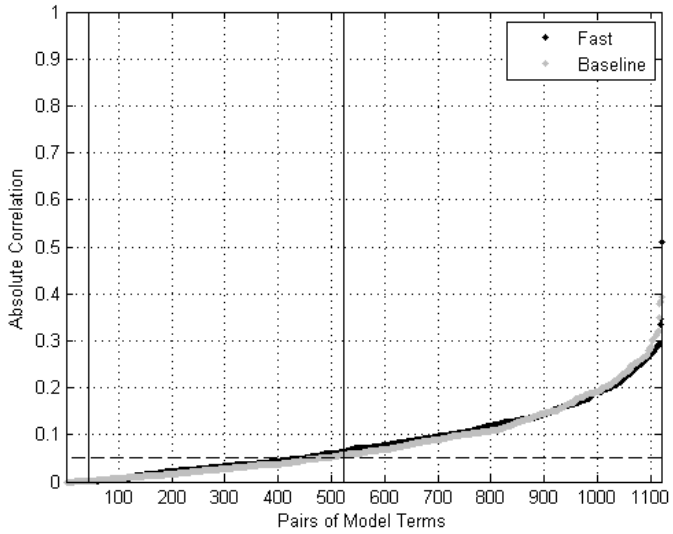


Figure 46. Absolute Correlations for 73-point Stage NOAB-IV

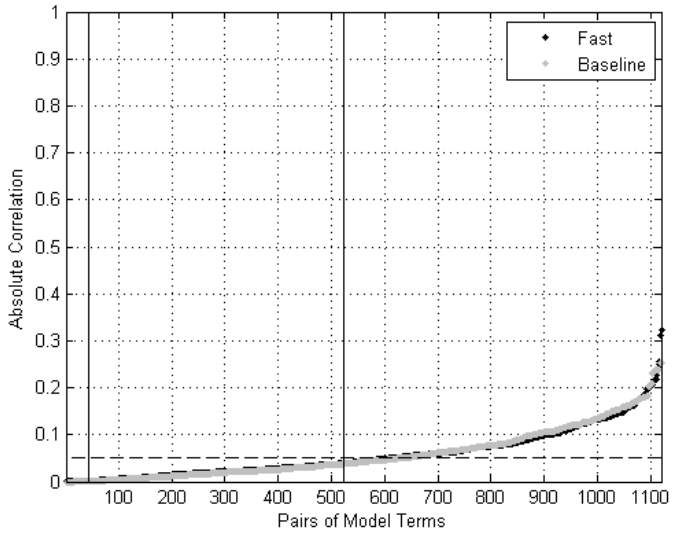


Figure 47. Absolute Correlations for 120-point Stage NOAB-IV

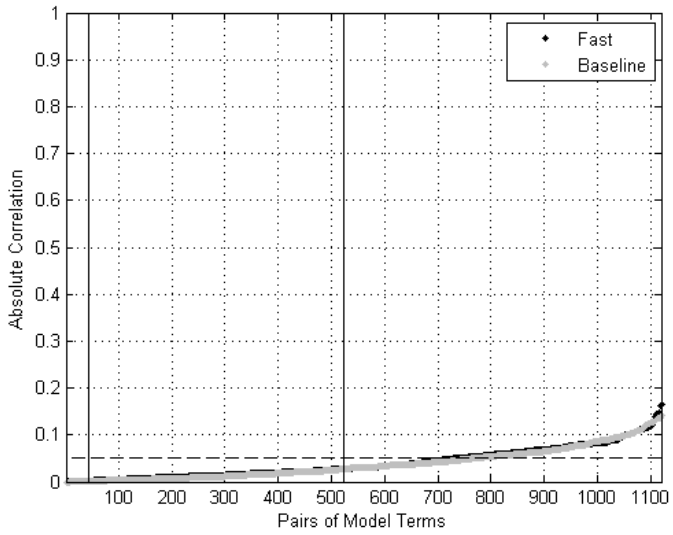


Figure 48. Absolute Correlations for 168-point Stage NOAB-V

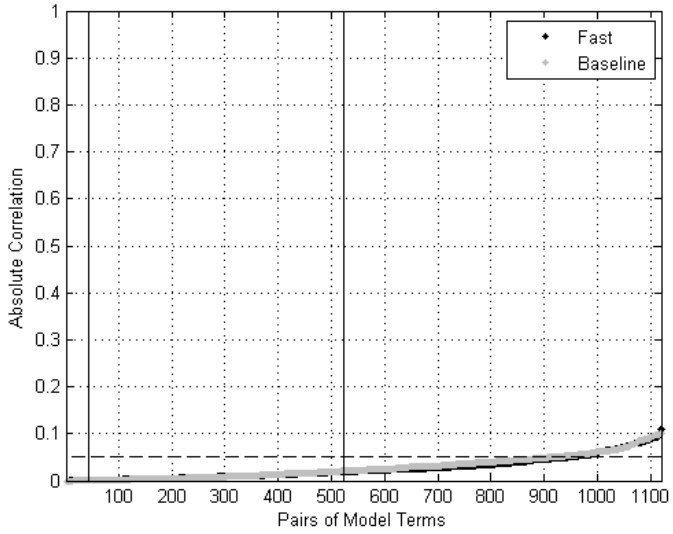


Figure 49. Absolute Correlations for 240-point Stage NOAB-V

Bibliography

[1] H. Vieira Jr., S. M. Sanchez, K. H. Kienitz, and M. C. N. Belderrain, "Efficient, nearly orthogonal-and-balanced, mixed designs: an effective way to conduct trade-off analyses via simulation," *Journal of Simulation*, vol. 7, no. 4, pp. 264–275, 2013.

[2] U.S. Air Force, "Intelligence, Surveillance, and Reconnaissance," 2013. [Online]. Available: http://www.af.mil/News/ArticleDisplay/tabid/223/Article/466894/intelligence-surveillance-and-reconnaissance.aspx. [Accessed: 28-Sep-2016].

[3] Air Force Doctrine Document (AFDD) 2-0, *Global Integrated Intelligence, Surveillance, and Reconnaissance Operations*. 2012.

[4] C. Cui, M. Hu, J. D. Weir, and T. Wu, "A recommendation system for meta-modeling: A meta-learning based approach," *Expert Systems with Applications*, vol. 46, pp. 33–44, 2016.

[5] C. Cui, T. Wu, M. Hu, J. D. Weir, and X. Li, "Short-term building energy model recommendation system: A meta-learning approach," *Applied Energy*, vol. 172, pp. 251–263, 2016.

[6] K. A. Smith-Miles, "Cross-disciplinary perspectives on meta-learning for algorithm selection," *ACM Computing Surveys*, vol. 41, no. 1, pp. 1–25, 2008.

