2022-06

EXTENDING AND IMPROVING DESIGNS FOR LARGE-SCALE COMPUTER EXPERIMENTS

Parker, Jeffrey D., Jr.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

EXTENDING AND IMPROVING DESIGNS FOR LARGE-SCALE COMPUTER EXPERIMENTS

by

Jeffrey D. Parker Jr.

June 2022

Dissertation Supervisors: Thomas W. Lucas
W. Matthew Carlyle

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2022	**3. REPORT TYPE AND DATES COVERED** Dissertation

4. TITLE AND SUBTITLE EXTENDING AND IMPROVING DESIGNS FOR LARGE-SCALE COMPUTER EXPERIMENTS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jeffrey D. Parker Jr.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

This research develops methods that increase the inventory of space-filling designs (SFDs) for large-scale computer-based experiments. We present a technique enabling researchers to add sequential blocks of design points effectively and efficiently to existing SFDs. We accomplish this through a quadratically constrained mixed-integer program that augments cataloged or computationally expensive designs by optimally permuting and stacking columns of an initial base design to minimize the maximum absolute pairwise correlation among columns in the new extended design. We extend many classes of SFDs to dimensions that are currently not easily obtainable. Adding new design points provides more degrees of freedom for building metamodels and assessing fit. The resulting extended designs have better correlation and space-filling properties than the original base designs and compare well with other types of SFDs created from scratch in the extended design space. In addition, through massive computer-based experimentation, we compare popular software packages for generating SFDs and provide insight into the methods and relationships among design measures of correlation and space-fillingness. These results provide experimenters with a broad understanding of SFD software packages, algorithms, and optimality criteria. Further, we provide a probability-distribution model for the maximum absolute pairwise correlation among columns in the widely used maximin Latin hypercube designs.

14. SUBJECT TERMS computer experiments, design of experiments, mixed integer programming, orthogonality, optimization, simulation, space-filling designs	**15. NUMBER OF PAGES** 277
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

EXTENDING AND IMPROVING DESIGNS FOR LARGE-SCALE COMPUTER EXPERIMENTS

Jeffrey D. Parker Jr.
Lieutenant Colonel, United States Marine Corps
BS, United States Naval Academy, 2006
MS, Operations Research, Naval Postgraduate School, 2015

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2022

Approved by:

Thomas W. Lucas
Department of
Operations Research
Dissertation Chair,
Dissertation Supervisor

W. Matthew Carlyle
Department of
Operations Research
Dissertation Supervisor

Raymond R. Buettner
Department of
Information Sciences

Alejandro S. Hernandez
Department of Systems
Engineering

W. David Kelton
Department of
Operations Research

Approved by:

W. Matthew Carlyle
Chair, Department of Operations Research

Michael E. Freeman
Vice Provost of Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This research develops methods that increase the inventory of space-filling designs (SFDs) for large-scale computer-based experiments. We present a technique enabling researchers to add sequential blocks of design points effectively and efficiently to existing SFDs. We accomplish this through a quadratically constrained mixed-integer program that augments cataloged or computationally expensive designs by optimally permuting and stacking columns of an initial base design to minimize the maximum absolute pairwise correlation among columns in the new extended design. We extend many classes of SFDs to dimensions that are currently not easily obtainable. Adding new design points provides more degrees of freedom for building metamodels and assessing fit. The resulting extended designs have better correlation and space-filling properties than the original base designs and compare well with other types of SFDs created from scratch in the extended design space. In addition, through massive computer-based experimentation, we compare popular software packages for generating SFDs and provide insight into the methods and relationships among design measures of correlation and space-fillingness. These results provide experimenters with a broad understanding of SFD software packages, algorithms, and optimality criteria. Further, we provide a probability-distribution model for the maximum absolute pairwise correlation among columns in the widely used maximin Latin hypercube designs.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

2^k	Full Factorial Design with Two Levels per Factor
CDF	Cumulative Distribution Function
CPU	Central Processing Unit
DoD	Department of Defense
DOE	Design of Experiments
GB	Gigabyte
GEV	Generalized Extreme Value
GHz	Gigahertz
JMP	"John's Macintosh Project" Statistical Package from the SAS Institute
JSL	JMP's Scripting Language
LH	Latin Hypercube
LHD	Latin Hypercube Design
LHS	Latin Hypercube Sampling
MaxPro	Maximum Projection
MaxProLHD	Maximum Projection Latin Hypercube Design
MIP	Mixed Integer Program
Mm	Maximin
mM	Minimax
MmLHD	Maximin Latin Hypercube Design
NOLH	Nearly Orthogonal Latin Hypercube
NOLHD	Nearly Orthogonal Latin Hypercube Design
PYOMO	Python Optimization Modeling Objects
RAM	Random Access Memory
RLH	Random Latin Hypercube
SFD	Space-Filling Design
SFM	Space-Filling Measure
UD	Uniform Design

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The mission of the Department of Defense (DOD) is "to provide the military forces needed to deter war and ensure our nation's security" (Department of Defense 2021). To do this, the DOD makes choices regarding the composition of our armed forces and how they should fight, if necessary. Since real-world experimentation is often infeasible or prohibitively expensive, the DOD's decision-making process is often underpinned by computer experiments using complex models. To enhance the information provided to decision makers through computer experimentation, this research develops methods that increase the inventory of effective and efficient experimental designs developed for complex, high-dimensional computer models.

Before computer models are run, users specify the inputs to the model and the responses they will collect. The inputs are made via an $n \times k$ design matrix $\mathbf{X}$, which stipulates the computer model's input settings for n experiments involving k factors (i.e., input variables). The insights one can glean depend critically on the properties of $\mathbf{X}$. Experimenters seek designs that (1) have zero or minimal correlations among columns in $\mathbf{X}$ "to allow independent assessment of the effects of the different inputs" (Moon et al. 2012, p. 378) and improve the performance of many statistical methods; (2) are space-filling (i.e., sample throughout the input space); (3) enable the fitting of a variety of diverse metamodels; and (4) minimize metamodel bias and allow us to detect it when it occurs (Santner et al. 2018). Many of these properties can be attained with special-purpose space-filling designs (SFDs); therefore, they are extensively used in computer experimentation. Algorithms that construct SFDs are typically computationally burdensome, contain randomness in their construction, or require commercial licenses. Thus, researchers often catalog designs to make them readily available for widespread use.

This dissertation introduces a new technique, which we name `permute_and_stack`, which enables experimenters to leverage existing SFDs to add sequential batches of design points efficiently. We accomplish this through a quadratically constrained mixed integer program that augments cataloged or computationally expensive SFDs by optimally permuting and stacking columns of an initial base design matrix ($\mathbf{X}^0$)

to minimize the maximum absolute pairwise correlation (ρ_{map}) among columns in the new extended design. Lower ρ_{map} values are preferred, with $\rho_{map} = 0$ indicating column orthogonality. The extended designs, with additional design points, provide more degrees of freedom for building metamodels and assessing fit.

The idea of extending designs by permuting and stacking the columns of $\mathbf{X}^0$ is not original. A straightforward approach, known as shift-and-stack, is a simple algorithm that extends designs by shifting each of the columns in $\mathbf{X}^0$ over by one (forward or backward) and appends the shifted matrix to $\mathbf{X}^0$ (Sanchez and Sanchez 2019). The heuristic shift-and-stack can be applied iteratively to provide a series of extended designs. This research began by investigating if using heuristics to reorder the columns of $\mathbf{X}^0$ improves the resultant designs from shift-and-stack when applied to cataloged nearly orthogonal Latin hypercubes (NOLHs) (Cioppa and Lucas 2007). There are over 10^{76} possible ten-stack designs ($(11!)^{10}$) when the base design $\mathbf{X}^0$ is the cataloged 33×11 NOLH. Figure ES1 shows how much better the `permute_and_stack` optimization reduces ρ_{map} values than various column-reordering heuristics using shift-and-stack.

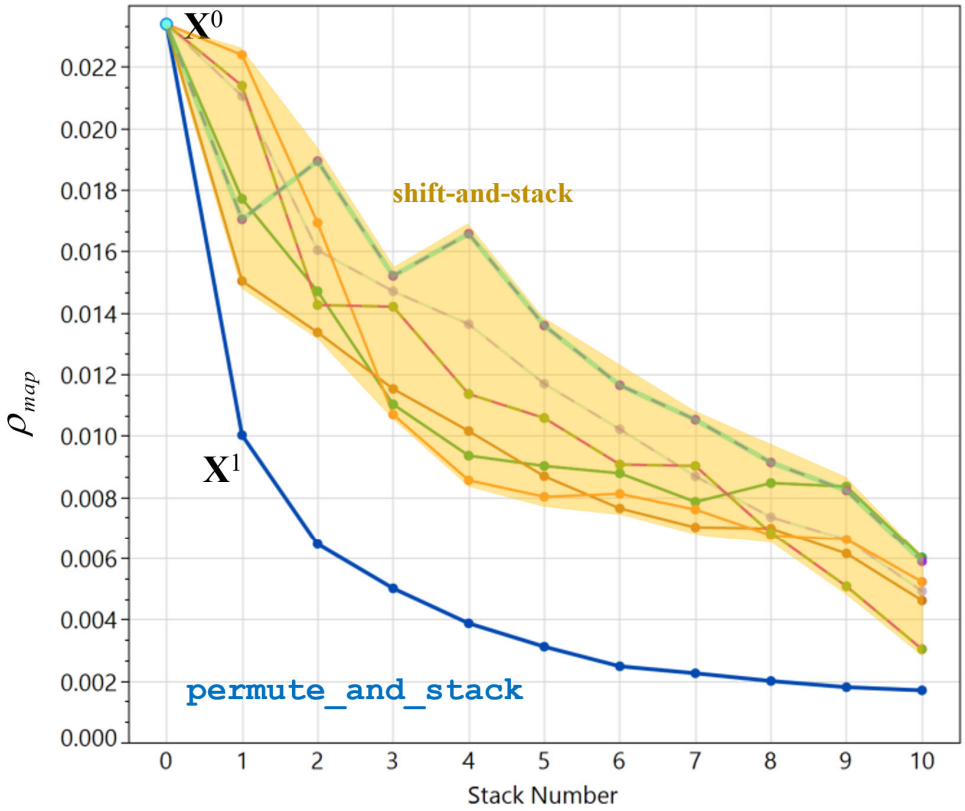


Figure ES1. The 33×11 NOLH design $\mathbf{X}^0$ and 110 extended designs using `permute_and_stack` (blue) and ten applications of shift-and-stack (light colors) using column reordering heuristics. $\mathbf{X}^1$ is the extended design after one application of `permute_and_stack`.

To assess the potential of `permute_and_stack`, this research explored how `permute_and_stack` performs in reducing ρ_{map} when sequentially adding batches (or stacks) of n design points (DPs) for several cutting-edge classes of cataloged and computer-generated SFDs. The reductions in ρ_{map} are contrasted with values obtained by applying shift-and-stack to the base designs using several correlation-based column-reordering heuristics as well as random permutations of the columns. Where feasible, we also evaluate `permute_and_stack` against newly generated SFDs in the extended design space. In all cases, `permute_and_stack` yields better designs in terms of ρ_{map} than do the alternatives. However, new designs generated in the extended space may produce better designs by some of the space-filling measures. For all classes and design dimensions

explored, the set of multi-objective (i.e., ρ_{map} and a measure of space-fillingness) non-dominated designs includes `permute_and_stack` designs.

In evaluating `permute_and_stack`'s performance, we generated and assessed many measures of design quality in nearly a million SFDs of numerous types and sizes. A wide-ranging review of the literature has found nothing as comprehensive in quantifying and displaying the relationships (or lack thereof) for so many measures in multiple SFD classes and sizes as is contained in this dissertation. This directly addresses the knowledge gap identified by Jin et al. (2003, p. 554) that "[w]hile there are many optimality criteria available in the literature, the comparison between different design criteria is certainly one of the most important problems in the field of design of computer experiments and deserves a thorough future investigation."

Since almost all of the algorithms that build SFDs have stochastic elements, we provide summary statistics or plots of the distributions of ten measures of design quality for several classes and sizes of SFDs. Results include the observation that across the design classes, for many measures, design-class preference can change with n and k. Moreover, high variances and outliers in the plots reveal the risks associated with generating one "optimal" design with the stochastic software used to construct SFDs. Researchers need to generate many SFDs to ensure they obtain a design with good properties for their application.

The ultimate goal of this research is to provide experimenters with good designs. Toward that end, this research provides empirical guidance on the minimum ρ_{map} a researcher can expect to attain when generating G independent SFDs for numerous classes and multiple sizes. We also explore the asymptotic distribution of the minimum ρ_{map} as G increases in Latin hypercube designs (LHDs), as well as LHDs that seek to maximize the minimum (Mm) distance between any two design points (i.e., MmLHDs) for designs with $k = 20$ and $n = 200$. These may be the two most widely used types of SFDs. The results suggest that asymptotically, as G gets larger, the minimum ρ_{map} from the G designs converges to a Weibull distribution. This research only scratches the surface of what is

possible. The analysis above can be applied to any measure of design characteristics for any class of SFD.

References

Cioppa TM, Lucas TW (2007) Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* 49(1):45–55.

DoD (2021) The Department of Defense releases the president's fiscal year 2022 defense budget. *US Dep. Def.* Retrieved (August 12, 2021), https://www.defense.gov/Newsroom/Releases/Release/Article/2638711/the-department-of-defense-releases-the-presidents-fiscal-year-2022-defense-budg/.

Jin R, Chen W, Sudjianto A (2003) An efficient algorithm for constructing optimal design of computer experiments. *Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.* 545–554.

Moon H, Dean AM, Santner TJ (2012) Two-stage sensitivity-based group screening in computer experiments. *Technometrics* 54(4):376–387.

Sanchez PJ, Sanchez SM (2019) Orthogonal second-order space-filling designs with insights from simulation experiments to support test planning. *Qual. Reliab. Eng. Int.* 35(3):854–867.

Santner TJ, Williams BJ, Notz WI (2018) *The Design and Analysis of Computer Experiments*, 2nd ed. (Springer, New York).

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

To begin, I want to thank those in the Marine Corps leadership who took a risk and afforded me the opportunity to engage in this research.

Next, I must express my most profound admiration and indebtedness to the faculty of the Operations Research Department for their invaluable instruction. My deepest gratitude goes to my advisors, Professors Matthew Carlyle and Thomas Lucas, for guiding me through this dissertation. I also want to thank Professors Raymond Buettner, Alejandro Hernandez, and David Kelton for being on my advisory committee and for their expert direction and recommendations. The patience, assistance, and guidance all of you offered were crucial in developing this research. I feel blessed to have spent so much quality time with so many brilliant and distinguished professionals; and I am humbled by your dedication to serving your students.

I thank my wife, Christina, for her boundless patience and constant love during this journey. I would not be where I am today without your unwavering support. I would also like to offer my eternal gratitude to my parents, sister, and in-laws for sacrificing to help our family—especially given the geographic challenges! Thanks to Sandy and Paul Terpeluk, I was able to focus, and my kiddos spent months filled with love and memories on Gobbler Hill Farm despite my absence. Lastly, I send my love to my children—and as my five-year-old son, Paul, would say, “I love you to God and back.”

Thank you all very much!

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THE DEPARTMENT OF DEFENSE RELIES ON COMPUTATIONAL MODELS

> Our mission is to provide the military forces needed to deter war and ensure our nation's security.
>
> —Department of Defense (2021)

To accomplish the Department of Defense's (DoD's) mission during an environment of increasing great power competition, the DoD has 2.146 million military personnel operating in over 160 countries and an annual budget of $715 billion—including $112 billion (the largest ever) for research, development, test, and evaluation (RDT&E) and $14.7 billion for science and technology (DoD 2021). Large-scale computational experimentation for knowledge acquisition spans the DoD enterprise. Senior leaders in the DoD need relevant information quickly and accurately to best use these enormous resources to create capabilities that enable it to meet its mission. Therefore, DoD employs many thousands of research analysts to study thousands of variables on how U.S. forces should be configured, maintained, and, if necessary, fight. Where possible, and because physical experimentation is often prohibitively expensive or infeasible, the analyses are frequently underpinned by computer experimentation (Morgan et al. 2018).

Attempting to model something as complex as warfare is a daunting challenge. In live combat, numerous diverse troops and platforms employ a variety of tactics, techniques, and procedures (TTPs) to meet mission objectives. The DoD strives to model potential conflict virtually with numerous simulated entities interacting over time and space in complex, nonlinear, and stochastic ways. The number of input variables in some DoD models is measured in the tens of thousands and beyond (Saeger and Hinch 2001). This modeling-and-analysis challenge is compounded by high levels of uncertainty inherent in warfare. Moreover, the run times associated with some of these models can be burdensome—sometimes measured in hours or even days (Lukemire et al. 2021, Morgan et al. 2018, Sanchez et al. 2012).

Efficiently obtaining timely information from complex high-dimensional models with long run times requires well-designed experiments to get the most information and best precision out of a fixed run-time budget (or, equivalently to get the required information at the required precision within the shortest run times). This research increases the options available to computational experimenters by extending the inventory of space-filling designs (SFDs). During this study, hundreds of thousands of SFDs were generated and evaluated using a variety of algorithms, design dimensions, and measures of design quality. The results provide a broad understanding of SFD software packages, algorithms, and optimality criteria. Together, the expanded inventory of SFDs, and an improved understanding of their quality measures, provide DoD analysts and other computational experimenters with additional tools for use in supporting decision makers.

B. HIGH-DIMENSIONAL COMPUTATIONAL EXPERIMENTATION

> Many applications involve a large number of input variables. As such, finding space-filling designs with a limited number of design points that provide a good coverage of the entire high dimensional input space is a hopeless undertaking.
>
> —Lin and Tang (2015)

As computers become ever more powerful and affordable, scientists, engineers, governments, and businesses increasingly use intricate computational models to better understand complex systems and phenomena. Computer models often contain large numbers of input variables, nonlinear relationships, numerous and diverse response variables, and stochastic (or pseudo-random) elements for the purpose of capturing real-world complexities and uncertainties. Learning efficiently from such models is facilitated by sophisticated design of experiments (DOE) created specifically for computer models; see Santner et al. (2018) and Sanchez et al. (2020).

Over the past few decades, dramatic advances in DOE have been developed specifically for high-dimensional computer models (Dean et al. 2015, Fang et al. 2006, Santner et al. 2018). Some of these sophisticated designs are computationally expensive to create, contain randomness in their construction, or require commercial licenses to generate or use (e.g., Fang et al. 2000a, 2000b, Joseph et al. 2015, Joseph and Hung 2008,

MacCalman et al. 2017, Morris and Mitchell 1995, Wang et al. 2021, Xiao and Xu 2018, Zhou et al. 2019).

Many computer-generated designs are cataloged and made available online to make these efficient designs readily available for widespread use. For example, https://spacefillingdesigns.nl/ contains a collection of L_2-maximin (Mm) space-filling Latin hypercube designs (LHDs) for between 11 and 20 factors and up to seven design points (van Dam et al. 2009). Fang et al. (2000b) provide nearly 1,000 space-filling uniform designs (UDs) for up to 29 factors and 30 design points (https://www.math.hkbu.edu.hk/UniformDesign/). The SEED Center for Data Farming (https://harvest.nps.edu) provides scores of nearly orthogonal and space-filling LHDs in easy-to-use spreadsheets. However, these cataloged designs provide experimenters with only a finite number of design options. Moreover, long processing times associated with some design-generating software may also limit choices.

This research uses mixed-integer programming (MIP) to construct new designs from existing designs by adding blocks of new design points sequentially that (nearly) minimize the maximum absolute pairwise correlation among columns in the new extended designs. This limits the worst-case aliasing when fitting a first-order model (Steinberg and Lin 2006). This investigation also empirically explores variability in space-filling measures (SFMs) for space-filling designs (SFDs) created by the SAS' JMP (SAS Institute 2018) software and popular R packages and examines relationships between types of SFDs and their SFMs—and proposes simple methods to improve the quality of the designs they create.

C. RESEARCH QUESTIONS

Large-scale computer experimentation is widely used in the DoD. While catalogs of SFDs or software to create them are available, they may not cover all the design dimensions needed, especially for a sequential "model-test-model" approach (McCool et al. 1995). Moreover, many algorithms for generating SFDs have randomness associated with them. Thus, the quality of the design and subsequent analyses depend on chance. By understanding and accommodating for stochasticity in design generation, we can improve

the quality of the design used and insights gleaned. This research uses mixed-integer programming (MIP) and the analysis of space-filling and correlation measures from hundreds of thousands of SFDs of multiple types and sizes to gain insights into the following research questions.

- In the context of batch sequential designs, what are the relative merits of column reordering heuristics applied to a base design to improve design quality?
- How do measures of design quality change with additional blocks of sequential design points?
- For many classes of SFDs (cataloged or generated), can optimization techniques provide us with an extended design with better correlation or space-filling properties than can existing heuristics?
- For many classes of SFDs (cataloged or generated), can optimization techniques provide us with an extended design that has better correlation properties than an SFD generated from the beginning in the dimension of the extended design?
- What is the variability in SFMs when using popular software packages to construct SFDs?
- What are the relationships among space-filling and correlation measures for various classes and sizes of SFDs?
- Which design-construction methods and parameter settings efficiently create high-quality SFDs?

D. ORGANIZATION OF THE DISSERTATION

This dissertation flows as follows. Chapter II provides a literature review and discusses the experimental setting, metamodeling, and the large body of research on SFDs

and the criteria for assessing them. Chapter III contains the motivation, mathematical model, and a new algorithm that batch sequentially augments existing SFDs by optimally permuting and stacking columns of the design matrix to minimize the maximum absolute pairwise correlation (ρ_{map}) among columns in the new extended design. Chapter IV shows the outcome of applying the new algorithm to sequentially extend several classes of SFDs. Chapter V explores variability and relationships among different space-filling and correlation measures, spanning many space-filling design classes and sizes for hundreds of thousands of designs constructed using JMP and R software packages. Chapter VI fits a distribution model that describes the maximum absolute pairwise correlation among columns of Latin hypercube designs constructed using the maximin-distance criterion. Chapter VII contains conclusions, recommendations, and suggestions for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND AND LITERATURE REVIEW

> The most popular experimental designs for computer experiments are Latin hypercube designs (LHDs, McKay et al. 1979), which has uniform one-dimensional projections and avoids replications on every dimension. According to practical needs, there are various types of efficient LHDs (aka. optimal LHDs), including space-filling LHDs, maximum projection LHDs and orthogonal LHDs.
>
> —Wang et al. (2021), p. 1

This chapter introduces the experimental setting, discusses metamodels, defines key terms and notation, presents measures used to assess space-filling designs (SFDs), and reviews relevant research designing high-dimensional computer experiments.

A. EXPERIMENTAL SETTING

This research addresses situations involving computer experiments in which users select the input values for experimental investigations over a rectangular region. The $n \times k$ *design matrix*, $\mathbf{X}$, specifies the computer model's input settings for n experiments involving k continuous factors (i.e., input variables). Row i of $\mathbf{X}$, denoted x_i, is called a *design point* (DP). DP x_i specifies the settings for each of the k factors for which the computational model will be run, for $i = 1,\ldots, n$. The c^{th} column of $\mathbf{X}$, which we label as X_c, provides the settings for factor c over the n DPs. x_{ic} specifies the value in DP i for factor c. Furthermore, we always scale column input such that $x_i \in [0,1]^k$, with χ representing the k-dimensional unit cube that comprises the experimental region. Finally, let y_i be the output or response from experiments run at the i^{th} DP x_i.

B. METAMODELS

To obtain insight from computer models, researchers often build "cheap surrogate[s]" or metamodels of the responses as functions of the input variables based on the results of computational experiments (Law and Kelton 2007, Sacks et al. 1989). The metamodel can be used to (1) build an understanding of the relationships among factors and responses of the computer model, (2) predict future outcomes, or (3) design future

experiments. When the responses are reasonably smooth, readily interpretable polynomial metamodels can effectively identify and quantify the most important factors and interactions (Kleijnen 2015, Sanchez et al. 2020). For more complicated responses, analytically flexible Gaussian process (GP) metamodels are frequently preferred to approximate computational models and make predictions (Fang et al. 2006 and Santner et al. 2018).

The types and accuracy of metamodels that can be fit depend critically on the design matrix, **X**. For example, if factor *c* has only two levels, nonlinear effects are unidentifiable. For situations with little *a priori* knowledge of the forms of possibly multiple diverse responses, we desire designs that "allow one to fit a variety of models and should provide information about all portions of the experimental region" (Santner et al. 2018, p. 148). We also prefer designs that minimize metamodel bias and allow us to detect it when it occurs. In addition, we favor column-orthogonal designs since multicollinearity among the columns of **X** hinders many statistical procedures (Ryan 2007). A careful choice of **X** is required to address these design objectives. Special-purpose SFDs often possess desirable design attributes (Sanchez et al. 2012). Therefore, SFDs are widely used by practitioners conducting high-dimensional computer experiments.

This research focuses on situations in which experimenters seek the ability to fit many wide-ranging metamodels due to a dearth of prior information on the types of responses that may be generated. If, on the other hand, there is a single response of interest and the form of the metamodel is known, one can often construct **X** to optimize a particular quality metric of the resultant design. The resulting **X**'s are typically referred to as *optimal designs* or *alphabetic optimal designs*; see, Atkinson et al. (2007), Guttorp and Lindgren (2009), Kiefer (1959), Pukelsheim (2006). For example, for so-called *I-optimal* designs, **X** is constructed to minimize the average of the variance of the predicted response. Santner et al. (2018) caution that *model bia*s occurs when the "assumed model is incorrect" (p. 147). Model bias may result in erroneous conclusions.

C. MEASURES OF DESIGN CHARACTERISTICS

> Which experiment setup should we choose? Although the question is straightforward, the answer may not always be easy, as different criteria have been developed to compare competing designs…
>
> —Hinkelmann and Kempthorne (2008), p. 59

This section discusses commonly used measures of design characteristics for large-scale computational experimentation. These measures can be grouped into two broad categories: correlation-based criteria and space-filling measures (SFMs). This section is organized as follows: a discussion on what constitutes a good design, an overview of correlation-based measures, an explanation of the concept of a design's *space-fillingness*, and an introduction to measures of space-fillingness.

1. What Constitutes a Good Design?

In the context of computer experimentation, the relationships between multiple responses and the factors are often unknown, complex, and diverse. Given that we seek designs for a breadth of varied metamodels, we primarily evaluate designs based on properties of **X**. Indeed, Lucas et al. (2015) argue that there are risks to successfully addressing real problems when "model assumptions and structure" (p. 297) are introduced at the expense of realism. Extending ideas from Sacks et al. (1989) and Santner et al. (2003, 2018), Moon et al. (2012, p. 378) explain that we prefer designs that (1) have zero or minimal correlations among columns in **X** "to allow independent [or nearly independent] assessments of the [main] effects of the different inputs" and (2) are space-filling "to [e]nsure all regions of the input space are explored."

2. Correlation-Based Measures

Measures of correlation have been used in evaluating designs since the science of design of experiments began (Box 1978; Fisher 1926, 1971; Montgomery 2013). Minimal correlations among the columns in **X** provide uncorrelated estimates of the coefficients (i.e., avoids confounding) for a main effects metamodel, increases the precision of effects estimates, and enhances the performance of many statistical procedures (Kim and Loh

2003, Ryan 2007). This subsection introduces the measures we use to quantify correlations in candidate designs. The correlation between columns X_c and X_d of **X** is given by Equation (1):

$$\rho_{c,d} = \frac{\sum_{i=1}^{n}\left[(x_{ic} - \bar{X}_c)(x_{id} - \bar{X}_d)\right]}{\sqrt{\left[\sum_{i=1}^{n}(x_{ic} - \bar{X}_c)^2\right]\left[\sum_{i=1}^{n}(x_{id} - \bar{X}_d)^2\right]}} \quad . \tag{1}$$

Here $\bar{X}_c$ and $\bar{X}_d$ are the means of columns *c* and *d*, respectively. We express the maximum absolute pairwise (*map*) correlation as

$$\rho_{map} = \max\left\{|\rho_{cd}|,\ c \neq d\right\}. \tag{2}$$

The value ρ_{map} quantifies how close to orthogonal the columns of **X** are, with ρ_{map} = 0 indicating column orthogonality. An objective of many design construction algorithms is to minimize ρ_{map} given a set of constraints (Cioppa and Lucas 2007). Kleijnen (2015, p. 47) explains, "zero correlation…simpli[fies] the statistical analysis." By minimizing ρ_{map}, we bound the worst-case pairwise correlation between input factors. A design with $\rho_{map} \leq 0.05$ is called *nearly orthogonal* (NO); see Hernandez (2008) and Hernandez et al. (2012a). Others, such as Owen (1994) and Sun et al. (2019), use the average squared or average absolute correlation as their measure of a design's correlation. For a design with *k* columns, let $\overline{|\rho|} = \sum_{c=1}^{k-1}\sum_{d=c+1}^{k}\left|\rho_{c,d}\right| / \left[k(k-1)\right]$ be the average of absolute correlations between columns in **X** (Sun et al. 2019, Wang et al. 2021) and let $\overline{\rho^2} = \sum_{c=1}^{k-1}\sum_{d=c+1}^{k}\rho_{c,d}^2 / \left[k(k-1)\right]$ (Owen 1994) be the average squared correlation between the columns. As with ρ_{map}, lower values of $\overline{|\rho|}$ and $\overline{\rho^2}$ are preferred and **X** is orthogonal when they are equal to zero. Researchers who have developed space-filling designs that seek to minimize correlations include: Cioppa and Lucas (2007), Dean et al. (2015), Joseph and Hung (2008), Little et al. (2019), MacCalman (2013), Sanchez and Sanchez (2019), Vieira et al. (2013), and Wang et al. (2020).

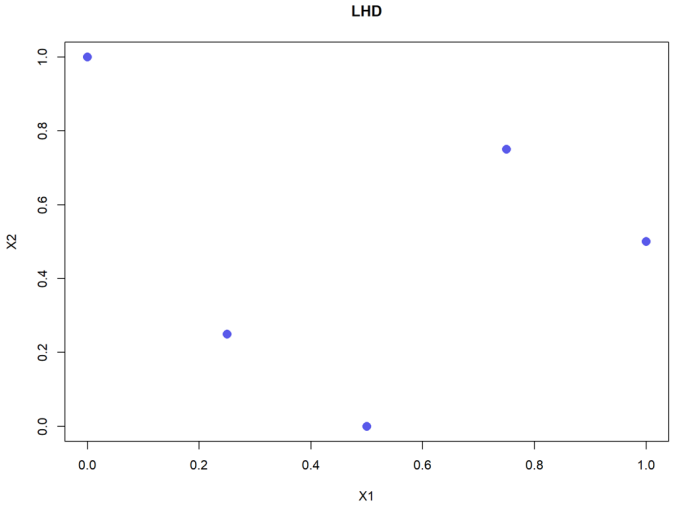
For designs with many factors, a low average absolute or average squared correlation does not guarantee that all pairwise correlations are small. For example, Ye (2021) provides a 33×11 Latin hypercube design (LHD) with an $\overline{|\rho|}$ of 0.015, but a ρ_{map} of 0.417. Because we do not want even one highly correlated pair of columns in **X**, we chose ρ_{map} as our preferred measure of correlation. Other measures, such as $\overline{|\rho|}$, for example, can allow one bad correlation to be masked by a low average value (Savage 2002). One formal justification for this worst-case objective as a general evaluation criterion (as opposed to a seemingly more comprehensive metric, e.g., average- or median-case approach) is that ρ_{map} ensures a low aliasing value (Steinberg and Lin 2006) across the design matrix. For example, consider two designs that we denote **A** and **B**, each having the same $\overline{|\rho|}$ or $\overline{\rho^2}$ values. On the one hand, the alias matrix of **A**, for fitting a first-order model; i.e., the columns 1, 2,…, $k-1$, k, has correlations between each pair of its columns less than 0.05. On the other hand, the alias matrix of **B**, for fitting the same model, has orthogonal columns for all but two columns, which are perfectly correlated (confounded); i.e., $\rho = \pm 1.0$. A concern with a seemingly more comprehensive metric is that it can change with size k.

3. What Makes a Design Space-Filling?

There are several qualitative descriptions of space-filling. Joseph (2016, p. 29) writes that an SFD has design "points everywhere in the experimental region with as few gaps or holes as possible." Santner et al. (2018, p. 149) state that space-filling intuitively means "evenly spread," with the design points (DPs) becoming "increasingly dense" in the experimental region as more samples are taken. It follows that an SFD is more likely to identify interesting, localized effects.

We will extensively use scatter plots to visualize space-fillingness. That is, are there "gaps or holes" in the design space and are the design points (DPs) "evenly spread"? Figure 1 shows the DPs in a 5×2 design matrix and the corresponding design matrix, **X**. Each blue dot represents a coordinate pair (i.e., DP) in the two-dimensional χ. Each DP corresponds to an input combination of the two factors where the computer model will be run. This

design is a 5×2 Latin hypercube design (LHD) (McKay et al. 1979) generated using the *DiceDesign* (Dupuy et al. 2015) R software package. LHDs and other design types and generating software are discussed later in this chapter.



DP	X_1	X_2
1	0.75	0.75
2	0.00	1.00
3	1.00	0.50
4	0.50	0.00
5	0.25	0.25

Figure 1. An example of a 5×2 LHD (McKay et al. 1979) generated using the *DiceDesign* (Dupuy et al. 2015) R software package. Each blue dot represents a point at which the computer model will be run.

As in Figure 1, for a design with two dimensions, we can visualize the entire design space. For $k > 2$, we will use scatter plot matrices to visualize the projections into all of the $\binom{k}{2}$ two-dimensional subspaces defined by all pairs of factors. Figure 2 shows side-by-side pairwise plots for two 16×4 designs of different types. In this case, there are six two-dimensional subspaces. Each pairwise plot has two subpanels (i.e., scatter plots) for each pair of factors, with the factors identified along the diagonal. The left plot is an LHD (McKay et al. 1979), and the right plot is a 2^k factorial design (Box and Wilson 1951). The *DiceDesign* (Dupuy et al. 2015) and *AlgDesign* (Wheeler 2019) R software packages generated these designs, respectively.

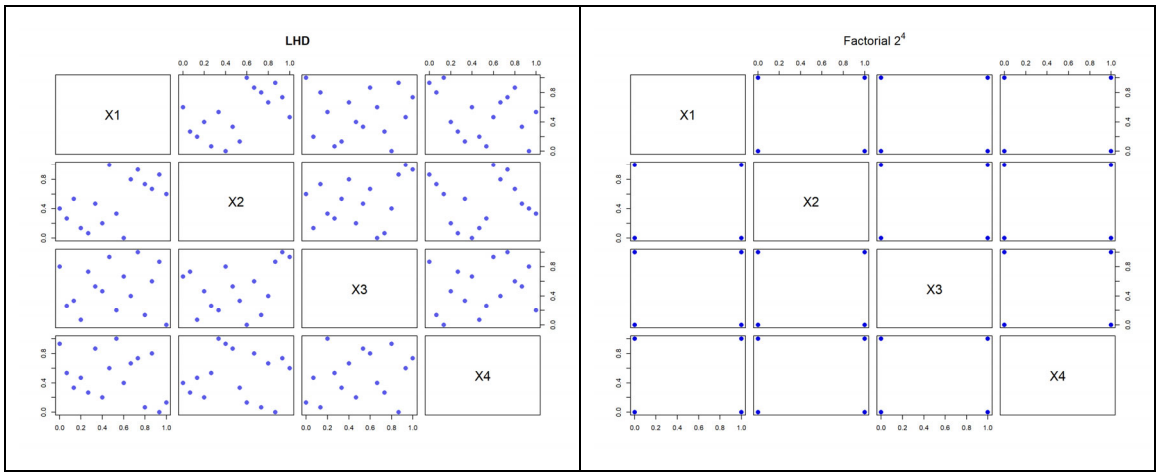


Figure 2. Two pairwise plots of 16×4 designs: An LHD (McKay et al. 1979) (left) and a 2^k Factorial (right). We see that the LHD has much better space-fillingness.

From Figure 2, the LHD (left) has more of its 16 DPs scattered throughout the design space, with less "white space." Conversely, the 2^k factorial design (Box and Wilson 1951) (right) samples only at the corners; i.e., the low and high settings for each factor. Each of the projections into the 2^k factorial design's two-dimensional subspaces has DPs at only four places, with four DPs "stacked" at each dot in the scatter plots. If it is known that the response contains "only linear main effects and interactions," the 2^k factorial design may be the "most efficient" design (Sanchez et al. 2020, p. 1132). If we use the LHD design when the response is, in fact, only linear main effects and interactions, we lose statistical power. However, if there are nonlinear responses, especially for localized effects in the interior, they cannot be identified with a 2^k factorial design, while the LHD enables an analyst to fit a breadth of diverse and complex models to multiple different responses (Hinkelmann and Kempthorne 2008, Santner et al. 2018). As Santner et al. (2018, p. 148) caution, "[b]ecause one does not know the true relationship between the response and inputs, designs should allow one to fit a variety of models." That is, SFDs can be advantageous when the true relationship between the model's responses and inputs are unknown.

4. Space-Filling Measures (SFMs)

There are multiple approaches to quantifying the space-fillingness of a design. The approaches generally fall into two broad classes, measures of distance and discrepancy (or uniformity). This section introduces the SFMs used in this research.

Using distance measures is the most common approach to quantifying the space-fillingness of a design (Joseph 2016). Intuitively, for a fixed number of design points in χ, large inter-point distances are desirable. The p^{th} order distance between any two design points x_i and x_j is defined as

$$d_p(x_i, x_j) = \left(\sum_{c=1}^{k} | x_{ic} - x_{jc} |^p \right)^{1/p}, \tag{3}$$

where $p = 1$ is the Manhattan distance and $p = 2$ is the Euclidean distance.

The *nearest neighbor* (NN) *distance* of the i^{th} DP, x_i, is $\gamma_i = \min_{i \neq j} d_p(x_i, x_j)$. The value $\bar{\gamma}$ is the average of the γ_i values over all n DPs. Designs obtained using distance criteria, such as $\bar{\gamma}$, usually strive to spread out (i.e., maximize $\bar{\gamma}$) DPs within the bounded χ.

One way to measure the largest gap or "white space" in $\mathbf{X}$ is to calculate the maximum distance of any point x in χ from its nearest DP; that is $\max_{x \in \chi} \min_i d_p(x, x_i)$. A common design goal is to find the design that minimizes this "worst" distance for all possible designs of the same dimension (i.e., n and k). The design $\mathbf{X}_{\text{mM}}$ that achieves this is called the *minimax* (mM) *distance design* (Johnson et al. 1990).

Finding $\mathbf{X}_{\text{mM}}$ is computationally challenging for large designs, as we must consider distances to all possible points in χ (Joseph 2016). Thus, *maximin* (Mm) *distance designs* (Johnson et al. 1990) are more common in practice since they need to compute only the distances among the n design points (Joseph 2016, p. 30). However, finding an Mm distance design is still challenging for large designs. $\mathbf{X}_{\text{Mm}}$ is called an Mm distance design

if it achieves the maximum $\min_{i \neq j} d_p(x_i, x_j)$ among all possible designs of the same dimension (Johnson et al. 1990).

Morris and Mitchell (1995) introduced the ϕ_p criterion, another popular distance measure that converges to the Mm distance criterion in the limit; i.e., as $p \rightarrow \infty$ of Equation (3), which is to be minimized. Let $d_1, \ldots, d_s$ be the ordered list (smallest to largest) of the s unique distances between design points. Additionally, let $J_1, \ldots, J_s$ be an index list that counts the occurrences of d_i. That is, J_i is the number of inter-sited distances of size d_i. Finally, ϕ_p is defined as:

$$\phi_p = \left[\sum_{i=1}^{s} J_i d_i^{-p} \right]^{1/p}. \tag{3}$$

Morris and Mitchell (1995) purport that their criterion is less "cumbersome" (p. 387) than the Mm distance in searching for MmLHDs. Concerning p, Morris and Mitchell (1995, p. 390) explain that values of p as large as 50 can be required for larger designs, while Wang et al. (2021) write "[i]n practice, $p = 15$ often suffices" (p. 4). Morris and Mitchell (1995) recommend their criterion for "situations of 'effect sparsity' and in situations where all or most inputs are important" (p. 397).

However, as Joseph (2016) points out, Mm and mM distance designs can have projections onto subspaces that "can be poor" (p. 31). Having good space-filling properties in low-dimensional subspaces is especially desirable in situations in which only a small number of input columns have a significant effect on the response.

There are several other distance measures of space-fillingness in the literature, as discussed in Santner et al. (2018). Of note are the distance measures that consider the space-fillingness of projections of a design into low-dimensional subspaces, such as the *maximum projection* (MaxPro) *designs* of Joseph et al. (2015) and the uniform projection designs of Sun et al. (2019). The MaxPro distance measure of space-fillingness is defined as

$$\psi(\mathbf{X})=\left\{\frac{1}{\binom{n}{2}}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{\prod_{c=1}^{k}\left(x_{ic}-x_{jc}\right)^{2}}\right\}^{1/k} . \quad (4)$$

The MaxPro design goal is to find the design that minimizes $\psi(\mathbf{X})$ as defined in Equation (4). We see from (4) that for any factor *c*, if there exists *i* and *j* such that $x_{ic} = x_{jc}$, $\psi(\mathbf{X})$ will evaluate to infinity. Moreover, when any $x_{ic} - x_{jc}$ is close to zero, the MaxPro distance measure will be large and subject to numeric instability. Thus, for all factors *c*, this distance measure severely penalizes having two input values close. Ba and Joseph (2018) construct MaxPro (Joseph et al. 2015) designs with and without restricting **X** to be an LHD. The latter are called MaxProLHDs.

Coverage (*Cov*) is a unitless space-filling distance criterion that measures the dissimilarity between **X** and a regular mesh grid (i.e., coverage equals zero for a mesh grid **X**), which is defined as

$$\text{coverage}(\mathbf{X})=\frac{1}{\bar{\gamma}}\left(\frac{1}{n}\sum_{i=1}^{n}\left(\gamma_i-\bar{\gamma}\right)^2\right)^{1/2}, \quad (5)$$

where γ_i and $\bar{\gamma}$ are the nearest-neighbor distance and average-nearest-neighbor distance defined previously (Dupuy et al. 2015, p. 4). Citing Dupuy et al. (2015), Sanchez and Sanchez (2019) explain that "smaller [coverage] values are indicative of better space-filling behavior" (p. 861).

Another approach to measuring the space-fillingness of a design is to quantify how uniformly the design points are spread throughout the experimental region (Fang 1980). To be precise, let $z = \{z_1,\ldots,z_k\}$ be a point in $\chi = [0,1]^k$, R_z the region $[0,z_1)\times\ldots\times[0,z_k)$, *vol*($R_z$) the volume of region R_z, and $N(\mathbf{X}, R_z)$ the number of design points of **X** within R_z. Then, the joint cumulative distribution function of the multivariate uniform distribution over χ is $F(z) = \prod_{c=1}^{k} z_c = vol(R_z)$. The *star discrepancy* (also known as the L_∞ *discrepancy*) of design **X** is defined as

$$D^*(\mathbf{X}) = \max_{z \in \chi} \left| \frac{N(\mathbf{X}, R_z)}{n} - vol(R_z) \right|. \tag{6}$$

$D^*(\mathbf{X})$ measures the discrepancy between the empirical joint cumulative distribution function of the DPs in **X** and the joint cumulative distribution function of a multivariate uniform distribution. It is equivalent to the Kolmogorov-Smirnov statistic commonly used as a "test of goodness of fit" (Massey 1951, p. 68). The objective is to construct a design of fixed dimensions with $D^*(\mathbf{X})$ as close to zero as possible.