[7] M. Cote, "Screening and Sufficiency in Multiobjective Decision Problems with Large Alternative Sets," Master's Thesis. Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2010.

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

[9] J. P. C. Kleijnen, S. M. Sanchez, T. W. Lucas, and T. M. Cioppa, "State-of-the-art review: a user's guide to the brave new world of designing simulation experiments," *INFORMS Journal on Computing*, vol. 17, no. 3, pp. 263–289, 2005.

[10] S. M. Sanchez, P. J. Sanchez, and H. Wan, "Simulation experiments: better insights by design," in *Proceedings of the 2014 Summer Simulation Conference*, 2014.

[11] D. C. Montgomery, *Design and Analysis of Experiments*. Hoboken, New Jersey: John Wiley & Sons, 2013.

[12] A. D. MacCalman, "Flexible space-filling designs for complex system simulations," Doctoral Dissertation. Naval Postgraduate School, Monterey, California, 2013.

[13] C. R. Rao, "Hypercubes of strength d leading to confounded designs in factorial experiments," *Bulletin of the Calcutta Mathematical Society*, vol. 38, no. 3, pp. 67–78, 1946.

[14] C. R. Rao, "Factorial experiments derivable from combinatorial arrangements of arrays," *Supplement to the Journal of the Royal Statistical Society*, vol. 9, no. 1, pp. 128–139,

1947.

[15] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal Arrays: Theory and Applications*. Springer Science & Business Media, 2012.

[16] G. E. P. Box and K. B. Wilson, "On the experimental attainment of optimum conditions," *Journal of Royal Statistical Society Series B*, vol. 13, pp. 1–45, 1951.

[17] G. E. P. Box and D. W. Behnken, "Some new three level designs for the study of quantitative variables," *Technometrics*, vol. 2, no. 4, pp. 455–475, 1960.