Since $D^*(\mathbf{X})$ is computationally burdensome, following Hickernell (1998), we use the *modified* L_2 *discrepancy* $(ML_2)^2$ to quantify the uniformity of a design. A nice feature of this measure is that "projection uniformity over all subdimensions can be considered" (Fang et al. 2000a, p. 243). Formally,

$$\left(ML_2\right)^2 = \left(\frac{4}{3}\right)^k - \frac{2^{1-k}}{n} \sum_{i=1}^{n} \prod_{c=1}^{k} \left(3 - x_{ic}^2\right) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \prod_{c=1}^{k} \left[2 - \max(x_{ic}, x_{jc})\right]. \tag{7}$$

Another commonly used measure of discrepancy (e.g., JMP design generation software) is the $(CL_2)^2$ or *centered L2 discrepancy*, Hickernell (1998), which is defined as

$$\begin{aligned} \left(CL_2\right)^2 = & \left(\frac{13}{12}\right)^k \\ & - \frac{2}{n} \sum_{i=1}^{n} \prod_{c=1}^{k} \left(1 + \frac{1}{2} |x_{ic} - 0.5| - \frac{1}{2} |x_{ic} - 0.5|^2\right) \\ & + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \prod_{c=1}^{k} \left(1 + \frac{1}{2} |x_{ic} - 0.5| + \frac{1}{2} |x_{jc} - 0.5|^2 - \frac{1}{2} |x_{ic} - x_{jc}|\right). \end{aligned} \tag{8}$$

The design goal for both the $(ML_2)^2$ and $(CL_2)^2$ measures of discrepancy is to find the design **X** that minimizes their values.

Design imbalance is a measure used to assess columns of **X** containing categorical or discrete factors. Nearly orthogonal-and-balanced (NOAB) designs (Vieira et al. 2011, 2013) combine nearly orthogonal Latin hypercube (NOLH) lattice sampling with discrete or categorical input. Citing Bathke (2004), Vieira et al. (2013) explain that one-dimensional

balanced designs are suitable and capable of efficiently exploring various response surfaces containing discrete input.

For column X_c, with n DPs containing l_c levels $1,\ldots,l_c$ and counts of occurrences of a particular group $(w_{i,c})$, the maximum imbalance for X_c is the following:

$$\delta_c = \max_{i=1,\ldots,l_c} \left| \frac{w_{i,c} - (n/l_c)}{n/l_c} \right|. \tag{9}$$

D. LATIN HYPERCUBE DESIGNS (LHDS)

Wang et al. (2021) state that "[t]he most popular experimental designs for computer experiments are Latin hypercube designs (LHDs)." McKay et al. (1979) introduced LHDs in their seminal paper on designing computer experiments. LHDs guarantee good (or even optimal) one-dimensional space-fillingness for each input factor by dividing its range into n equally spaced intervals and obtaining one observation from each interval. The interval observations can be a fixed value, such as the center of the interval, or chosen randomly according to a user-specified distribution (often a uniform) within the interval. We call the latter approach Latin hypercube sampling (LHS). Simple LHD variants can be easily constructed for most any n and k; for example, having each X_c be an independent random permutation of the integers 1 through n scaled to the factor's range. In such cases, all design points reside on a k-dimensional lattice. For designs of size n and k, there are a total of $(n!)^{k-1}$ possible LHDs.

Due to their ease of construction and flexibility in fitting metamodels, LHDs are widely available in many simulation software packages. However, randomly created LHDs can have very poor overall space-fillingness or high correlations among columns (Lin and Tang 2015). Thus, there have been many efforts to build or improve upon the correlation or space-filling properties of LHDs.

Algorithms that reduce correlations among columns of LHDs include Florian (1992), Hernandez et al. (2012a), Iman and Conover (1982), Little et al. (2019), MacCalman et al. (2017), and Owen (1994). Beginning with Ye (1998), several researchers

have developed algebraic methods to construct column orthogonal LHDs (OLHDs), including Bingham et al. (2009), Butler 2001, Georgiou (2009), Georgiou and Efthimiou (2014), Lin et al. (2009), Pang et al. (2009), Steinberg and Lin (2006, 2015), Sun et al. (2009), and Sun and Tang (2017a, 2017b). These algorithms can be good for mathematically "identifying efficient LHDs" for large n and k (Wang et al. 2021). However, a limitation of many of these algebraic approaches is severe constraints on possible n and k combinations.

As noted above, designs optimized for space-filling distance measures may not have good space-filling in low-dimensional projections. Conversely, LHDs have excellent one-dimensional projections for all factors but are not guaranteed to have good high-dimensional space-fillingness. Consequently, in attempts to obtain the benefits of both approaches, many researchers have developed algorithms to optimize space-filling measures with $\mathbf{X}$ constrained to be an LHD.

E. ALGORITHMS FOR BUILDING SPACE-FILLING DESIGNS

This subsection provides an overview of several space-filling design (SFD) construction algorithms. Constructing large-scale designs that optimize the space-filling measures (SFMs) above is typically extremely challenging due to nonlinear objective functions, integer constraints, and many decision variables. Most often, heuristic stochastic search algorithms do not guarantee that an optimal design will be found within a practical amount of time (Jin et al. 2003 and Lin and Tang 2015). Depending on the random-number-generator seed, initial conditions for the search, stopping criteria, processing time allotted, computing power, and search-algorithm input parameters, different "optimal" solutions with substantially disparate measures of design quality can be obtained. As Jin et al. (2003) write, "[s]earching the optimal design of experiments within a class of designs (e.g., LHD), even though more tractable than searching in the entire sample space without any restrictions, is still difficult to solve exactly" (p. 546). Moreover, there may be multiple optimal solutions with respect to a given criterion, e.g., maximin distance, especially for lattice restricted designs. For example, through an exhaustive search of small 5×3 LHDs, Joseph and Hung (2008) "found that there are 142 different designs according to the

maximin criterion" (p. 179). These multiple "optimal solutions" may exhibit substantially different values for other design quality measures.

One widely used stochastic optimization method in SFD construction is simulated annealing (SA) (Bohachevsky et al. 1986, Kirkpatrick et al. 1983). For example, Morris and Mitchell (1995) used SA to search for an LHD that maximizes the minimum Euclidean distance between design points, which they call a maximin LHD (MmLHD). Their procedure begins with a randomly generated LHD and stochastically looks for improvement through random exchanges of two entries from a randomly chosen column. However, as Lin and Tang (2015, p. 600) note, "these designs are approximate maximin Latin hypercube designs. No general method is available to construct exact maximin Latin hypercube designs of flexible run sizes." Ba and Joseph (2018) also used SA based algorithms to search for LHDs that optimize their MaxPro criterion (Joseph et al. 2015). Joseph and Hung (2008) incorporated both Mm distance and correlation into their SA algorithm. Other SA-based algorithmic methods that leverage stochastic optimization include Ba (2015), Ba et al. (2015), Leary et al. (2003), Qian (2012), and Sun et al. (2019).

Another commonly used stochastic optimization approach for constructing SFDs is genetic algorithms (GAs), also known as evolutionary algorithms. Wang et al. (2021, p. 7) explain that a GA is "a nature-inspired meta-heuristic optimization algorithm which mimics Charles Darwin's idea of natural selection" (Goldberg and Holland 1988 and Holland 1992). As one example, MacCalman et al. (2017) used a GA involving selection, crossover, mutation, and fitness of the new population as well as "many thousands of hours" on high-performance computing equipment to find and catalog LHDs that are nearly orthogonal for a complete second-order regression model. Liefvendahl and Stocki (2006) used a GA to generate MmLHDs that incorporated a distance criterion from Audze and Eglais (1977) in their fitness function. In the context of GA applied to experimental design, other examples of leveraging the effectiveness and efficiency of GA include Goldfarb et al. (2005), Heredia-Langner et al. (2003), and Heredia-Langner et al. (2004).

Optimization algorithms for generating SFDs containing stochastic elements are also found in proprietary commercial software. For example, JMP generates space-filling

designs using a coordinate-exchange algorithm (Meyer and Nachtsheim 1995) to "optimize one of several optimality criteria" (SAS Institute 2018). Citing JMP's DOE Guide,

> Each iteration of the algorithm involves testing every value of every factor in the design to determine if replacing that value increases the optimality criterion. If so, the new value replaces the old. This process continues until no replacement occurs for an entire iteration. To avoid converging to a local optimum, the whole process is repeated several times using a different random start. (SAS Institute 2018, p. 99)

The parameter *Number of Starts* controls the number of design regenerations to improve the overall design for a desired "optimality" criterion.

Lin and Tang (2015) and Wang et al. (2021) provide excellent reviews of SA, GAs, and other stochastic search algorithms used in constructing LHDs and other space-filling designs and show many comparison results. The big takeaways are that solutions are usually not guaranteed to be optimal and there is tremendous variability in design measures from one random generation to the next.

F. SOFTWARE PACKAGES FOR CONSTRUCTING SPACE-FILLING DESIGNS

This subsection lists software packages used to construct SFDs in this research, beginning with the commercial software package JMP (SAS Institute 2018). The space-filling design command below DOE in JMP's interface can generate numerous possible design classes. Table 1 lists JMP SFDs that this research explores (SAS Institute 2018). A list of R software packages investigated during this research is in Table 2.

Table 1. A partial list of JMP space-filling designs.

Design title in JMP	Goal	Design class
Sphere Packing	Maximize the minimum Euclidean (L_2) distance between DPs	Mm-distance
Latin hypercube	Maximize the minimum Euclidean (L_2) distance between DPs and adds column constraints to ensure even spacing between the levels; i.e., constrained sphere packing	Mm-distance LH
Uniform	Minimize the $(CL_2)^2$ discrepancy (Hickernell 1998)	Uniform

Table 2. Examples of R software packages used to construct space-filling designs.

R Package	Description	Design class
DiceDesign	Constructs multiple design types: maximin distance designs; Latin hypercube designs (LHD), either randomly sampled, maximin, or minimal discrepancy	Mm-distance, random LHD, or MmLHD
maximin	Sun et al. (2019) and Sun and Gramacy (2021) generate designs under the Mm distance criterion.	Mm-distance
MaxPro	Recommended construction creates maximum projection (MaxPro) distance designs	MaxPro
minimaxdesign	Constructs minimax (mM) designs and minimax projection designs using PSO (particle swarm optimization) (Chen 2013)	mM-distance
MOLHD	Hou and Lu (2018) generate "the optimal Maximin Latin hypercube designs"	MmLHD
UniDOE	Minimize a given discrepancy (e.g., $(CL_2)^2$ of Hickernell (1998))	Uniform

As we have seen, there are many different approaches to creating and assessing SFDs. There are also helpful R software packages, e.g., *AlgDesign* (Wheeler 2019) for generating comparative factorial designs, which are usually not considered space-filling.

Figure 3 displays six 16×4 SFDs constructed to optimize a variety of space-filling measures (SFMs) using multiple R software packages. The design classes are labeled: (a) Mm distance, (b) mM distance, (c) MmLHD, (d) uniform design (UD), (e) MaxProLHD, and (f) MaxPro. The R software packages are: (a) *maximin* (Sun et al. 2019 and Sun and Gramacy 2021), (b) *minimaxdesign* (Mak 2021), (c) *MOLHD* (Hou and Lu 2018), (d) *DiceDesign* (Dupuy et al. 2015), and (e) - (f) *MaxPro* (Ba and Joseph 2018 and Joseph et al. 2015). The names of R software packages are italicized in this dissertation.

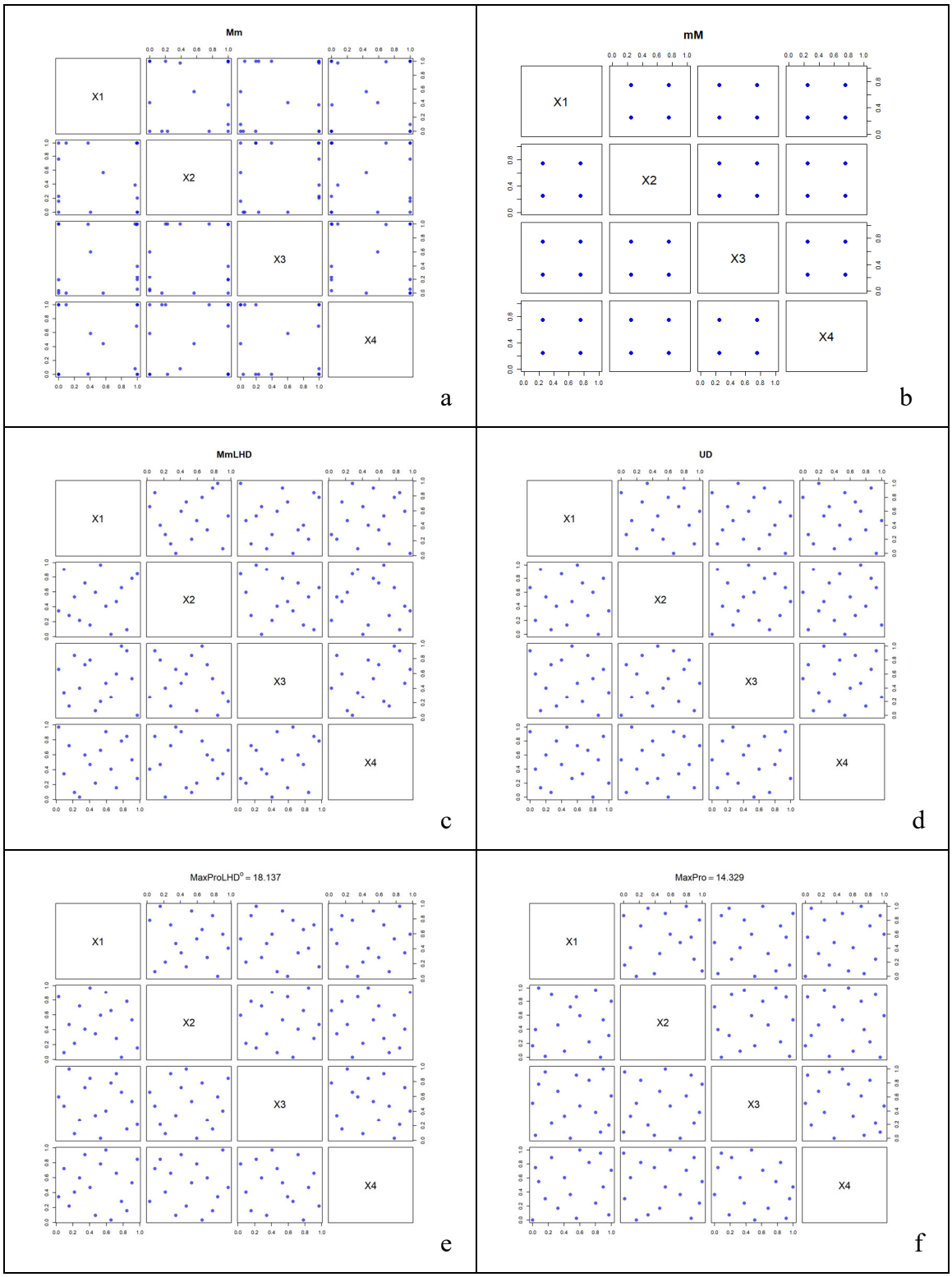


Figure 3. Six pairwise plots of 16×4 SFDs: (a) Mm distance, (b) mM distance, (c) MmLHD, (d) uniform design (UD), (e) MaxProLHD, and (f) MaxPro.

Looking visually across the six design types in Figure 3, we see that some of the SFDs are qualitatively different—at least in their two-dimensional space-fillingness. The comparative scatter plots of Figure 3 reveal that (a) and (b) do not fill all portions of the experimental space like designs (c) - (f) do, with many large gaps in the two-dimensional projections. The last four designs visually appear comparable (evenly spreading the design points) in the two-dimensional projections. However, for larger designs, measures of design characteristics (e.g., correlation, distance, discrepancy, and projectivity) are required to compare properties and identify high-quality designs.

Quantitative measures provide another tool that complements visual assessments, especially when comparative plots lose their fidelity, e.g., large n and k (as typically encountered in large-scale computational experimentation). This research uses several measures to assess design quality for multiple SFD classes of different design sizes. Table 3 contains ten measures on a representative design from each of eight design classes when $n = 16$ and $k = 14$. The columns specify the design type, number of DPs (n), number of columns (k), and ten quality measures. The first five quality measures include two correlation-based measures (ρ_{map} and $\overline{|\rho|}$), two discrepancy-based measures (the modified L_2-discrepancy $(ML_2)^2$ and the centered L_2-discrepancy $(CL_2)^2$), and coverage (Cov). The last five columns of Table 3 are distance-based measures, which include MaxPro, ϕ_p (Morris and Mitchell 1995) for $p = 15$ and $p = 50$ (Wang et al. 2021), maximin (Mm) distance (Johnson et al. 1990), and lastly $\overline{\gamma}$ (Dupuy et al. 2015, Sanchez and Sanchez 2019). For each measure, the best and worst designs are respectively identified by italics and bolding.

Table 3. Comparative table for the eight classes of SFD of Figures 2 and 3 (a – f) and for ten measures of design characteristics.

Design class	n	k	ρ_{map}	$\overline{\lvert\rho\rvert}$	$(ML_2)^2$	$(CL_2)^2$	Cov	MaxPro	$\phi_{p=15}$	$\phi_{p=50}$	Mm	$\overline{\gamma}$
LHD	16	4	**0.4441**	**0.1618**	0.0355	0.0166	0.1472	33.4131	**2.8991**	**2.8348**	**0.3528**	**0.4510**
2^k	16	4	*0.0000*	*0.0000*	**0.7191**	**0.6152**	*0.0000*	**Inf**	*1.2606*	*1.0718*	*1.0000*	*1.0000*
Mm	16	4	0.1344	0.0792	0.3531	0.2561	0.0523	**Inf**	1.4431	1.2388	0.8492	0.8722
mM	16	4	*0.0000*	*0.0000*	0.2141	0.1170	*0.0000*	**Inf**	2.5213	2.1435	0.5000	0.5000
MmLHD	16	4	0.2706	0.1294	0.0165	0.0119	0.0313	31.0771	2.2452	1.9407	0.5266	0.5537
UD	16	4	0.1106	0.0409	0.0152	0.0092	**0.1509**	35.9343	2.7915	2.7356	0.3656	0.4760
MaxProLHD	16	4	0.1882	0.0951	*0.0130*	*0.0088*	0.0947	18.1373	2.4751	2.3852	0.4193	0.5095
MaxPro	16	4	0.1829	0.0916	0.0433	0.0238	0.1096	*14.5094*	2.2021	2.1369	0.4680	0.5783

Lower values are preferred for all the measures except for the last two from the right; i.e., Mm and $\overline{\gamma}$. Low correlation, discrepancy, coverage, and the MaxPro criterion are desirable in practice. However, larger values are preferred in the context of the design's Mm distance and average nearest neighbor. When these values are large, typically good inter-point (and average inter-point) spread exists. There is large variation across the design classes for most measures. Interestingly, a design class can perform superbly for some measures and quite poorly for others.

Most of the literature creates designs based primarily on one measure. However, a few formal multi-criteria efforts focused on both space-filling and column-correlation measures. Cioppa and Lucas (2007) used massive random generation and correlation reduction techniques to find and catalog nearly orthogonal LHDs that minimize $(ML_2)^2$ in design dimensions where orthogonal LHDs were known to exist. Joseph and Hung (2008) applied simulated annealing to construct designs that optimized a weighted combination of a distance and correlation measure. Wang et al. (2018) and Wang et al. (2020) show connections between maximin distance, projection uniformity, and column orthogonality.

G. CATALOGS OF SPACE-FILLING DESIGNS

To make computationally expensive SFDs readily available for widespread use, many computer-generated designs are cataloged and typically made available online. For example, https://spacefillingdesigns.nl/ contains a collection of L_2-maximin (Mm) space-filling Latin hypercube designs (LHDs) for between 11 and 20 factors and up to seven

design points (van Dam et al. 2009). Fang et al. (2000b) provide nearly 1,000 space-filling uniform designs (UDs) for up to 29 factors and 30 design points (https://www.math.hkbu.edu.hk/UniformDesign/). To create this library of uniform designs, Fang et al. (2000b) used a stochastic threshold-accepting heuristic procedure that strives to find designs that minimize measures of discrepancy where the (unscaled) columns of the design matrix correspond to a permutation of the first *n* natural numbers, as in lattice LHDs. The SEED Center for Data Farming (https://harvest.nps.edu) provides scores of nearly orthogonal and space-filling LHDs, as well as other designs, constructed by a host of stochastic heuristic optimization methods and mathematical programming in easy-to-use spreadsheets. Of course, even extensive catalogs provide experimenters with comparatively few design options given the vastness of the potential application space.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PERMUTE AND STACK—CONSTRUCTING PORTFOLIOS OF DESIGNS

> Since we will never have the correct model, we cannot expect to run a single experiment and learn all that we need to learn from that experiment. Indeed, [George] Box (1993) quoted R. A. Fisher: "The best time to design an experiment is after you have done it." Thus, experimentation should (ideally) be sequential…
>
> —Ryan (2007), p. 4

The cataloged space-filling designs (SFDs) available from several sources have been created for a limited number of design point and factor combinations. We can extend these designs by adding more design points, with the goal of extracting more information from our experimentation. The question then becomes, "how can we extend a given design to best improve its measures of correlation and space-fillingness?" This chapter introduces a mixed-integer programming (MIP) algorithm that enables experimenters to sequentially add blocks of n design points to an original $n \times k$ base design to minimize ρ_{map} in the new, extended designs.

A. MOTIVATION

The motivation for this research is to find a better way to extend computationally expensive or cataloged designs than the heuristic approaches currently in use. The key idea is that any reordering (or permutation) of the columns of an $n \times k$ base design matrix, $\mathbf{X}^0$, will generate a column reordered $n \times k$ design with equivalent space-filling and correlation properties. By appending (or stacking) the column reordered design under the base design, we can create a new $2n \times k$ design matrix. The challenge is how best to permute the columns for large designs since there are $k!$ possible orderings of the columns of $\mathbf{X}^0$. For example, assuming the base design has 16 factors, then there are 16! (i.e., 20.9 trillion) possible orderings.

A straightforward approach, known as shift-and-stack, is a simple algorithm to extend designs by shifting each of the columns in $\mathbf{X}^0$ over by one (forward or backward)

and appending the shifted matrix to $\mathbf{X}^0$ to form an extended design matrix. When shifting columns forward, the first column of $\mathbf{X}^0$ rotates to the last column in the shifted matrix. When shifting columns backward, the last column of $\mathbf{X}^0$ rotates to the first column in the shifted matrix. Sanchez and Sanchez (2019) use the shift-and-stack method (p. 859) to form "seven stacks of a seven-factor optimal frequency-based design (FBD)" to improve its space-filling characteristics (p. 861). Others using this approach include Lin (2018), Kesler et al. (2019), and Sanchez et al. (2020). Throughout this work, unless otherwise specified, forward shift-and-stack is used.

B. MATHEMATICAL MODEL

This section introduces a new sequential optimization method based on a quadratically constrained mixed integer program (MIP) that improves design stacking by matching columns of a given base design to the columns of a design being extended. The matching is done in a manner that minimizes the maximum absolute pairwise correlation (ρ_{map}) among columns in the new extended design. Our mathematical model is based on a reformulation of the quadratic assignment problem (Lawler 1963). The relationship to the quadratic assignment problem makes our problem *NP*-hard; see Garey and Johnson (1979).

The process begins with a given base $n \times k$ SFD, which we label $\mathbf{X}^0$. Each column specifies the settings for a given input factor over the n runs. For any permutation p of the column indices $1,\ldots, k$, p_i is the column in the i^{th} position in the permutation. Let $\mathbf{X}^p$ be the $n \times k$ matrix resulting from permuting the columns of $\mathbf{X}^0$ according to p. Next, consider the $2n \times k$ matrix, $\mathbf{X}^1$, resulting from appending (or stacking) $\mathbf{X}^p$ below $\mathbf{X}^0$; see Equation (10):

$$\mathbf{X}^1 = \begin{bmatrix} \mathbf{X}^0 \\ \mathbf{X}^p \end{bmatrix}. \tag{10}$$

Column c of this stacked design consists of column c of $\mathbf{X}^0$ on top of column c of $\mathbf{X}^p$ or, equivalently, on top of column p_c of $\mathbf{X}^0$. We label these columns X_c^0, X_c^p, and $X_{p_c}^0$, respectively. Column c of $\mathbf{X}^1$ is as follows:

$$X_c^1 = \begin{bmatrix} X_c^0 \\ X_c^p \end{bmatrix} = \begin{bmatrix} X_c^0 \\ X_{p_c}^0 \end{bmatrix}. \tag{11}$$

The larger *one-stack* SFD, $\mathbf{X}^1$, has twice as many DPs, thereby making the experimental region χ denser and providing more degrees of freedom for fitting metamodels.

The $k!$ possible $\mathbf{X}^p$ matrices make an exhaustive search for the $\mathbf{X}^1$ with the best properties difficult for large *k*. In what follows, X_c^0 and X_f^0 denote columns *c* and *f* of $\mathbf{X}^0$ and X_d^p and X_g^p denote columns *d* and *g* of $\mathbf{X}^p$. If permutation *p* places column *d* in position *c* (i.e., if $p_c = d$, so that $X_c^p = X_{p_c}^0 = X_d^0$) and, similarly, places column *g* in position *f*, we use $\ddot{\rho}_{(c,d),(f,g)}$ to denote the pairwise correlation of columns *c* and *f* in the resulting stacked design. Here, (*c*, *d*) denotes the resultant stacked column *c* when the column X_d^0 is appended below column X_c^0 and (*f*, *g*) represents the resultant stacked column *f* when the column X_g^0 is appended below column X_f^0. We calculate $\ddot{\rho}_{(c,d),(f,g)}$ by applying Equation (1) to the stacked columns.

Consider the following simple example. For $k = 3$, each column *c* has three columns that can be stacked under it. This results in $k^2 = 9$ possible column pairs. For example, if (*c*, *d*) = (1, 3), the first stacked column in $\mathbf{X}^1$ consists of X_3^0 appended below X_1^0. Furthermore, if (*f*, *g*) = (2, 1), the second stacked column in $\mathbf{X}^1$ consists of X_1^0 appended below X_2^0. This particular stacking is the result of permutation $p = \langle 3,1,2 \rangle$.

For a particular permutation, p^1, the i^{th} entry is written as p_i^1 (e.g., $p_2^1 = 1$ in the previous example). Moreover, this stacking idea can be applied iteratively. For a sequence of *S* permutations $p^1, \ldots, p^s, \ldots, p^S$, where $S \geq 1$, write $\mathbf{X}^{p^S}$ for the s^{th} stack as specified by the s^{th} permutation of the columns of $\mathbf{X}^0$. That is, $\mathbf{X}^{p^S}$ is $\mathbf{X}^0$ with its columns reordered according to p^s.

Given the $sn \times k$ stacked design, $\mathbf{X}^{s-1}$, define $\mathbf{X}^s$ recursively as

$$\mathbf{X}^{s} = \begin{bmatrix} \mathbf{X}^{s-1} \\ \mathbf{X}^{p^s} \end{bmatrix}. \tag{12}$$

By appending $\mathbf{X}^{p^s}$ to $\mathbf{X}^{s-1}$, we create design matrix $\mathbf{X}^{s}$ with $(s+1)n$ rows (i.e., DPs) and k columns.

Of course, the recursion process begins by adding the first stack:

$$\mathbf{X}^{1} = \begin{bmatrix} \mathbf{X}^{0} \\ \mathbf{X}^{p^1} \end{bmatrix}, \tag{13}$$

where $\mathbf{X}^0$ is our given $n \times k$ base design.

For iteration s, building $\mathbf{X}^{s}$ involves assigning each column of the base design $\mathbf{X}^0$ to a position in the new, extended stacked design. At each step, we minimize the objective function ρ_{map} and equate its value to the variable v. We express the linearized formulation of this optimization problem in model MINNU.

Model MINNU:

INDICES

c columns (alias d, f, g) $c = 1, \ldots, k$

DATA

$\mathbf{X}^0$ base design matrix

$\mathbf{X}^{s-1}$ stacked design matrix to be extended

CALCULATED DATA

$\ddot{\rho}_{(c,d),(f,g)}$ correlation $\rho_{c,f}$, for columns c and f in the resulting stacked design, if base design column d is stacked into position c and base design column g is stacked into position f

DECISION VARIABLES:

$W_{(c,d)}$ equals 1 (0 otherwise) if column d is assigned to position c

$Y_{(c,d),(f,g)}$ equals 1 (0 otherwise) if both columns d and g are assigned to positions c and f, respectively

ν a variable representing the value ρ_{map}

OBJECTIVE and CONSTRAINTS

$$\min \quad \nu \tag{A0}$$

$$\text{s.t.} \quad \sum_{c \in k} W_{(c,d)} = 1 \qquad \forall d \in k \tag{A1}$$

$$\sum_{d \in k} W_{(c,d)} = 1 \qquad \forall c \in k \tag{A2}$$

$$Y_{(c,d),(f,g)} \leq W_{(c,d)} \qquad \forall c,d,f,g \in k \tag{A3}$$

$$Y_{(c,d),(f,g)} \leq W_{(f,g)} \qquad \forall c,d,f,g \in k \tag{A4}$$

$$Y_{(c,d),(f,g)} \geq W_{(c,d)} + W_{(f,g)} - 1 \qquad \forall c,d,f,g \in k \tag{A5}$$

$$\nu \geq \rho_{(c,d),(f,g)} Y_{(c,d),(f,g)} \qquad \forall c,d,f,g \in k, f \neq c \tag{A6}$$

$$\nu \geq -\rho_{(c,d),(f,g)} Y_{(c,d),(f,g)} \qquad \forall c,d,f,g \in k, f \neq c \tag{A7}$$

$$W_{(c,d)} \in \{0,1\} \qquad \forall c,d \in k \tag{A8}$$

$$Y_{(c,d),(f,g)} \in \{0,1\} \qquad \forall c,d,f,g \in k \tag{A9}$$

The objective (A0) represents the maximum absolute correlation between any pair of columns in the resulting stacked design (i.e., $\mathbf{X}^S$) given by the solution to the model. Each constraint (A1) ensures that base design column d is assigned to precisely one position in the stacked design. Each constraint (A2) ensures that each position c in the stacked design is assigned to exactly one column from the base design. For a particular pair

of positions *c* and *f*, and for a particular pair of base design columns *d* and *g*, constraints (A3-A5) ensure that:

$$Y_{(c,d),(f,g)} = W_{(c,d)} W_{(f,g)}, \tag{14}$$

using a standard linearization technique that takes advantage of the fact that all variables involved are binary (Bazaraa et al. 2004, Bertsimas and Tsitsiklis 1997, Lawler 1963). Constraints (A6) and (A7) guarantee that the objective value will be at least as large as the absolute value of any single pairwise correlation; coupled with the fact that the objective is to be minimized, this guarantees that the optimal value will be equal to the maximum absolute pairwise correlation, ρ_{map}.

The linearization we use in Equation (14) is well known in the literature. Its primary drawback is that it introduces a large number of auxiliary variables $Y_{(c,d),(f,g)}$ into the model (on the order of k^4). However, for moderate values of *k,* our models easily fit into memory and can be solved to a small optimality gap in a reasonable amount of time. The model MINNU has $k^4 + k^2$ binary variables and one continuous variable. MINNU also contains $k(5k^3 - 2k^2 + 2)$ constraints. The model size does not depend explicitly on either *n* or *s* because we precalculate all $\ddot{\rho}_{(c,d),(f,g)}$ values.

Solutions to the model use the variables introduced in Equation (14) to linearize the pairwise correlation $\rho_{c,f}$, where

$$\rho_{c,f} = \sum_{d,g} \ddot{\rho}_{(c,d),(f,g)} W_{(c,d)} W_{(f,g)} = \sum_{d,g} \ddot{\rho}_{(c,d),(f,g)} Y_{(c,d),(f,g)}, \tag{15}$$

given the precalculated data $\ddot{\rho}_{(c,d),(f,g)}$. The computational requirements to compute $\ddot{\rho}_{(c,d),(f,g)}$ grow linearly with *n* (and the number of stacks, *S*). Therefore, the time to solve MINNU is driven primarily by the number of columns, *k.* The solution provided; i.e., the $Y_{(c,d),(f,g)}$ values, gives the measurably optimal permutation solution p^s, which determines $\mathbf{X}^{p^s}$.

C. PERMUTE AND STACK ALGORITHM: ITERATIVELY APPLYING MINNU

We apply this stacking idea sequentially, where each added stack is designed to minimize ρ_{map} and improve the space-filling properties of the resulting design at the cost of extra runs. Our algorithm takes as input a base design $\mathbf{X}^0$ and an integer number of stacks, S, to create, and applies the model MINNU sequentially, first to the base design and then to each resulting stacked design in turn to yield a sequence of designs from one stack to S stacks. These stacks are output as the sequence of permutations p^s that indicate the measurably optimal column assignments for each successive stack s.

To solve MINNU for a particular stack, we must first calculate all of the $\ddot{\rho}_{(c,d),(f,g)}$ values that can result from each of the pairs of possible column-to-position assignments. We provide a subroutine `calculate_rho_dict` that takes as input an $n \times k$ base design $\mathbf{X}^0$ and the $sn \times k$ stacked design matrix to be extended $\mathbf{X}^{s-1}$, and returns a dictionary, `rho_dict`, of the resulting pairwise correlations.

Our main routine is given in the `permute_and_stack` algorithm, which successively solves MINNU for an increasing sequence of designs and calls `calculate_rho_dict` as a subroutine before each model solve. We summarize each of these algorithms below.

The algorithm (Algorithm 1) subroutine `calculate_rho_dict` is as follows.

Algorithm 1 `calculate_rho_dict`: An algorithm to compute the dictionary, $\{\ddot{\rho}_{(c,d),(f,g)}\}$

Input: $n \times k$ base design, $\mathbf{X}^0$, and the $sn \times k$ stacked design matrix to be extended, $\mathbf{X}^{s-1}$

Output: `rho_dict` := $\{\ddot{\rho}_{(c,d),(f,g)}\}$, resulting pairwise correlations of all possible pairs of columns and assignments in the extended design space.

Dictionaries:

```
rho_dict = {}
stack_col_mean = {}
stack_col_sd = {}
permutations = {}
for c = 1, ..., k do
    for d = 1, ..., k do
        stack_col_mean[c,d] = mean of stacked column c on top of base column d
        stack_col_sd[c,d] = standard deviation of stacked column c on top of base column d
        add (c, d) to permutations
    end
end
for (c,d) in permutations do
    for (f,g) in permutations do
        rho_dict[c, d, f, g] = correlation between columns c and f in the stacked design
        (see Equation 1)
    end
end
return rho_dict
```

The algorithm (Algorithm 2) `permute_and_stack` for iteratively solving MINNU is as follows.

Algorithm 2 `permute_and_stack`

Input: base design, $\mathbf{X}^0$, and number of stacks, S

Output: $p^s, \mathbf{X}^s$ for $s = 1, 2, \ldots, S$,

```
begin
    for s = 1, 2, ..., S do
        rho_dict = calculate_rho_dict (X^0, X^{s-1})
        Solve MINNU, using rho_dict values, to obtain p^s
        append stack X^{p^s} to X^{s-1}, creating stacked design, X^s
    end
end
```

An application of `permute_and_stack` takes as input a base design $\mathbf{X}^0$ and the number of stacks to be created, *S*. It then iteratively determines `rho_dict` and solves

MINNU, generating a sequence of permutations p^s and stacked designs $\mathbf{X}^s$, for $s = 1,\ldots, S$. Each iteration consists of (1) finding and storing the column arrangement p^s of $\mathbf{X}^0$ (i.e., $\mathbf{X}^{p^s}$) that minimizes ρ_{map} and (2) extending the $\mathbf{X}^{s-1}$ matrix by appending the n rows of $\mathbf{X}^{p^s}$ to create the s^{th} stacked design $\mathbf{X}^s$.

For each stack, the resulting problem MINNU is sufficiently difficult that there is no guarantee that we will find an optimal solution within a finite time limit. However, the software to solve this model uses the branch-and-bound algorithm, which keeps track of both an upper and lower bound on the value of an optimal solution. The difference between the upper and lower bound is referred to as the optimality gap, and we can specify an acceptable gap as one of the inputs to the software. If the software terminates with a solution within this gap, we have a *nearly optimal* solution. We use the CPLEX solver (CPLEX 2009), which has a default optimality gap value that stops when a feasible integer solution has been proven to be within ten percent of the optimal solution value. We use the software package *PYOMO* (Bynum et al. 2021, Hart et al. 2011) to construct our models, and we also use it to set parameters, such as the optimality gap, to other values (e.g., 1%, 5%, and 10%).

D. A SEQUENCE OF DESIGNS: AN EXAMPLE OF THE PERMUTE AND STACK ALGORITHM

> All the above arguments point to the desirability of a sequence of moderately sized designs and reassessment of the results as each group of experiments becomes available.
>
> —Santner et al. (2018), p. 148

This section illustrates `permute_and_stack` extending a cataloged 8×7 uniform design (UD). We show how the correlation and space-filling measures improve with successive stacks and compare `permute_and_stack` with simple forward shift-and-stack and UDs in the stacked dimensions.

1. Extending a Cataloged Uniform Design

We consider a cataloged 8×7 uniform design (UD) from Fang et al. (2000b), see http://www.math.hkbu.edu.hk/UniformDesign/. With $k = 7$, there are 5,040 unique orderings of the columns. Thus, this example is for explanatory purposes since an exhaustive search over all possible stacks (i.e., permutations) is feasible.

Figure 4 shows the UD matrix (i.e., $\mathbf{X}^0$) and two extensions of it. On the left side of Figure 4 (grey) are two stacks resulting from the simple forward shift-and-stack heuristic applied to the columns of $\mathbf{X}^0$ as specified by the catalog. Original column labels in the stacks are kept for explanatory purposes. Forward shift-and-stack starts by shifting the columns of $\mathbf{X}^0$ to the left one column to make the shifted design that is appended under $\mathbf{X}^0$ (Sanchez and Sanchez 2019). The column in the 1^{st} position cannot be shifted left; thus, it is placed in the last (i.e., 7^{th}) position. Next, the shifted design is appended (or stacked) below $\mathbf{X}^0$. We label the resultant $2n \times k$ forward shift-and-stack design as $(\mathbf{X}_s)^1$. This process may be repeated k times before replicating $\mathbf{X}^0$—resulting in $k - 1$ possible stacks before the original ordering of $\mathbf{X}^0$ is obtained. The 2^{nd} extended design resulting from another application of forward shift-and-stack is denoted $(\mathbf{X}_s)^2$.

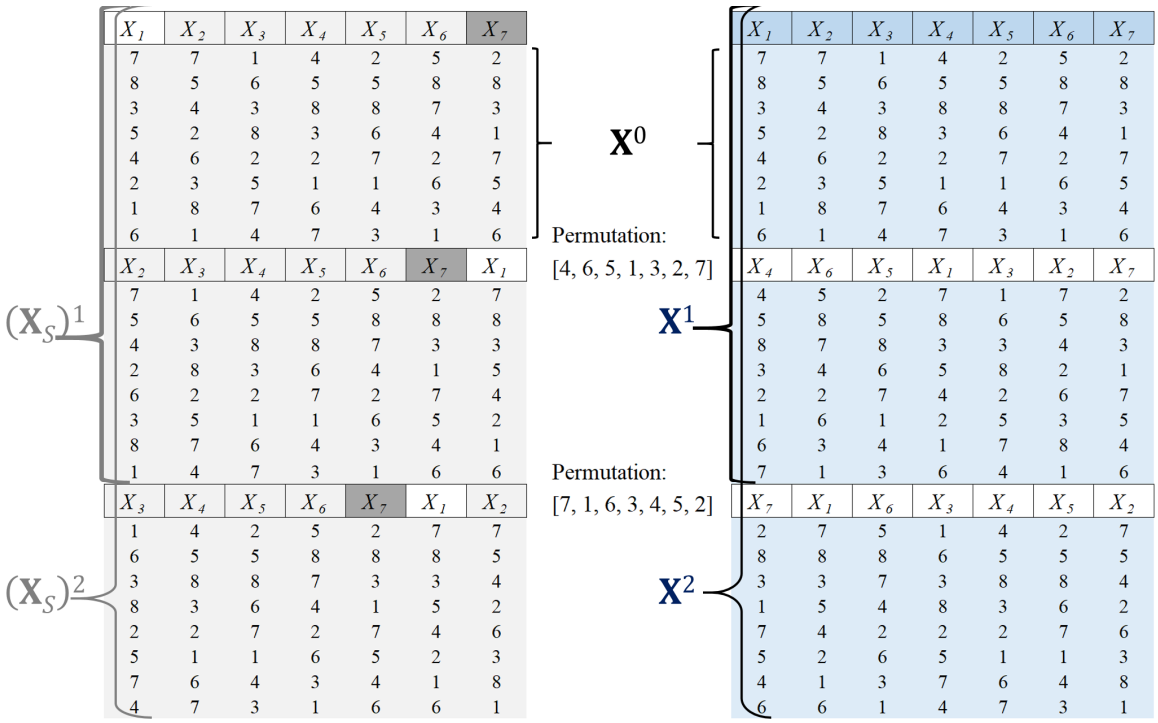


Figure 4. Two extensions of the 8×7 base uniform design (UD) $\mathbf{X}^0$ in matrix form: The left (grey) results from forward shift-and-stack. The right (blue) results from the `permute_and_stack` algorithm.

On the right side of Figure 4, we show two extensions to the base UD, $\mathbf{X}^0$, using the `permute_and_stack` algorithm (blue). Column labels are kept for explanatory purposes. These stacks are permutation solutions, where the chosen column concatenations minimize ρ_{map}. Here, permutations [4,6,5,1,3,2,7] and [7,1,6,3,4,5,2] result from two solves of MINNU for the first and second stacks, respectively. The designs ($\mathbf{X}^1$ and $\mathbf{X}^2$) result from one execution of the `permute_and_stack` algorithm.

2. Properties of the Extended Uniform Design

This section shows how some key SFD quality measures improve after using forward shift-and-stack and `permute_and_stack`. Reading row-wise, Table 4 shows two correlation (ρ_{map} and $\overline{|\rho|}$) and two space-filling ($(ML_2)^2$ and $(CL_2)^2$) measures for the UD extensions constructed in Figure 4. For the given cataloged $\mathbf{X}^0$, we have $\rho_{map} = 0.3095$, $\overline{|\rho|} = 0.0930$, $(ML_2)^2 = 0.5143$, and $(CL_2)^2 = 0.1870$. Lower values are desired for all of

these measures. After one stack, by using the `permute_and_stack` optimization, we achieve a 69.2 percent decrease in ρ_{map} for $\mathbf{X}^1$. Similarly, after one stack, using the heuristic forward shift-and-stack, we obtain a 34.6 percent reduction in ρ_{map} for $(\mathbf{X}_S)^1$. In this two-extension example, with $S = 2$ (i.e., 24×7 stacked designs), `permute_and_stack` decreases ρ_{map} by 82.0 percent while shift-and-stack reduces this measure by 53.8 percent. We also observe that the extended designs from `permute_and_stack` also yield better performance in the other measures; i.e., $\overline{|\rho|}$, $(ML_2)^2$, and $(CL_2)^2$—though by only a small amount for the discrepancy measures.