[18] A. T. Hoke, "Economical second-order designs based on irregular fractions of the 3 n factorial," *Technometrics*, vol. 16, no. 3, pp. 375–384, 1974.

[19] K. G. Roquemore, "Hybrid designs for quadratic response surfaces," *Technometrics*, vol. 18, no. 4, pp. 419–423, 1976.

[20] S. M. Sanchez and P. J. Sanchez, "Very large fractional factorials and central composite designs," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 15, no. 4, pp. 362–377, 2005.

[21] J. Kiefer and J. Wolfowitz, "Optimum designs in regression problems," *The Annals of Mathematical Statistics*, vol. 30, no. 2, pp. 271–294, 1959.

[22] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[23] R. L. Iman and W. J. Conover, "A distribution-free approach to rank correlation among input variables," *Communications in Statistics - Simulation and Computation*, vol. 11, no. 3. pp. 311–334, 1982.

[24] A. Florian, "An efficient sampling scheme: updated Latin hypercube sampling," *Probabilistic Engineering Mechanics*, vol. 7, no. 2, pp. 123–130, 1992.

[25] A. B. Owen, "Controlling correlations in Latin hypercube samples," *Journal of the American Statistical Association*, vol. 89, no. 428, pp. 1517–1522, 1994.

[26] B. Tang, "A theorem for selecting OA-based Latin hypercubes using a distance criterion," *Communications in Statistics - Theory and Methods*, vol. 23, no. 7, pp. 2047–2058, 1994.

[27] M. C. Shewry and H. P. Wynn, "Maximum entropy sampling," *Journal of Applied Statistics*, vol. 14, no. 2, pp. 165–170, 1987.

[28] M. E. Johnson, L. M. Moore, and D. Ylvisaker, "Minimax and maximin distance designs," *Journal of Statistical Planning and Inference*, vol. 26, no. 2, pp. 131–148, 1990.

[29] K.-T. Fang, "The uniform design application of number-theoretic methods in experimental design," *Acta Mathematicae Applicatae Sinica*, vol. 3, no. 4, pp. 363–372, 1980.

[30] K.-T. Fang, D. K. J. Lin, P. Winker, and Y. Zhang, "Uniform design: theory and application," *Technometrics*, vol. 42, no. 3, pp. 237–248, 2000.

[31] K. Q. Ye, “Orthogonal column Latin hypercubes and their application in computer experiments,” *Journal of the American Statistical Association*, vol. 93, no. 444, pp. 1430–1439, 1998.

[32] D. M. Steinberg and D. K. J. Lin, “A construction method for orthogonal Latin hypercube designs,” *Biometrika*, vol. 93, no. 2, pp. 279–288, 2006.

[33] J. K. Ang, “Extending orthogonal and nearly orthogonal Latin hypercube designs for computer simulation and experiments,” Master’s Thesis. Naval Postgraduate School, Monterey, California, 2006.

[34] T. M. Cioppa and T. W. Lucas, “Efficient nearly orthogonal and space-filling Latin hypercubes,” *Technometrics*, vol. 49, no. 1, pp. 45–55, 2007.

[35] A. S. Hernandez, T. W. Lucas, and M. Carlyle, “Constructing nearly orthogonal Latin hypercubes for any nonsaturated run-variable combination,” *ACM Transactions on Modeling and Computer Simulation*, vol. 22, no. 4, pp. 1–17, 2012.

[36] A. D. MacCalman, H. Vieira, and T. Lucas, “Second-order nearly orthogonal Latin hypercubes for exploring stochastic simulations,” *Journal of Simulation*, vol. 11, no. 2, pp. 137–150, 2017.

[37] A. M. Law, *Simulation Modeling and Analysis*, 5th ed. New York: McGraw-Hill, 2015.

[38] T. M. Cioppa, “Efficient Nearly Orthogonal and Space-filling Experimental Designs for High-dimensional Complex Models,” Doctoral Dissertation. Naval Postgraduate School, Monterey, California, 2002.

[39] R. Dorfman, “The detection of defective members of large populations,” *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[40] M. S. Patel, “Group-screening with more than two stages,” *Technometrics*, vol. 4, no. 2, pp. 209–217, 1962.

[41] F. E. Satterthwaite, “Random balance experimentation,” *Technometrics*, vol. 1, no. 2, pp. 111–137, 1959.

[42] J. E. Jacoby and S. Harrison, “Multi-variable experimentation and simulation models,” *Naval Research Logistics Quarterly*, vol. 9, no. 2, pp. 121–136, 1962.

[43] B. Bettonvil, “Factor screening by sequential bifurcation,” *Communications in Statistics - Simulation and Computation*, vol. 24, no. 1, pp. 165–185, 1995.

[44] G. Taguchi, *Introduction to Quality Engineering Designing Quality into Products and Processes*. Japan: Asian Productivity Organization, 1988.

[45] N. Flournoy, “A Clinical Experiment in Bone Marrow Transplantation: Estimating a Percentage Point of a Quantal Response Curve,” in *Case Studies in Bayesian Statistics*, New York: Springer, 1993, pp. 324–336.

[46] I. Verdinelli and K. Chaloner, “Bayesian experimental design : a review,” *Statistical Science*, vol. 10, no. 3, pp. 273–304, 1995.

[47] J. N. Srivastava, “Designs for searching non-negligible effects,” in *A Survey of Statistical Design and Linear Models*, J. N. Srivastava, Ed. North Holland, Amsterdam: Elsevier Science B. V., 1975, pp. 507–519.

[48] K. Chatterjee, L.-Y. Deng, and D. K. J. Lin, “Resolution v.2 search designs,” *Communications in Statistics - Theory and Methods*, vol. 29, no. 5–6, pp. 1143–1154, 2000.

[49] L. W. Schruben, “Simulation optimization using frequency domain methods,” in *Proceedings of the 1986 Winter Simulation Conference*, 1986, pp. 366–369.

[50] D. J. Morrice, “A comparison of frequency domain methodology and conventional factor screening methods,” *Operations Research Letters*, vol. 17, pp. 165–174, 1995.

[51] F. J. Hickernell, “A generalized discrepancy and quadrature error bound,” *Mathematics of Computation*, vol. 67, no. 221, pp. 299–322, 1998.

[52] J. P. C. Kleijnen, S. M. Sanchez, T. W. Lucas, and T. M. Cioppa, “A user’s guide to the brave new world of designing simulation experiments,” *Tilberg University. Center for Economic Research Discussion Paper*, vol. 2003–1, 2003.

[53] IBM Corporation, “IBM ILOG CPLEX Optimization Studio V12.6.1 documentation,” 2014. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html. [Accessed: 28-Sep-2016].

[54] I. M. Sobol’, “On the distribution of points in a cube and the approximate evaluation of integrals,” *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[55] SEED Center for Data Farming, “SEED Center,” 2016. [Online]. Available: http://harvest.nps.edu. [Accessed: 15-Aug-2016].

[56] C. C. Wakeman, “Discrete event simulation modeling and analysis of key leader engagements,” Master’s Thesis. Naval Postgraduate School, Monterey, California, 2012.

[57] D. O. Marlow, S. M. Sanchez, and P. J. Sanchez, “Testing aircraft fleet management policies using designed simulation experiments,” in *MODSIM2015, 21st International Congress on Modelling and Simulation*, 2015.

[58] R. L. Keeney, *Value-focused Thinking: A Path to Creative Decision Making*. Cambridge, Massachusetts: Harvard University Press, 1996.

[59] G. F. Piepel, “Discussion of ‘Response surface design evaluation and comparison’ by C.M. Anderson-Cook, C.M. Borror, and D.C. Montgomery,” *Journal of Statistical Planning and Inference*, vol. 139, no. 2, pp. 653–656, 2009.

[60] H. Guo and A. Mettas, “Design of experiments and data analysis,” in *2010 Reliability and Maintainability Symposium*, 2010.

[61] J. F. Lawless, *Statistical Models and Methods for Lifetime Data*. New York: John Wiley & Sons, 2011.

[62] J. Schmee and G. J. Hahn, “A simple method for regression analysis with censored data,” *Technometrics*, vol. 21, no. 4, pp. 417–432, 1979.

[63] H. Koul, V. Susarla, and J. Van Ryzin, “Regression analysis with randomly right-censored data,” *The Annals of Statistics*, vol. 9, no. 6, pp. 1276–1288, 1981.