Table 4. The correlation and space-filling properties of the 8×7 uniform design (UD) and its extensions. Read rowwise to compare forward shift-and-stack to `permute_and_stack`.

	n	k	ρ_{map}	$\overline{\lvert\rho\rvert}$	$(ML_2)^2$	$(CL_2)^2$
$\mathbf{X}^0$	8	7	0.309 5	0.093 0	0.514 3	0.187 0
$(\mathbf{X}_S)^1$	16	7	0.202 4	0.066 9	0.349 0	0.118 5
$\mathbf{X}^1$	16	7	0.095 2	0.054 4	0.322 4	0.109 4
$(\mathbf{X}_S)^2$	24	7	0.142 9	0.046 1	0.284 8	0.093 8
$\mathbf{X}^2$	24	7	0.055 6	0.031 0	0.263 1	0.085 4

In this illustrative case, Fang et al. (2000b) provide 16×7 and 24×7 UDs. However, they provide no UDs with $n > 30$. In the context of a sequential "model-test-model" application, the desired design dimensions can quickly exceed their catalog's limits (e.g., three stacks yield $n = 32$).

Table 5 shows the properties of the forward shift-and-stack and `permute_and_stack` designs resulting from three extensions. The resultant `permute_and_stack` design, $\mathbf{X}^3$, is nearly orthogonal; i.e., $\rho_{map} = 0.0476 \leq 0.05$ (Hernandez et al. 2012a). The $\rho_{map} = 0.1369$ obtained from two applications of the simple forward shift-and-stack heuristic is nearly three times larger. Moreover, $\mathbf{X}^3$ is constructed

with one execution of the `permute_and_stack` algorithm, and the optimal permutations can be stored to effectively extend the catalog to many more designs. These added DPs provide additional degrees of freedom in fitting metamodeling, detecting bias, and gaining insights.

Table 5. Design property comparison after three iterations of forward shift-and-stack and `permute_and_stack`. By extending the base design using optimization, we decrease ρ_{map} from 0.3095 to 0.0476. The forward shift-and-stack design has a ρ_{map} value nearly three times higher. Note: There is no 32×7 in Fang et al.'s (2000b) online catalog, see http://www.math.hkbu.edu.hk/UniformDesign/.

	n	k	ρ_{map}	$\overline{\lvert\rho\rvert}$	$(ML_2)^2$	$(CL_2)^2$
$(\mathbf{X}_S)^3$	32	7	0.136 9	0.036 3	0.250 3	0.080 1
$\mathbf{X}^3$	32	7	0.047 6	0.030 0	0.233 2	0.073 7

Another possibility is to reorder the columns of $\mathbf{X}^0$ randomly (with all possible reorderings equally likely) and append them to the base design. To obtain a good sequence in a multiple-stack design, one would need to arrive at good random matchings with each stack. Here, good means a low or high value specific to the performance measure of interest—in this case a low ρ_{map}. The likelihood of achieving a near-optimal ρ_{map} decreases with additional stacks as the possible combinations grow. To show this, 1,000 random three-stack designs were made to extend the 8×7 UD analyzed above. Figure 5 shows the results of this three-stack experiment, with each stack being a random permutation of the columns of $\mathbf{X}^0$. None of the 1,000 arrived-at randomly-stacked designs achieve the ρ_{map} value from `permute_and_stack`, the blue line; moreover, the average is nearly three times higher.

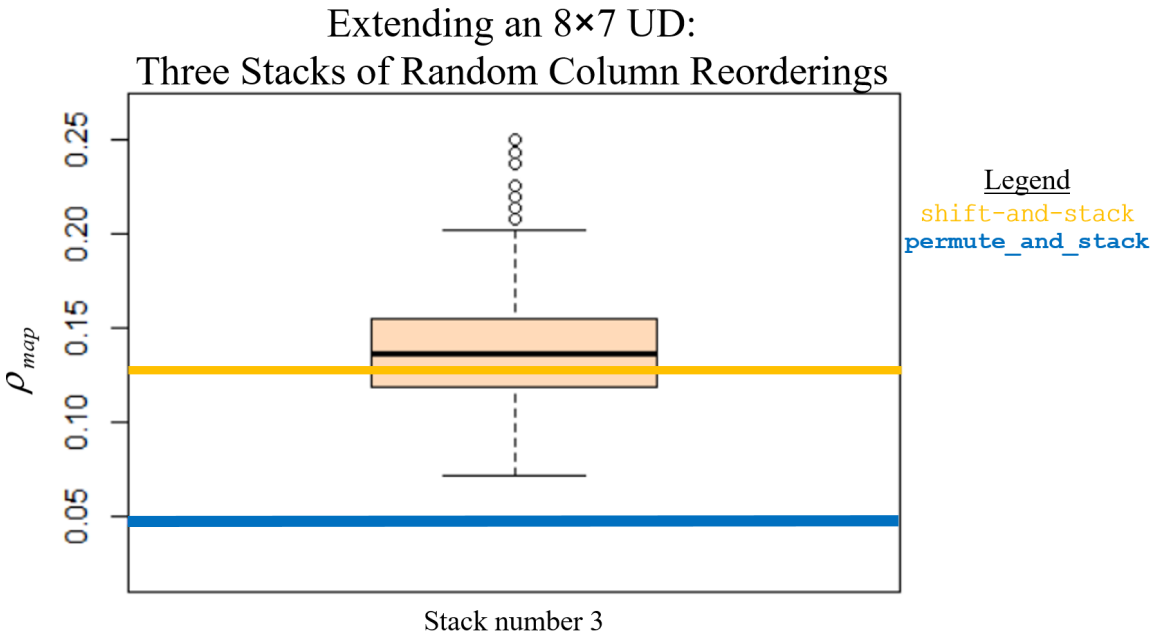


Figure 5. Three approaches: (1) Box plot of 1,000 ρ_{map} values for the three-stack 8×7 UD randomly extended experiment, (2) after three applications of shift-and-stack (ρ_{map} = 0.137) (orange), and (3) `permute_and_stack` (ρ_{map} = 0.048) (blue).

The odds of obtaining a near-optimal design (in terms of ρ_{map}) by stacking random permutations of the columns of $\mathbf{X}^0$ are extremely low and will almost certainly decrease as k increases given that there are $k!$ possible random permutations for each stack. If there are s stacks, there are $(k!)^s$ possible random extensions. Figure 6 presents an additional three-stack experiment for the 33×11 NOLH (nearly orthogonal Latin hypercube) design (Cioppa and Lucas 2007) and the results under `permute_and_stack` (blue line). The benefits of `permute_and_stack` can be seen, as the minimum ρ_{map} for 1,000 random three-stack 33×11 NOLH designs extended is more than double that obtained through optimization. We also see that ten heuristic variations of shift-and-stack (detailed in Chapter IV) outperform randomly extended 33×11 NOLH designs, but by not nearly as much as `permute_and_stack`.

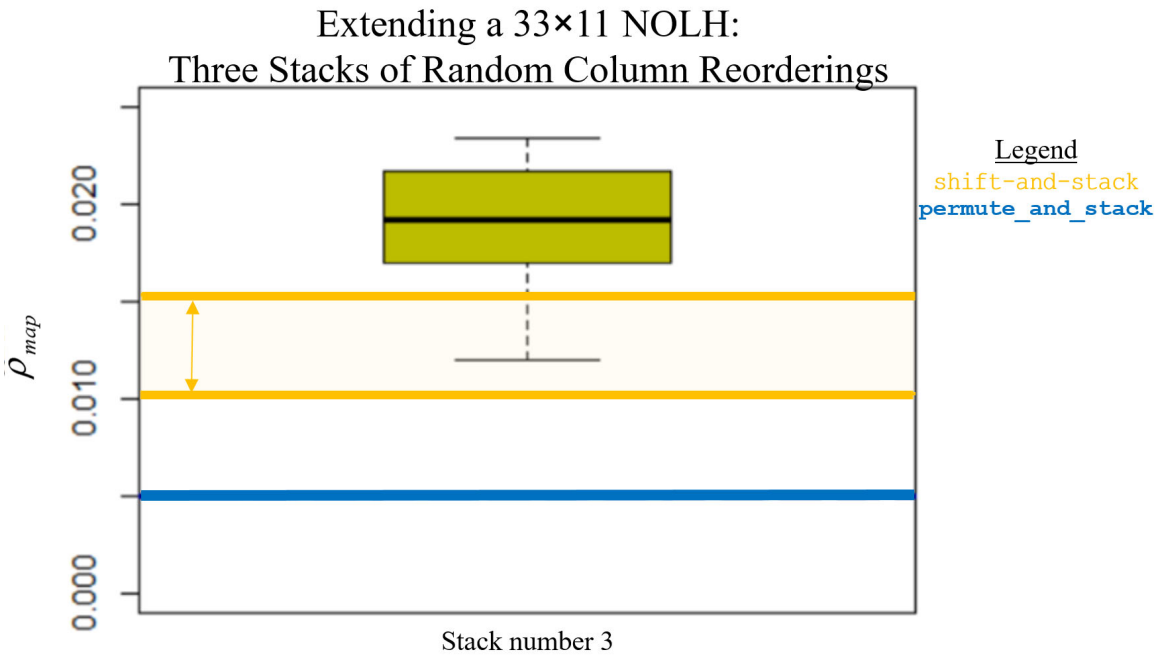


Figure 6. Three approaches: (1) Box plot (yellow) of 1,000 ρ_{map} values for the three-stack 33×11 NOLH randomly extended experiment, (2) the range of ρ_{map} values after three applications of shift-and-stack (orange), and (3) the results of `permute_and_stack` (blue).

E. PARAMETERS AND TIMING

This subsection touches on the CPLEX optimization parameters and some experiments on the processing time of the `permute_and_stack` algorithm. The experiments were run on a Microsoft Surface Studio 2 with four 7th Generation Intel ® Core™ i7-7820HQ @2.9GHz processors, 32 Gb of RAM, and the Windows 10 Enterprise operating system. The program software used for the optimization is the Python Optimization Modeling Objects (Pyomo) software. Pyomo is a capable mathematical modeling language with rich libraries (Bynum et al. 2021, Hart et al. 2011). Custom libraries produced for this research include the column reordering heuristic, shift-and-stack, and measures of design characteristics, which can all be imported.

Table 6 provides a partial list of input parameters for the CPLEX solver. Read columnwise, for a brief purpose description, default value, and example syntax for interactive scripting (e.g., Pyomo).

Table 6. A partial list of CPLEX parameters and the required interactive script syntax.

Purpose:	Default	Interactive Script	Values
solver selection	cplex	pyo.SolverFactory('cplex')	cplex
controls tade-off	0	'emphasis mip '	= 0, 1 …, 4
frequency to apply heuristic	0	'mip strategy heuristicfreq'	= -1, 0, or $\mathbb{Z}^+$
absolute tolerance on the gap	1e-06	'mip tolerances absmipgap '	= e.g., 0.01
relative MIP gap tolerance	1e-04	'mip tolerances mipgap '	= e.g., 0.01
symmetry breaking reductions	-1	'preprocessing symmetry '	= -1, 0, …, 5
maximum optimizer run time in seconds	NA	'timelimit '	= 7200
	Example:	`opt.options['emphasis mip'] = 1`	

The optimization was accomplished using the CPLEX solver, Version 12.10 (CPLEX 2009). However, other solvers are possible, e.g., Gurobi (Gurobi Optimization, LLC 2022) and COIN-OR (COIN-OR 2016). The parameter `emphasis mip` has five possible values that adjust the trade-off between feasibility, optimality, best bound, and hidden feasible solutions. If no value is provided, the default is zero, balancing feasibility and optimality to find a measurably optimal solution. To prioritize feasibility over optimality, optimality over feasibility, best-bound optimality (feasible solutions receive less emphasis), or high-quality hidden feasible solutions the user sets `emphasis mip` to 1, 2, 3, or 4, respectively. For large-scale computational experiments, we recommend setting this parameter to 1. Setting the value of `mip strategy heuristicfreq` to -1 turns off the periodic heuristic, to 0 (zero) applies the heuristic frequency at a rate chosen by CPLEX, and to any $\mathbb{Z}^+$ (e.g., 10) calls the heuristic every ten nodes (i.e., 0, 10, 20, …). We mostly left this at zero.

Concerning termination conditions (outside integer optimality), there are two tolerance options `absmipgap` and `mipgap`. The gap is computed as: `(best integer - best node) * objsen / (abs (best integer) + 1e-10)`, see CPLEX (2009) for details. The gap is the difference between the best integer feasible solution and a bound on the optimal solution. Option `absmipgap` is an absolute tolerance on the gap. The default value of `absmipgap` is 1e-6. The more common option is `mipgap`, a relative

tolerance on the gap, and the default value is 1e-4. Note `objsen` specifies whether the optimization is a minimization (as in this case) or a maximization.

We use default options in `preprocessing symmetry`. The default symmetry breaking is set to a value of –1, allowing CPLEX to decide the level of symmetry breaking. Symmetry-breaking reductions can be turned off (value set to 0) or increased for five possible levels of aggressiveness (i.e., 1-5). Option `timelimit`; i.e., optimizer time limit, can be set for any number of seconds. For example, two hours of solver time is shown in Table 6, which implies that if $s = 3$, then the resultant optimizer effort is for at most six hours. We set `timelimit` to two hours (7,200 seconds) for most of the optimization routines. For more information regarding additional parameters or options, see CPLEX (2009). We remark that more experimentation on how CPLEX's parameters affect `permute_and_stack's` performance and processing times would be valuable.

Solve time and final ρ_{map} results are shown in Table 7 as a function of `mipgap` for three extensions of uniform designs (UDs) for a dozen design sizes using CPLEX Version 12.10 (CPLEX 2009) as the solver. For $k \leq 10$, solve times are minimal (on the order of 30 seconds or less) regardless of the size of n or `mipgap` setting. For larger k, solve times increase dramatically if the algorithm is allowed to run until the `mipgap` tolerance is achieved. For example, three extensions of the 200×20 UD were solved within a `mipgap` of 0.10 after 1.08 days. The resultant 800×20 UD extended reduces the ρ_{map} value for $\mathbf{X}^0$ of 0.040 to a ρ_{map} of 0.008 for $\mathbf{X}^3$. For a timing comparison, JMP reports that it will take over 21 hours to generate one 200×20 UD. Importantly, CPLEX provides the best-found solution (and design) if stopped before the `mipgap` is attained. In most of our experiments, the `timelimit` option was set to 7,200 (two hours), and we have obtained low ρ_{map} values in designs for k as large as 30. In all cases looked at, the solutions found (i.e., final ρ_{map} values) are relatively insensitive to `mipgap`.

Table 7. Time comparison after three iterations of `permute_and_stack` applied to cataloged uniform designs.

	k	7			10			15			20		
		Solve Time	ρ_{map}		Solve Time	ρ_{map}		Solver Time	ρ_{map}		Solver Time	ρ_{map}	
n	`mipgap`	[mm:ss]	$\mathbf{X}^0$	$\mathbf{X}^3$	[mm:ss]	$\mathbf{X}^0$	$\mathbf{X}^3$	[mm:ss]	$\mathbf{X}^0$	$\mathbf{X}^3$	[mm:ss]	$\mathbf{X}^0$	$\mathbf{X}^3$
	0.50	00 : 05	0.62	0.11	00 : 37	0.51	0.08	11:03	0.45	0.07	1.2 days	0.41	0.07
	0.10	00 : 05		0.12	00 : 30		0.08	09:05		0.07	1.3 days		0.07
$\mathbf{4} * (k+1)$	0.05	00 : 05		0.11	00 : 30		0.08	09:06		0.07	—		—
	0.01	00 : 05		0.11	00 : 30		0.08	09:06		0.07	—		—
	0.50	00 : 05	0.13	0.02	00 : 32	0.19	0.03	11:03	0.15	0.02	1.45 days	0.13	0.02
	0.10	00 : 05		0.02	00 : 34		0.03	15:16		0.02	1.70 days		0.02
$\mathbf{4} * (3k+2)$	0.05	00 : 05		0.02	00 : 34		0.03	15:16		0.02	—		—
	0.01	00 : 05		0.02	00 : 34		0.03	15:52		0.02	—		—
	0.50	00 : 05	0.07	0.02	00 : 35	0.05	0.008	17:03	0.05	0.008	1.45 days	0.04	0.008
	0.10	00 : 05		0.02	00 : 34		0.008	28:09		0.008	1.08 days		0.008
$\mathbf{4} * (10k)$	0.05	00 : 05		0.02	00 : 34		0.008	37:16		0.008	—		—
	0.01	00 : 05		0.02	00 : 35		0.008	37:16		0.008	—		—

F. PERMUTE AND STACK SUMMARY

This chapter introduced the `permute_and_stack` optimization algorithm that enables experimenters to sequentially add blocks of n design points to an original $n \times k$ base design ($\mathbf{X}^0$) to minimize ρ_{map} in the new, extended designs. The added DPs also increase the space-fillingness of the extended designs. There are $k!$ unique permutations of the columns at each stack, with $(k!)^s$ total possible designs $\mathbf{X}^s$. Therefore, total enumeration is not viable for building large stacked designs. It is also unlikely that random permutations will produce a near-optimal $\mathbf{X}^s$. In addition, a series of easy-to-implement heuristics, building off of the shift-and-stack method, also failed to produce near-optimal extended designs. Hence, the value of `permute_and_stack` as shown above for the 8×7 UD**.** In the following chapter, we apply `permute_and_stack` to eight design types (a mix of cataloged and software-generated) for nine design sizes (i.e., n and k values) and assess the resultant designs using ten measures.

IV. PERMUTE AND STACK: EXTENDING AND IMPROVING DESIGNS FOR LARGE-SCALE COMPUTER EXPERIMENTS

> Our sequential customized designs are no longer LHDs (even though the first stage may be a LHD)… . In sequential designs we learn about the behavior of the underlying system as we experiment with this system and collect data.
>
> —Kleijnen (2015), p. 203

This chapter explores how `permute_and_stack` performs in reducing ρ_{map} when sequentially adding batches (or stacks) of n design points (DPs) to space-filling designs (SFDs). The reductions in ρ_{map} are contrasted with values obtained by applying shift-and-stack to several correlation-based column-reordering heuristics. Where feasible, we also evaluate `permute_and_stack` against newly generated SFDs in the extended design space. The extended design space refers to the dimensions of the design matrix $\mathbf{X}^s$ after s extensions (or stacks), which now has $(s+1)n$ rows and k columns.

The approach we take depends on whether we are extending a fixed cataloged design or a software-generated design. For cataloged designs, we compare several design sizes of the nearly orthogonal Latin hypercube (NOLH) designs (Cioppa and Lucas 2007) and uniform designs (UDs) (Fang et al. 2000a, 2000b). For software-generated designs, since there is a stochastic component, following Wang et al. (2020), we generate 100 base SFDs for each type. A sample size of 100 provides reasonable estimates of the distributions of performance measures studied, including small estimated standard errors of their means. The performance comparisons are made on designs containing 33 DPs and 11 factors (or columns). This allows us to compare measures among design classes. While not shown, similar results are obtained for other design sizes. The software-generated SFD types we extend and assess include MmLHDs (Morris and Mitchell 1995); maximum projection (MaxPro) designs (Joseph et al. 2015, Ba and Joseph 2018); sphere-packing Mm distance designs (Johnson et al. 1990); uniform designs (UDs) (Fang et al. 2000a, Zhang et al. 2018)

using both JMP and R software packages (e.g., the *UniDOE* R package of Zhang et al. (2018)); and random Latin hypercube designs (LHDs) (McKay et al. 1979).

A. A BASELINE TO COMPARE RESULTS

To assess the potential of `permute_and_stack`, this chapter explains how it performs (in terms of reducing ρ_{map}) in extending space-filling designs (SFDs) in comparison to shift-and-stack for several correlation-based column-reordering heuristics. We do this since we saw in Chapter III, Section D.2 that random permutations of the columns of $\mathbf{X}^0$ in constructing stacked designs were unsuccessful at obtaining the minimum ρ_{map}. This subsection defines the column-reordering heuristics used.

There is little documentation on how column-reordering heuristics may affect the properties of the resultant shift-and-stack design. The simple forward shift-and-stack approach from Sanchez and Sanchez (2019) uses the given ordering of the columns of $\mathbf{X}^0$, which we label as configuration B for the base ordering. Since the order of the columns does not change the design's correlation and space-filling measures, we investigated whether a simple easy-to-implement heuristic reordering of the columns of $\mathbf{X}^0$ could improve upon the designs constructed using shift-and-stack. Cioppa (2002) originally suggested this method. We define and examine nine correlation-based column-reordering heuristics to improve ρ_{map} values for stacked designs after applying shift-and-stack.

Let $\mathbf{R}$ be the $k \times k$ correlation matrix of $\mathbf{X}^0$. We use the non-diagonal values in $\mathbf{R}$ to reorder the columns before applying shift-and-stack. For each column, we calculate its maximum or average pairwise correlation as well as the maximum or average pairwise absolute correlation. We use the term "signed correlation" for reorderings using the raw correlation values (which can be positive or negative). Since the correlation values involve pairs of columns, we use random selection to break ties in selecting the order. We then arrange the columns of $\mathbf{X}^0$ by these values, either ascending, descending, or alternating. For example, the column-reordering heuristic labeled AD (for Absolute Descending) arranges the columns of $\mathbf{X}^0$ according to the descending maximum absolute pairwise correlation values of the columns of $\mathbf{R}$. The column-reordering heuristic labeled AA (for

Absolute Ascending) arranges the columns of $\mathbf{X}^0$ using the ascending maximum absolute correlation values of the columns of **R**.

Table 8 shows **R** for the 8×7 cataloged uniform design (UD) from Fang et al. (2000b), which serves as $\mathbf{X}^0$ in this example. Table 9 shows the labels and values of the heuristics used to reorder the columns of $\mathbf{X}^0$. For example, for SD (Signed Descending), the values are arranged from the largest to the smallest of the columns' signed pairwise correlations. In this case, columns 4 and 5 of $\mathbf{X}^0$ are randomly assigned to positions 1 and 2 in the new reordering corresponding to SD. When shift-and-stack is applied to SD, a column with the lowest signed correlation is appended under a column with the highest signed correlation. Table 10 shows the column reordering of $\mathbf{X}^0$ corresponding to SD.

Table 8. **R** for the 8×7 uniform design (UD) from Fang et al. (2000b).

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
X_1	1.000	−0.166	−0.214	0.071	−0.047	0.166	0.190
X_2	−0.166	1.000	−0.238	−0.023	0.023	0.095	0.023
X_3	−0.214	−0.238	1.000	−0.023	0.023	0.071	−0.095
X_4	0.071	−0.023	−0.023	1.000	0.309	0.023	−0.047
X_5	−0.047	0.023	0.023	0.309	1.000	0.071	0.000
X_6	0.166	0.095	0.071	0.023	0.071	1.000	−0.023
X_7	0.190	0.023	−0.095	−0.047	0.000	−0.023	1.000

Table 9. Ten correlation-based column-reordering heuristics using **R** from Table 8.

Label	Description	Example
B	As is (i.e., the column order given by $\mathbf{X}^0$)	
SD	Max signed correlation, Descending	0.309, 0.309, . . . , 0.095, 0.071
AD	Max absolute correlation, Descending	0.309, 0.309, . . . , 0.190, 0.166
SA	Max signed correlation, Ascending	0.071, 0.095, . . . , 0.309, 0.309
AA	Max absolute correlation, Ascending	0.166, 0.190, . . . , 0.309, 0.309
aSD	Avg signed correlation, Descending	0.067, 0.063, . . . , -0.048, -0.079
aAD	Avg absolute correlation, Descending	0.143, 0.111, . . . , 0.075, 0.063
aSA	Avg signed correlation, Ascending	-0.079, -0.048, . . . , 0.063, 0.067
aAA	Avg absolute correlation, Ascending	0.063, 0.075, . . . , 0.111, 0.143
AltH	Alternating Max, Min correlation	0.309, -0.238, . . . , -0.023, 0.000

Table 10. $\mathbf{X}^0$ (left) and SD reordered $\mathbf{X}^0$ (right).

X_1	X_2	X_3	X_4	X_5	X_6	X_7
7	7	1	4	2	5	2
8	5	6	5	5	8	8
3	4	3	8	8	7	3
5	2	8	3	6	4	1
4	6	2	2	7	2	7
2	3	5	1	1	6	5
1	8	7	6	4	3	4
6	1	4	7	3	1	6
$\mathbf{X}^0$ with its given column ordering, which we label case (B)						

X_5	X_4	X_7	X_1	X_6	X_2	X_3
2	4	2	7	5	7	1
5	5	8	8	8	5	6
8	8	3	3	7	4	3
6	3	1	5	4	2	8
7	2	7	4	2	6	2
1	1	5	2	6	3	5
4	6	4	1	3	8	7
3	7	6	6	1	1	4
$\mathbf{X}^0$ with its columns reordered by the Signed Descending (SD) heuristic						

The SD column reordering of $\mathbf{X}^0$ (Table 10 right) shows an example of one of the column-reorderings of the design. The correlation values used to generate SD are 0.309, 0.309, 0.190, 0.190, 0.166, 0.095, and 0.071. The first two columns of the SD reordered $\mathbf{X}^0$ contain X_5 and X_4 from the original $\mathbf{X}^0$. These columns contain the maximum signed correlation value of 0.309. Thus, there can be several orders for a given column-reordering heuristic. In this case, due to tied pairwise correlations, four reorderings satisfy the SD column reordering configuration. We select one of those four reorderings at random, with each having the same likelihood of being chosen.

B. EXTENDING CATALOGED DESIGNS

This subsection shows how well `permute_and_stack` does in extending cataloged designs. A fundamental limitation of cataloged designs is that they may not be available in the desired dimensions. For example, the uniform designs (UDs) from Fang et al. (2000b) have a maximum DP size (n) of 30. Also, the maximin L_2 distance designs from van Dam et al. (2009) are for limited n and k.

1. Extending Cataloged Nearly Orthogonal Latin Hypercubes (NOLHs)

> Despite the tremendous amount of uncertainty about potential future conflicts, decisions must be made (such as what equipment to purchase, how to organize units, and how to use future forces) that will affect large sums of money and affect many lives.
>
> —Kleijnen et al. (2005), p. 269

Providing computer experimenters with more options by extending the catalog of space-filling and nearly orthogonal Latin hypercubes (NOLHs) at https://harvest.nps.edu was the genesis of this research. Due to their ability to allow analysts to explore a breadth of diverse metamodels and an easy-to-use design spreadsheet (Sanchez 2005), these designs have underpinned scores of simulation-based national security studies (Kleijnen et al. 2005, MacCalman 2013, Sanchez et al. 2012). Thousands of computing hours were used to create space-filling NOLHs of the following sizes: 17×7, 33×11, 65×16, 129×22, and 257×29 (Cioppa and Lucas 2007). While these designs have proven useful, they allow relatively few options compared to potentially useful n and k combinations. Hence, shift-and-stack has been extensively used to leverage the work that went into constructing these designs to increase the number of design points (DPs), see Lin (2018), Kesler et al. (2019), and Sanchez et al. (2020, p. 1137), for example.

With the 33×11 NOLH in the spreadsheet as the base design $\mathbf{X}^0$, Figure 7 plots ρ_{map} values for up to ten extensions using `permute_and_stack` and the ten correlation-based column-reordering heuristics defined above. We see that `permute_and_stack` yields extended designs of size $33(s+1)\times 11$ that have lower ρ_{map} values than every shift-and-stack alternative for all s. We also observe that none of the heuristics consistently outperforms the others and that ρ_{map} can increase in an extension. We observe the latter in the nonmonotonic behavior of ρ_{map} for the absolute ascending (AA) and absolute descending (AD) shift-and-stack column-reordering heuristics. For $s = 1$, the extended design $\mathbf{X}^1$ obtained through `permute_and_stack` has a ρ_{map} of 0.010, while the ten $(\mathbf{X}_s)^1$ designs from shift-and-stack have ρ_{map} values ranging from 0.015 to 0.022. As

additional stacks (i.e., batches of DPs) are added, the ρ_{map} values seem to be converging towards zero.

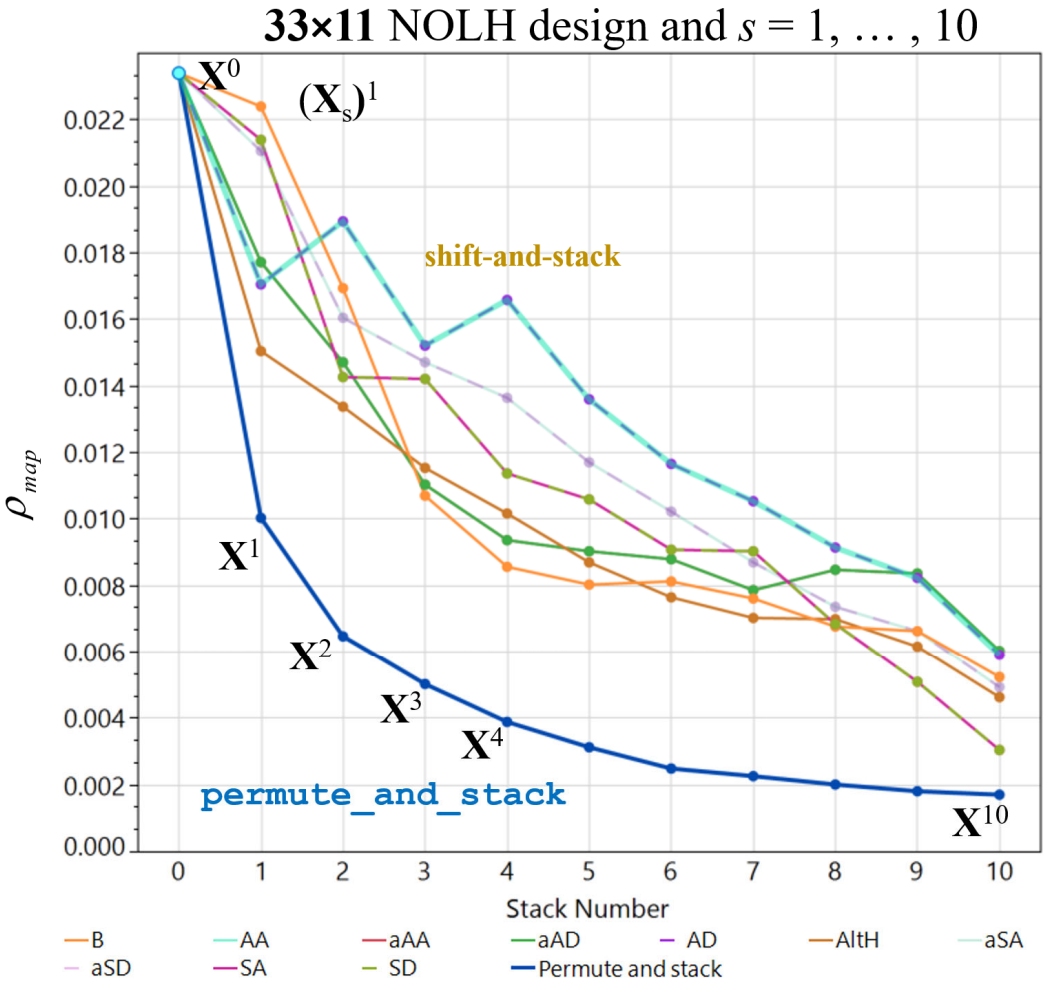


Figure 7. The 33×11 NOLH design $\mathbf{X}^0$ and 110 extended designs using `permute_and_stack` (blue) and ten applications of shift-and-stack (light colors).

Figure 8 plots ρ_{map} values for up to ten extensions using `permute_and_stack` and shift-and-stack with the ten column-reordering heuristics for 33×11, 65×16, 129×22, and 257×29 space-filling NOLHs. We omit the 17×7 NOLH since it is column-orthogonal; thus, any stacking of a permutation of its columns yields an orthogonal design. For all four

initial base designs, `permute_and_stack` outperforms all the alternatives at all extensions, with greater improvement occurring with smaller NOLHs.

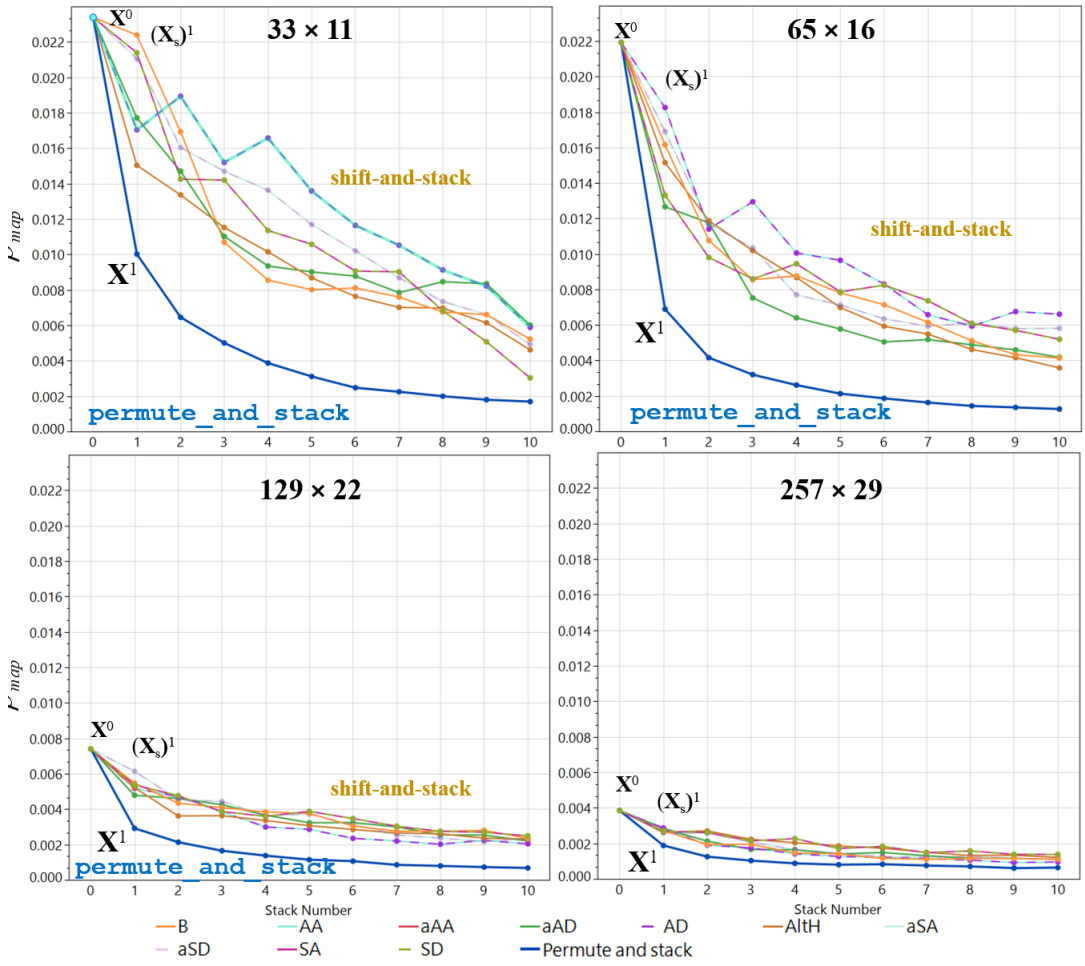


Figure 8. Comparative plot between `permute_and_stack` (blue) and shift-and-stack (light colors) for 33×11, 65×16, 129×22, and 257×29 space-filling NOLHs. The `permute_and_stack` designs dominate shift-and-stack in terms of ρ_{map}.

We have seen previously that good correlation does not necessarily mean good space-filling (recall the 2^k designs). What follows is a multi-criteria comparison, ρ_{map} and $(ML_2)^2$ discrepancy, for NOLH designs extended. Figure 9 shows 44 extensions for up to four stacks resulting from `permute_and_stack` and applying shift-and-stack to the ten column reorderings of $\mathbf{X}^0$. Preferred outcomes are in the bottom left of the figure; i.e., lower

ρ_{map} and $(ML_2)^2$ values. The improvement sequences can be described as beginning in the top right and ending towards the bottom left. We see that additional stacks tend to improve both measures. For each stack number, we see that `permute_and_stack` obtains a lower ρ_{map} and all of the designs have similar $(ML_2)^2$ values.

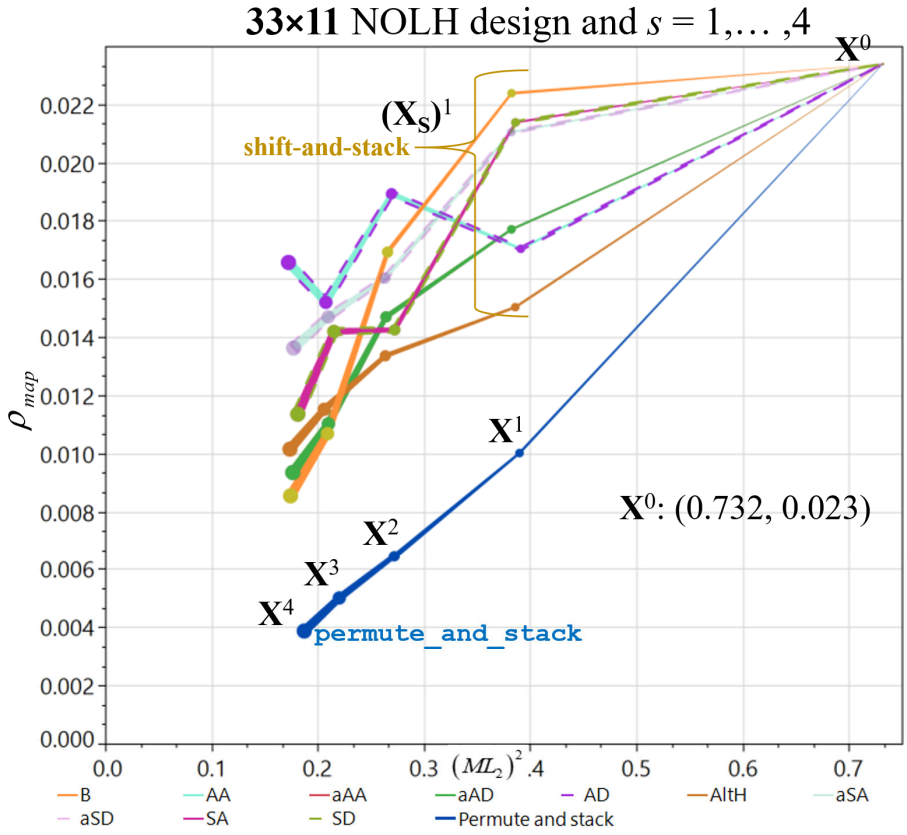


Figure 9. Multi-objective ρ_{map} and $(ML_2)^2$ comparison for extended NOLH designs up to four applications of shift-and-stack and `permute_and_stack`.

For each of the 33×11, 65×16, 129×22, and 257×29 space-filling NOLHs, `permute_and_stack` has been applied to obtain ten optimal extensions. Appendix A shows the ρ_{map} and $(ML_2)^2$ results for these four designs for up to ten stacks. In applying `permute_and_stack` to larger designs (e.g., $k = 22$ or 29), the optimization was limited to two hours of search time for each stack, even if the MIP gap tolerance had not been met. The results show that `permute_and_stack` can produce excellent design extensions

(in terms of ρ_{map}) even when the optimization doesn't have time to run to completion—as evidenced by the fact that `permute_and_stack` has produced good results for k as large as 29. How well it performs for even larger designs as a function of the allotted search time is left for future research.

While the stacked designs are no longer Latin hypercubes, for each factor, the DPs are balanced across n values (or levels). That is, each of the n input values for every factor appears exactly $1 + s$ times in a column of $\mathbf{X}^s$. This property holds for any shifted-and-stacked LHD.

Given $\mathbf{X}^0$, storing the NOLH extended designs, $\mathbf{X}^1,\ldots,\mathbf{X}^{10}$, is straightforward using the corresponding permutations $p^1,\ldots, p^{10}$ for the ten new designs. Specifically, the permutation sequence specifies the optimal stacks to be appended, $\mathbf{X}^{p^1},\ldots,\mathbf{X}^{p^{10}}$, for each stack. Table 11 lists the optimal permutations and the resultant ρ_{map} values for the ten extensions (i.e., appended stacks) of the cataloged 33×11 NOLH designs. The final stacked design's ρ_{map} value in Table 11 is approximately seven percent of the ρ_{map} value of $\mathbf{X}^0$. With ten added stacks, the final design has 363 DPs. It is worth noting that the original base NOLHs all have a point in the center of χ. Thus, there are s replicates of the center point in $\mathbf{X}^s$, which can be removed. Appendix C provides the optimal permutations and the resultant ρ_{map} values for up to ten extensions for the other NOLHs in the catalog.

Table 11. Optimal permutations for the 33×11 NOLH through ten stacks and the corresponding ρ_{map} values for $\mathbf{X}^s$.

Optimal `permute_and_stack` permutations $p^1,\ldots, p^{10}$:
1: '[9,8,4,1,10,3,2,0,5,6,7]', 2: '[2,7,10,8,1,0,4,9,5,6,3]', 3: '[10,5,2,0,7,9,1,6,4,8,3]', 4: '[3,0,7,10,8,6,9,4,1,2,5]', 5: '[8,4,10,9,6,0,5,1,2,7,3]', 6: '[10,0,4,3,2,9,6,7,8,1,5]', 7: '[3,5,10,4,1,6,9,0,7,8,2]', 8: '[1,8,3,9,7,5,2,4,10,0,6]', 9: '[2,6,5,9,1,3,0,10,8,7,4]', 10: '[8,7,0,6,5,2,1,4,9,10,3]'.
ρ_{map}:
1: [0.0100], 2: [0.0064], 3: [0.0050], 4: [0.0038], 5: [0.0031], 6: [0.0024], 7: [0.0022], 8: [0.0020], 9: [0.0018], 10:[0.0017].

2. Extending Cataloged Uniform Designs (UDs)

This section extends cataloged, discrepancy-based, uniform designs (UDs) (Fang et al. 2006, Fang et al. 2000a, 2000b) to sequentially add batches of n DPs that achieve superior correlations in the extended space using `permute_and_stack`. Figure 10 presents the cataloged 8×7 UD example of Chapter III and four applications of shift-and-stack for the ten column-reordering heuristics and `permute_and_stack`, as was discussed previously. We see that constructing a design to minimize discrepancy does not guarantee low correlation, as the ρ_{map} value of $\mathbf{X}^0$ is greater than 0.30. However, just one extension with `permute_and_stack` reduces ρ_{map} to 0.0952. As before, `permute_and_stack` outperforms all the column-reordering heuristics by a substantial amount at each stack. After adding three stacks, the extended design satisfies the nearly-orthogonal criterion.