[64] F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown, “Algorithm runtime prediction: methods & evaluation,” *Artificial Intelligence*, vol. 206, pp. 79–111, 2014.

[65] E. Biganzoli, P. Boracchi, L. Mariani, and E. Marubini, “Feed forward neural networks for the analysis of censored survival data: A partial logistic regression approach,” *Statistics in Medicine*, vol. 17, no. 10, pp. 1169–1186, 1998.

[66] B. D. Ripley and R. M. Ripley, “Neural networks as statistical methods in survival analysis,” in *Clinical Applications of Artificial Neural Networks*, New York: Cambridge University Press, 2001, pp. 237–255.

[67] Y. Goldberg and M. R. Kosorok, “Support vector regression for right censored data,” *arXiv preprint arXiv:1202.5130*, 2012. [Online]. Available: https://arxiv.org/abs/1202.5130.

[68] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Hoboken, New Jersey: John Wiley & Sons, 2016.

[69] K. Chaloner and K. Larntz, “Optimal Bayesian design applied to logistic regression experiments,” *Journal of Statistical Planning and Inference*, vol. 21, no. 2, pp. 191–208, 1989.

[70] M. Saleh and R. Pan, “Constructing efficient experimental designs for generalized linear models,” *Communications in Statistics - Simulation and Computation*, vol. 45, no. 8, pp. 2827–2845, 2016.

[71] M. Konstantinou, S. Biedermann, and A. Kimber, “Optimal designs for two-parameter nonlinear models with application to survival models,” *Statistica Sinica*, vol. 24, pp. 415–428, 2014.

[72] S. E. Burke, “Optimal Design of Experiments for Dual-Response Systems,” Doctoral Dissertation. Arizona State University, Tempe, Arizona, 2016.

[73] T. Hill and P. Lewicki, *Statistics: Methods and Applications: A Comprehensive Reference for Science, Industry, and Data Mining*. StatSoft, Inc., 2006.

[74] T. J. Mitchell, “Computer construction of ‘D-optimal’ first-order designs,” *Technometrics*, vol. 16, no. 2, pp. 211–220, 1974.

[75] P. Goos and B. Jones, *Optimal Design of Experiments: A Case Study Approach*. New York: John Wiley & Sons, 2011.

[76] L. Lu, C. M. Anderson-Cook, and T. J. Robinson, “Optimization of designed experiments based on multiple criteria utilizing a Pareto frontier,” *Technometrics*, vol. 53, no. 4, pp. 353–365, 2011.

[77] C. M. Anderson-Cook, C. M. Borror, and D. C. Montgomery, “Response surface design evaluation and comparison,” *Journal of Statistical Planning and Inference*, vol. 139, no. 2, pp. 629–641, 2009.

[78] P. A. Parker, “Discussion–‘Response surface design evaluation and comparison’ by Christine M. Anderson-Cook, Connie M. Borror, Douglas C. Montgomery,” *Journal of Statistical Planning and Inference*, vol. 139, no. 2, pp. 645–646, 2009.

[79] P. Goos, “Discussion of ‘Response surface design evaluation and comparison,’” *Journal of Statistical Planning and Inference*, vol. 139, no. 2, pp. 657–659, 2009.

[80] C. M. Anderson-Cook, C. M. Borror, and D. C. Montgomery, “Rejoinder for ‘Response surface design evaluation and comparison,’” *Journal of Statistical Planning and Inference*, vol. 139, no. 2, pp. 671–674, 2009.

[81] L. Lu, C. M. Anderson-Cook, and T. J. Robinson, “A case study to demonstrate a Pareto Frontier for selecting a best response surface design while simultaneously optimizing multiple criteria,” *Applied Stochastic Models in Business and Industry*, vol. 28, no. 3, pp. 206–221, 2012.

[82] G. Derringer and R. Suich, “Simultaneous optimization of several response variables,” *Journal of Quality Technology*, vol. 12, no. 4, pp. 214–219, 1980.

[83] A. Ammeri, W. Hachicha, F. Masmoudi, and H. Chabchoub, “A comprehensive literature classification of simulation optimisation methods,” *International Conference on Multiple Objective Programming and Goal Programming, MOPGP10*, 2010.

[84] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[85] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[86] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. New York: John Wiley & Sons, 1949.

[87] D. J. Fonseca, D. O. Navaresse, and G. P. Moynihan, “Simulation metamodeling through artificial neural networks,” *Engineering Applications of Artificial Intelligence*, vol. 16, no. 3, pp. 177–183, 2003.

[88] G. Matheron, “Principles of geostatistics,” *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[89] N. A. C. Cressie, *Statistics for Spatial Data*. New York: John Wiley & Sons, 1993.

[90] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer simulation experiments,” *Statistical Science*, vol. 4, no. 4, pp. 409–423, 1989.

[91] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. New York: Springer, 2003.

[92] B. Ankenman, B. L. Nelson, and J. Staum, “Stochastic kriging for simulation

metamodeling," *Operations Research*, vol. 58, no. 2, pp. 371–382, 2010.

[93] J. P. C. Kleijnen, *Design and Analysis of Simulation Experiments*. New York: Springer, 2007.

[94] J. P. C. Kleijnen, "Kriging metamodeling in simulation: A review," *European Journal of Operational Research*, vol. 192, pp. 707–716, 2009.

[95] J. Staum, "Better simulation metamodeling: the why, what, and how of stochastic kriging," in *Proceedings of the 2009 Winter Simulation Conference*, 2009, pp. 119–133.

[96] M. H. Kutner, C. Nachtsheim, and J. Neter, *Applied Linear Regression Models*. New York: McGraw-Hill, 2004.

[97] R. R. Barton, "Metamodels for simulation input-output relations," *Proceedings of the 1992 Winter Simulation Conference*, vol. 9, pp. 289–299, 1992.

[98] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–141, 1991.

[99] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, California: Wadsworth & Brooks, 1984.

[100] J. Gareth, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*. New York: Springer, 2013.

[101] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *Journal of Geophysical Research*, vol. 76, no. 8, pp. 1905–1915, 1971.

[102] H. Fang, M. Rais-Rohani, Z. Liu, and M. F. Horstemeyer, "A comparative study of metamodeling methods for multiobjective crashworthiness optimization," *Computers and Structures*, vol. 83, no. 25–26, pp. 2121–2136, 2005.

[103] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.

[104] S. M. Clarke, J. H. Griebsch, and T. W. Simpson, "Analysis of support vector regression for approximation of complex engineering analyses," *Journal of Mechanical Design*, vol. 127, no. 6, pp. 1077–1087, 2005.

[105] A. Ben-Tal, "Characterization of Pareto and Lexicographic Optimal Solutions," in *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen, Königswinter, West Germany, August 20--24, 1979*, G. Fandel and T. Gal, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1980, pp. 1–11.

[106] A. Charnes and W. W. Cooper, *Management Models and Industrial Applications of Linear Programming*. New York: John Wiley & Sons, 1961.

[107] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[108] J. R. Rice, “The algorithm selection problem,” *Advances in Computers*, vol. 15, pp. 65–118, 1976.

[109] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[110] W. Duch and K. Grudzinski, “Meta-learning: searching in the model space,” in *Proceedings of the International Conference on Neural Information Processing*, 2001, pp. 235–240.

[111] C. Köpf, C. Taylor, and J. Keller, “Meta-analysis: from data characterisation for meta-learning to meta-regression,” in *Proceedings of the PKDD Workshop on Data Mining, Decision Support, Meta- Learning and ILP*, 2000.

[112] K. Leyton-Brown, E. Nudelman, and Y. Shoham, “Learning the empirical hardness of optimization problems: The case of combinatorial auctions,” *Eighth International Conference on Principles and Practice of Constraint Programming (CP’02)*, pp. 556–572, 2002.

[113] B. Bischl, P. Kerschke, L. Kotthoff, M. Lindauer, Y. Malitsky, A. Fréchette, H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney, and J. Vanschoren, “ASlib: A benchmark library for algorithm selection,” *Artificial Intelligence*, vol. 237, pp. 41–58, 2016.

[114] C. Lemke, M. Budka, and B. Gabrys, “Metalearning: a survey of trends and technologies,” *Artificial Intelligence Review*, vol. 44, no. 1, pp. 117–130, 2015.

[115] T. A. F. Gomes, R. B. C. Prudêncio, C. Soares, A. L. D. Rossi, and A. Carvalho, “Combining meta-learning and search techniques to select parameters for support vector machines,” *Neurocomputing*, vol. 75, no. 1, pp. 3–13, 2012.