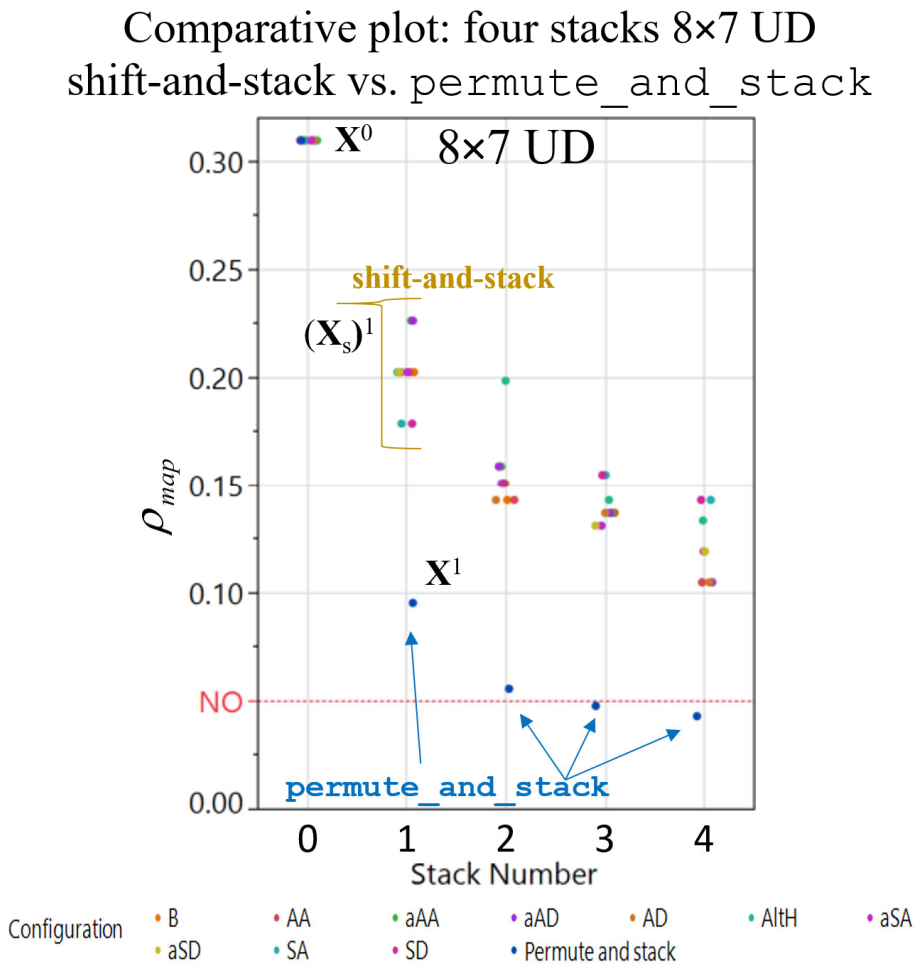


Figure 10. Four shift-and-stack and `permute_and_stack` applications to an 8×7 uniform design (UD) example from Fang et al. (2000b). Jitter is added to the points so that ties can be identified. The nearly orthogonal (NO) criterion is marked with a dashed line at 0.05.

Figure 11 shows various $n \times k$ cataloged UDs and four applications of shift-and-stack and `permute_and_stack`. For each design size, `permute_and_stack` dramatically exceeds all of the column-reordering shift-and-stack heuristics in reducing ρ_{map}. All of the $\mathbf{X}^1$ designs in Figure 11 surpass the UD catalog's limit of 30 DPs. These new extended designs allow researchers to leverage the extensive effort that went into making the cataloged UDs.

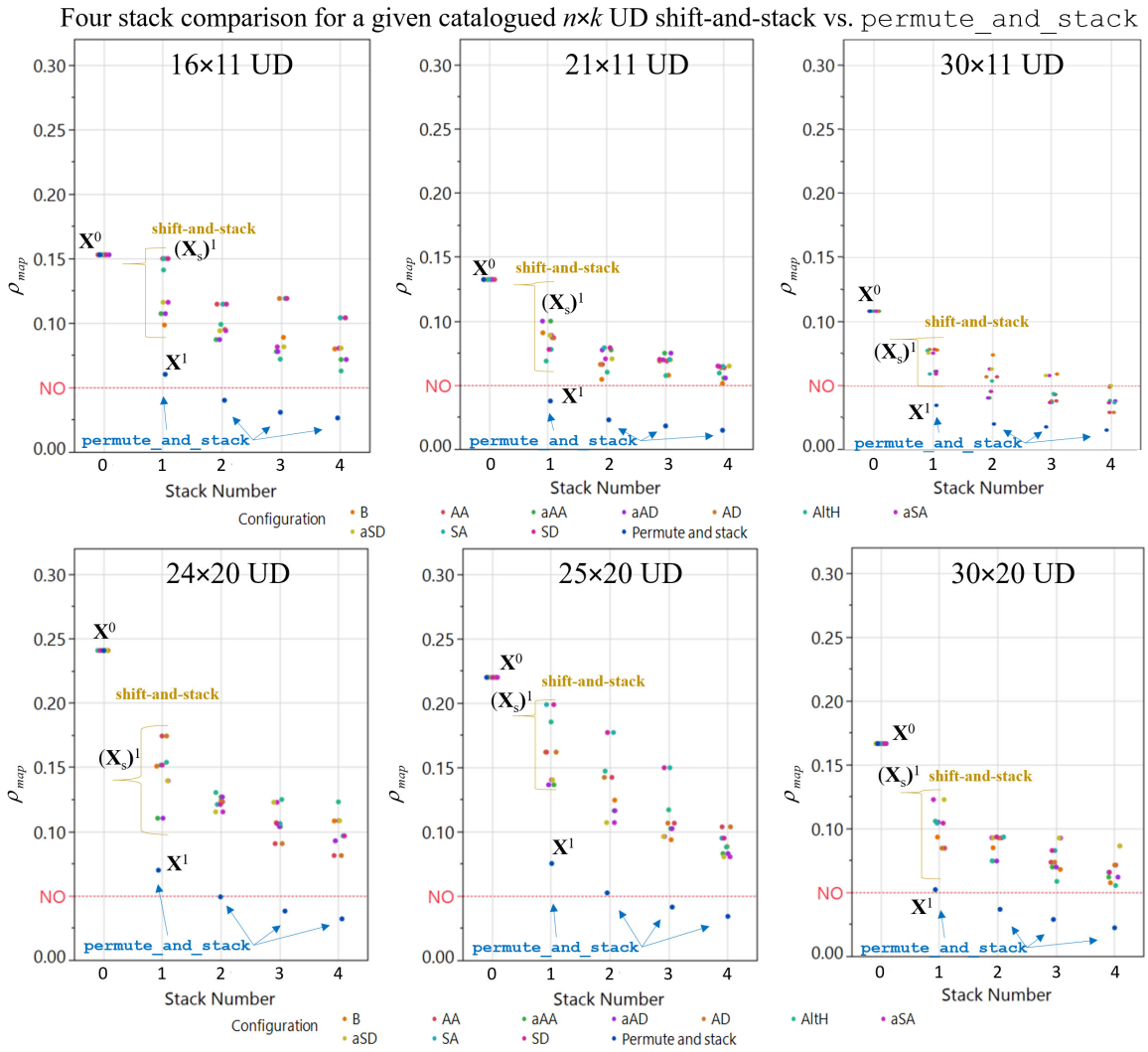


Figure 11. Four applications of shift-and-stack and `permute_and_stack` to UDs of size 16×11, 21×11,…, and 30×20 from the UD catalog of Fang et al. (2000b).

Figure 12 shows how the `permute_and_stack` and shift-and-stack heuristics do with respect to the correlation measure ρ_{map} and the discrepancy measure $(ML_2)^2$. Ideally, we would use a 33×11 base UD to match the size of our previous NOLH example. However, since $n > 30$, no such design is available at Fang et al. (2000b). Therefore, we use a cataloged 30×11 UD for the multi-objective assessment to provide a relatively close comparison. We see that `permute_and_stack` always yields the lowest ρ_{map} and the $(ML_2)^2$ values are similar at each stack.

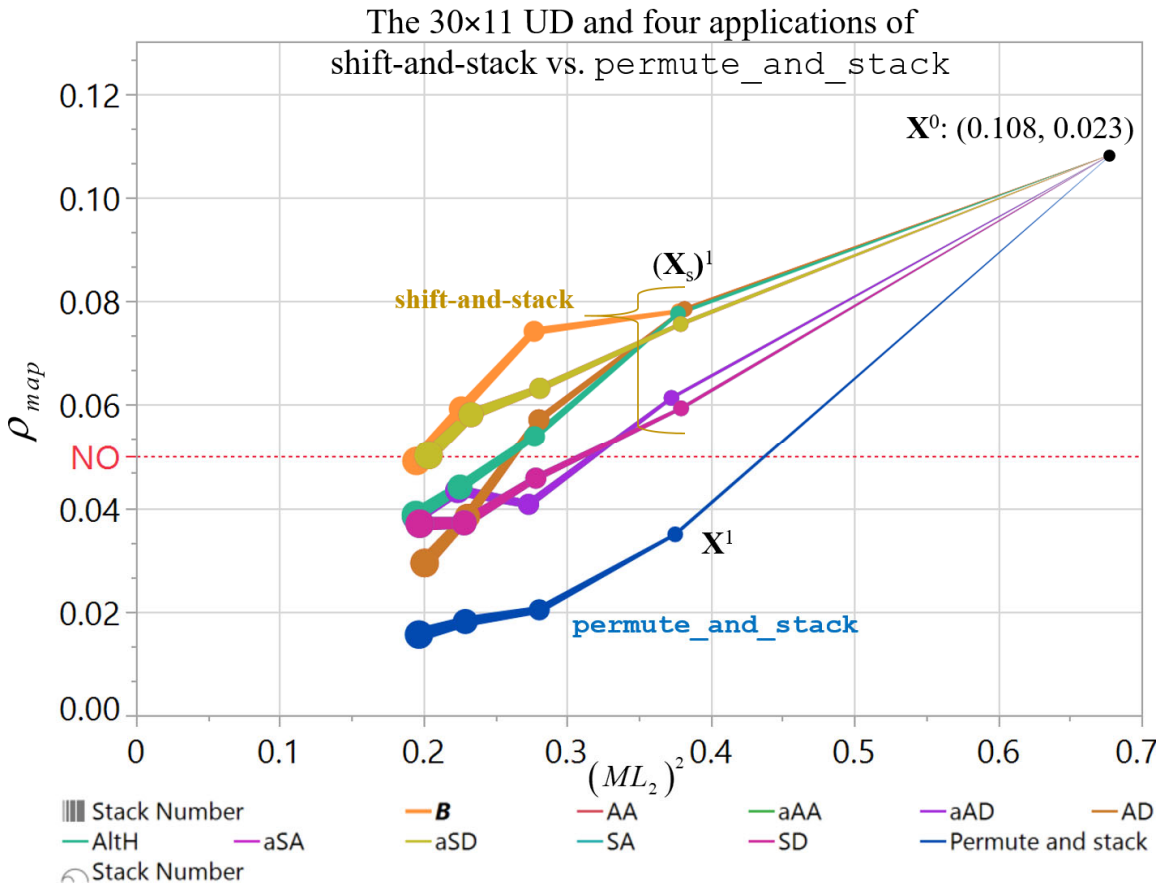


Figure 12. Multi-objective (ρ_{map} and $(ML_2)^2$) 30×11 UD comparison and resultant UDs extended after four applications of shift-and-stack and `permute_and_stack`.

Constructing a UD (Fang et al. 2000a) is nontrivial. It involves stochastically searching for the optimum UD by applying a threshold-accepting algorithm to minimize $(CL_2)^2$ (Fang et al. 2000b). After considerable computational effort, the UD that minimizes $(CL_2)^2$ was added to the catalog (Fang et al. 2000b). Although these designs achieve extraordinary discrepancy measures, the limited number of DPs constrain their utility. In addition, since the optimization algorithm does not consider correlation, some higher-than-desired correlations may result. Generally, we expect `permute_and_stack` to be of greater value in reducing ρ_{map} for SFDs that do not consider correlation in their construction, like UDs.

C. EXTENDING STOCHASTICALLY GENERATED SPACE-FILLING DESIGNS USING JMP AND R SOFTWARE PACKAGES

This section extends several classes of SFDs obtained from JMP and R software packages. Since there is stochasticity in the construction process, we create 100 designs for each class as Wang et al. (2020, p. 2) did. As before, the design dimensions are 33×11. After empirically assessing the distribution of important measures of design characteristics, we extend all 100 base designs by applying `permute_and_stack` and shift-and-stack for the ten column-reordering heuristics. We also assess how `permute_and_stack` performs in comparison to designs created in the extended design space.

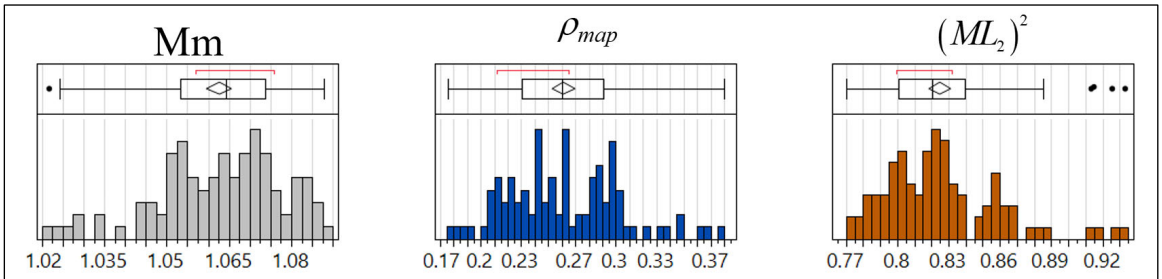
1. Extending JMP's MmLHDs

This subsection flows as follows. We first discuss why and how we create 100 MmLHDs in JMP. Then, we show design-generation stochasticity for multiple design characteristic measures. After that, we see how shift-and-stack performs (with respect to ρ_{map}) for our ten correlation-based column-reordering heuristics. Next, we compare the ρ_{map} values obtained by using the shift-and-stack heuristics to `permute_and_stack` for both a sequence of stacks and designs generated by JMP in the extended space. Finally, we look at how `permute_and_stack` affects both the correlation measure ρ_{map} and the space-filling measure $(ML_2)^2$ for multiple stacks.

We study the MmLHD first because it "seems to be the most commonly used experimental design for computer experiments in practice because of its simplicity and availability in software packages" (Joseph 2016, p. 31). When using a design generator that includes randomness, how should an experimenter select the design to use? One approach is to select the best design from many randomly generated designs, such as the "100 maximin L_2-distance LHD(n,k)'s by the R package *SLHD* [Sliced LHD] (Ba 2015) with default settings" as Wang et al. (2020, p. 2) did. Moreover, there can be multiple optimal solutions for a given design size. Citing Morris and Mitchell (1995), Joseph and Hung (2008, p. 179) "found that there are 142 different designs according to the maximin criterion" in MmLHDs of size 5×3.

To generate 100 MmLHDs efficiently, we utilize JMP's Scripting Language (JSL), save the designs and their random-number-generator seeds, and collect space-filling measures (SFMs) for follow-on data analysis. We compare 100 space-filling MmLHDs for $n = 33$ and $k = 11$ using JMP and its default settings.

Figure 13 summarizes multiple key design measures for the 100 33×11 MmLHDs. The Mm distance criterion is "maximized" during construction. The ρ_{map} value provides information on the design's correlation. The $(ML_2)^2$ value measures the design's discrepancy. The histograms in Figure 13 show the empirical distributions of the Mm, ρ_{map}, and $(ML_2)^2$ measures. Of note is the variability in these measures, in particular for ρ_{map}.



Mean	1.0627	Mean	0.2614	Mean	0.8247
Std Dev	0.0150	Std Dev	0.0417	Std Dev	0.0329
Upper 95% Mean	1.0657	Upper 95% Mean	0.2697	Upper 95% Mean	0.8313
Lower 95% Mean	1.0597	Lower 95% Mean	0.2531	Lower 95% Mean	0.8182
N	100	N	100	N	100
Minimum	1.0217	Minimum	0.1755	Minimum	0.7703
Maximum	1.0879	Maximum	0.3797	Maximum	0.9332
Median	1.0643	Median	0.2599	Median	0.8205

Figure 13. Mm, ρ_{map}, and $(ML_2)^2$ measures for 100 33×11 MmLHDs from JMP. Note that ρ_{map} ranges from 0.176 to 0.380.

JMP built all 100 designs intending to maximize the minimum Euclidean distance between any two DPs (i.e., the Mm criterion) while the DPs are constrained to an LHD. The best design has an Mm distance of 1.088. Here, the Mm distances for the 100 randomly constructed 33×11 MmLHDs range from 1.022 to 1.088, with a mean value of 1.063 and a 95 percent confidence interval (CI) on the mean of [1.059, 1.065]. The estimated standard error on the mean of Mm distance is 0.0015. The ρ_{map} values vary between 0.176 and 0.380,

with a mean value of 0.261 and 95 percent CI on the mean of [0.253, 0.270]. The estimated standard error on the mean of ρ_{map} is 0.0042. Concerning ρ_{map}, the worst-case design has more than double the ρ_{map} of the best-case design. The $(ML_2)^2$ discrepancy ranges from 0.770 to 0.933, with a mean value of 0.825 and a 95 percent CI on the mean of [0.818, 0.831]. The estimated standard error on the mean of $(ML_2)^2$ discrepancy is 0.0033. These results empirically display the consequence of randomness in design generation, as only the random-number-generator seeds were varied. We also see that a sample size of 100 is sufficient to obtain an understanding of the distributions of design measures. Moreover, Figure 14 shows that none of the 100 33×11 MmLHDs even approach the nearly orthogonal standard of $\rho_{map} \leq 0.05$, even though such an LHD is possible (Cioppa and Lucas 2007).

Figure 14 displays the ρ_{map} values for the 100 33×11 MmLHDs constructed by JMP in two plots (left and right). Low ρ_{map} values are preferred. The strip chart shows individual points (left), visually reinforcing the 100 random executions of JMP. There are 87 unique ρ_{map} values. An important takeaway from Figure 14 is the observed variability of ρ_{map}. An experimenter needs to understand this to ensure that they generate enough designs to obtain one with good properties. The box plot (right) displays the variability of ρ_{map} using quartiles. We will use box plots for most of our visualizations below.

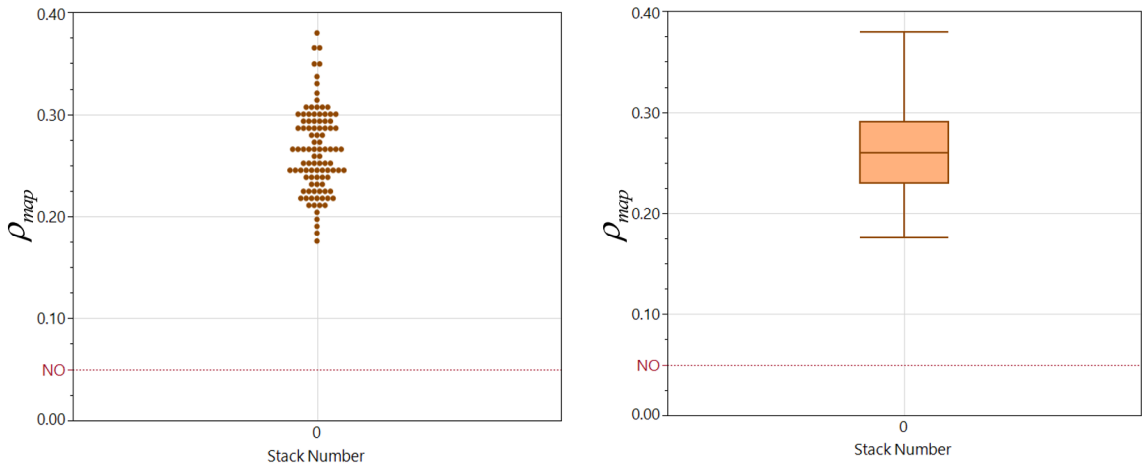


Figure 14. Strip chart and box plot of ρ_{map} values for 100 33×11 MmLHDs constructed using JMP. The individual points (left) show the ρ_{map} values of 100 $\mathbf{X}^0$ matrices. We see considerable variably in ρ_{map}, with none of the designs being nearly orthogonal.

Figures 15–17 show how adding sequential stacks generally reduces ρ_{map}. We explain how sequential stacking improves these 100 designs by seeing how ρ_{map} values tend to decrease as additional stacks are added using forward shift-and-stack with the base design configuration (B); i.e., without applying our column-reordering heuristics. Figure 15 shows the distributions of ρ_{map} for up to four iterations of forward shift-and-stack for 100 33×11 MmLHDs. A knee of the curve appears to be located near the second or third iteration. There also appears to be a decrease in variability (in terms of ρ_{map}) as the number of stacks (s) increases. This may follow from the increased number of DPs; i.e., $(s+1)n$, and the fact that the decreasing ρ_{map} values are bounded below by zero.

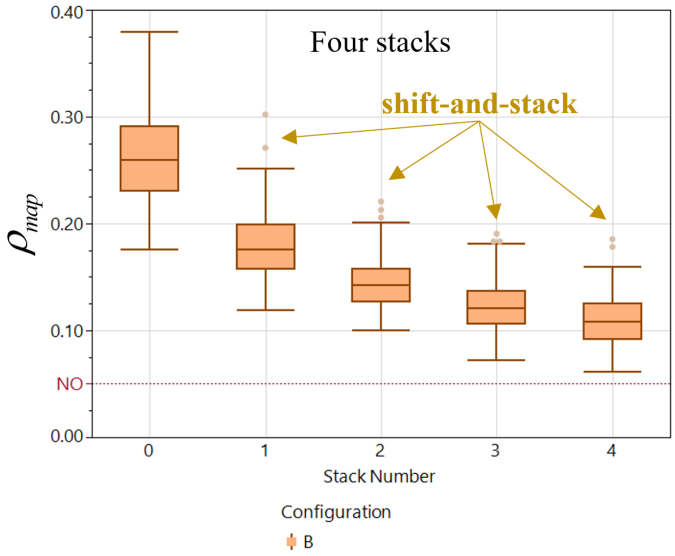


Figure 15. Four forward shift-and-stack iterations with box plots of ρ_{map} values for 100 33×11 MmLHDs using the base design configuration (B) of $\mathbf{X}^0$.

Figure 16 shows four iterations of shift-and-stack for the ten column-reordering heuristics in Table 9, starting with 100 33×11 MmLHDs constructed using JMP.

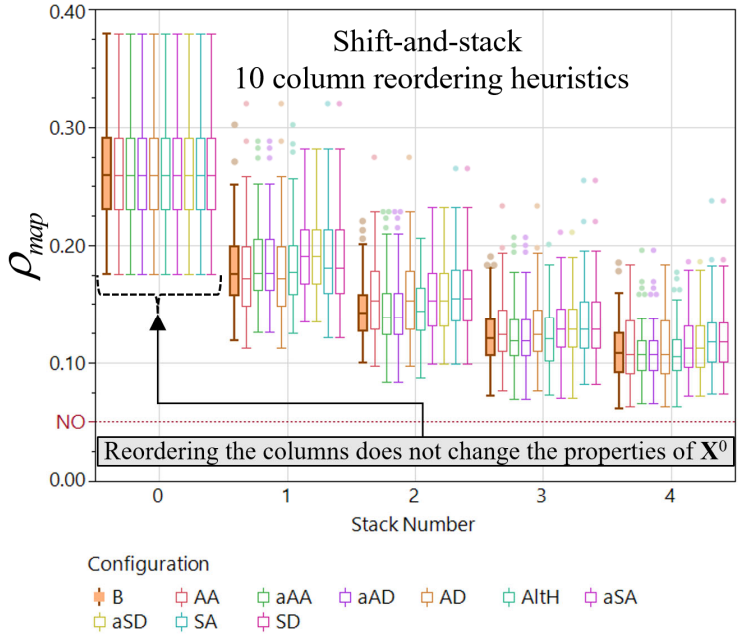


Figure 16. Box plots of ρ_{map} values for ten column-reordering heuristics through four iterations of shift-and-stack for 100 33×11 MmLHDs.

Figure 16 shows box plots of ρ_{map} values for the 4,000 new designs resulting from the ten heuristics and four iterations of shift-and-stack on 100 33×11 MmLHDs. The big takeaways are: (1) all ten heuristics have similar performance in terms ρ_{map}, (2) ρ_{map} tends to decrease as additional stacks are appended, and (3) there is substantial variability in ρ_{map} values at all stages.

Our inability to find a simple column-reordering heuristic drove us to consider an optimal matching of columns when stacking, ultimately leading to the `permute_and_stack` algorithm. Figure 17 shows how much better `permute_and_stack` performs in reducing ρ_{map} and its variability compared to all the shift-and-stack heuristics. By the third optimal permute and stack, the average ρ_{map} is 0.047, which is nearly orthogonal (i.e., $\leq$ 0.05)—this is an 82 percent reduction in correlation from the average of the 100 original 33×11 MmLHDs. The knee in the ρ_{map} `permute_and_stack` curve looks to be at roughly the third or fourth stack. Additionally, the "worst" `permute_and_stack` design is almost always better than the best shift-and-stacked design. Finally, we see that ρ_{map}'s variability is significantly lower when applying `permute_and_stack`.

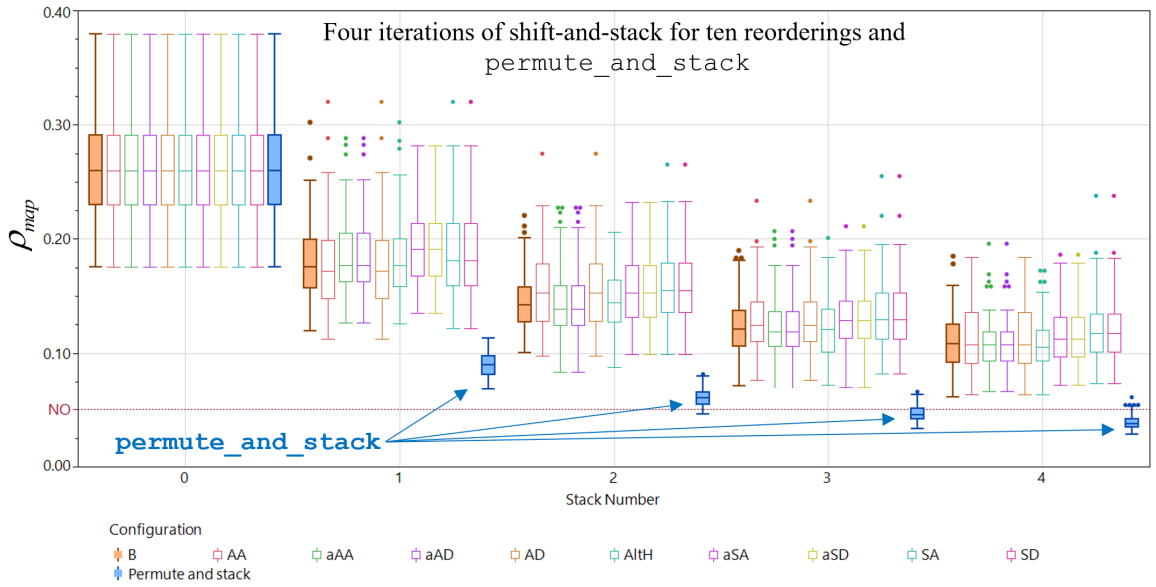


Figure 17. Box plots of ρ_{map} values at each of four stacks given the 100 33×11 MmLHDs using shift-and-stack for the ten column-reordering heuristics and `permute_and_stack` (blue). The `permute_and_stack` algorithm dramatically outperforms shift-and-stack in reducing ρ_{map} values and variability.

For many classes of designs, we cannot easily create designs of the new larger dimension, such as, the cataloged UDs or NOLHs. However, for MmLHDs generated by JMP, we can readily construct designs in the new dimensions. How will these new software-generated designs compare to `permute_and_stack`? To assess this, we use JMP to create 100 MmLHDs of size $(s+1)33\times11$ for $s=1,\ldots,4$ to compare to `permute_and_stack`. That is, we collect the ρ_{map} measures for 100 constructed MmLHDs of sizes 66×11, 99×11, 132×11, and 165×11. Figure 18 shows a comparative plot for four iterations of our ten shift-and-stack heuristics (light colors), `permute_and_stack` (blue), and the newly constructed MmLHDs of like size (grey) using JMP's default settings. We see that `permute_and_stack` dominates the newly constructed MmLHDs of like size with respect to ρ_{map}.

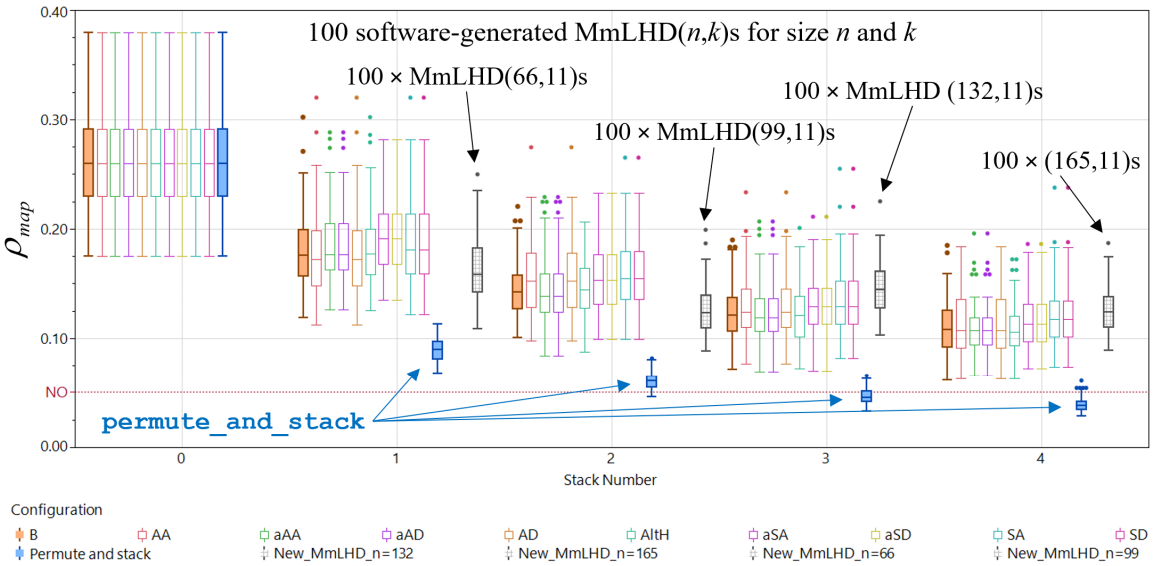


Figure 18. Comparative plot between `permute_and_stack` (blue), shift-and-stack (light colors), and 100 new (s+1)33×11 MmLHDs (grey). The best designs (in terms of ρ_{map}) are the `permute_and_stack` designs. None of the software-generated MmLHDs of size (s+1)33×11 are nearly orthogonal.

While `permute_and_stack` considers only correlation, we desire designs with low column correlations and good space-filling properties. To determine better designs for multiple objectives, we present a multi-objective (i.e., ρ_{map} and $(ML_2)^2$) empirical study of the effectiveness of shift-and-stack, `permute_and_stack`, and using software to create a new design in the extended space. Non-dominated multi-objective observations are shown for a given design size to help inform on the Pareto front (Lu et al. 2011, Pareto 1906) of these measures for JMP's MmLHDs.

Figure 19 plots the ρ_{map} and $(ML_2)^2$ values from the 100 33×11 MmLHDs JMP created. The y-axis (ρ_{map}) ranges from 0.00 to 0.40 and the x-axis ($(ML_2)^2$) from 0.00 to 0.95. We desire lower values for each measure. Thus, designs in the lower left of the plot are preferred. The nearly orthogonal (NO) cutoff value, ρ_{map} = 0.05, is labeled to denote the desired target. Observations in Figure 19 range from [0.16, 0.38] for ρ_{map} and [0.75, 0.95] for $(ML_2)^2$, which provide a relative reference prior to adding stacks with shift-and-

stack and `permute_and_stack`. There is a lot of variability in these "optimal" software-generated designs. An ellipse containing 90 percent of the observations is added.

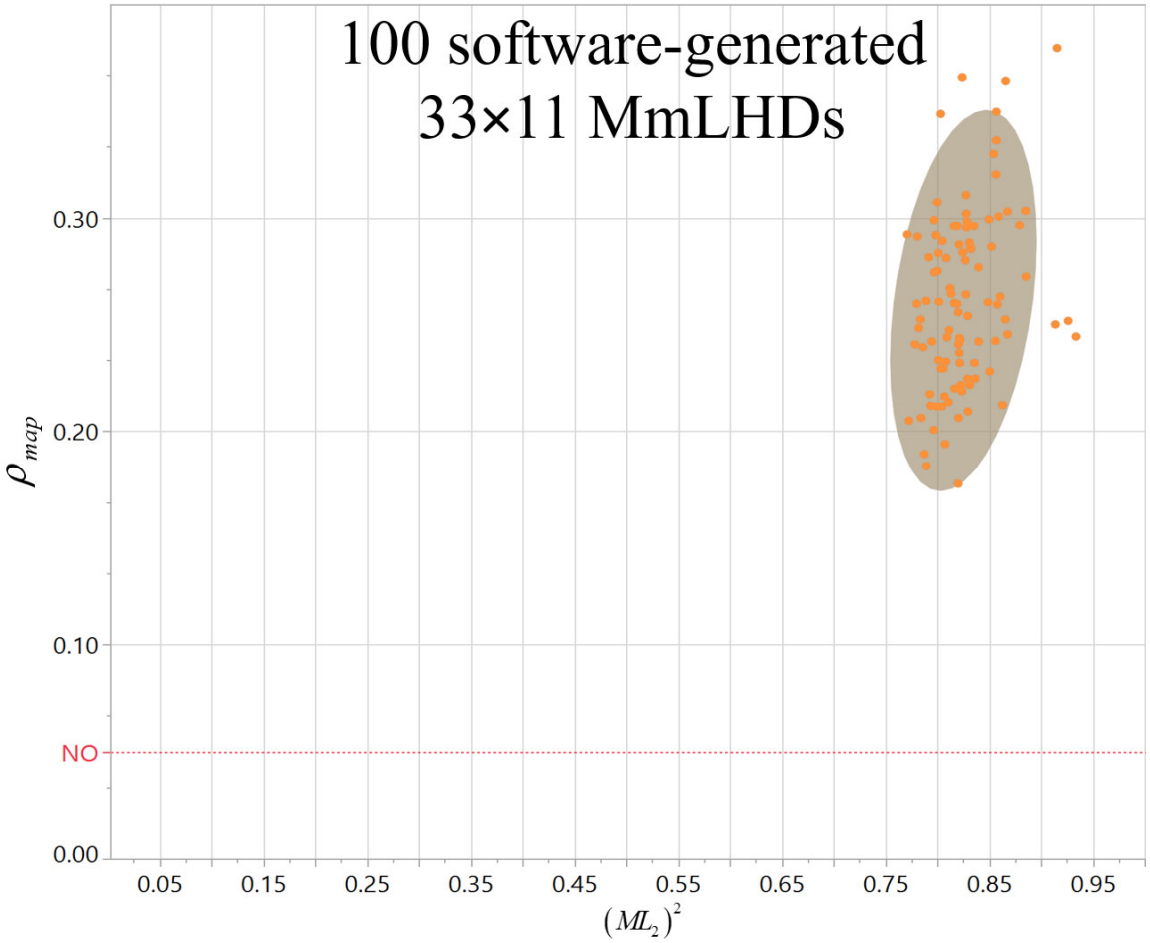


Figure 19. 100 33×11 MmLHDs prior to shift-and-stack and `permute_and_stack`. The shaded 90 percent ellipse shows an approximate ρ_{map} range of [0.17, 0.35] and $(ML_2)^2$ of [0.75, 0.90].

Subsequent figures show ρ_{map} and $(ML_2)^2$ values as stacks are appended using `permute_and_stack` and the shift-and-stack heuristics as well as those of 100 new design generations in JMP of size $(s+1)33\times11$. We begin with a single stack; i.e., $s = 1$. Figure 20 displays the ρ_{map} and $(ML_2)^2$ statistics for the 100 $\mathbf{X}^1$ designs obtained by `permute_and_stack` (in blue), the 1,000 $(\mathbf{X}_s)^1$ designs from our ten shift-and-stack heuristics (in a variety of colors), and the 100 new 66×11 MmLHDs from JMP. To help

visualize the distributions, we display ellipses that contain 90 percent of the observations. We see that all shift-and-stack heuristics are comparable, as most ellipses overlap. As we have seen above, `permute_and_stack` consistently produces the lowest ρ_{map} values. As expected, with a finer lattice, the 66×11 MmLHDs have the best $(ML_2)^2$ space-filling measures. Which design is preferred depends on how a practitioner values these two measures. Both `permute_and_stack` and the 66×11 MmLHDs show less variability (i.e., smaller ellipses).

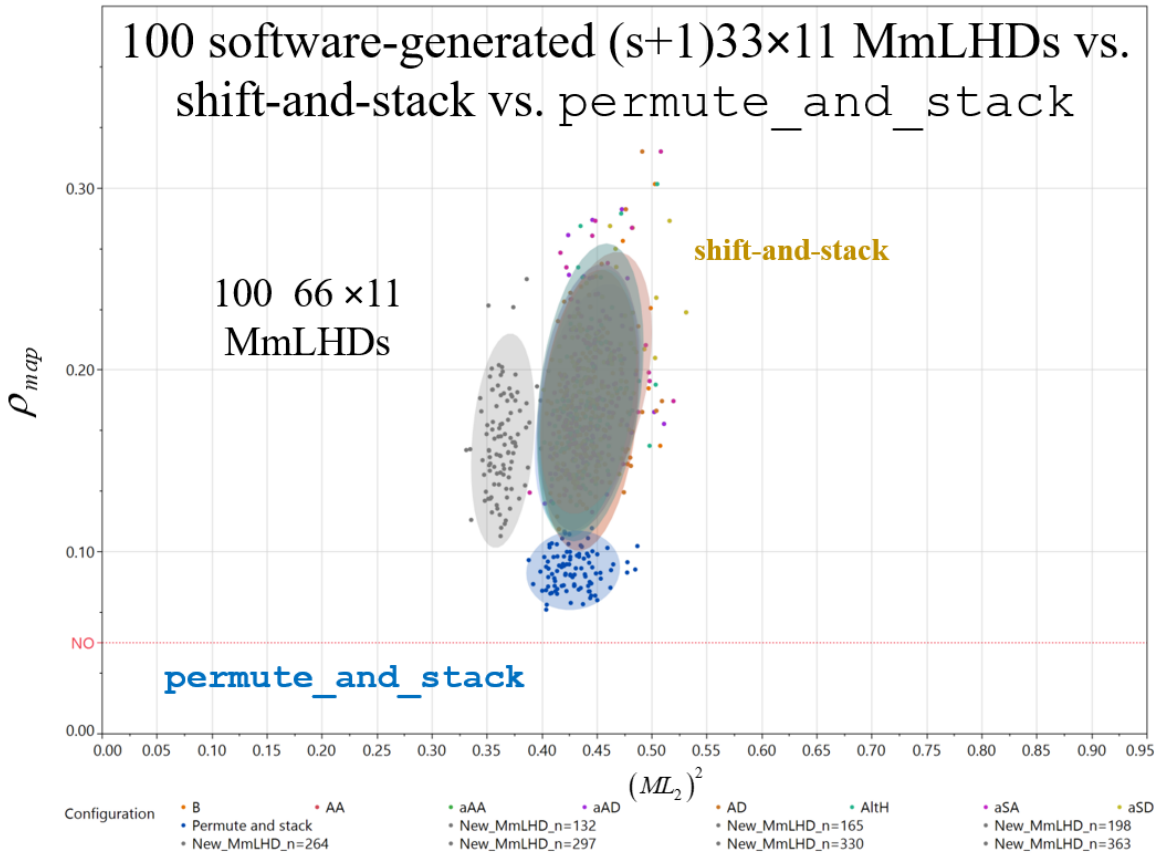


Figure 20. 100 33×11 MmLHDs after one application ($s = 1$) of forward shift-and-stack (light colors) and `permute_and_stack` (blue). 100 new 66×11 MmLHD constructions are plotted individually (grey).

Figure 21 displays the ρ_{map} and $(ML_2)^2$ values for the 100 $\mathbf{X}^s$ designs obtained by `permute_and_stack` and the 1,000 $(\mathbf{X}s)^s$ designs from our shift-and-stack heuristics

when s = 4. The plot also includes the 100 "big new designs," i.e., the 100 165×11 MmLHDs developed directly in JMP. Looking across the figures, we see improvement in each measure with more stacks (i.e., added DPs). Designs constructed using `permute_and_stack` perform far better than the ten shift-and-stack heuristics and the 165×11 MmLHDs with respect to ρ_{map}. All of the designs that are nearly orthogonal result from `permute_and_stack`. As expected, with five times more levels for each factor, the 165×11 MmLHDs have the best $(ML_2)^2$ values. From the 1,200 designs, only six are non-dominated—three of which were obtained by `permute_and_stack`. Appendix B contains the full sequence of four extensions.

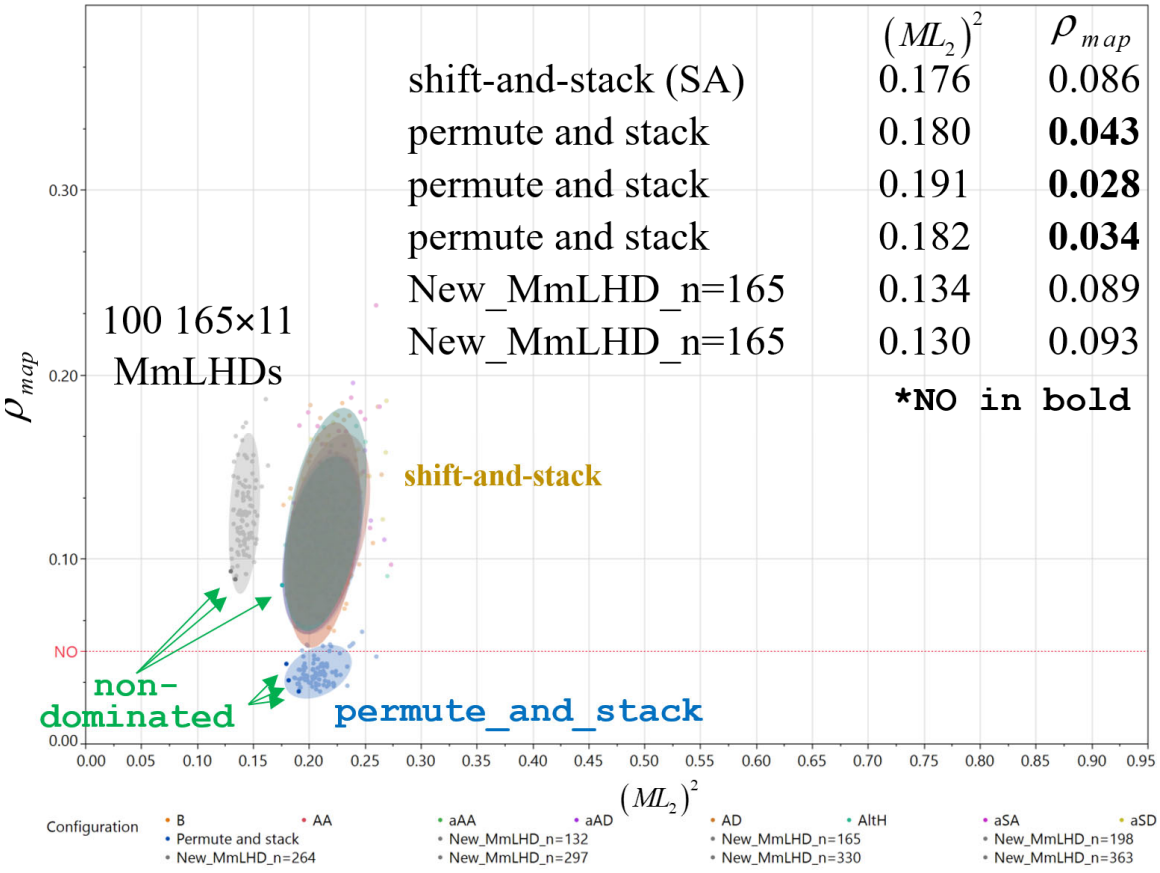


Figure 21. 100 33×11 MmLHDs with four extensions using shift-and-stack (colors) and `permute_and_stack` (blue) compared with 100 165×11 MmLHDs. The shaded ellipses contain 90 percent of the design measure coordinates for each of the dozen generation methods. From the 1,200 designs, only six are non-dominated (darkened).

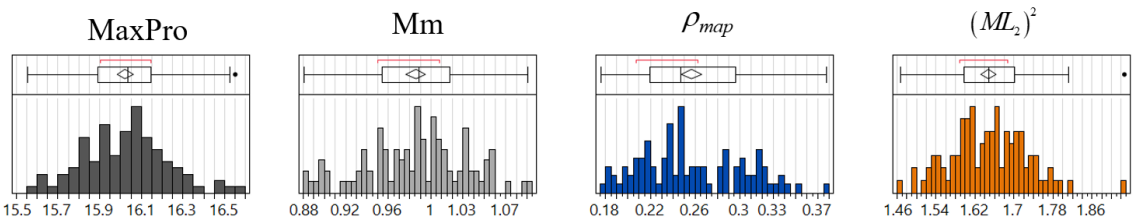
Empirically, Figure 21 reveals the trade-off possibilities for this difficult problem. For such situations, Jin et al. (2003, p. 546) write that it "is more practical to solve optimal design (of experiments) problems approximately." Here, only the six non-dominated designs (darkened) need to be considered if the goal is attaining a best (in terms of ρ_{map} and $(ML_2)^2$) 33×11 MmLHD extended in four batches or a 165×11 MmLHD. From the non-dominated designs, the best non-`permute_and_stack` ρ_{map} is 207% higher than the lowest ρ_{map} achieved using `permute_and_stack`. From the non-dominated designs, the best non-`permute_and_stack` $(ML_2)^2$ is 28% lower. For a given design type, size $n \times k$, and set of properties, we observe that the edge of the envelope created by the non-dominated points bound the multi-criteria Pareto front (Lu et al. 2011, Pareto 1906).

2. Extending R's Maximum Projection (MaxPro) Designs

Maximum projection (MaxPro) designs are a relatively new approach to generating SFDs that strive to "maximize space-filling properties to all subsets of factors." (Joseph et al. 2015, p. 371). The MaxPro optimality criterion of Equation (4) is a distance-based measure that "fav[o]r[s] more points towards the boundaries than at the cent[er]" (Joseph et al. 2015, p. 376), which can result in poor uniformity. This subsection explores how `permute_and_stack`, shift-and-stack using the ten column-reordering heuristics, and MaxPro designs generated in the extended space perform in reducing ρ_{map}. The MaxPro designs are generated by the R software package *MaxPro* (Ba and Joseph 2018) using the default settings.

Ba and Joseph (2018) recommend using a two-step approach in generating MaxPro designs. The first step is to use simulated annealing (SA) to build a MaxPro design with the design matrix constrained to an LHD. The result is an initial MaxProLHD, which serves as the starting point for the second step, an SA optimization without the LHD restriction. We follow Ba and Joseph's (2018) advice in constructing these designs and use the name MaxPro to denote the resultant design. We note that MaxPro designs typically have unevenly spaced levels, unlike the MaxProLHDs.

Figure 22 summarizes the distributions of multiple key design measures for 100 33×11 MaxPro designs produced by the R package *MaxPro*. As with MmLHDs in the previous section, we see substantial variability in the measures resulting from the 100 "optimal" 33×11 MaxPro designs. The best (minimum) MaxPro value achieved has a MaxPro value of 15.556. The mean MaxPro value is 16.025, and the 95 percent CI on the mean is [15.985, 16.025]. The worst (maximum) MaxPro value obtained is 16.553. Having explored the variability in the criterion for optimizing the designs, we consider other key quality measures. The Mm distance criterion for the 100 33×11 MaxPro designs ranges from 0.881 to 1.091. The worst ρ_{map} encountered is $\rho_{map} = 0.378$. The best ρ_{map} obtained is 0.176. The average ρ_{map} is 0.257. The $(ML_2)^2$ discrepancy ranges from 1.467 to 1.927, with a mean value of 1.648 and a 95 percent CI on the mean of [1.632, 1.663]. These results empirically illustrate the variability in multiple design properties resulting from the *MaxPro* R software package, as only the random-number-generator seeds were varied.



Mean	16.025	Mean	0.986	Mean	0.257	Mean	1.648
Std Dev	0.202	Std Dev	0.048	Std Dev	0.046	Std Dev	0.078
Upper 95% Mean	16.065	Upper 95% Mean	0.995	Upper 95% Mean	0.266	Upper 95% Mean	1.663
Lower 95% Mean	15.985	Lower 95% Mean	0.976	Lower 95% Mean	0.248	Lower 95% Mean	1.632
N	100	N	100	N	100	N	100
Minimum	15.556	Minimum	0.881	Minimum	0.176	Minimum	1.467
Maximum	16.553	Maximum	1.091	Maximum	0.378	Maximum	1.927
Median	16.037	Median	0.989	Median	0.247	Median	1.648

Figure 22. MaxPro, Mm distance, ρ_{map}, and $(ML_2)^2$ distributions for 100 33×11 MaxPro designs from the *MaxPro* R software package. The average ρ_{map} is 0.257, and it ranges from 0.176 to 0.378.

Figure 23 displays the ρ_{map} values for 100 33×11 MaxPro designs after applying `permute_and_stack`, shift-and-stack using the ten column-reordering heuristics, and MaxPro designs generated in the extended space of size $(s+1)33\times11$ for $s = 1,\ldots, 4$ stacks. The `permute_and_stack` designs have consistently lower ρ_{map} values than the shift-

and-stack heuristics and the new designs built in the extended space. A majority of the `permute_and_stack` designs achieve near orthogonality by $s = 3$ (i.e., three extensions). We also see much less variability in ρ_{map} values with `permute_and_stack` designs.

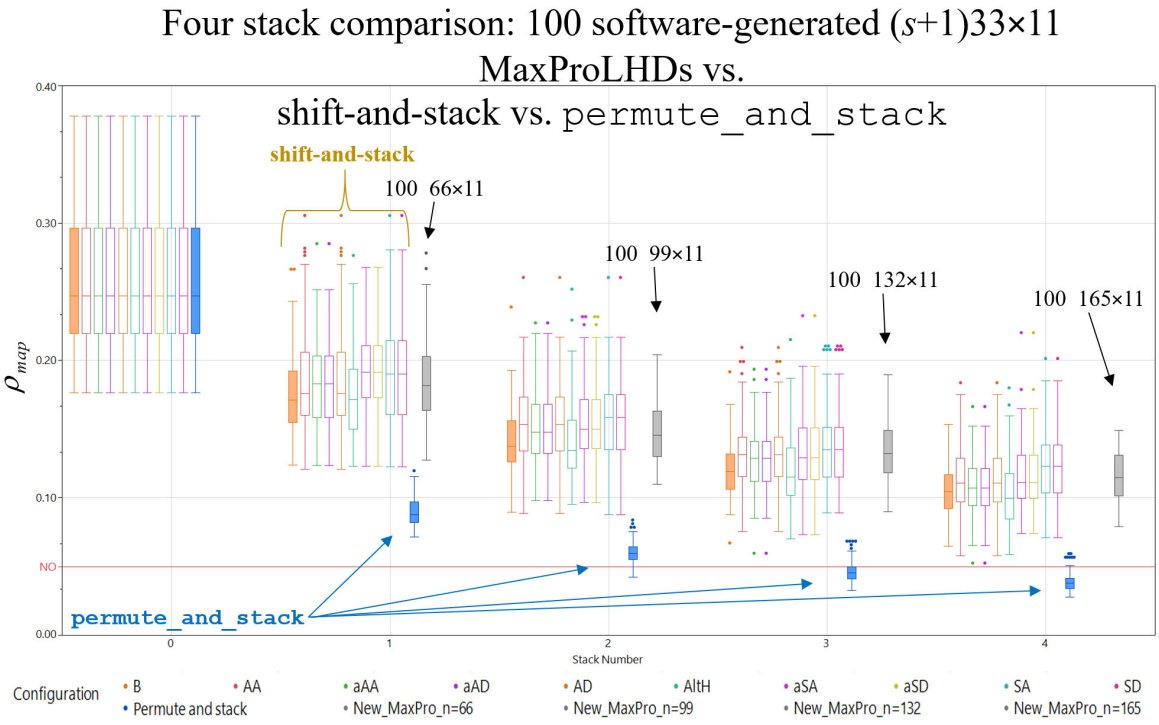


Figure 23. Box plots of ρ_{map} values at each of four stacks given 100 33×11 MaxPro designs using shift-and-stack (with ten column-reordering heuristics), `permute_and_stack` (blue), and 100 new $(s+1)33\times11$ MaxPro designs (grey) for $s = 1, \ldots, 4$. None of the MaxPro designs satisfy the nearly orthogonal criterion.

Figure 24 shows multiple design criteria (i.e., ρ_{map} and $(ML_2)^2$) considered together after applying `permute_and_stack` and shift-and-stack using the ten column-reordering heuristics for $s = 2$. The plot also contains 100 MaxPro designs constructed by the *MaxPro* R software package in the extended space. As before, designs in the lower-left corner are preferred. Once again, all of the shift-and-stack heuristics perform similarly. The `permute_and_stack` designs have by far the best ρ_{map} values, while the 99×11

MaxPro designs have the best $(ML_2)^2$ statistics. There are a total of 14 non-dominated designs, seven from `permute_and_stack` and seven from the 99×11 MaxPro designs. The best $(ML_2)^2$ value in the non-dominated set of MaxPro designs is 34.8 percent less (i.e., better) than the best $(ML_2)^2$ in the non-dominated set of `permute_and_stack` designs. However, this same non-dominated MaxPro design has a 207.1 percent higher ρ_{map} than the `permute_and_stack` design with the best ρ_{map} and 122.4% greater ρ_{map} than the `permute_and_stack` design with the best $(ML_2)^2$. The non-dominated points provide information on the tradespace and to what is possible for these two measures for SFDs of this size.

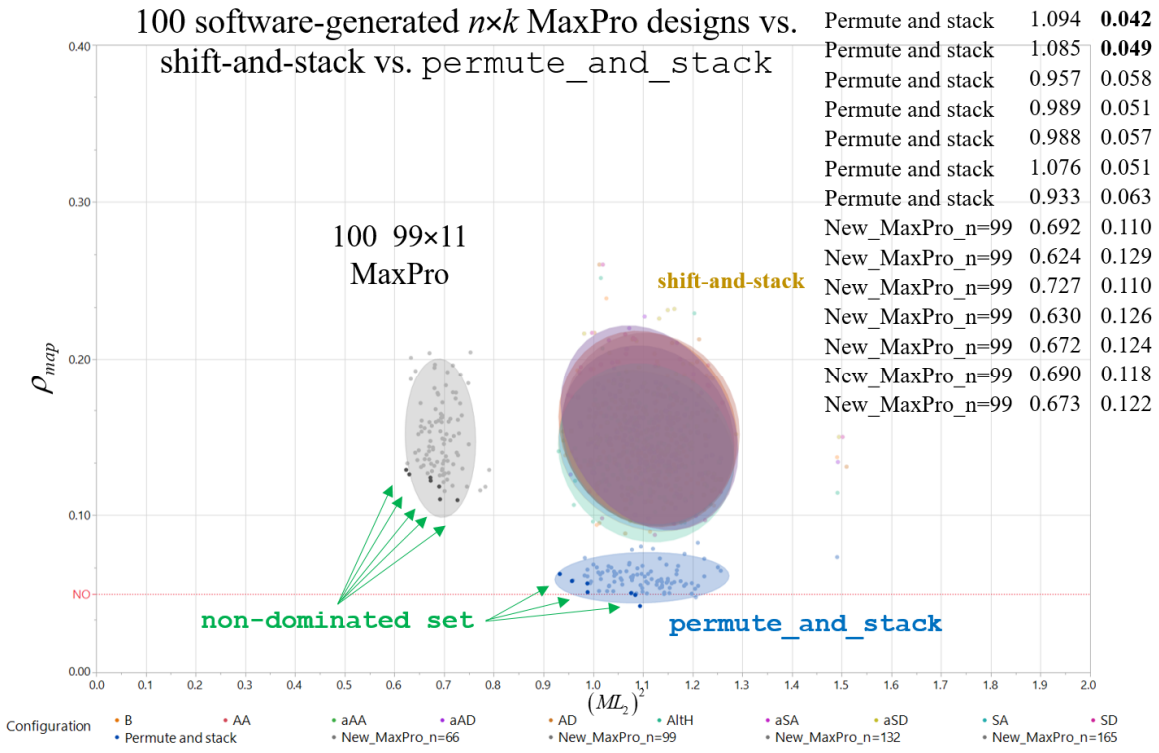
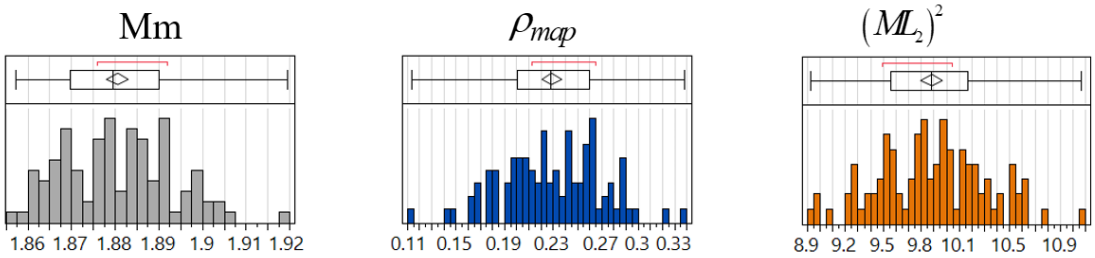


Figure 24. 100 33×11 MaxPro designs with two extensions ($s = 2$) of shift-and-stack with the ten column-reordering heuristics (colors) and `permute_and_stack` (blue) compared with 100 99×11 MaxPro designs. None of the 100 99×11 MaxPro designs generations (grey) in the extended space satisfy the nearly orthogonal criterion.

3. Extending JMP's Sphere-Packing (Mm distance) Designs

This section compares the performance of `permute_and_stack` and shift-and-stack in another design class—the sphere-packing design, which is JMP's Mm distance design. We compare 100 space-filling sphere-packing designs for $n = 33$ and $k = 11$ using JMP and its default settings for up to four stacks. We also see how `permute_and_stack` performs relative to sphere-packing designs created in the extended design space.

Figure 25 summarizes the distributions of multiple key design measures for 100 33×11 sphere-packing designs. We see that the Mm distances here are larger (as this is the design goal) than for the other SFDs of similar size. For example, the greatest MmLHD Mm distance achieved has a Euclidean distance of only 1.088, which is 43.33 percent lower than the best sphere-packing design, with a minimum Euclidean distance of 1.920. These results make sense, as other design classes optimize different criteria (e.g., MaxPro or UD) or have additional constraints (e.g., MmLHD). Of note is that ρ_{map} ranges from 0.114 to 0.338. We also see that the $(ML_2)^2$ values are much higher than in the other software-generated SFDs we study. This follows from the optimization criterion, where the DPs are pushed to the boundaries of the design space, which results in gaps in the interior and higher-than-desired uniformity values. For the 100 sphere-packing designs, the minimum $(ML_2)^2$ value is 8.923, whereas for the MmLHD the maximum $(ML_2)^2$ value is 0.933, almost an order of magnitude lower.



Mean	1.881	Mean	0.229	Mean	9.887
Std Dev	0.012	Std Dev	0.041	Std Dev	0.431
Upper 95% Mean	1.883	Upper 95% Mean	0.237	Upper 95% Mean	9.973
Lower 95% Mean	1.878	Lower 95% Mean	0.221	Lower 95% Mean	9.801
N	100	N	100	N	100
Minimum	1.857	Minimum	0.114	Minimum	8.923
Maximum	1.920	Maximum	0.338	Maximum	11.065
Median	1.879	Median	0.228	Median	9.881

Figure 25. Mm, ρ_{map}, and $(ML_2)^2$ measures for 100 33×11 sphere-packing (JMP's Mm distance) designs. Note that the mean ρ_{map} is 0.229, ranging from 0.114 to 0.338.

Figure 26 illustrates how much better `permute_and_stack` performs in reducing ρ_{map} and its variability compared to shift-and-stack and new sphere-packing designs created by JMP in the extended space. Once again, by the third stack, most of the `permute_and_stack` designs are nearly orthogonal.

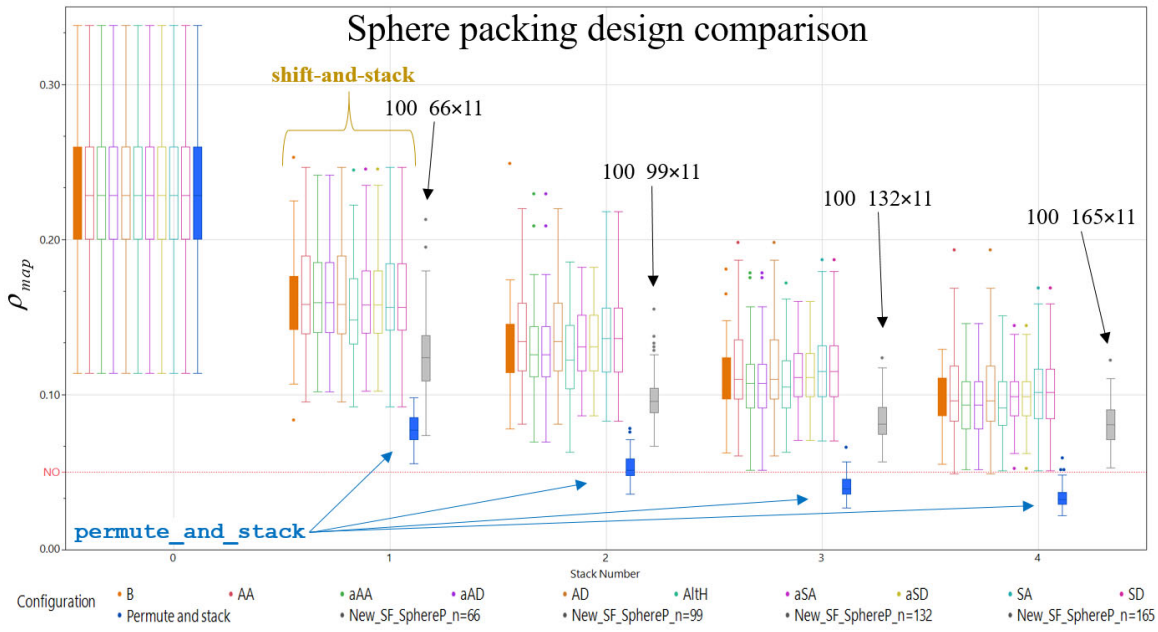


Figure 26. Comparative plot between `permute_and_stack` (blue), shift-and-stack (colors), and 100 (s+1)33×11 sphere-packing designs (grey). The best designs (in terms of ρ_{map}) are the `permute_and_stack` designs.

Figure 27 is a comparative plot of the distributions of ρ_{map} and $(ML_2)^2$ for 100 `permute_and_stack` (blue), 1,000 shift-and-stack with the column-reordering heuristics (colors), and 100 new sphere-packing designs in the extended space (grey) for $s = 2$. Once again, `permute_and_stack` yields the lowest ρ_{map} values, and the sphere-packing designs created in the extended space have the best $(ML_2)^2$ values. From these 1,200 designs, there are only five non-dominated ones. Which design is preferred depends on the experimenter's goals. These results provide trade-off options. The lowest ρ_{map} value in the non-dominated set of 99×11 sphere-packing designs is 86.1 percent higher than the minimum in the non-dominated set achieved by extending the 33×11 sphere-packing designs twice using `permute_and_stack`. Conversely, the minimum $(ML_2)^2$ value in the non-dominated set of 99×11 sphere-packing designs is 14.8% better than the minimum $(ML_2)^2$ value in the non-dominated set achieved using two applications of `permute_and_stack` to extend the 33×11 sphere-packing designs.

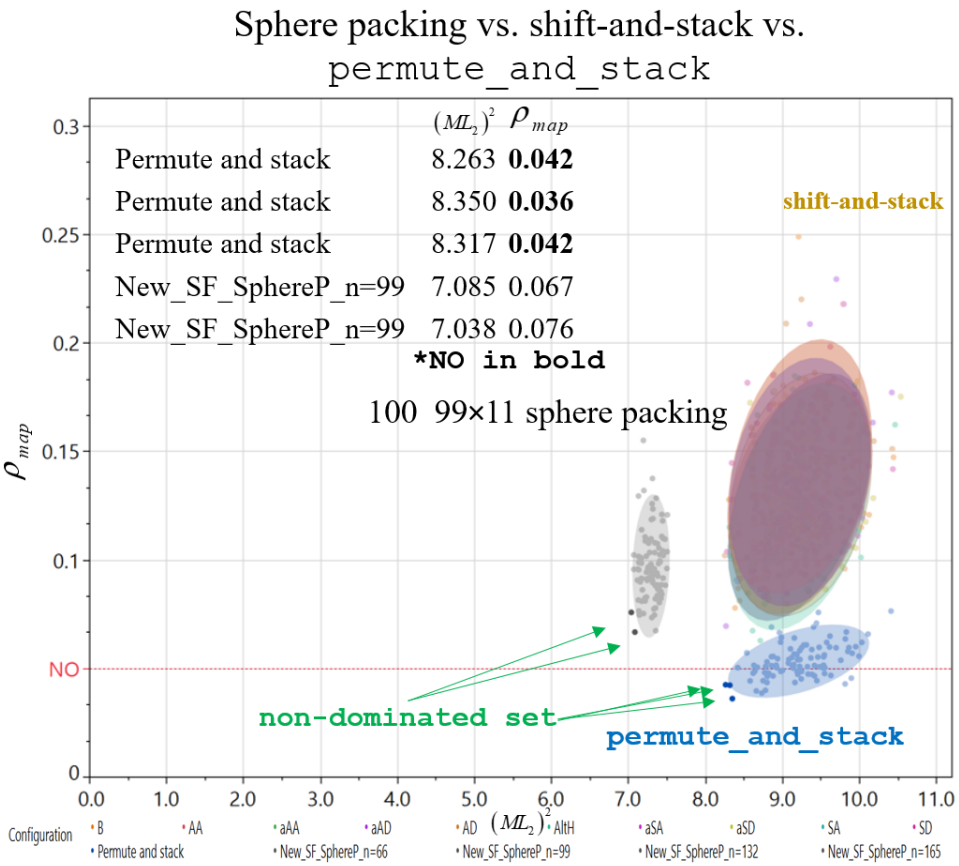


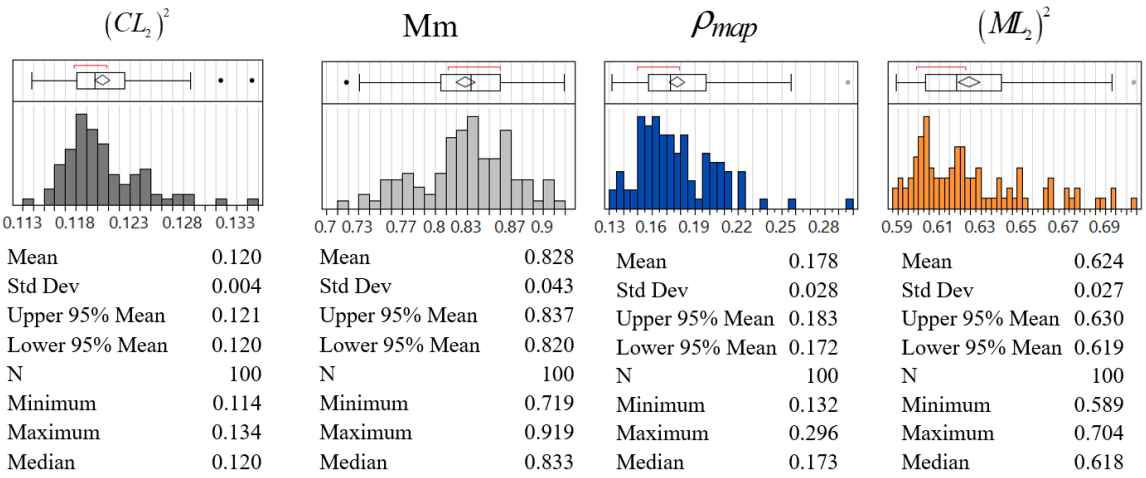
Figure 27. 100 (s+1)33×11 sphere-packing designs with s = 2 extensions using shift-and-stack (colors) and `permute_and_stack` (blue), as well as new generations for the extended space (grey). Note that many of `permute_and_stack` designs satisfy the $\rho_{map} \leq 0.05$ criteria.

4. Extending R's UniDOE Uniform Designs (UDs)

Zhang et al. (2018) introduced the R package *UniDOE*, which generates uniform designs (UDs) for a variety of distance and discrepancy criteria, including Mm distance, $(CL_2)^2$ discrepancy, and the mixture L_2 discrepancy. Of these, the mixture L_2 discrepancy defined by Zhou et al. (2013) is the default optimality criterion value for the software as the criterion is "more reasonable" (Ke et al. 2015, p. 741) than other uniformity criteria. Thus, the *UniDOE* package in R for the default settings was used to create 100 33×11 UDs.

Figure 28 shows the distributions of measures of design quality that result. *UniDOE* builds UDs from a stochastic and adaptive threshold accepting algorithm that augments

integer LHDs, optimizing the mixture L_2 discrepancy for uniformity, which Ke et al. (2015) explain overcomes some of the limitations of the $(CL_2)^2$. Ke et al. (2015) write that the $(CL_2)^2$ "covers the points near the center insufficiently and this measurement will cause some problems when data is high dimensional…lead[ing] to some unreasonable results" (p. 742) compared to the mixture L_2 discrepancy measure.



	$(CL_2)^2$	Mm	ρ_{map}	$(ML_2)^2$
Mean	0.120	0.828	0.178	0.624
Std Dev	0.004	0.043	0.028	0.027
Upper 95% Mean	0.121	0.837	0.183	0.630
Lower 95% Mean	0.120	0.820	0.172	0.619
N	100	100	100	100
Minimum	0.114	0.719	0.132	0.589
Maximum	0.134	0.919	0.296	0.704
Median	0.120	0.833	0.173	0.618

Figure 28. $(CL_2)^2$, Mm distance, ρ_{map}, and $(ML_2)^2$ measures for 100 33×11 uniform designs generated from the *UniDOE* package in R. Note that ρ_{map} ranges from 0.132 to 0.296.

The $(CL_2)^2$ measure distribution ranges from 0.114 to 0.134. The lowest mean $(ML_2)^2$ discrepancy (compared to MaxPro, MmLHD, and sphere packing) is achieved by the R package *UniDOE*, which generates designs that have an $(ML_2)^2$ value of 0.624, on average. The mean values of $(ML_2)^2$ for MaxPro, MmLHD, and sphere-packing designs are 1.648, 0.825, and 9.887, respectively. Interestingly, the MaxPro optimization results in poorer uniformity than does the Mm optimization of the MmLHD.

Figure 29 shows four extensions of shift-and-stack's ten configurations, incorporating the column-reordering heuristics, `permute_and_stack`, and new UD generations for the extended space using the R package *UniDOE*. Note that by the second stack `permute_and_stack`'s average ρ_{map} measure is less than 0.05. It is apparent from the figure that the *UniDOE* R software package reduces the average value of ρ_{map} and its

variability as n increases, even though the optimality criterion deals with measures of uniformity. In fact, after four extensions, a 165×11 *UniDOE* UD is very competitive with `permute_and_stack` with respect to ρ_{map}.

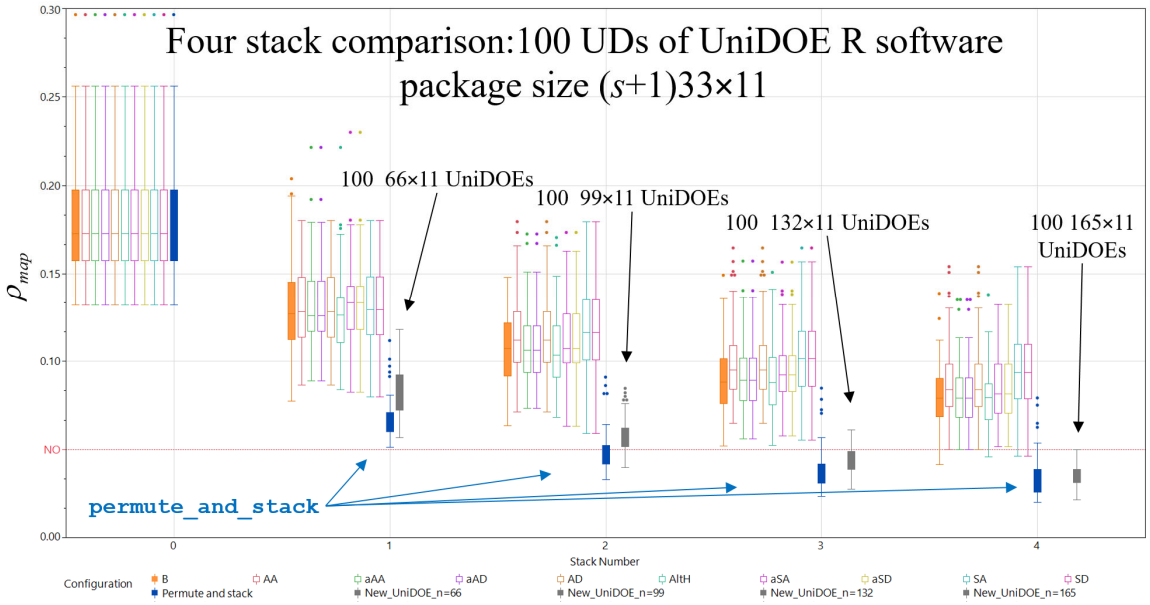


Figure 29. Comparative box plots of 100 design generations using the *UniDOE* R package for the default settings compared to shift-and-stack and `permute_and_stack`. Two extensions of the `permute_and_stack` algorithm result in an average $\rho_{map} < 0.05$.

A comparative plot of ρ_{map} and $(ML_2)^2$ for two extensions ($s = 2$) using `permute_and_stack` and shift-and-stack is given in Figure 30. The figure also includes 100 UDs constructed in the extended space (grey). The R package *UniDOE* generates the base 100 33×11 and 100 99×11 UDs for the extended space. Design extensions of the base 100 33×11 UDs are shown for `permute_and_stack` (blue) and shift-and-stack using the ten column-reordering heuristics (colors). The nine darkened observations in Figure 30 are the non-dominated set of observations. Note that all of the non-dominated designs are nearly orthogonal. From the non-dominated set, the best *UniDOE* 99×11UD has an $(ML_2)^2$ value 37 percent lower than the best obtained by `permute_and_stack`. Also, from the non-dominated set, the best *UniDOE* 99×11 UD

has a ρ_{map} value 21 percent higher than the best obtained by `permute_and_stack`. Across the 100 99×11 UDs, the average ρ_{map} is 0.085, while it is 0.047 after two extensions of the base 100 33×11 UDs using `permute_and_stack`.

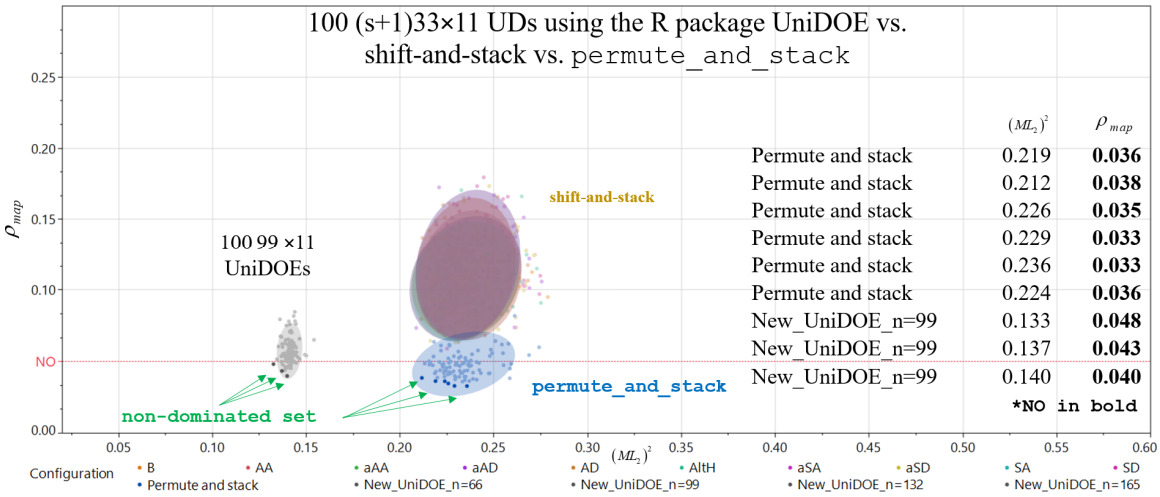


Figure 30. Multi-objective (ρ_{map} and $(ML_2)^2$) comparative plot for UDs generated using the R package *UniDOE*. Note the fraction of observations that are nearly orthogonal for the ellipses shown.

5. Extending JMP's Uniform Designs (UDs)

Santner et al. (2018) explain that JMP constructs uniform designs using "the $(CL_2)^2$ discrepancy measure of Hickernell (1998)" (p. 200). Creating 100 JMP uniform designs (UDs) is less straightforward and requires JMP's scripting language (JSL). Figure 31 summarizes four measures of design characteristics for 100 33×11 UDs generated by JMP. The optimality criterion minimized during construction is the $(CL_2)^2$. JMP provides this value in each design's diagnostic report.

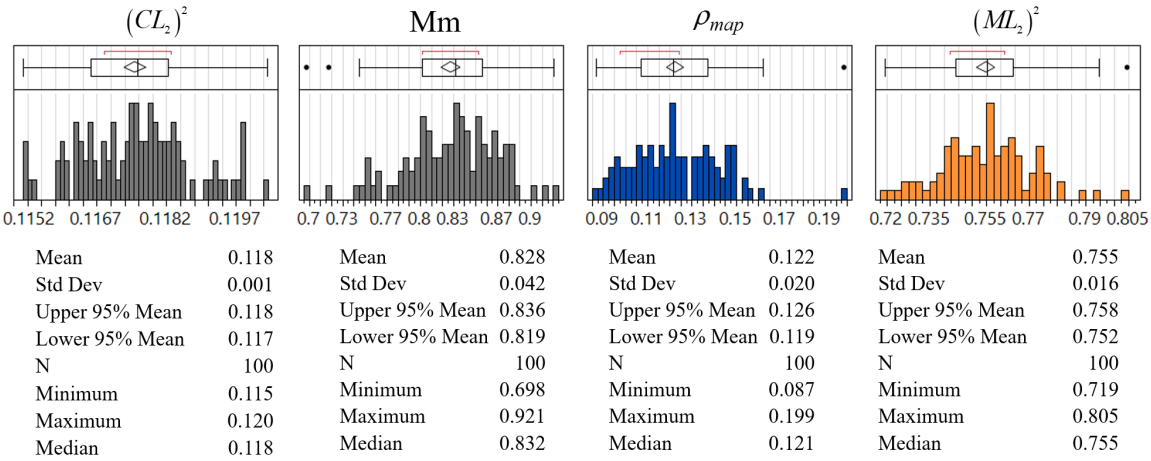


Figure 31. $(CL_2)^2$, Mm distance, ρ_{map}, and $(ML_2)^2$ measures for 100 33×11 JMP uniform design (UD) constructions. Note that the mean ρ_{map} for these designs ranges from 0.087 to 0.199.

Comparing the uniformity achieved between UDs constructed using the two different software packages—the R package *UniDOE* and JMP—we observe similar results for $(CL_2)^2$. Specifically, the mean value of $(CL_2)^2$ for UniDOE UDs is 0.120 and for JMP 0.118. However, when uniformity is measured using $(ML_2)^2$, these measures are less close. The mean $(ML_2)^2$ discrepancy achieved using the R package *UniDOE* is 0.624, whereas the average $(ML_2)^2$ for JMP's UDs is 0.755, which is noticeably worse.

Figure 32 presents a comparative plot for three iterations of shift-and-stack (using ten column-reordering heuristics) (colors), `permute_and_stack` (blue), and JMP UD generations in the extended space (grey) using default settings. While JMP generates UDs to optimize $(CL_2)^2$, it also improves the average ρ_{map} values for designs as n increases, similar to the R package *UniDOE*.

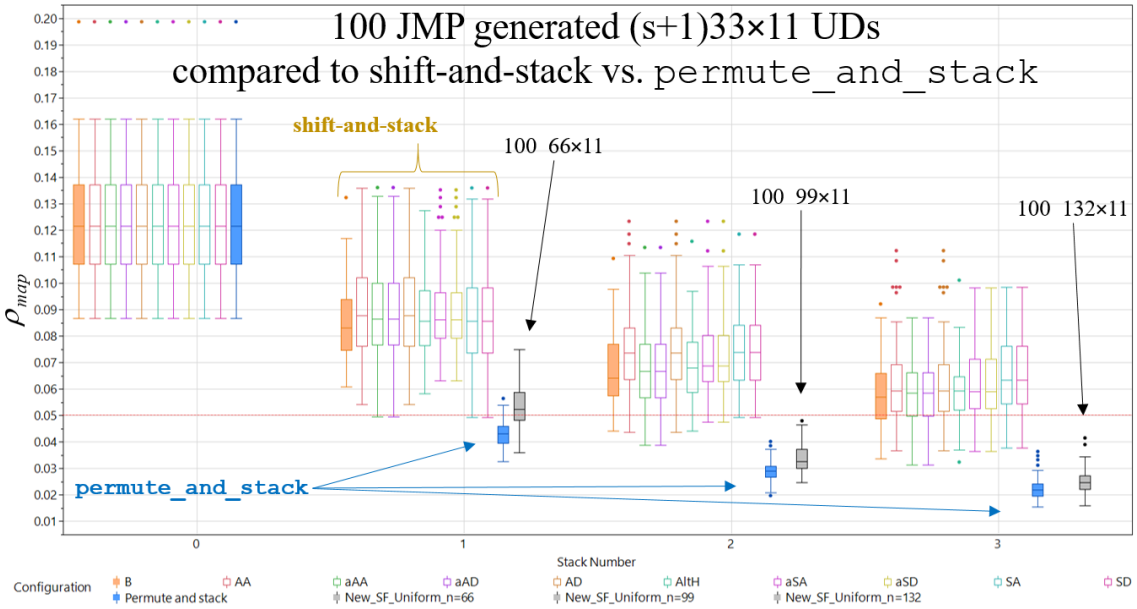


Figure 32. Comparative plot between `permute_and_stack` (blue), shift-and-stack (light colors), and 100 new (s+1)33×11 UDs of JMP (grey). The best designs (in terms of ρ_{map}) are the `permute_and_stack` designs, which are nearly orthogonal on average after s = 1.

Both software packages successfully achieve steady improvement in these measures; however, there is also a steady increase in the time required to construct these designs. Figure 33 is the time estimate (21 hours and 48 minutes) for JMP to complete the calculations necessary to generate one 200×20 UD, which can be terminated early to "deliver the best design found so far."

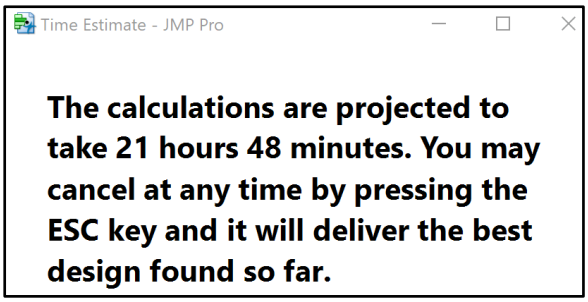
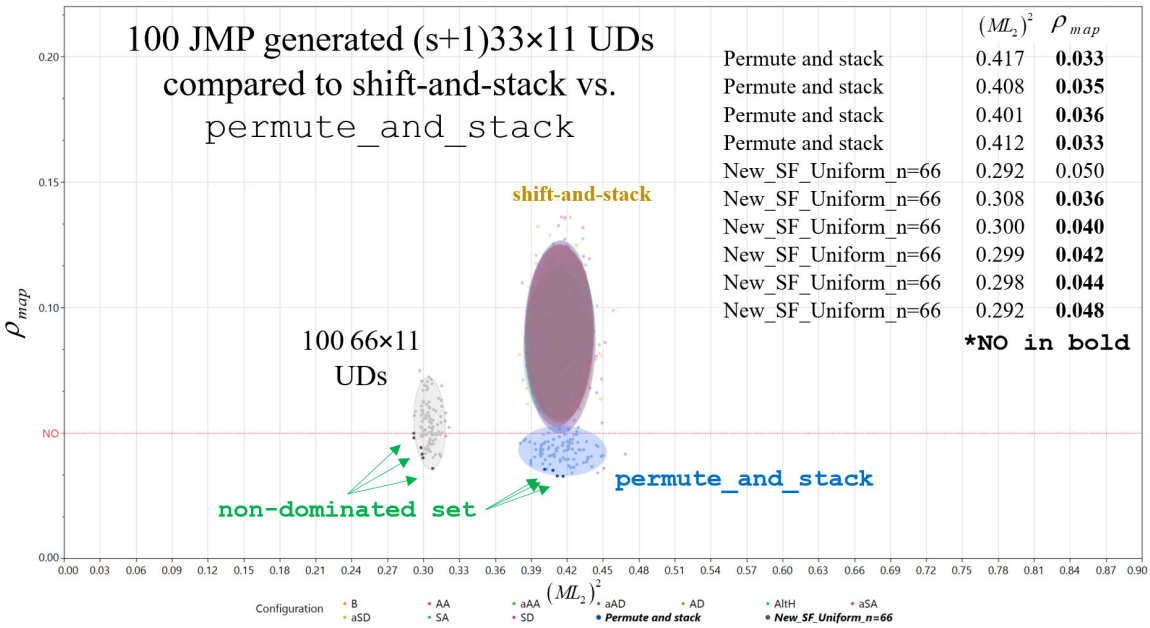


Figure 33. Time diagnostic report to generate one 200×20 UD in JMP.

A portfolio of batch-sequential designs can be generated with one execution of `permute_and_stack`, which takes less time than designs generated from scratch for the extended space or large n (e.g., n = 363). The time required to generate 100 200×20 large-scale UDs in JMP (see Figure 33) is 88.06 days.

Figure 34 shows a comparative plot of ρ_{map} and $(ML_2)^2$ for two extensions (s = 2). From the 1,200 design observations of Figure 34, only ten coordinates are non-dominated, including four from `permute_and_stack`. The mean ρ_{map} in the `permute_and_stack` designs is 0.048. The mean ρ_{map} in the 66×11 UDs is 0.057.