[116] G. Loterman and C. Mues, “Selecting accurate and comprehensible regression algorithms through meta learning,” in *Proceedings - 12th IEEE International Conference on Data Mining Workshops, ICDMW 2012*, 2012, pp. 953–960.

[117] A. L. D. Rossi, A. C. P. de L. F. de Carvalho, C. Soares, and B. F. de Souza, “MetaStream: a meta-learning based method for periodic algorithm selection in time-changing data,” *Neurocomputing*, vol. 127, pp. 52–64, 2014.

[118] X. Wang, K. Smith-Miles, and R. Hyndman, “Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series,” *Neurocomputing*, vol. 72, no. 10, pp. 2581–2594, 2009.

[119] M. Matijaš, J. A. Suykens, and S. Krajcar, “Load forecasting using a multivariate meta-learning system,” *Expert Systems with Applications*, vol. 40, no. 11, pp. 4427–4437, 2013.

[120] M. Kück, S. F. Crone, and M. Freitag, “Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, no. June, pp. 1499–1506.

[121] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu,

"Hyper-heuristics: a survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[122] S. Poursoltan and F. Neumann, "A feature-based prediction model of algorithm selection for constrained continuous optimisation," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 1461–1468.

[123] K. Smith-Miles and L. Lopes, "Measuring instance difficulty for combinatorial optimization problems," *Computers and Operations Research*, vol. 39, no. 5, pp. 875–889, 2012.

[124] M. Feurer, J. T. Springenberg, and F. Hutter, "Initializing Bayesian hyperparameter optimization via meta-learning," in *Proceedings of the 29th Conference on Artificial Intelligence (AAAI 2015)*, 2015, pp. 1128–1135.

[125] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems : A survey on methods and challenges," *Information Sciences*, vol. 317, pp. 224–245, 2015.

[126] M. A. Muñoz and K. A. Smith-Miles, "Performance analysis of continuous black-box optimization algorithms via footprints in instance space," *Evolutionary Computation*, 2016.

[127] B. F. de Souza, C. Soares, and A. C. P. L. F. de Carvalho, "Meta-learning approach to gene expression data classification," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, no. 2, pp. 285–303, 2009.

[128] L. P. F. Garcia, A. C. P. L. F. de Carvalho, and A. C. Lorena, "Noise detection in the meta-learning level," *Neurocomputing*, vol. 176, pp. 14–25, 2016.

[129] C. Romero, J. L. Olmo, and S. Ventura, "A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets," in *Proceedings of the 6th International Conference on Educational Data Mining*, 2013, pp. 268–271.

[130] C. F. Tsai and Y. F. Hsu, "A meta-learning framework for bankruptcy prediction," *Journal of Forecasting*, vol. 32, no. 2, pp. 167–179, 2013.

[131] A. Abbasi, C. Albrecht, A. Vance, and J. Hansen, "Metafraud: a meta-learning framework for detecting financial fraud," *MIS Quarterly*, vol. 36, no. 4, pp. 1293–1327, 2012.

[132] L. Kotthoff, "Algorithm selection for combinatorial search problems: a survey," *AI Magazine*, pp. 48–60, 2014.

[133] D. Bursztyn and D. M. Steinberg, "Comparison of designs for computer experiments," *Journal of Statistical Planning and Inference*, vol. 136, no. 3, pp. 1103–1119, 2006.

[134] A. J. Mason, "OpenSolver - an open source add-in to solve linear and integer progammes in Excel," in *Operations Research Proceedings 2011*, 2012, pp. 401–406.

[135] OpenSolver, "OpenSolver for Excel," 2017. [Online]. Available: https://opensolver.org/. [Accessed: 26-Jan-2018].

[136] R. Jin, W. Chen, and A. Sudjianto, “On sequential sampling for global metamodeling in engineering design,” in *Proceedings of DETC*, 2002, pp. 539–548.

[137] L. Pronzato and W. G. Müller, “Design of computer experiments: space filling and beyond,” *Statistics and Computing*, vol. 22, no. 3, pp. 681–701, 2012.