	$(ML_2)^2$	ρ_{map}
Permute and stack	0.417	**0.033**
Permute and stack	0.408	**0.035**
Permute and stack	0.401	**0.036**
Permute and stack	0.412	**0.033**
New_SF_Uniform_n=66	0.292	0.050
New_SF_Uniform_n=66	0.308	**0.036**
New_SF_Uniform_n=66	0.300	**0.040**
New_SF_Uniform_n=66	0.299	**0.042**
New_SF_Uniform_n=66	0.298	**0.044**
New_SF_Uniform_n=66	0.292	**0.048**

***NO in bold**

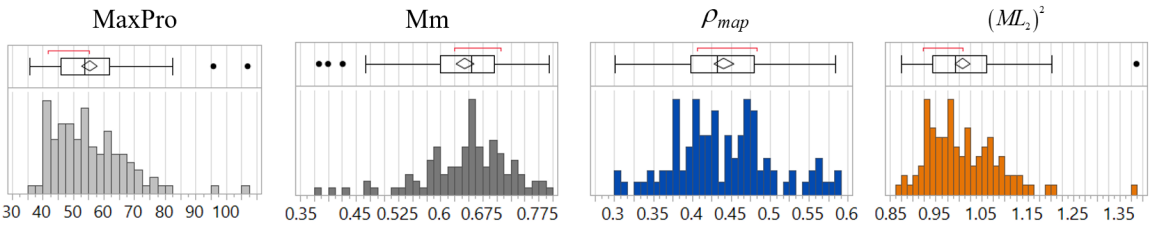
Figure 34. 100 33×11 UDs generated using JMP with one extension using shift-and-stack (colors) and `permute_and_stack` (blue) compared with 100 66×11 UDs in the extended space.

6. Extending Random Latin Hypercube Designs (LHDs)

Latin hypercube designs (LHDs) (McKay et al. 1979) are ubiquitous in computational experimentation as constructing an n×k LHD is straightforward. Moreover, "when a discrete uniform is used on the [columns], the one-dimensional projections are optimum space-filling designs" (Cioppa and Lucas 2007, p. 47). All 100 Latin hypercube

(LH) designs (LHDs) constructed in this section result from custom functions developed in R; however, online code exists for generating LHDs (e.g., the `rLHD()` function in the R software package *LHD*).

Figure 35 displays several measures of design quality for 100 33×11 random LHDs. The histograms show the empirical distributions of the MaxPro, Mm distance, ρ_{map}, and $(ML_2)^2$ measures of the designs. As expected, since there is a vast number of candidate designs ($33!^{11}$) and these designs are not optimizing any criterion, we observe significant variability across all measures. The MaxPro criterion values vary between 35.71 and 107.01, with a mean value of 55.34 and a 95 percent CI on the mean of [52.90, 57.78]. The mean MaxPro measure for LHDs is over three times that for MaxPro designs. The average Mm distance is 0.644, 60.6 percent of the mean achieved by the MmLHD. The ρ_{map} values range between 0.301 and 0.585, with a mean value of 0.441 and a 95 percent CI on the mean of [0.428, 0.453]. These are the highest values of all the designs we considered in this chapter. All 100 of these 33×11 LHDs have ρ_{map} values greater than ten times the equivalent-sized NOLH. The $(ML_2)^2$ discrepancy ranges from 0.875 to 1.387, with a mean value of 1.009 and a 95 percent CI on the mean of [0.993, 1.025]. This is 61.7 percent greater than the average $(ML_2)^2$ value for *UniDOE's* 33×11 UDs.



Mean	55.34	Mean	0.644	Mean	0.441	Mean	1.009
Std Dev	12.28	Std Dev	0.080	Std Dev	0.064	Std Dev	0.081
Upper 95% Men	57.78	Upper 95% Mean	0.660	Upper 95% Mean	0.453	Upper 95% Mean	1.025
Lower 95% Mean	52.90	Lower 95% Mean	0.628	Lower 95% Mean	0.428	Lower 95% Mean	0.993
N	100	N	100	N	100	N	100
Minimum	35.71	Minimum	0.384	Minimum	0.301	Minimum	0.875
Maximum	107.01	Maximum	0.792	Maximum	0.585	Maximum	1.387
Median	53.75	Median	0.655	Median	0.432	Median	0.991

Figure 35. MaxPro, Mm, ρ_{map}, and $(ML_2)^2$ measures for 100 33×11 LHDs. The mean ρ_{map} of these 100 LHDs is 0.441, ranging from 0.301 to 0.585.

Figure 36 shows how ρ_{map} tends to decrease when an original LHD is extended by `permute_and_stack` and shift-and-stack (with the ten column-reordering heuristics) for s = 1,2,…, 10. We see that ρ_{map} generally declines with the number of DPs for LHDs generated in the extended space (i.e., 66×11, 99×11,…, 363×11). Interestingly, for our largest designs, 363×11, all stacking methods outperform the one big design. The quickest and most significant reductions in ρ_{map} occur when `permute_and_stack` is used. However, it takes seven extensions for the average ρ_{map} in the 264×11 LHDs extended to meet the nearly orthogonal criterion. We also observe that `permute_and_stack` also decreases the variability in ρ_{map} far more than all other approaches.

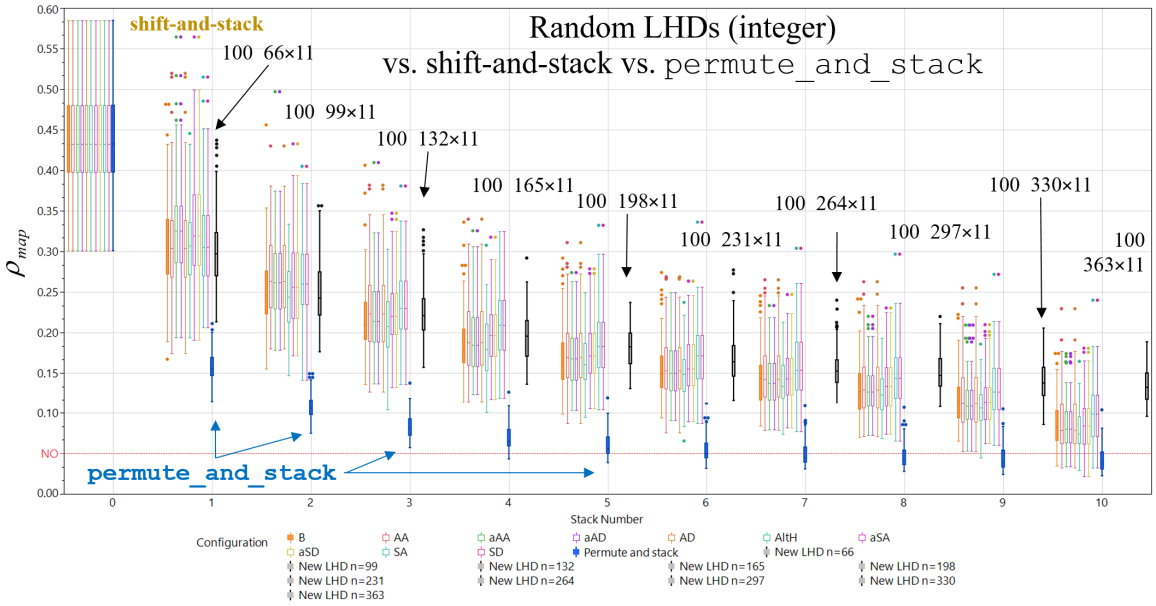


Figure 36. Comparative plot between `permute_and_stack` (blue), shift-and-stack (light colors), and 100 (s+1)33×11 LHDs (grey). The best designs according to ρ_{map} are the `permute_and_stack` designs.

Figure 37 displays a multi-objective comparative plot of ρ_{map} and $(ML_2)^2$ after one extension of 100 33×11 LHDs. Observation data for 100 66×11 LHDs constructed in the extended design space also appear comparable (90 percent ellipses mostly overlapped) to shift-and-stack (with ten column-reordering heuristics) for these measures. None of the designs in the figure are nearly orthogonal.

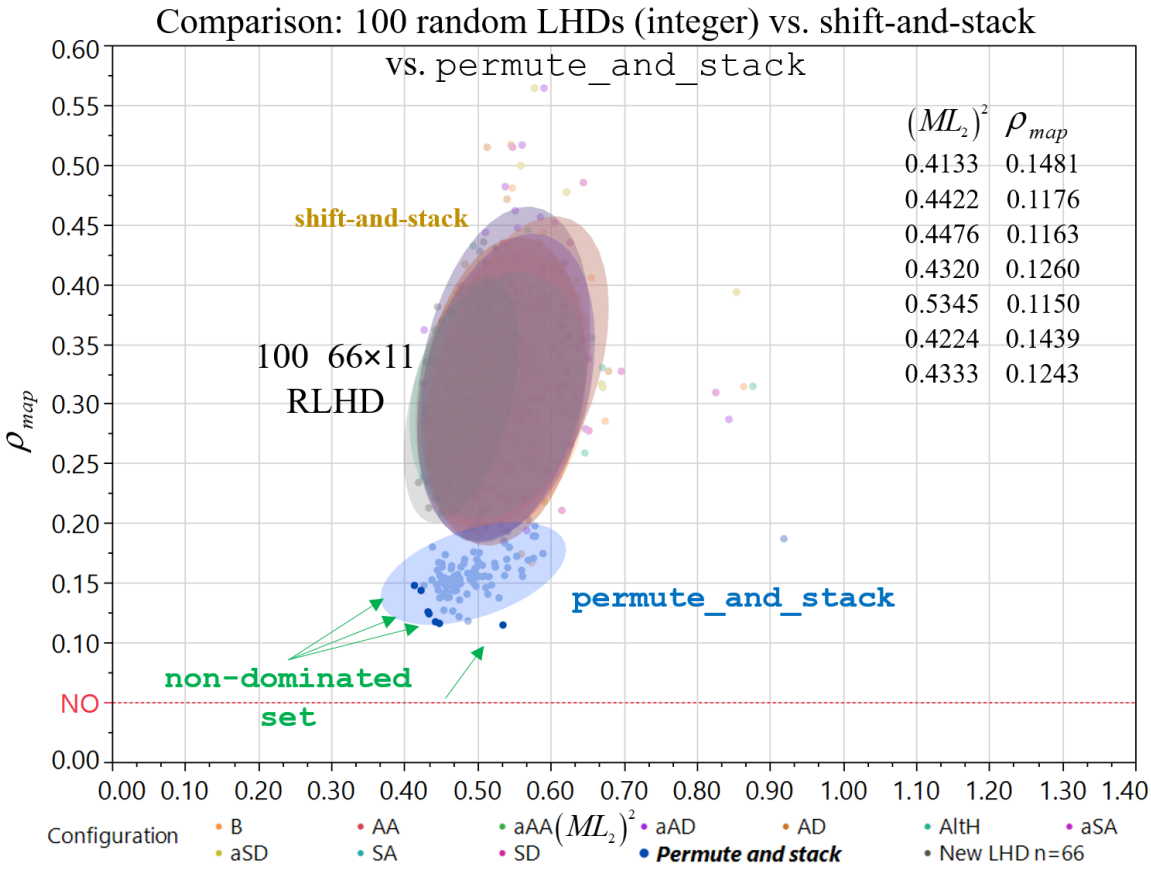


Figure 37. 100 33×11 random LHDs after one extension using shift-and-stack (light colors) and `permute_and_stack` (blue) compared with 100 66×11 LHDs. Shaded ellipses containing 90 percent of the 66×11 LHDs are barely noticeable (seven o'clock position) behind the design measure coordinates (and ellipses) of shift-and-stack. From the 1,200 designs, seven are non-dominated, all of which are `permute_and_stack` designs.

THIS PAGE INTENTIONALLY LEFT BLANK

V. LESSONS FROM MASSIVE EXPERIMENTATION ON SPACE-FILLING DESIGNS OF DIFFERENT TYPES AND SIZES

> While there are many optimality criteria available in the literature, the comparison between different design criteria is certainly one of the most important problems in the field of design of computer experiments and deserves a thorough future investigation.
>
> —Jin et al. (2003), p. 554

Despite the call from Jin et al. (2003) nearly two decades ago, there remains a dearth of knowledge on the relationships among design criteria for computer experiments. This chapter highlights lessons from massive experimentation on correlation and space-filling measures (SFMs) for many classes and sizes of designed computer experiments. The need for such experimentation became apparent to the author when accessing how well `permute_and_stack` does across many classes and sizes of SFDs. The plots in this chapter reveal higher than expected variances and significant outliers. This highlights the risks associated with generating only a single "optimal" design with stochastic software. Researchers need to generate many SFDs to ensure they obtain a design with good properties for their application.

A. INTRODUCTION

There are many classes of space-filling designs (SFDs) and measures to assess them. In these experiments, particular emphasis is given on the variability in space-filling and correlation measures in popular software-generated designs as well as relationships among them.

Finding optimal space-filling designs (SFDs) for large n and k is extremely challenging. Lin and Tang (2015) explain that due to the complexity of the optimization problem of creating large space-filling designs, most design-generation software contains stochastic aspects. Jin et al. (2003) write, "[s]earching the optimal design of experiments within a class of designs (e.g., LHD), even though more tractable than searching in the entire sample space without any restrictions, is still difficult to solve exactly" (p. 546).

Thus, stochastic search algorithms are almost always used in generating complex SFDs. Simulated annealing (SA) (Bohachevsky et al. 1986), threshold-accepting (TA) (Winker and Fang 1998), and genetic algorithms (GAs) (Goldberg and Holland 1988) all generate resultant designs that are "approximate" or "near-optimal" solutions. Moreover, the arrived-at designs' properties can vary significantly based on random-number-generator seeds, starting points, or search-algorithm parameter settings.

The research questions that drove this experimentation are:

(1) What is the variability in SFMs when using popular software packages to construct SFDs?

(2) What are the relationships among space-filling and correlation measures for various classes and sizes of SFDs?

(3) Which design software, construction methods, and parameter settings efficiently create quality SFDs?

To address these research questions, popular software packages are used to construct SFDs to inform on the variability and relationships among space-filling and correlation measures for various design classes and sizes. Many thousands of SFDs of different dimensions are created using JMP and R software packages. The design classes include random Latin hypercube designs (LHDs) (McKay et al. 1979); sphere-packing (Mm distance) designs in JMP (Johnson et al. 1990); MmLHDs in JMP (Morris and Mitchell 1995); MaxPro designs (Joseph et al. 2015) with the *MaxPro* R package (Ba and Joseph 2018), and uniform designs (UDs) using the *UniDOE* R package, see Fang et al. (2000a) and Zhang et al. (2018).

B. APPROACH METHODOLOGY: WHAT WE DID

1. Software-Generated Constructions

Extending the research of Lin and Tang (2015) and Wang et al. (2021), we present a comparative study to examine the stochastic aspects of SFD construction. For each design class, following Wang et al. (2020), we generate 100 designs using JMP or R (at their default settings) for $k = 5$, 10, and 20 and $n = k+1$, $3k+2$, and $10k$ to investigate our research questions. This allows us to explore the effects of the number of factors (k) as well as the

number of design points (DPs) per factor. When $n = k+1$, the designs are fully saturated and quite sparse; i.e., few DPs relative to k. Designs with $n = 3k+2$ are similar in DPs per factor as those evaluated in Chapter IV. In cases where $n = 10k$, there are more degrees of freedom for fitting metamodels. Having $n = 10k$ matches the rule-of-thumb guidance "to use a sample size of $10d$ when the input space is of dimension d." noted in Santner et al. (2018, p.149). With more DPs, the density or space-fillingness of points in the experimental region χ can be greater. For each design generated, measurements were recorded for ten design criteria.

The SFDs were constructed using JMP and several R packages. The generation was facilitated by writing scripts in R and JMP's scripting language (JSL). Using the "space-filling designer" in JMP (SAS, 2021), sphere-packing (Mm distance) designs and MmLHDs were constructed. In JMP, MmLHDs refers to a sphere-packing design constrained to an LHD (Latin hypercube design) with evenly spaced levels for each factor. In R, we generated MaxPro designs, uniform designs (UDs), and random LH sampling (LHS) designs using the software packages *MaxPro*, *UniDOE*, and *lhs*, respectively. Following the recommendation of Ba and Joseph (2018), the MaxPro designs are initialized with a MaxProLHD (i.e., first optimizing the MaxPro criterion with design matrix $\mathbf{X}$ constrained to an LHD). The R package *lhs* is used to construct designs using LHS as initially defined by McKay et al. (1979), which samples within strata as opposed to random LHDs in which the design points are constrained to a lattice. Custom R functions were used to generate LHDs.

Appendix D provides example scripts for generating 100 sphere-packing designs in JMP using JSL (JMP's Scripting Language) and constructing MaxPro designs using the `MaxProLHD()` function and R software package *MaxPro*. Since there is randomness in the construction algorithm, following Wang et al. (2020), 100 designs were created for each of $k = 5$, 10, and 20, and $n = k+1$, $3k+2$, and $10k$. Thus, a total of 900 sphere-packing designs were generated. The other five classes of designs studied also had 100 instances created at each of the same design dimensions.

Figure 38 shows the first ten entries of an output folder containing MaxPro designs of size 32×10. We label each construction with an ID from 1-100 (only 1-10 are shown) to facilitate analysis. For example, our first MaxPro design file is named `MaxPro_32_10_1.CSV`. We also store the random-number-generator seeds so that all designs can be reconstructed. This process is repeated for six types of SFDs, yielding 6×9×100 = 5,400 SFDs. In actuality, the number of designs created and evaluated is a bit larger, as some fractional and full factorial designs (which are typically not considered space-filling) were also constructed for comparison purposes. Figure 38 shows an example file that illustrates how these designs are stored for future research and the approach used to save stochastically generated designs with their random-number seed.

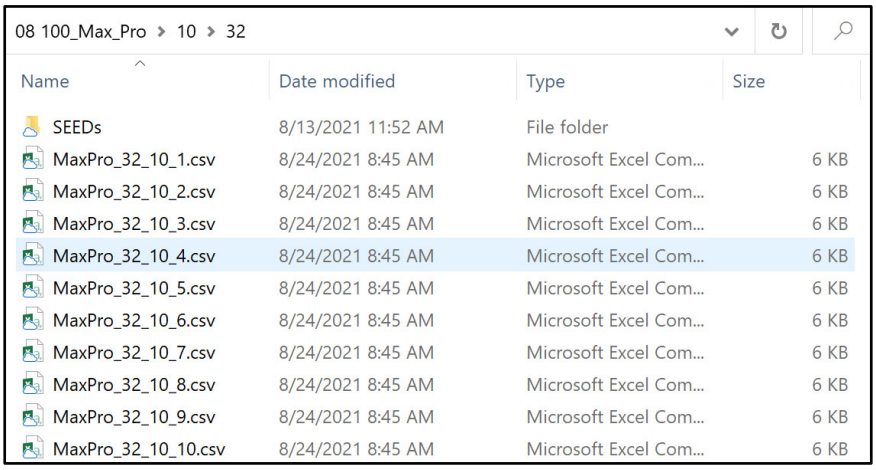


Figure 38. An example file folder containing the first ten of 100 32×10 MaxPro design generations.

2. Measurement Data

For each of the 5,400 SFDs created, we computed and collected ten measures of design characteristics. The results are stored in a single CSV (comma-separated values) file. Table 12 shows the first ten rows from the final 5,400×14 CSV data file. The first column specifies the design replication number (used in the design ID), which varies from one to 100. The second column contains the design class. Columns three and four specify the number of DPs (n) and factors (k), respectively. The next ten columns (columns E-N)

contain the design's characteristic measures. Columns E and F are measures of correlation. Columns G and H are measures of discrepancy. The design's coverage is given in column I. Columns J through N are distance measures. For the measures in columns E through L, lower values are preferred. For the measures in columns M and N, greater values are favored. That is, designs optimizing for ρ_{map}, $\overline{|\rho|}$, $(ML_2)^2$, $(CL_2)^2$, coverage, MaxPro, and ϕ_p select the **X** with the lowest value—while Mm- and $\bar{\gamma}$ - distance designs search for the **X** with the largest values.

Table 12. Ten rows from the final 5,400×14 CSV measurement data file.

ID	Class	n	k	rho_MAP	avg_absrho	ML_2	CL_2	Cov	MaxPro	Phip15	Phip50	Mm	avg_nn
1	MaxPro	100	10	0.0898	0.0407	0.3621	0.0708	0.1046	27.6067	1.7380	1.5609	0.6480	0.8418
2	MaxPro	100	10	0.1240	0.0460	0.4050	0.0734	0.0995	28.2875	1.7507	1.6094	0.6215	0.8313
3	MaxPro	100	10	0.1395	0.0505	0.4501	0.0764	0.0923	28.0889	1.7058	1.5270	0.6555	0.8437
4	MaxPro	100	10	0.1792	0.0450	0.4065	0.0739	0.0894	27.9895	1.7258	1.5833	0.6320	0.8391
5	MaxPro	100	10	0.1809	0.0383	0.4254	0.0719	0.0885	28.0248	1.6940	1.4739	0.6820	0.8405
6	MaxPro	100	10	0.1799	0.0430	0.4117	0.0777	0.0939	27.7300	1.7116	1.5314	0.6539	0.8370
7	MaxPro	100	10	0.1566	0.0491	0.4073	0.0730	0.0941	27.6819	1.7357	1.5507	0.6476	0.8184
8	MaxPro	100	10	0.1642	0.0425	0.4043	0.0751	0.0845	28.0364	1.6803	1.4482	0.6944	0.8447
9	MaxPro	100	10	0.1022	0.0362	0.4145	0.0785	0.0945	27.6872	1.7073	1.5544	0.6435	0.8414
10	MaxPro	100	10	0.1320	0.0417	0.3957	0.0732	0.0942	28.3035	1.7380	1.5606	0.6439	0.8226

Consider the first row of output for a MaxPro design (label ID is 1) of Table 12. This is the first construction out of 100 total. This MaxPro design is of size 100×10. The design's measurements are as follows: Correlation measurements (ρ_{map} = 0.090 and $\overline{|\rho|}$ = 0.041); discrepancy measurements (($ML_2)^2$ = 0.362 and $(CL_2)^2$ = 0.071); coverage (cov = 0.105); and distance measurements (MaxPro criterion = 27.61, $\phi_{p=15}$ = 1.738, $\phi_{p=50}$ = 1.561, Mm = 0.648, and $\bar{\gamma}$ = 0.842).

Measures are calculated directly using R functions, where possible. JMP provides diagnostic reports that contain measures such as the MaxPro criterion and Mm distance; however, while these values can be used to check custom R functions, they are not suited for subsequent analysis when measuring/comparing thousands of designs across ten measures because of the manual effort required to pull each value from JMP's diagnostic report. To overcome these challenges, the author automated a four step process: (1)

generate designs in JMP and store them using a label identifier (e.g., a CSV file) using JMP's scripting language (JSL), (2) leverage R (or other software package, e.g., MatLab) to import the designs, (3) compute measures of design characteristics, and (4) save measures to an output file for assessment. Interestingly, there are differences between the discrepancy function in R, `discrepancyCriteria()` (Dupuy et al. 2015), and JMP's discrepancy reported in the diagnostic. Thus, custom functions in R (to compute the $(ML_2)^2$ and $(CL_2)^2$ values) were created to facilitate the analysis. The custom functions match JMP's discrepancy measure; see the example script in Appendix D.

The author has not found any studies that compare more than a couple of measures, types, and dimensions. Here, we calculate ten of the most common metrics for some of the most popular designs and compare their properties. The author has found nothing nearly as comprehensive in the literature.

C. BUILDING INTUITION AND WINNOWING DOWN THE FIELD

Before displaying the results of our experiments, this section shows a few cases that help build intuition and focus the comparisons we look at. First, we look at coverage as a measure of space-fillingness and see that, by itself, it can be misleading. Then, we see the effect that optimizing using the MaxPro criterion has on the positioning of design points (DPs). Finally, we compare LH designs when the DPs are restricted to a lattice (LHDs) and when they are not (LHS designs).

1. Some SFMs Can Be Misleading

Coverage can be misleading as a measure of space-fillingness for some design types. For illustrative purposes, Figure 39 shows a scatter plot matrix of all the pairwise scatter plots of DPs of the *resolution IV* 8×4 fractional factorial 2^{4-1} design with $n = 8$ and $k = 4$.

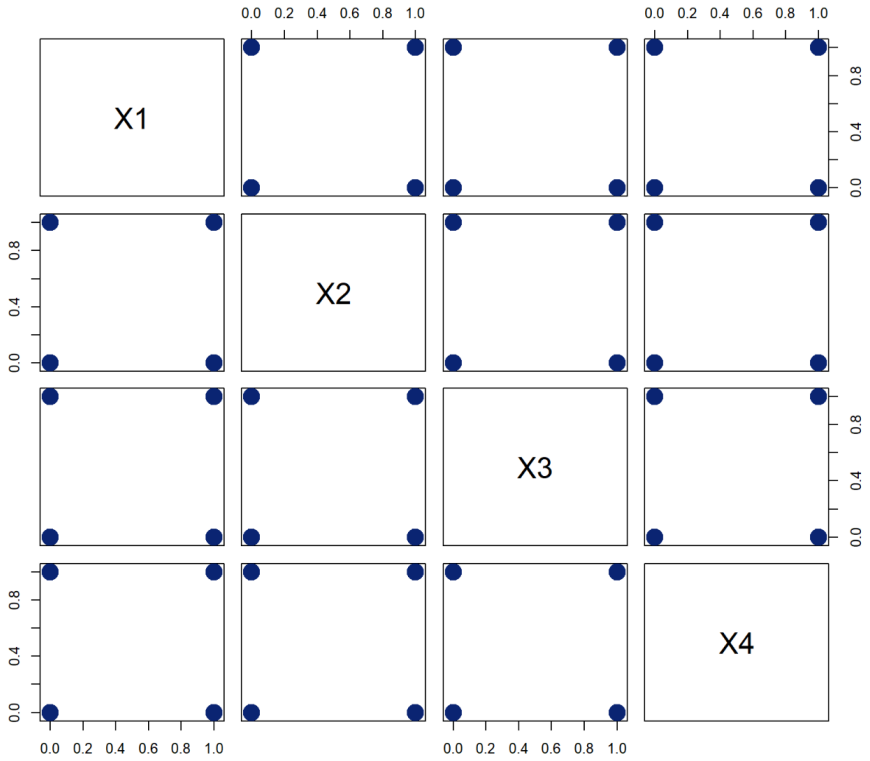


Figure 39. Scatter plot matrix for the *resolution IV* 8×4 Fractional factorial 2^{4-1} design. This design is column orthogonal and has optimal coverage; but, it is not generally considered space-filling.

The 2^{4-1} design has excellent coverage (i.e., *cov* = 0, which is the desired value), as defined by Dupuy et al. (2015). Nevertheless, the "white space" or gaps in the center of the pairwise scatter plots reveals that the design is not space-filling in terms of density or discrepancy in two-dimensional subspaces. The Mm distance criteria and average nearest-neighbor distance are also "optimal" for this design size (i.e., Mm = 1.414 and $\bar{\gamma}$ = 1.414), resulting from the DPs being pushed to the corners. We also observe that the design's correlation measures are also optimal, with ρ_{map} = 0.00 and $\overline{|\rho|}$ = 0.00. Finally, the MaxPro criterion = ∞, its worse possible value, due to replicated column entries in **X**. The MaxPro measure is designed to provide good space-fillingness in lower-dimensional spaces; therefore, it penalizes column entries that are close to each other. For some design classes, e.g., sphere-packing designs or stacked LHDs, the potential exists to encounter numeric instability with the MaxPro distance measure.

2. The MaxPro Criterion

An initial design is used as the starting point for the simulated-annealing search algorithm in the software generating MaxPro designs (Ba and Joseph 2018). Figure 40 shows a scatter plot for two factors from both an initial 16×4 MmLHD (blue circles) and the resultant MaxPro design (red triangles). Optimizing the MaxPro criterion appears to push DPs towards the boundaries. This behavior was also observed when using other designs as the starting point for the optimization using the MaxPro criterion. It is not clear how this affects other measures.

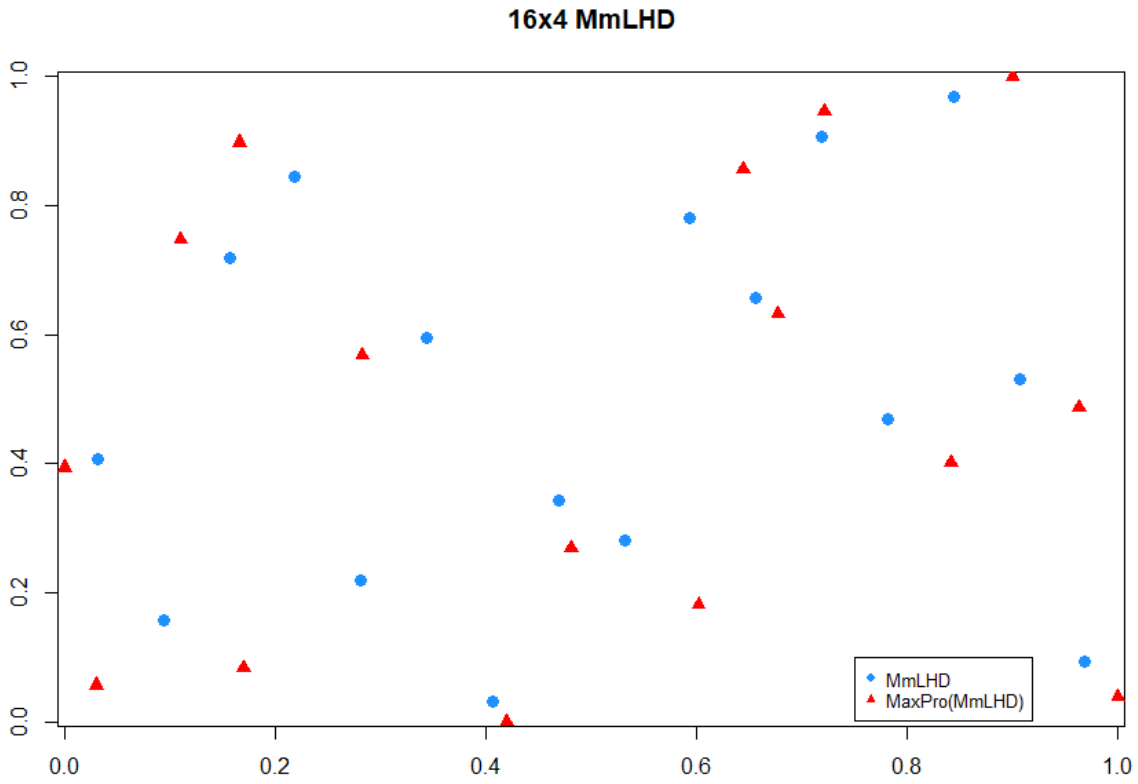


Figure 40. A two-dimensional projection of a 16×4 MmLHD (blue circles), which provides the starting point for the optimization and the resultant MaxPro design (red triangles).

Joseph et al. (2015) explain that MaxPro augmentations can arrive at designs with higher-than-desired discrepancy, which results from points' being pushed to the boundary space; however, they have improved projectivity for dimensions 2,…, k–1. The authors

recommend initializing design generation using a MaxProLHD (i.e., a design maximizing the MaxPro criterion constrained to be an LHD) when using the *MaxPro* R package. Joseph et al. (2015) explain, "the Latin hypercube restriction only makes the spacing of the levels equal, but somehow it improves the uniformity in all subspaces" (p. 376).

3. Measures of Discrepancy

In response to the challenges with using distance criteria, Fang et al. (2000a) recommend using discrepancy (i.e., $(ML_2)^2$ and $(CL_2)^2$), as initially defined by Hickernell (1998), for constructing uniform designs (UDs) and cataloged their best ones (Fang et al. 2000b). Similar to MaxProLHDs, Fang et al.'s (2000b) UDs leverage LHD structure. Table 13 presents sixteen cataloged UDs and their associated discrepancy measures for $(ML_2)^2$ and $(CL_2)^2$.

Table 13. Sixteen cataloged $n \times k$ UDs (Fang et al. 2000b) with their $(ML_2)^2$ and $(CL_2)^2$ values.

ID	n	k	$(ML_2)^2$	$(CL_2)^2$
1	8	7	5.14E-01	0.187
2	16	7	1.61E-01	0.052
3	24	7	8.23E-02	0.027
4	30	7	5.75E-02	0.019
5	12	11	2.67E+00	0.457
6	16	11	1.68E+00	0.283
7	21	11	1.14E+00	0.184
8	24	11	9.60E-01	0.153
9	30	11	6.79E-01	0.110
10	21	20	1.21E+02	3.039
11	24	20	8.81E+01	2.556
12	25	20	8.61E+01	2.395
13	30	20	6.53E+01	1.894
14	26	25	6.49E+02	8.432
15	30	25	6.45E+02	6.997
16	30	29	3.05E+03	18.682

Figure 41 plots the $(ML_2)^2$ values from Table 13 on the x-axis and the $(CL_2)^2$ on the y-axis. It shows a strong correlation across the design dimensions between $(ML_2)^2$ and $(CL_2)^2$ for the sixteen cataloged UDs (Fang et al. 2000b) of varying sizes. The author is unaware of any documentation on the degree of correlation between these measures used in arrived-at UDs using R software packages (e.g., *UniDOE*). Since both measures are used as surrogates for star discrepancy (see Equation [6]), it is of interest as to how well they match in quantifying the uniformity of a design.

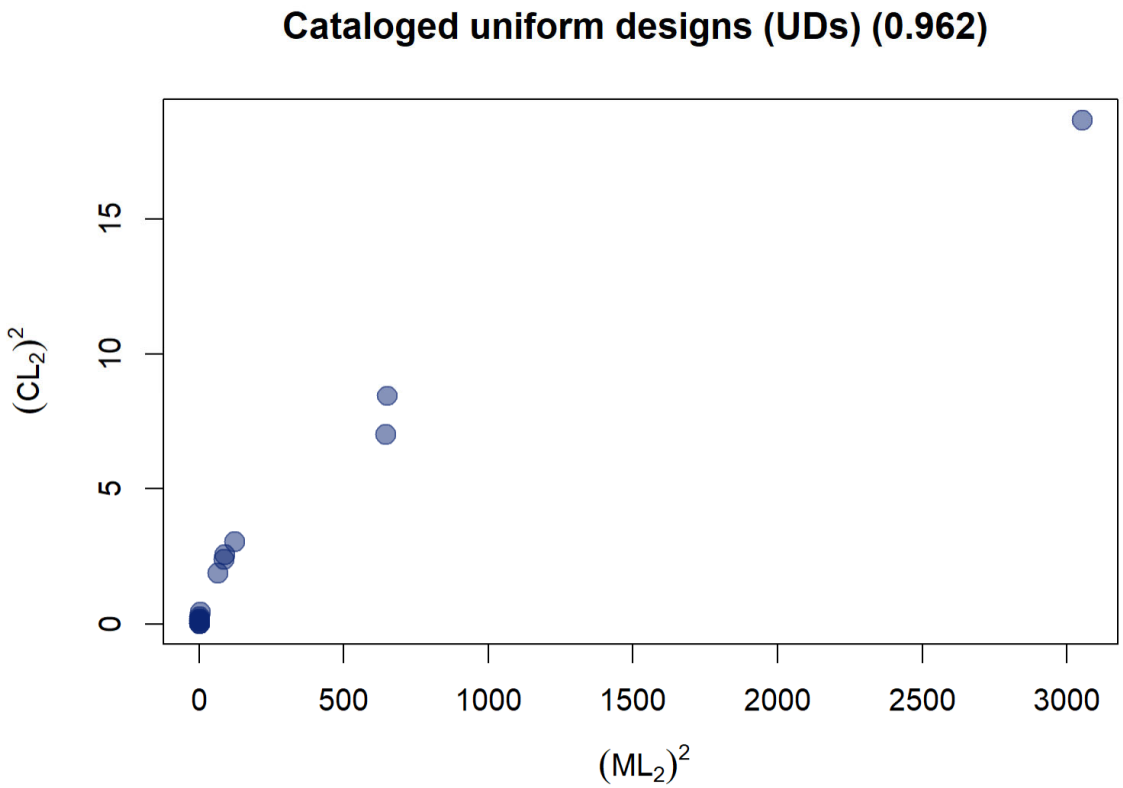


Figure 41. A scatter plot of $(ML_2)^2$ values and $(CL_2)^2$ values from Table 13. The correlation between these two discrepancy measures is 0.962, though driven largely by a few outlying points.

4. Results on LHD and LHS Designs

Wang et al. (2021) write, "[t]he most popular experimental designs for computer experiments are Latin hypercube designs (LHDs, McKay et al. (1979)" (p. 1). This

subsection investigates LHD and LHS design properties using correlation, discrepancy, and distance measures. The widespread popularity of Latin hypercube sampling (LHS) (McKay et al. 1979) and LHDs in computational experimentation result in numerous extensions and variations that utilize LH structure (Cioppa and Lucas 2007, Fang et al. 2000a, Helton and Davis 2003, Johnson et al. 1990, Joseph et al. 2015, MacCalman et al. 2017, McKay et al. 1979, Morris and Mitchell 1995, Shields and Zhang 2016, Sun and Tang 2017b, Wang et al. 2021, Ye 1998) and then take steps to improve space-fillingness. In their seminal paper on LHS, McKay et al. (1979) propose independently sampling within n strata for each input variable; that is, the points are not restricted to a lattice. We refer to this as Latin hypercube sampling. In our LHS designs, we sample from a uniform distribution in each stratum. In practice, a fixed value is often chosen in each stratum, such as the median. Doing so results in the DPs falling on a lattice, which we call Latin hypercube designs (LHDs) (Satterthwaite 1959). LHDs provide optimum space-fillingness (by many measures) in projections into each factor's subspace (Joseph 2016, p. 31).

In the context of LH structure, little is documented about how these two sampling methods affect correlation or space-filling measures (SFMs). We can see that neither method guarantees good SFMs, e.g., uniformly distributed DPs over the reverse diagonal (Hernandez et al. 2012a, p. 4). To see if we can limit the design classes we investigate to just one of these, we compare LHSs and LHDs over our nine design sizes. Specifically, which implementation, LHSs or LHDs, leads to the best results by our ten measures, or does it matter?

Figure 42 compares side-by-side box plots of ρ_{map} values for 100 LHSs and LHDs of size 6×5. There is very little difference between the two box plots.

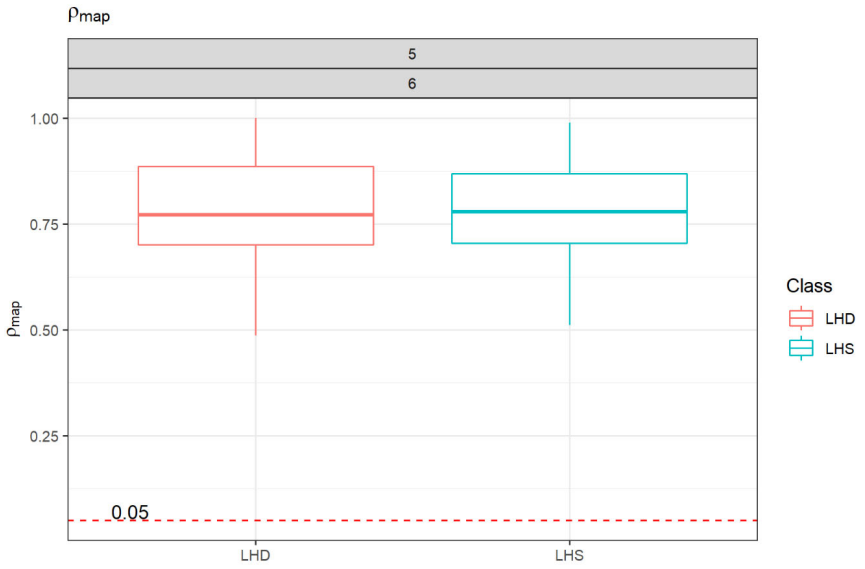


Figure 42. Side-by-side box plots of 100 LHSs and 100 LHDs for $k = 5$ and $n = 6$ (or k+1). These box plots display the results from 100 LHS and LHD designs.

Figure 43 (read columnwise) compares side-by-side box plots of 100 ρ_{map} values for LHSs and LHDs of size 6×5, 17×5, and 50×5. The ρ_{map} values decrease as n increases. The box plots in each panel are similar.

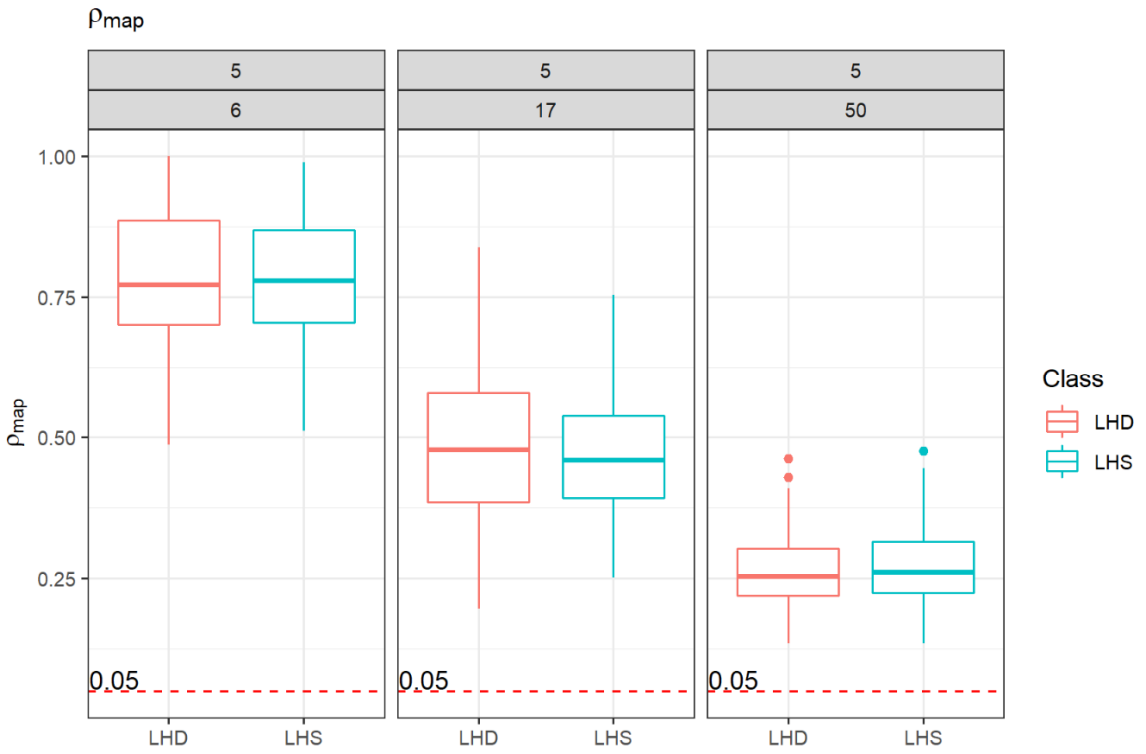


Figure 43. Side-by-side box plots of 100 LHSs and 100 LHDs for $k = 5$ and $n = 6$, 17, and 50 (columnwise).

The first row (top three panels) of Figure 44—read columnwise—compares side-by-side box plots of ρ_{map} values for 100 LHSs and LHDs of size 6×5, 17×5, and 50×5; i.e., $n = k+1$, $3k+2$, and $10k$ for $k = 5$. Subsequent three-panel-rows plot $k = 10$ and $k = 20$. Figure 44 shows that when n and k are small, the arrived-at LHSs and LHDs can have very poor ρ_{map} values. Roughly speaking, both LHSs and LHDs result in the same ρ_{map} distributions, which improves as n increases; however, interesting numerical behaviors do exist in other measures that ultimately caused us to limit our explorations to LHDs.

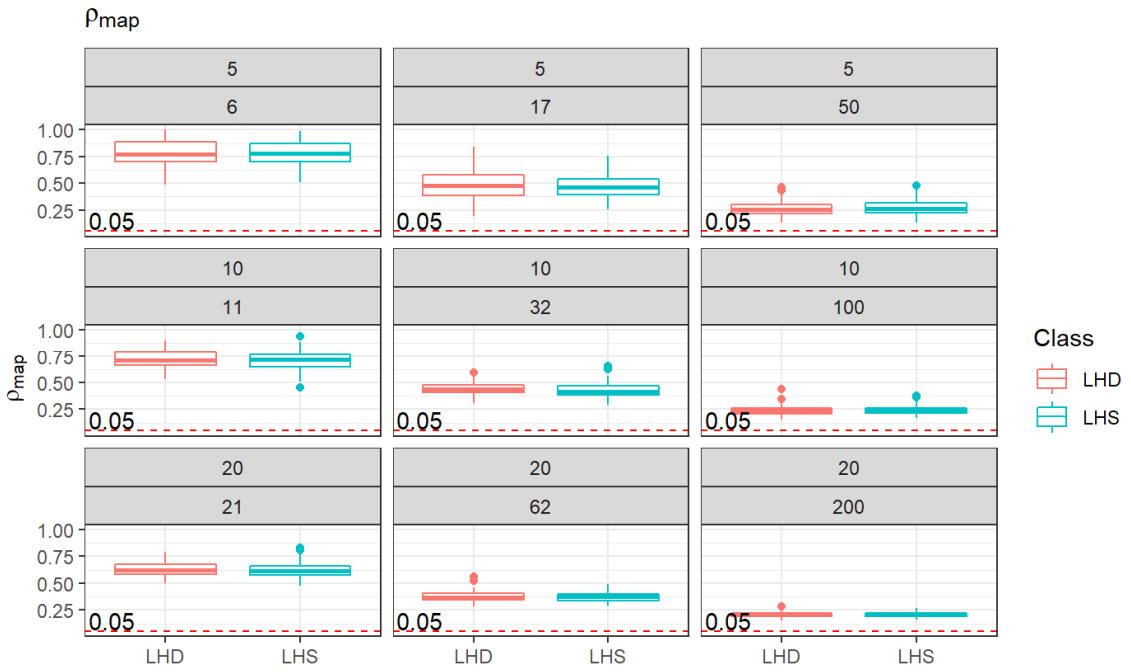


Figure 44. Side-by-side box plots of 100 LHSs and 100 LHDs for $k = 5$, 10, and 20 (rowwise) and $n = k+1$, $3k+2$, and $10k$ (columnwise). These box plots display the results from 900 LHS and 900 LHD designs.

Figures 45 and 46 illustrate two different numerical measures: Discrepancy $(ML_2)^2$ (Figure 45), and the MaxPro distance criterion (Figure 46), each containing side-by-side box plots. Smaller values are preferred for each. These figures reveal more outliers and larger values (i.e., underperformers) for LHS designs than LHDs. Note the outliers in the lower-left, 21×20, panel of Figure 45 and the top-right, 50×5, panel of Figure 46. We remark that each of the nine panels in the plots have the same y-axes. In general, similar performance is seen visually for the two data series (LHD and LHS) for these measures.

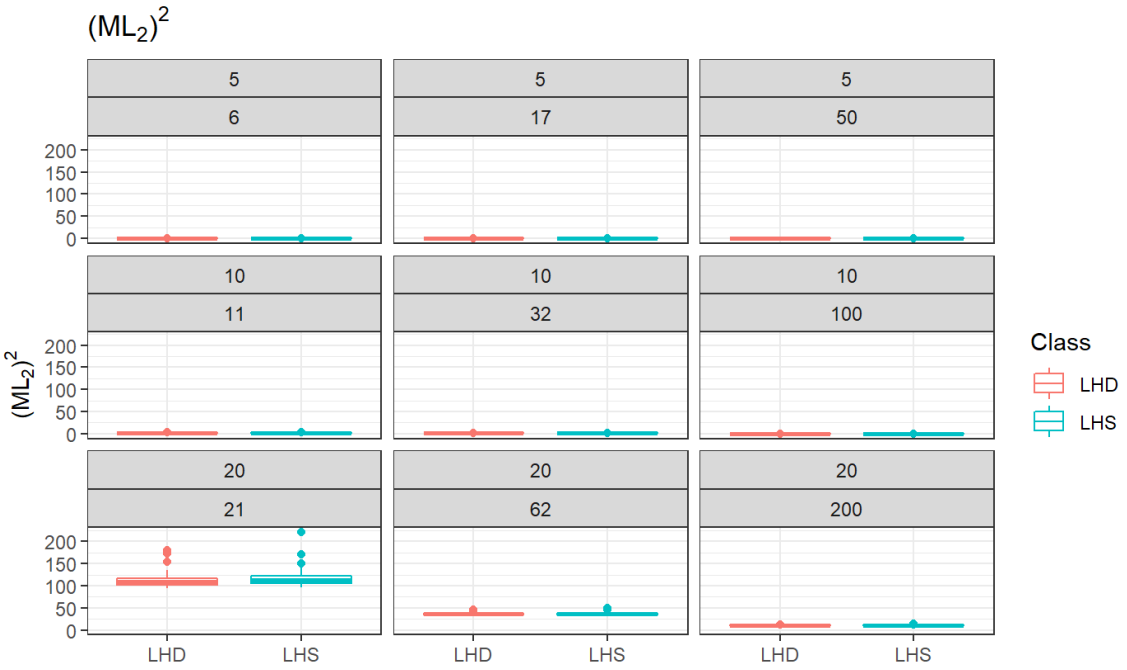


Figure 45. A comparison of discrepancy. Side-by-side box plots of the $(ML_2)^2$ discrepancy.

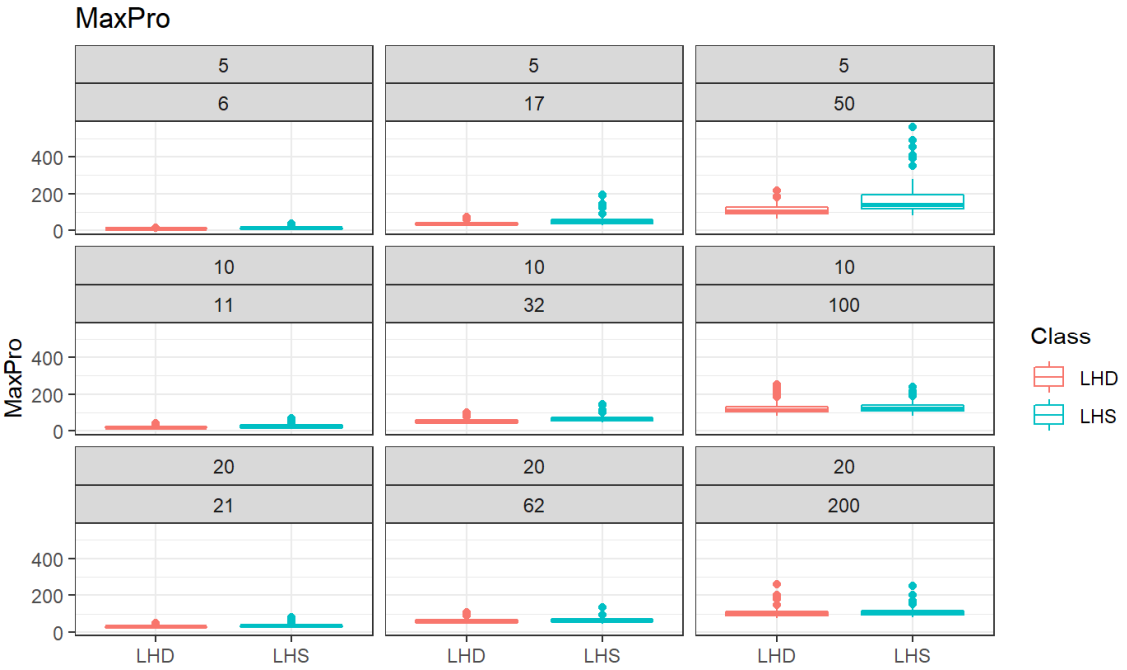


Figure 46. A comparison of distance. Side-by-side box plots of the MaxPro criterion.

A closer examination of the 6×5 panel in Figures 45 and 46 (i.e., $(ML_2)^2$ and MaxPro) is presented in Figure 47. For these two measures, at these two design sizes, LHDs are generally preferred over LHS designs. Empirically, the LHS box plots in Figure 47 show greater spread, contain a higher sample mean, and generally lead to designs with higher (poorer) $(ML_2)^2$ discrepancy and MaxPro distance.

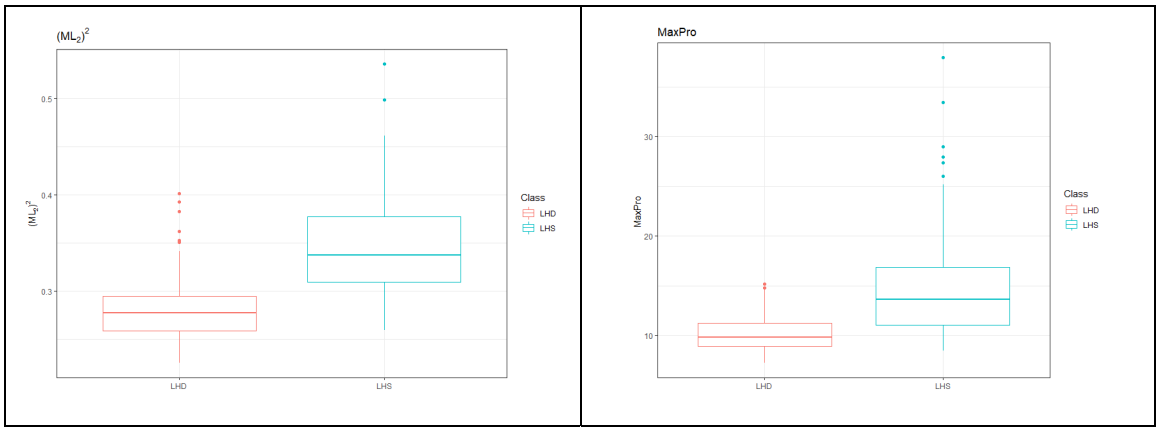


Figure 47. Side-by-side box plots of $(ML_2)^2$ discrepancy (left) and MaxPro criterion (right).

Experimentalists should know that, although both LHSs and LHDs result in sets of experimental points with good one-dimensional projections, LHSs can result in designs with DPs close to one another, which can affect discrepancy and the space-fillingness of projections into subspaces. Fang et al. (2000a), write "these designs are similar in one dimension but can be very different in higher dimensions" (p. 238). In sum, for most measures and design sizes there is little discernable difference between LHDs and LHSs. Where there is, LHDs generally perform a bit better. Thus, in what follows, we use LHDs. Appendix E contains the remaining LHD and LHS comparison.

D. NUMERICAL RESULTS AND MEASURES

This section is organized into two parts: The first part compares box plots of the distributions of a few key measures of design characteristics for popular design classes of various sizes. The second part provides scatter plots and correlations for a partial list of pairs of the most used measures from some of the most popular designs. Given that we

have ten measures, there are $\binom{10}{2} = 45$ pairs of design measures. Thus, while data have been collected for all of them, we analyze only a few in this chapter. Appendix G contains correlation data for each pair of our ten measures for six design classes of nine different sizes.

Designs are typically constructed based on one measure and might not fare well for other measures. To limit the combinations, we collect properties across five design classes, including LHDs, MaxPro augmented MaxProLHDs (labeled MaxPro), MmLHDs, sphere-packing designs (labeled SphereP), and UDs (labeled UniDOE). To make it clear, UniDOE without italics refers to a uniform designed experiment resulting from the *UniDOE* R package. These design classes are among the most used by practitioners. The LHDs were built using custom R functions that the author built. The MmLHDs and sphere-packing designs were created using JMP. The MaxPro and UDs were made using the R packages *MaxPro* (Ba and Joseph 2018) and *UniDOE* (Zhang et al. 2018), respectively. All calculations were done with the software packages' default settings. Much more can be done in exploring the parameters of the various search algorithms. An interesting insight from this section is that design preference varies by SFM and can change with design size (i.e., for different n and k).

1. Distributions of ρ_{map} for Five Design Classes of Various Sizes

Figure 48 shows side-by-side box plots for 500 6×5 designs—100 of each experimental design type. By this measure, for this size of design, sphere-packing designs tend to provide the lowest ρ_{map} values and have the least variability. The intuition is that JMP's sphere-packing algorithm uses the Mm Euclidean distance criterion and can achieve low ρ_{map} values by placing points in the corners of χ, similar to a fractional factorial design. The highest ρ_{map} values occur in LHDs. The ranges in ρ_{map} values obtained within each class highlight the need to generate multiple designs for each of these design types when generating a design. It is worth noting that none of these design construction methods explicitly consider correlation properties. Despite this, some of the sphere-packing designs are nearly orthogonal.

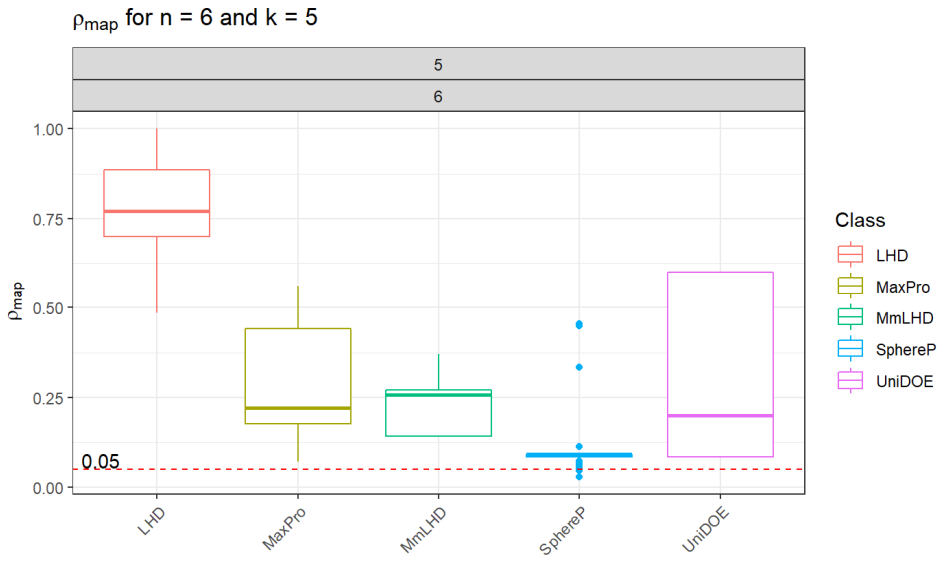


Figure 48. Side-by-side box plots of the empirical ρ_{map} distributions for 100 6×5 LHD, MaxPro, MmLHD, SphereP, and UniDOE designs. The ρ_{map} values across the designs range from near zero to almost one.

Figure 49 shows the empirical distributions of ρ_{map} measurements in our five design classes for $k = 5$ and $n = 6$, 17, and 50. Figure 48 is the leftmost plot in this graphic. As n increases, ρ_{map} values generally trend lower, as does their variance. There are exceptions though, as 17×5 sphere-packing designs have generally higher ρ_{map} values than 6×5 sphere-packing designs. We also see that the best design class by this measure switches as n increases. For $n = 50$, the R software package *UniDOE* designs perform best, with a majority of them being nearly orthogonal. When $n = 5$ or 17, the sphere-packing designs have the lowest ρ_{map} values. For all design sizes, LHDs fare worst. Interestingly, the third-best median ρ_{map} value switches from MaxPro to MmLHD between design sizes 17×5 and 50×5.

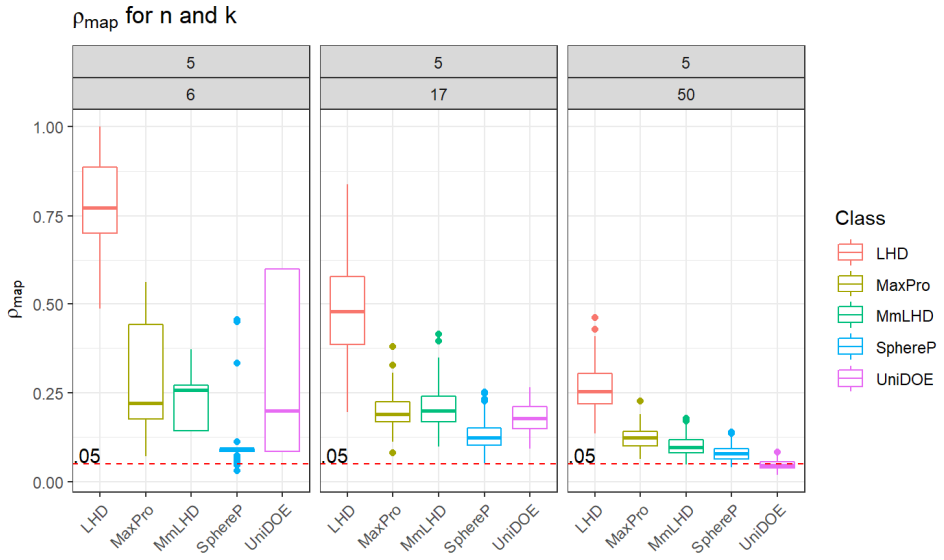


Figure 49. Box plots of ρ_{map} distributions for 100 6×5, 17×5, and 50×5 LHD, MaxPro, MmLHD, SphereP, and UniDOE designs. As n increases, ρ_{map} steadily improves.

Figure 50 appends two rows (for k = 10 and k = 20) to what is displayed in the previous figure, forming nine total panels of side-by-side box plots. These panels cover all the combinations of the three number of factors we explore and the three design points (DPs) density per factor. Specifically, shown are box plots of 100 observations for each design type for sizes 6×5, 17×5,…, 62×20, and 200×20. Rows of three panels (top, middle, and bottom rows) are for factors k =5, 10, and 20, which increase from top to bottom. Columns of three panels (left, middle, and right columns) are for DPs $n = k+1$, $3k+2$, and $10k$ per factor, which increases from saturated to higher-density designs from left to right. For large designs, those with higher n and k, we observe that the *UniDOE* R software package generates designs with the lowest ρ_{map} values, and are often nearly orthogonal. No other design type at any size produces more than an occasional nearly orthogonal design.

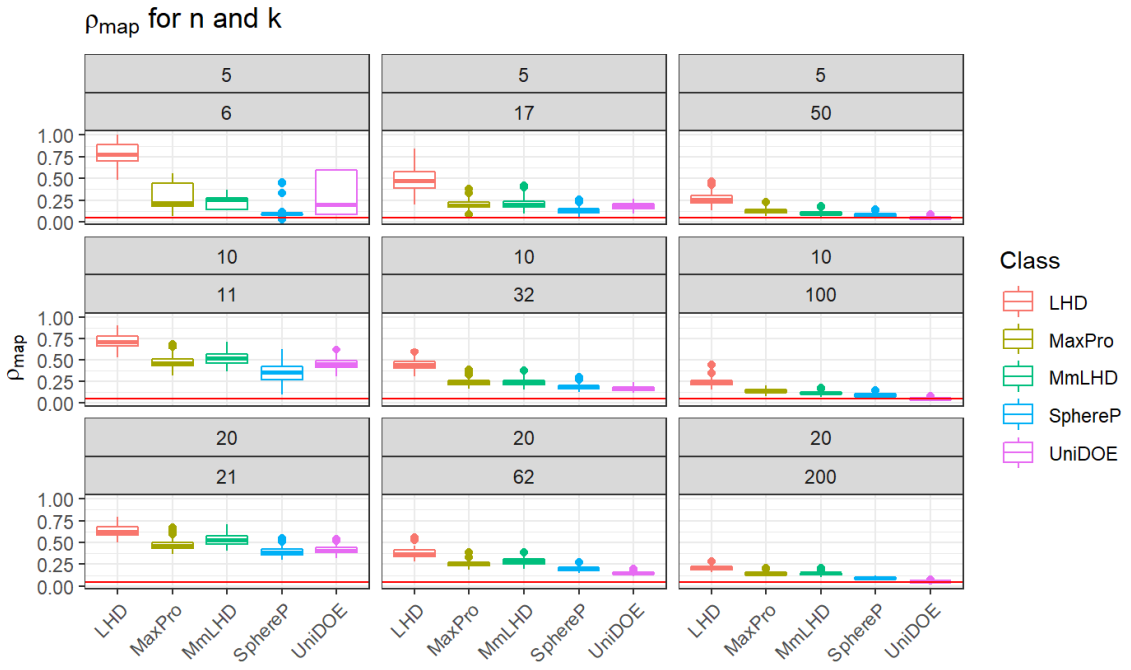


Figure 50. Box plots of ρ_{map} distributions for 100 6×5, 17×5,…, 200×20 LHD, MaxPro, MmLHD, SphereP, and UniDOE designs. For large n and k; *i.e.,* moving towards the lower right panels, we see that UniDOE designs do best with respect to ρ_{map}.

Looking across the nine panels, ρ_{map} ranges from near zero to almost one. Since we are most interested in designs with many factors, as many computer models have many input variables, we extract and display the bottom row (i.e., when $k = 20$) to identify better differences obscured by plot size in Figure 50. Figure 51 shows the box plots of ρ_{map} for each design type for designs of sizes 21×20, 62×20, and 200×20. We see that ρ_{map} values are greater than 0.30 for all these design-construction methods for saturated designs. Of note, nearly orthogonal LHDs of this dimension have been constructed by Hernandez et al. (2012a). Of course, their algorithm explicitly minimized ρ_{map}. As n increases, ρ_{map} values trend lower for all design classes and the variability decreases. We also see that UniDOE designs perform substantially better than the other methods for our largest designs. Having looked at a measure of correlation, we next look at a measure of distance.

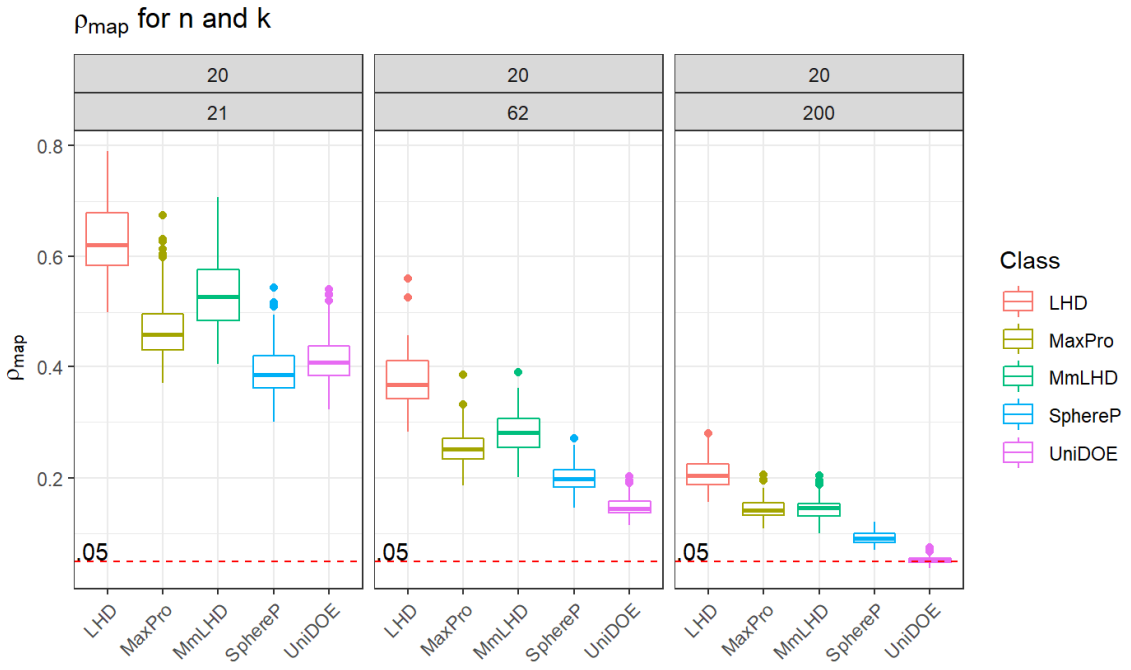


Figure 51. Box plots of ρ_{map} distributions for 100 LHD, MaxPro, MmLHD, SphereP, and UniDOE designs with $k = 20$. As n increases, ρ_{map} steadily improves, variability decreases, and UniDOE designs perform the best.

2. Distributions of Mm Distance for Five Design Classes of Various Sizes

Mm distance criteria (such as Euclidean, Manhattan, ϕ_p) are widely used in constructing and assessing SFDs. Figure 52 presents comparative box plots for the Euclidean Mm distance (Johnson et al. 1990) across nine design dimension panels. Panels in the same row show box plots for designs with the same number of columns. For example, the top row of three panels presents designs with $k = 5$ columns. Panels in the same column display box plots for designs with the same number of DPs per factor. Smaller designs are towards the top left and larger designs are towards the bottom right. Higher Mm distances are generally preferred for a given design size, as it means no two DPs are "close." As expected, the sphere-packing designs, which optimize over this measure without any constraints, other than that all DPs are in χ, perform best for all design sizes with respect to Mm Euclidean distance. The MmLHDs have the second-highest average Mm distance,

except for the saturated designs with $k = 10$ and 20, in which cases the MaxPro designs beat them. The LHDs generally have the lowest Mm distances.

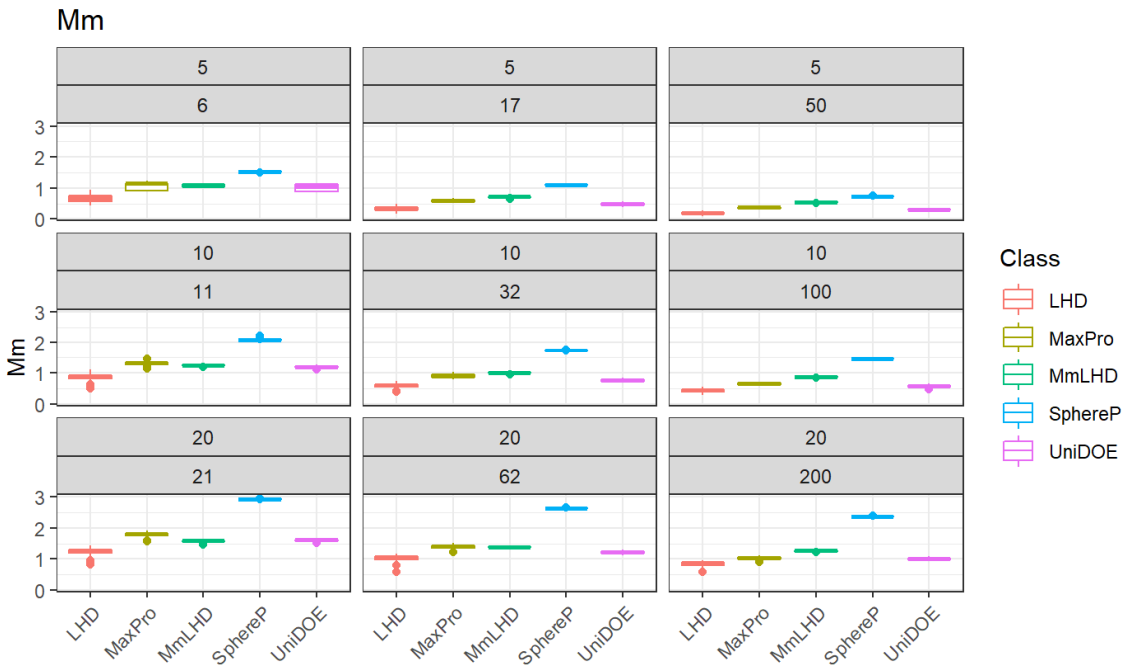


Figure 52. Mm distances for 100 LHD, MaxPro, MmLHD, SphereP, and UniDOE designs of various sizes. As n and k increase, SphereP designs achieve dominant Mm distance behavior, but less so in the few-factors ($k = 5$) case.

We have seen that sphere-packing designs perform best with respect to Mm distance for all design sizes explored, often by a considerable amount. Thus, to get more discrimination for the other design classes, Figure 53 shows the Mm distance distributions without the sphere-packing designs. When $n = 10k$, we see that MmLHDs perform best by this measure. A larger Mm distance is expected since MmLHDs are developed to optimize this measure (constrained to an LHD). However, somewhat surprising is that in fully saturated situations, the left column, the MaxPro designs have the largest Mm distances. In most panels, UniDOE designs are third and LHDs last.

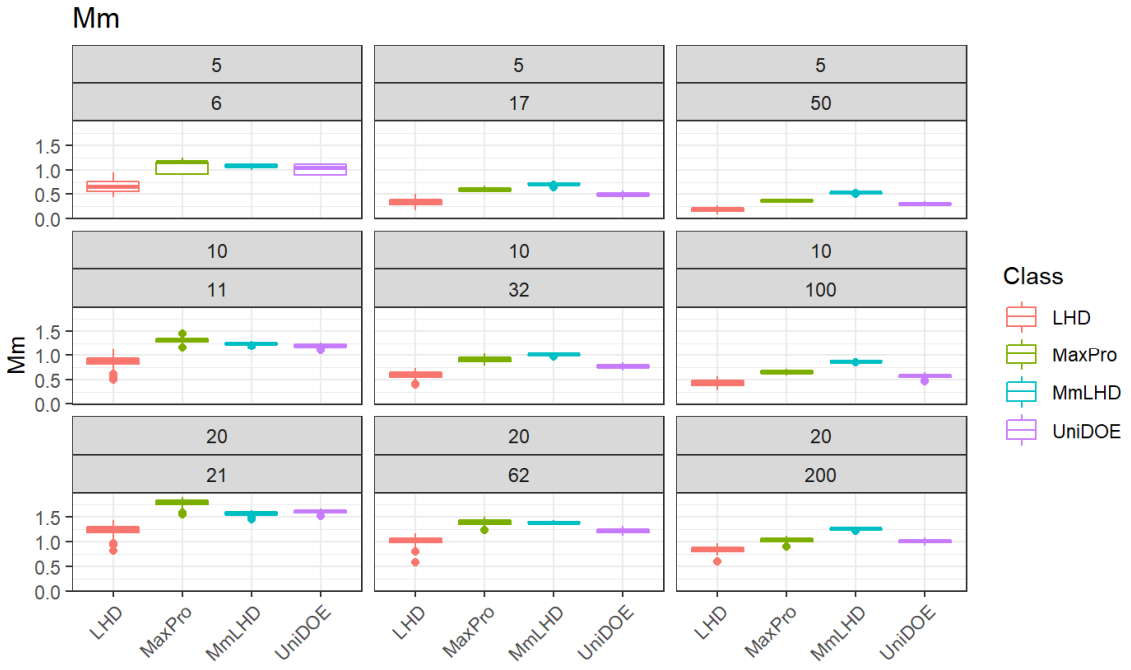


Figure 53. Mm distances for 100 LHD, MaxPro, MmLHD, and UniDOE designs of various sizes. For saturated n and k; i.e., the left column of panels, MaxPro designs achieve dominant Mm distance, which changes to MmLHDs for the $n = 10k$ case; i.e., the right column of panels.

In the 200×20 case, our largest design, with sphere-packing designs removed, the MmLHDs outperform MaxPro designs in this measure by a considerable amount, see Figure 54. We also see that there is much less variability in this measure for MmLHDs. The following section looks at the distributions of $(ML_2)^2$ discrepancy for five classes at nine design dimensions.

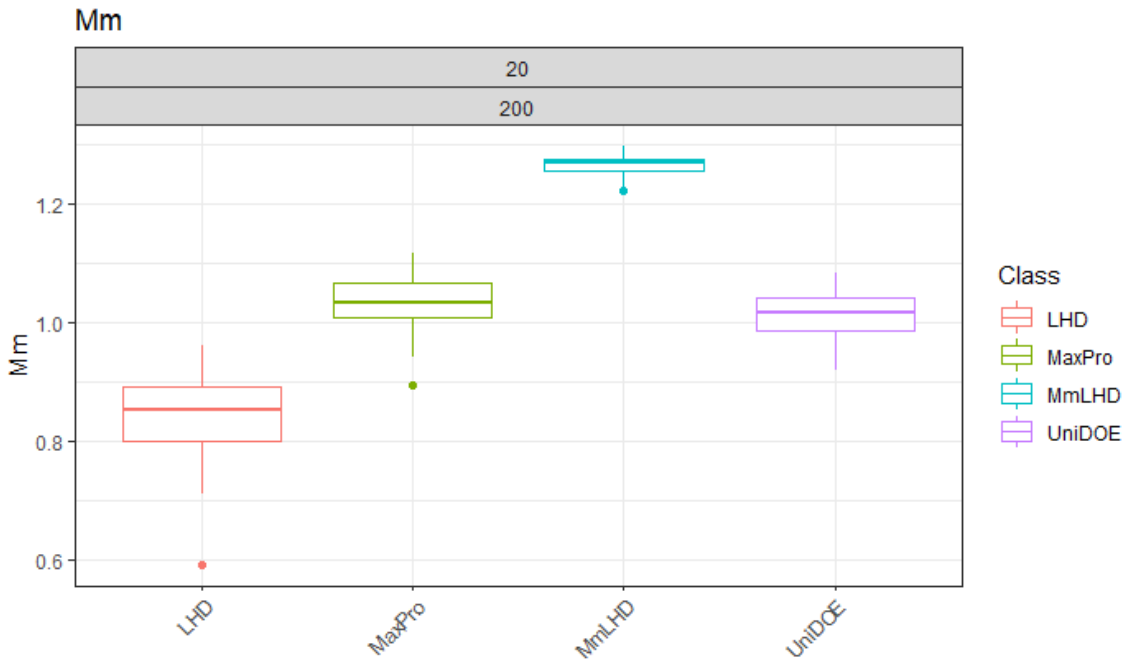


Figure 54. Mm distances for 100 200×20 LHD, MaxPro, MmLHD, and UniDOE designs. For large-scale computational experimentation, MmLHDs achieve dominant Mm distance among these.

3. Distributions of $(ML_2)^2$ Discrepancy for Five Design Classes of Various Sizes

$(ML_2)^2$ measures the disagreement (discrepancy) between the empirical distribution derived from the design points (DPs) compared to a theoretical uniform distribution, which is very different from a distance-based criterion. Figure 55 presents nine panels of side-by-side box plots for the $(ML_2)^2$ measure. Again, read rowwise for k = 5, 10, and 20 and columnwise for $n = k+1$, $3k+2$, and$10k$. From the bottom row, it is seen that optimizing for the Mm distance criterion (only) can result in a higher (poorer) discrepancy. The insight is that spreading-out DPs using the Mm distance criterion can increase the number of DPs near the boundary of the design space, resulting in an uneven empirical distribution of points, especially in the interior. This effect is most apparent with large designs (here for k = 20). Focusing on the bottom row, for the other design construction methods, we see that $(ML_2)^2$ values decrease rapidly as n increases. Finally, the difference in $(ML_2)^2$ values

between the sphere-packing designs and MmLHDs shows the impact of imposing an LH structure.

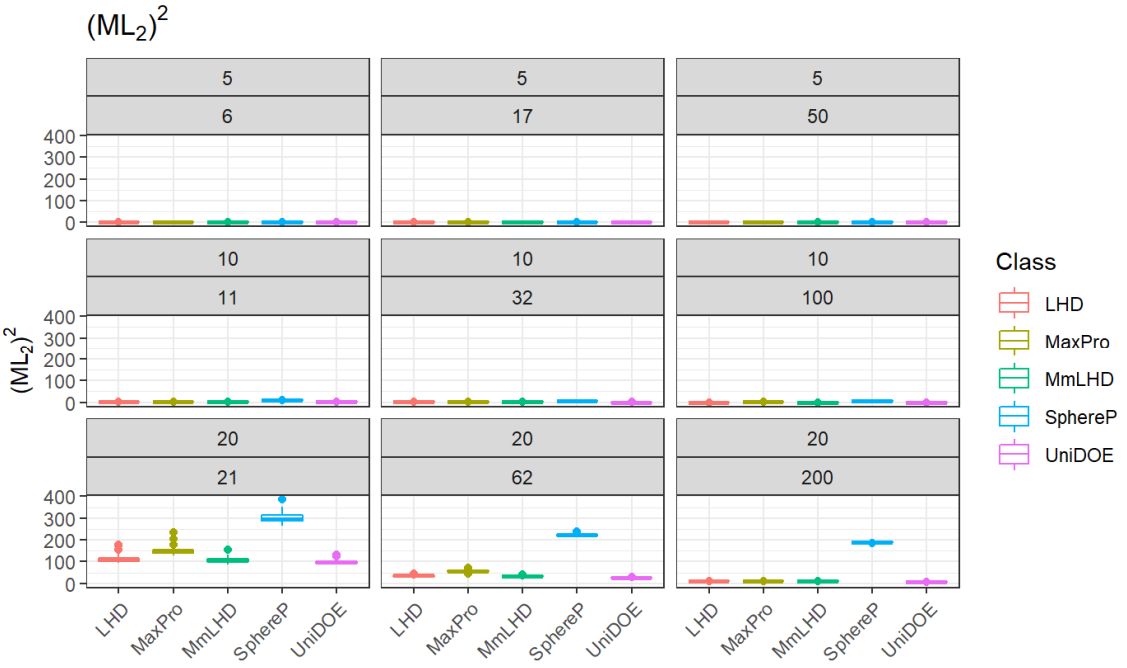


Figure 55. $(ML_2)^2$ values for 100 LHD, MaxPro, MmLHD, SphereP, and UniDOE designs of various sizes. For the 21×20 case (bottom left panel), SphereP designs achieve the highest $(ML_2)^2$ discrepancy.

To better discern differences among the other design classes, we remove sphere-packing Mm distance designs and consider rows one and three of Figure 55; i.e., $k = 5$ and $k = 20$, respectively. The results are displayed in Figures 56 and 57. Note the difference in the y-axes between Figure 56 and Figure 57. It is far more challenging to have DPs uniformly cover χ when n is small or k is large. UniDOE designs consistently achieve the lowest $(ML_2)^2$ value for our data, as they are constructed to minimize a discrepancy measure. We provide the exact values in Appendix F.

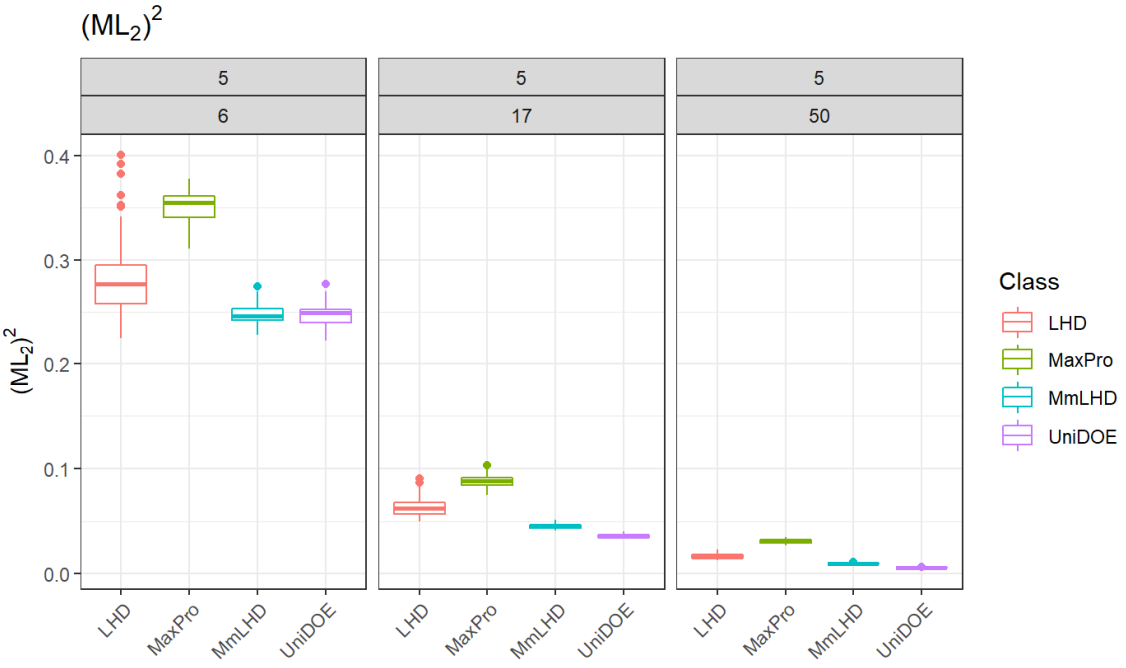


Figure 56. Side-by-side box plots of $(ML_2)^2$ discrepancy of LHD, MaxPro, MmLHD, and UniDOE designs of sizes 6×5, 17×5, and 50×5.

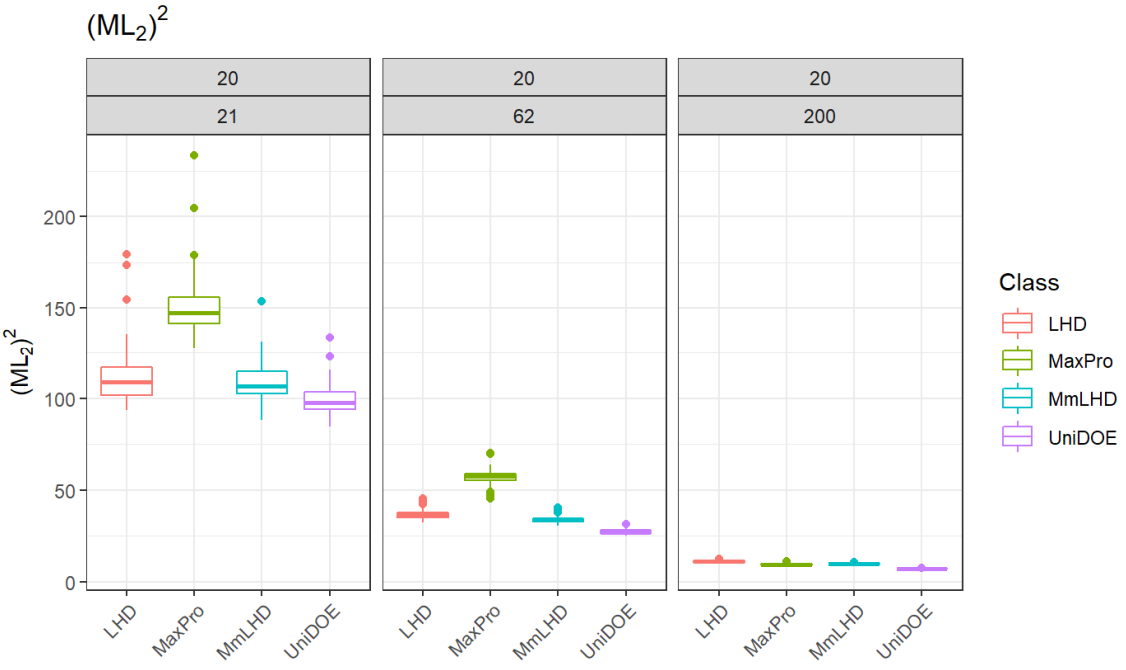


Figure 57. Side-by-side box plots of $(ML_2)^2$ discrepancy of LHD, MaxPro, MmLHD, and UniDOE designs of sizes 21×20, 62×20, and 200×20.