[138] M. Shamsuzzaman, M. G. Satish, and J. D. Pintér, “Distance correlation-based nearly orthogonal space-filling experimental designs,” *International Journal of Experimental Design and Process Optimisation*, vol. 4, no. 3/4, pp. 216–233, 2015.

[139] R. T. Johnson, D. C. Montgomery, and B. Jones, “An empirical study of the prediction performance of space-filling designs,” *International Journal of Experimental Design and Process Optimisation*, vol. 2, no. 1, pp. 1–18, 2011.

[140] F. A. C. Viana, “Things you wanted to know about the Latin hypercube design and were afraid to ask,” *10th World Congress on Structural and Multidisciplinary Optimization*, 2013.

[141] G. Rennen, B. Husslage, E. R. Van Dam, and D. H. Dick, “Nested maximin Latin hypercube designs,” *Structural and Multidisciplinary Optimization*, vol. 41, no. 3, pp. 371–395, 2010.

[142] K. Crombecq, E. Laermans, and T. Dhaene, “Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling,” *European Journal of Operational Research*, vol. 214, no. 3, pp. 683–696, 2011.

[143] F. Xiong, Y. Xiong, W. Chen, and S. Yang, “Optimizing Latin hypercube design for sequential sampling of computer experiments,” *Engineering Optimization*, vol. 41, no. 8, pp. 793–810, 2009.

[144] S. Golchi and J. L. Loeppky, “Space Filling Designs for Constrained Domains,” 2015. [Online]. Available: http://arxiv.org/abs/1512.07328.

[145] J. L. Loeppky, L. M. Moore, and B. J. Williams, “Batch sequential designs for computer experiments,” *Journal of Statistical Planning and Inference*, vol. 140, no. 6, pp. 1452–1464, 2010.

[146] W. Duan, B. E. Ankenman, P. J. Sanchez, and S. M. Sanchez, “Sliced full factorial-based Latin hypercube designs as a framework for a batch sequential design algorithm,” *Technometrics*, vol. 59, no. 1, pp. 11–22, 2017.

[147] K. Kennedy, “Bridging the Gap Between Space-Filling and Optimal Designs. Design for Computer Experiments,” Doctoral Dissertation. Arizona State University, Tempe, Arizona, 2013.

[148] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[149] J. J. Liang, B. Y. Qu, and P. N. Suganthan, “Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization,” Technical Report. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University,

Singapore, 2013.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
22-03-2018	Doctoral Dissertation	Oct 2014 – Mar 2018

4. TITLE AND SUBTITLE

Experimental Designs, Meta-modeling, and Meta-learning for Mixed-Factor Systems with Large Decision Spaces

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Little, Zachary C., Mr.

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-ENS-DS-18-M-137

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

Air Force Simulation and Analysis Facility
ATTN: Timothy Menke
2303 8th Street
Wright-Patterson AFB OH 45433
(937) 938-3772, Timothy.Menke@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

SIMAF (AFLCMC/XZS)

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A:
Approved For Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Many Air Force studies require a design and analysis process that can accommodate for the computational challenges associated with complex systems, simulations, and real-world decisions. For systems with large decision spaces and a mixture of continuous, discrete, and categorical factors, nearly orthogonal-and-balanced (NOAB) designs can be used as efficient, representative subsets of all possible design points for system evaluation, where meta-models are then fitted to act as surrogates to system outputs. The mixed-integer linear programming (MILP) formulations used to construct first-order NOAB designs are extended to solve for low correlation between second-order model terms (i.e., two-way interactions and quadratics). The resulting second-order approaches are shown to improve design performance measures for second-order model parameter estimation and prediction variance as well as for protection from bias due to model misspecification with respect to second-order terms. Further extensions are developed to construct batch sequential NOAB designs, giving experimenters more flexibility by creating multiple stages of design points using different NOAB approaches, where simultaneous construction of stages is shown to outperform design augmentation overall. To reduce cost and add analytical rigor, meta-learning frameworks are developed for accurate and efficient selection of first-order NOAB designs as well as of meta-models that approximate mixed-factor systems.

15. SUBJECT TERMS

Design of experiments, nearly orthogonal-and-balanced, mixed-integer linear programming, pairwise correlation, meta-model

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Dr. Jeffery D. Weir, AFIT/ENS
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	178	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			(937) 255-3636 x4523 jeffery.weir@afit.edu