We also observe from these figures that MaxPro designs generally have poor $(ML_2)^2$ measures. The goal of the MaxPro criterion is design matrices with increased projectivity into factor subspaces; however, poorer $(ML_2)^2$ values can result, which can even be worse than the discrepancy reached randomly with LHDs. Joseph et al. (2015, p. 376) write:

> Uniformity is not as important as the maximin and minimax distance measures in a computer experiment, because the primary objective of a computer experiment is approximation and not integration. Thus, the poor performance of maximum projection designs under the uniformity measure is not of great concern.

4. Multiobjective Results on $(ML_2)^2$ and ρ_{map}

As we have seen above, design classes can perform well in some measures and poorly in others. Experimenters may desire designs that perform well in multiple measures, such as Joseph and Hung (2008). Here, we look at $(ML_2)^2$ and ρ_{map} for different design classes and sizes. The combination results in one correlation measure and one discrepancy measure. These two measures are the ones that Cioppa and Lucas (2007) sought to minimize in LHDs; i.e., space-filling NOLHDs. Before moving to the figures with nine sizes, we begin with one design size to highlight how poorly sphere-packing designs perform in terms of $(ML_2)^2$ to show why they are omitted from subsequent plots. Figure 58 presents multiobjective scatter plots for comparing $(ML_2)^2$ and ρ_{map} for 100 observations from each of five design classes when $n = 50$ and $k = 5$. The scatter plot on the left includes sphere-packing designs, which have by far the largest $(ML_2)^2$ values. The scatter plot on the right excludes them to obtain more resolution on the remaining four classes. Since we seek low $(ML_2)^2$ and ρ_{map} values, designs in the lower left of the plots are preferred.

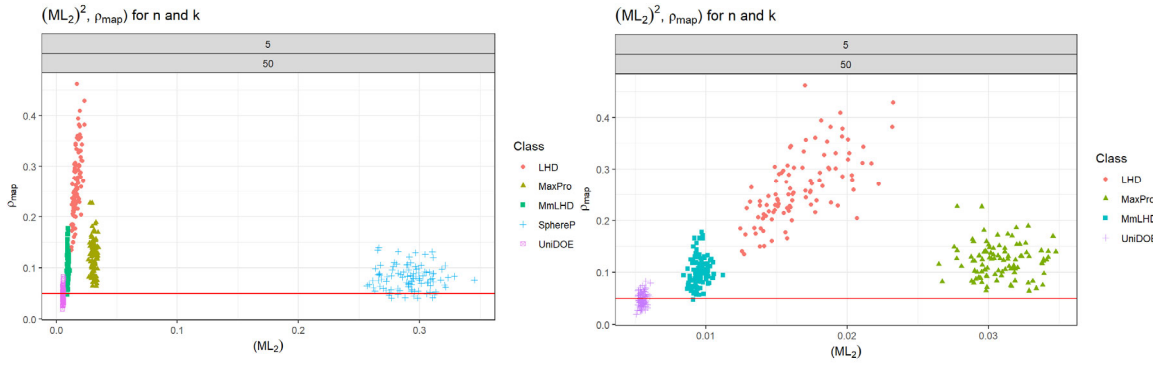


Figure 58. Multiobjective scatter plots comparing $(ML_2)^2$ and ρ_{map} for designs with dimensions $n = 50$ and $k = 5$. Five design types are shown in the scatter plot on the left and four on the right.

Figure 58 shows that 50×5 sphere-packing designs can have a low ρ_{map}, even sometimes meeting the nearly-orthogonal standard, but consistently have the largest $(ML_2)^2$. Once removed, we see clear patterns when comparing the other design classes. In terms of $(ML_2)^2$ values, UniDOE designs are the best, followed by MmLHDs, LHDs, and MaxPro designs, respectively. UniDOEs also have the best ρ_{map} values, followed by broadly comparable results for MmLHDs and MaxPro Designs. The highest ρ_{map} values occur in LHDs. All of the nondominated DPs at this design size are UniDOE designs. What about other design sizes?

Figure 59 presents scatter plots for a multiobjective comparison of ρ_{map} and $(ML_2)^2$ of five design types over nine design dimensions. Note that the x-axis is $(ML_2)^2$ and the y-axis is ρ_{map} in each panel. Design observations plotted in the lower-left corner of each chart contain the best-performing designs by these measures. As we have seen previously, $(ML_2)^2$ values are greatest in large, saturated designs. We see that sphere-packing designs have substantially higher $(ML_2)^2$ values than in comparably-sized other design classes, especially for large k—so much so that differences among the measures for other design classes are obscured. For $k = 5$ or 10, the highest ρ_{map} values occur in LHDs for all n.

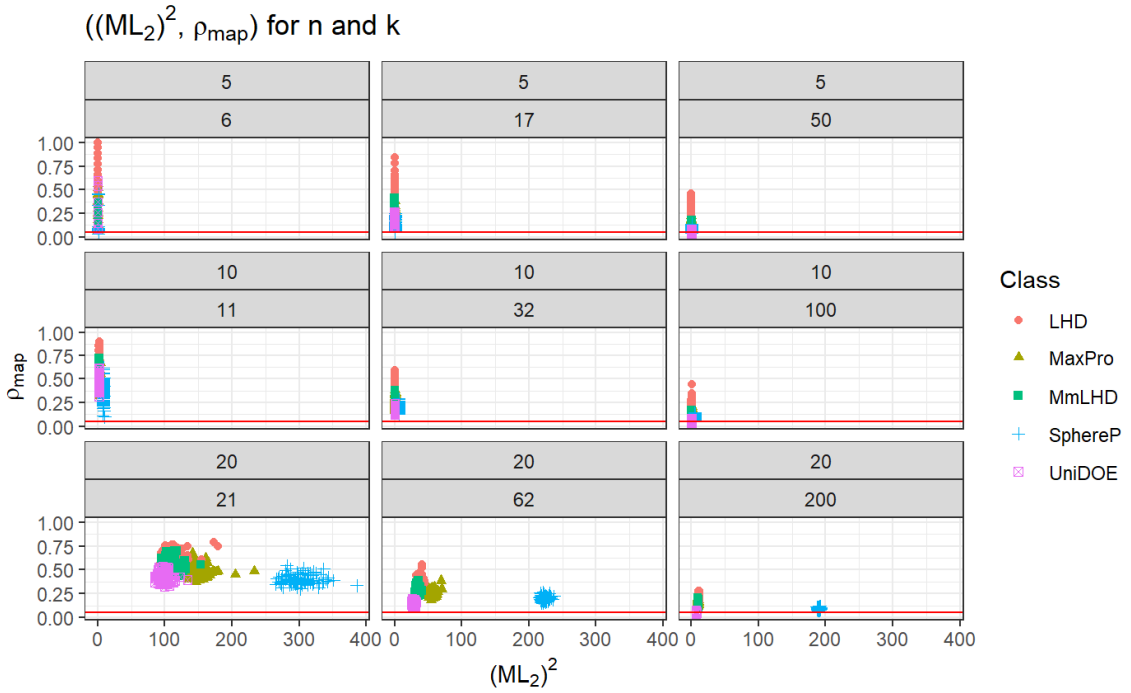


Figure 59. Multiobjective scatter plots of ρ_{map} and $(ML_2)^2$.

To obtain more resolution in our comparisons, Figures 60 and 61 highlight various portions of Figure 59. First, Figure 60 repeats Figure 59 with the sphere-packing designs removed, thereby zooming in on the x-axis. Here we see that the $(ML_2)^2$ values for MaxPro designs can also be large enough to hide differences among the other design classes or sizes. We also see UniDOE designs appearing more in the preferred lower left regions of the plots, especially for large k. This is seen more clearly in Figures 61 and 62, which respectively show the three panels when $k = 20$ and just the scatter plot for the largest design size (i.e., 200×20). For the latter figure, we see that all UniDOE designs beat all other designs in these two measures, even though it optimizes for only the discrepancy.

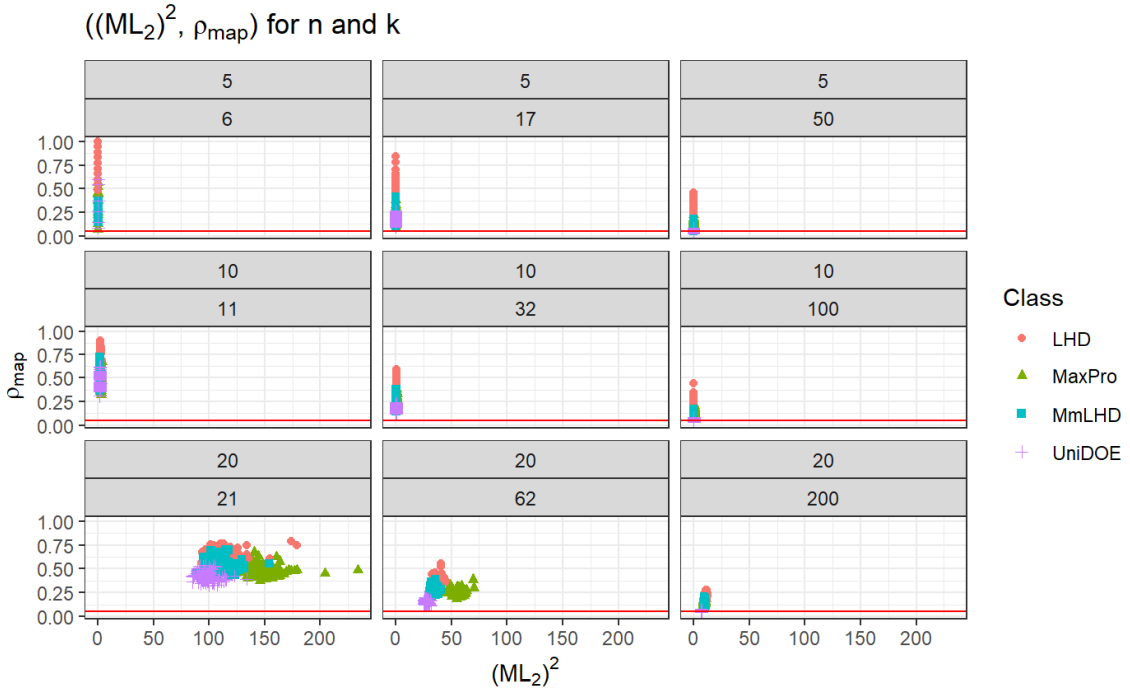


Figure 60. Multiobjective scatter plots of ρ_{map} and $(ML_2)^2$ reduced to four design classes.

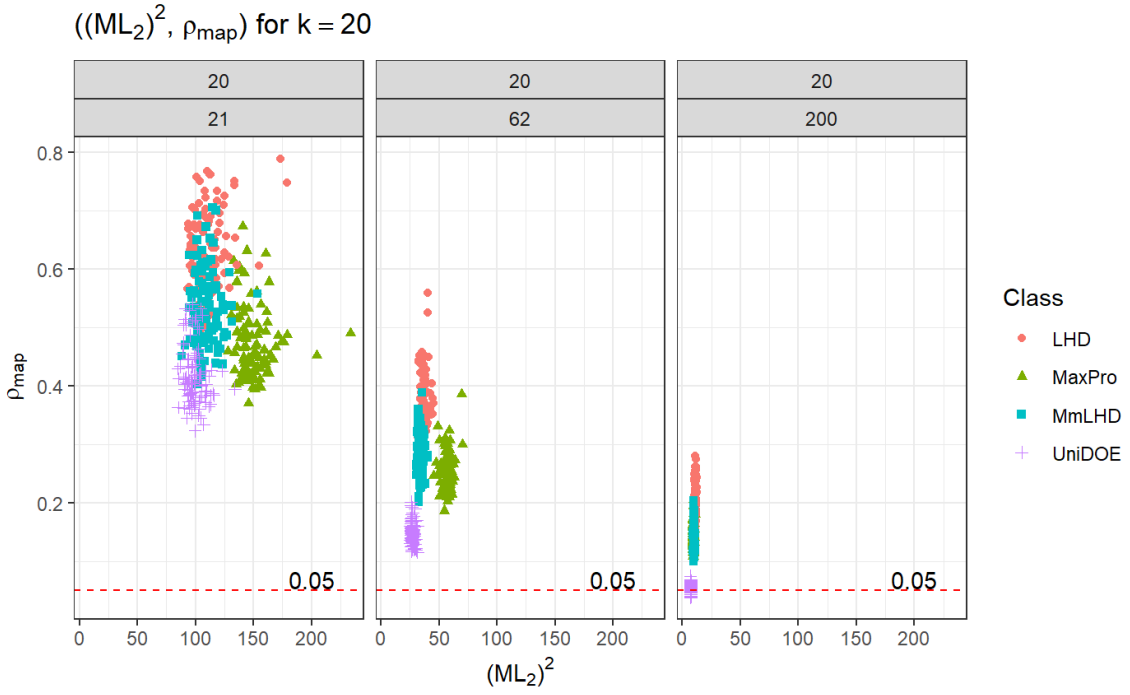


Figure 61. Scatter plots for $k = 20$ (in terms of ρ_{map} and $(ML_2)^2$).

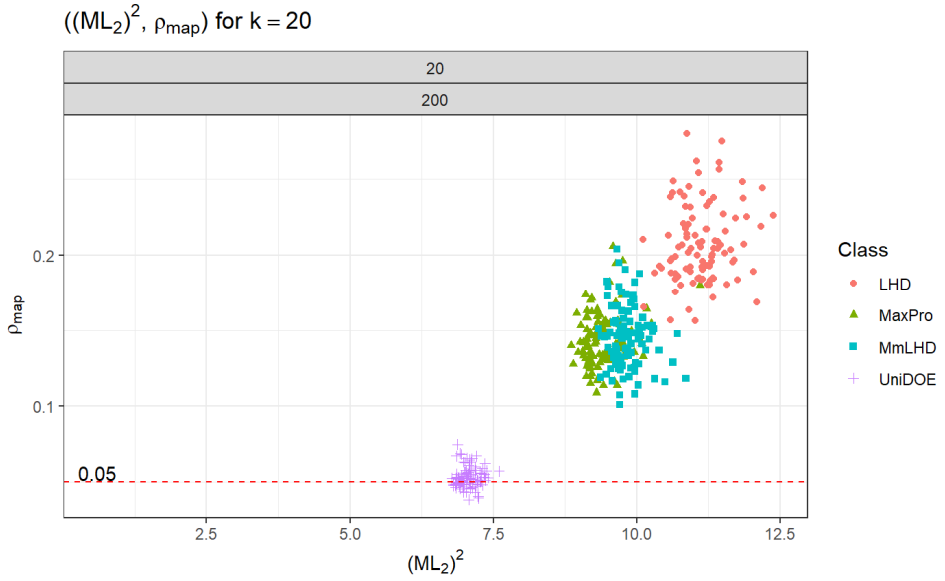


Figure 62. ρ_{map} and $(ML_2)^2$ scatter plots of 400 200×20 designs.

In large-scale computational experimentation, the *UniDOE* R software package consistently arrives at designs with low ρ_{map} and discrepancy ($(ML_2)^2$), as seen in Figure 62. For large degrees of freedom (DOF) (i.e., when $n = 10k$), the MaxPro criterion improves projectivity and lowers discrepancy. While they have significantly overlapping ρ_{map} values for most design sizes, the top performer between MaxPro designs and MmLHDs with respect to $(ML_2)^2$ changes dramatically as n increases when $k = 20$. Specifically, when $n = k+1$ or $3k+2$, the MmLHDs almost always have lower $(ML_2)^2$ values, while this ordering reverses when $n = 10k$.

This exploration focused on just one pair of measures (ρ_{map} and $(ML_2)^2$). There are $\binom{10}{2} - 1 = 44$ other combinations of two measurement relationships we could have looked at to understand better the quality of SFDs resulting from the differing algorithms for multiple design sizes. Next is an assessment of the relationships among a few pairs of measures (e.g., ρ_{map} and $|\overline{\rho}|$) for a subset of the designs.

E. CORRELATIONS AMONG SELECT MEASURES FOR MMLHDS, SPHERE-PACKING DESIGNS, AND LHDS

The design-quality measures we have looked at fall into three broad categories; correlation, discrepancy, and distance. For each category, several options are available; for example, ρ_{map}, $\overline{|\rho|}$, and $\overline{\rho^2}$ have been used by various authors to minimize or assess a design's correlation. Does it matter which we use? What about relationships among measures in different categories? Do designs with low discrepancy measures also tend to have good distance and correlation measures? If so, does this depend on the design class and dimensions? There is a dearth in the literature addressing such questions. As a first step towards answering questions such as these, this section explores correlations among a subset of our design metrics for three classes of SFDs. Appendix G tabularizes all correlations examined; that is, between all ten measures (yielding 45 pairwise comparisons) for six classes of designs of nine sizes. Appendix G includes both LHSs and LHDs.

1. Correlation Analysis Motivation and Approach

Jin et al. (2003 p. 554) write, "there are many optimality criteria available in the literature, the comparison between different design criteria is certainly one of the most important problems in the field of design of computer experiments and deserves a thorough future investigation." With so many measures, types, and sizes of SFDs, these explorations use simple correlation to measure linear dependence.

Table 14 shows Pearson's correlation coefficients (Pearson 1895 and Pearson and Filon 1898) between ρ_{map} and itself (top row), $\overline{|\rho|}$ (middle row), and $(ML_2)^2$ (bottom row) for each of six SFD types of size 200×20. For example, we see that $r(\overline{|\rho|}, \rho_{map}) = 0.202$ for MmLHDs of this size. Since this empirical outcome is the result of a random process, we need to determine if this positive association is likely real or simply the result of chance.

Table 14. Pearson correlation coefficients, $r(x, y)$, for six design classes of dimension 200×20.

Correlations $\mathbf{r(x, y)}$								
	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
$10k$	ρ_{map}	ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.202	0.303	0.27	0.107	0.286	0.41
		$(ML_2)^2$	-0.107	0.266	0.102	0.012	0.155	0.244

Figure 63 plots all 100 observations of the $\overline{|\rho|}$ and ρ_{map} pairs for the 200×20 MmLHDs. We see a very weak linear relationship, as expected with $r(\overline{|\rho|}, \rho_{map}) = 0.202$. It turns out that this weak positive correlation is statistically significant at the 0.05 level; more on this later. The lack of a strong positive correlation means that these two correlation measures can yield dramatically different rankings of these designs.

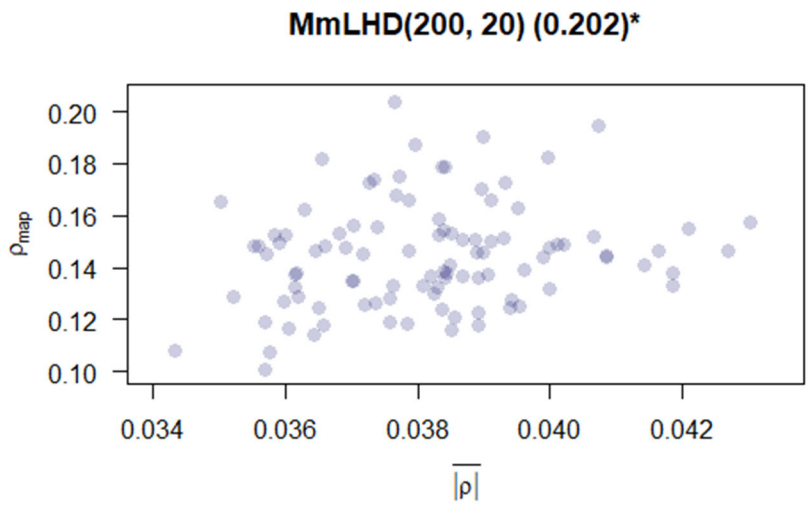


Figure 63. Scatter plot of $\overline{|\rho|}$ versus ρ_{map} for the 100 200×20 MmLHDs from JMP. For the data, $r(\overline{|\rho|}, \rho_{map}) = 0.202$ and the p-value ≤ 0.05.

We show many similar plots in what follows, so some explanation is worthwhile. Following Wang et al. (2020, p. 10), a shorthand notation for the design's correlation between variables x and y is MmLHD(n, k) ($r(x, y)$), which is Figure 63's main title. That is, MmLHD(200, 20) (0.202) denotes that the design examined is the MmLHD class for $n = 200$ and $k = 20$—and the resultant $r(x, y)$ is 0.202. In this case, x and y are paired vectors corresponding to the $\overline{|\rho|}$ and ρ_{map} values of the 100 MmLHDs, respectively. In this work, we append significance stars to denote the level of statistical significance (if any) when testing the null hypothesis that $r(x, y) = 0$ versus the alternative hypothesis that $r(x, y) \neq 0$. Lastly, the scatter plot shown in Figure 63 is scaled to the observational data; i.e., the ranges of the variables x and y. Subsequent scatter plots are labeled and rescaled following this format.

2. Parametric Approach

This subsection discusses Pearson's correlation coefficient in evaluating linear relationships among design measures and applies a commonly used parametric significance test (Pearson 1895 and Pearson and Filon 1898). The test uses Pearson's correlation coefficient (i.e., Equation [1]) and assumes that the two random variables are normally distributed. In such a case, when X and Y are independent; i.e., $\rho(X, Y) = 0$, then the test statistic $t = \dfrac{r(x, y)\sqrt{n-2}}{\sqrt{1-r(x, y)^2}}$ has a t distribution with $n - 2$ degrees of freedom, where $r(x, y)$ is the sample correlation coefficient (Fisher 1925). Since $\overline{|\rho|}$ and ρ_{map} are both limited to [0, 1], they cannot have a bivariate normal distribution. The question is, are they close enough to being normally distributed that Pearson's test will provide reliable results? To assess this, we inquire as to the robustness of the procedure and we test for normality.

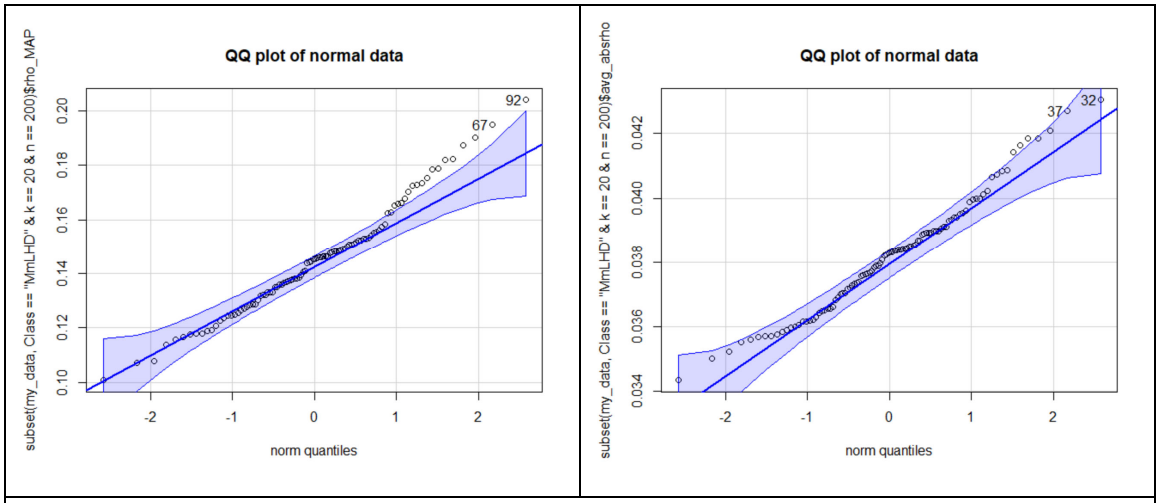
As for the robustness of the procedure, Kowalski (1972, p. 2) quotes E. S. Pearson on the robustness of the normality assumption:

> [A]fter studying samples of sizes 20 and 30 from two "considerably non-normal distributions" (each a mixture of bivariate normal distributions) with respective correlation coefficients of 0.5346 and 0.4626, concluded that "the normal bivariate surface can be mutilated and distorted to a remarkable degree without affecting the frequency distribution of r."…"these results

emphasize the insensitivity of the distribution of r to change in the population form."

Additionally, Hey (1938) concluded: "considerable non-normality in the original distribution will not affect the distributions of correlation" (Kowalski 1972, p. 2). Therefore, this parametric statistic appears to be an appropriate statistical method even if the data are not perfectly normally distributed.

To evaluate how close $\overline{|\rho|}$ and ρ_{map} are to being normally distributed, we use quantile-quantile (q-q) plots and the Shapiro-Wilk test for normality (Shapiro and Wilk 1965). Figure 64 shows the q-q plots for ρ_{map} (left) and $\overline{|\rho|}$ (right) for the 200×20 MmLHD data series using the *stats* R software package (R Core Team 2021). The numerical results (test statistic and associated p-value) of the Shapiro-Wilk normality test are given below each q-q plot.



Shapiro-Wilk normality test	
W = 0.9794, p-value = 0.1193	W = 0.9793, p-value = 0.1174

Figure 64. Two quantile-quantile (q-q) plots and Shapiro-Wilk normality test results for ρ_{map} (left) and $\overline{|\rho|}$ (right) for the MmLHD(200, 20) data series.

The visual comparison of the empirical quantiles against the theoretical quantiles and the results from the Shapiro-Wilk normality test; i.e., fail to reject at classically used

significance levels, are evidence that the data are "nearly" normally distributed. Given the q-q plots and the results of the Shapiro-Wilk normality test, we find distributions of ρ_{map} and $\overline{|\rho|}$ to be appropriately close enough to normal to apply the test, especially given the robustness of the procedure.

In the context of the parametric test, we provide the values for "significance stars," which we put after $r(x,y)$ values in the main titles of our plots. If the p-value falls within the range of alpha in parentheses (e.g., 0.01 – 0.05), then we append the corresponding symbol (e.g., "*") to indicate one of the following significance levels:

- Alpha (0.000 - 0.001): symbol is "***"
- Alpha (0.001 - 0.01): symbol is "**"
- Alpha (0.01 - 0.05): symbol is "*"
- Alpha (0.05 - 0.10): symbol is "."
- Alpha (0.10 - 1.00): (no symbol),

Interpreting correlation, e.g., $r(\overline{|\rho|}, \rho_{map}) = 0.202$* (see Figure 63), is dependent on statistical significance. Here, the significance star (*) after the correlation value indicates the level of statistical significance alpha is in the interval (0.01 - 0.05). The 95 percent confidence interval (CI) on the population correlation coefficient is: [0.005, 0.383], which notably does not contain zero.

Concerning the categorical statements about the value $r(x,y)$, Schober et al. (2018, p. 1765) explain that "most researchers would probably agree that a coefficient of < 0.1 indicates a negligible and > 0.9 a very strong relationship, values in between are disputable." We like the stratifications (cutoff points) and qualitative interpretations provided by Schober et al. (2018, p. 1765) for values of $|r(x,y)|$. Specifically,

- |r(x,y)| = 0.00-0.10 is negligible,
- |r(x,y)| = 0.10-0.39 is weak,

- |r(x,y)| = 0.40-0.69 is moderate,
- |r(x,y)| = 0.70-0.89 is strong, and
- |r(x,y)| = 0.90-1.00 is very strong correlation.

Thus, MmLHD(200, 20) (0.202)* means a weak statistically significant correlation at the 0.05 level for ρ_{map} and $\overline{|\rho|}$ (per the interpretations above). The absence of a stronger correlation was a bit surprising to the author, given that $r(x, y)$ in this case consists of two measures (ρ_{map} and $\overline{|\rho|}$) that both use the pairwise correlations in **X**. The bottom line is that for the 100 200×20 MmLHDs JMP generated, while there is a small positive correlation, ρ_{map} and $\overline{|\rho|}$ yield very different rankings of which designs are preferred.

3. Nonparametric Approach

We can avoid distributional assumptions like multivariate normality by leveraging a nonparametric framework. Casella and Berger (2002) explain that the bootstrap is a "simple, general technique for obtaining a standard error of any statistic" (p. 519). The R software package *boot* (Canty and Ripley 2021 and Davison and Hinkley 1997) uses resampling to generate nonparametric *bootstrap* (Efron 1992) samples. Resampling aims to approximate the sampling distribution of the statistics of interest and quantify the uncertainty of an estimate (e.g., standard deviation) by repeatedly resampling (with replacement) from the observed data.

Consider the 100 observed $\overline{|\rho|}$ and ρ_{map} pairs from the 200×20 MmLHD data series represented by the blue dots in Figure 63. Our point estimate of the correlation is $r(\overline{|\rho|}, \rho_{map}) = 0.202$. Of course, there is uncertainty associated with this measure that we would like to quantify. To do so, we use the bootstrap method to estimate the sampling distribution of $r(\overline{|\rho|}, \rho_{map})$ by the following:

1. Resample from the original data (i.e., 100 observed $\overline{|\rho|}$ and ρ_{map} pairs) with replacement,
2. Compute the estimate of $r(\overline{|\rho|}, \rho_{map})$ using the 100 resampled pairs, and

3. Repeat steps (1) and (2) 10,000 times to approximate the sampling distribution of $r(|\overline{\rho}|, \rho_{map})$. To distinguish resampled correlations from the original sample, label the 10,000 resampled correlations as `t*`.

A histogram of the 10,000 bootstrap estimates (i.e., `t*`) gives us an estimate of the sampling distribution of $r(|\overline{\rho}|, \rho_{map})$ and can be used to construct an approximate confidence interval on $\rho(|\overline{\rho}|, \rho_{map})$ (Efron 1992). Figure 65 shows a histogram of the computed $r(|\overline{\rho}|, \rho_{map})$ values based on 10,000 bootstrap samples and a q-q plot. The values of `t*` range from –0.132 to 0.488, with a mean of 0.200 and a standard error of 0.0836. 98 of the 10,000 resampled correlations are less than or equal to zero. The histogram and q-q plot show that the `t*` values are very close to a normal distribution.

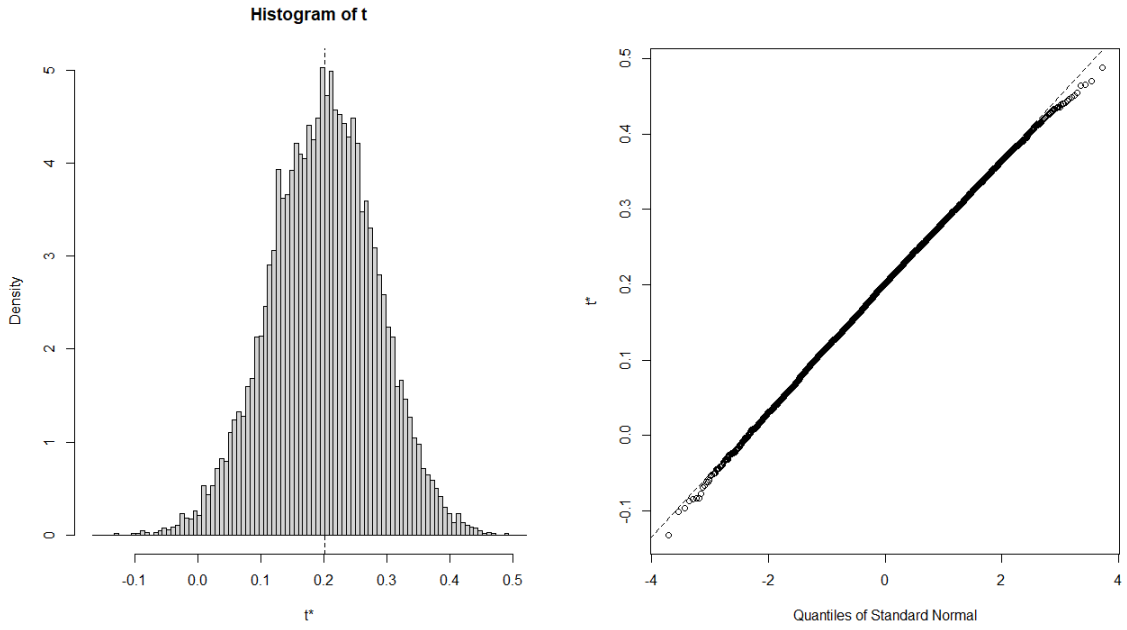


Figure 65. Histogram of the 10,000 resampled correlations (i.e., `t*`) for the 200×20 MmLHD data series. Note $r(|\overline{\rho}|, \rho_{map}) = 0.202$ for the original sample.

Table 15 presents four bootstrap CIs on $\rho(|\overline{\rho}|, \rho_{map})$ that the R function `boot` produces; see Canty and Ripley (2021) for a discussion. We highlight the percentile

interval (PI) that is defined by the middle 2.5 and 97.5 percentiles of `t*` and the bias-corrected and accelerated (BCa) interval that accounts for bias and skewness (Efron and Tibshirani 1986). The 95% bootstrap PI is [0.033, 0.362] and the 95% BCa interval is [0.038, 0.367]. Recall that the 95 percent CI from the parametric method is [0.005, 0.383]. The bootstrap CIs are a bit shorter, as often occurs in practice (Efron and Tibshirani 1986, p. 62). The parametric test is the most conservative (i.e., widest) of our intervals. Thus, we will use the parametric test to provide p-values and significance stars in our plots below.

Table 15. Bootstrap CIs of $r(|\overline{\rho}|, \rho_{map})$ replications for the 200×20 MmLHD data.

```
> boot.ci(boot.out=bootstrap_correlation05,
+         type=c("norm", 'basic', "perc", 'bca'))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = bootstrap_correlation05, type = c("norm",
    "basic", "perc", "bca"))

Intervals :
Level      Normal              Basic
95%   ( 0.0394,  0.3672 )   ( 0.0416,  0.3704 )

Level     Percentile            BCa
95%   ( 0.0328,  0.3616 )   ( 0.0379,  0.3665 )
Calculations and Intervals on Original Scale
```

4. Correlations between Measures for Various Design Classes and Sizes

This subsection displays scatter plots and provides the correlation values between a subset of measures for MmLHDs, sphere-packing designs, and LHDs. For all of them, we present information on nine design sizes. First, we look at relationships between measures of the same category for several design classes, specifically, the following:

- Correlation-based measures:
  - MmLHD(n, k) ($r(|\overline{\rho}|, \rho_{map})$)
  - SphereP(n, k) ($r(|\overline{\rho}|, \rho_{map})$)
  - LHD(n, k) ($r(|\overline{\rho}|, \rho_{map})$)

- Discrepancy-based measures:
    - MmLHD(n, k) ($r((CL_2)^2, (ML_2)^2)$)
    - SphereP(n, k) ($r((CL_2)^2, (ML_2)^2)$)
    - LHD(n, k) ($r((CL_2)^2, (ML_2)^2)$)
- Distance-based measures:
    - MmLHD(n, k) (r(Mm, MaxPro))
    - SphereP(n, k) (r(Mm, MaxPro))
    - LHD(n, k) (r(Mm, MaxPro))

Next, we look at combinations of measures of different types, e.g., discrepancy- and correlation-based measures. The following pairs of measures are plotted for three design classes.

- Correlation- and discrepancy-based measures (i.e., $(ML_2)^2$, ρ_{map}):
    - MmLHD(n, k) ($r((ML_2)^2, \rho_{map})$)
    - SphereP(n, k) ($r((ML_2)^2, \rho_{map})$)
    - LHD(n, k) ($r((ML_2)^2, \rho_{map})$)
- Correlation- and distance-based measures (i.e., Mm, ρ_{map}):
    - MmLHD(n, k) (r(Mm, ρ_{map}))
    - SphereP(n, k) (r(Mm, ρ_{map}))
    - LHD(n, k) (r(Mm, ρ_{map}))

- Discrepancy- and distance-based measures (i.e., $(ML_2)^2$, MaxPro):
  - MmLHD(n, k) ($r((ML_2)^2$, MaxPro))
  - SphereP(n, k) ($r((ML_2)^2$, MaxPro))
  - LHD(n, k) ($r((ML_2)^2$, MaxPro))

We continue this exploration by looking at two correlation measures ($\overline{|\rho|}$ and ρ_{map}) in MmLHDs. Figure 66 builds on Figure 63 by including two additional scatter plot panels for MmLHD(21, 20) and MmLHD(62, 20) (left and middle). Each panel consists of 100 observations, all with k =20. Note that each panel's x-axis, $\overline{|\rho|}$, and the y-axis, ρ_{map}, are rescaled as these values vary significantly by design size). As described above, the panel main titles follow Wang et al. (2020, p. 10); i.e., MmLHD(n, k) ($r(\overline{|\rho|}, \rho_{map})$) and include significance stars. We see that the small positive $r(\overline{|\rho|}, \rho_{map})$ values for the 21×20 and 62×20 designs are not statistically significant. Perhaps a larger sample size would bring out a statistically significant correlation; however, if correlated, the linear association will be very weak.

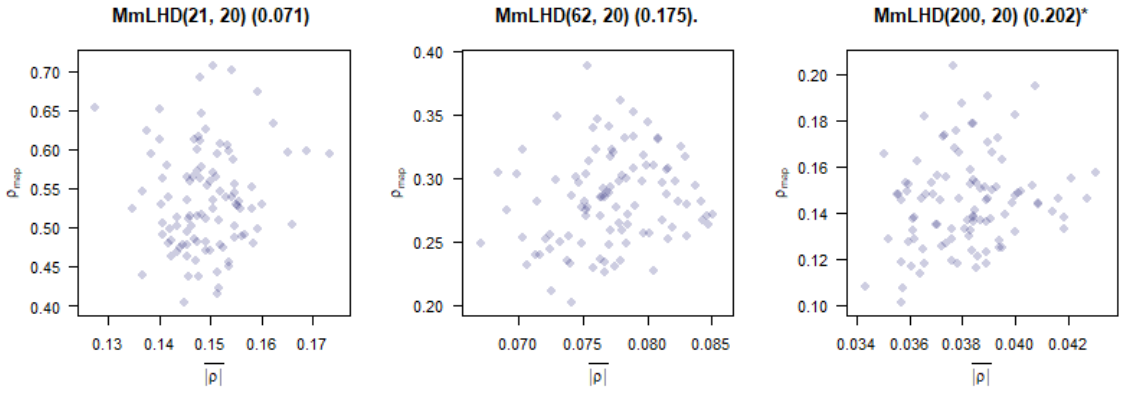


Figure 66. Three scatter plots of MmLHD(n, k) ($r(\overline{|\rho|}, \rho_{map})$) for the 21×20 (left), 62×20 (middle), and 200×20 (right) MmLHD data series. Note that the right panel is Figure 63.

Figure 67 builds rowwise on top of Figure 66 by including two additional three-panel rows of scatter plots for $k = 10$ (middle row) and $k = 5$ (top row). For example, the top left panel of the nine-panel plot is 100 $|\overline{\rho}|$ and ρ_{map} observations for MmLHDs of size 6×5. All nine correlations are positive, and the strongest correlations occur when $k = 5$. For larger designs ($k = 10$ or 20), the empirical correlation between $|\overline{\rho}|$ and ρ_{map} increases in n. When n is small (e.g., 6), some of the measures only have a limited number of possible outcomes.

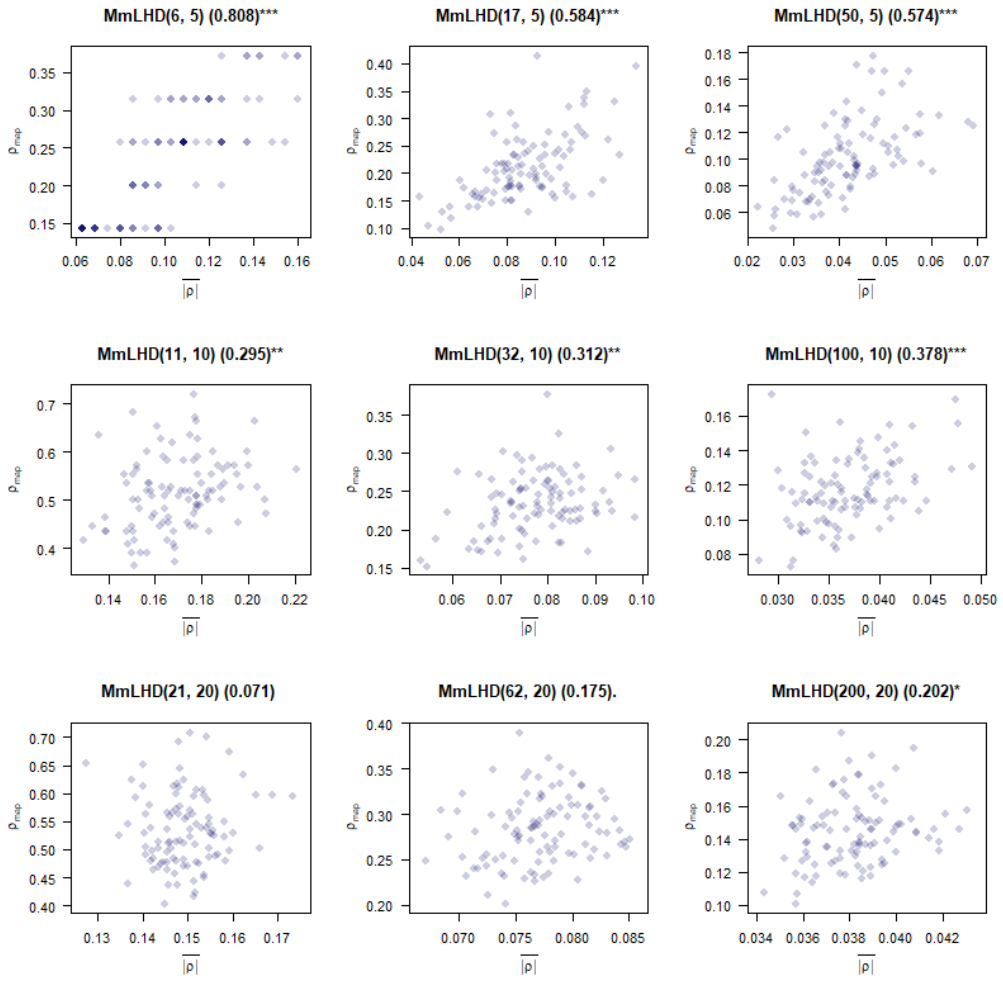


Figure 67. The combined nine-panel plot of MmLHD(n, k) ($r(|\overline{\rho}|, \rho_{map})$) scatter plots. Note the bottom three-panel-row is $k = 20$, which is Figure 66.

Figure 68 presents the combined nine-panel plot for the sphere-packing (Mm Euclidean distance) designs; i.e., SphereP(n, k) ($r(|\overline{\rho}|, \rho_{map})$) from JMP. Pearson's correlation coefficient, $r(|\overline{\rho}|, \rho_{map})$, for the SphereP(6, 5) data series is 0.994, which is the highest of any correlation found. In this case, the two measures are in almost perfect agreement in design ranking. Here too, all nine panels have a positive correlation. All correlations are statistically significant at the 0.01 level except for JMP's 62×20 sphere-packing design. For fixed k, as n increases, the value of $r(|\overline{\rho}|, \rho_{map})$ decreases, except in the 62×20 case.

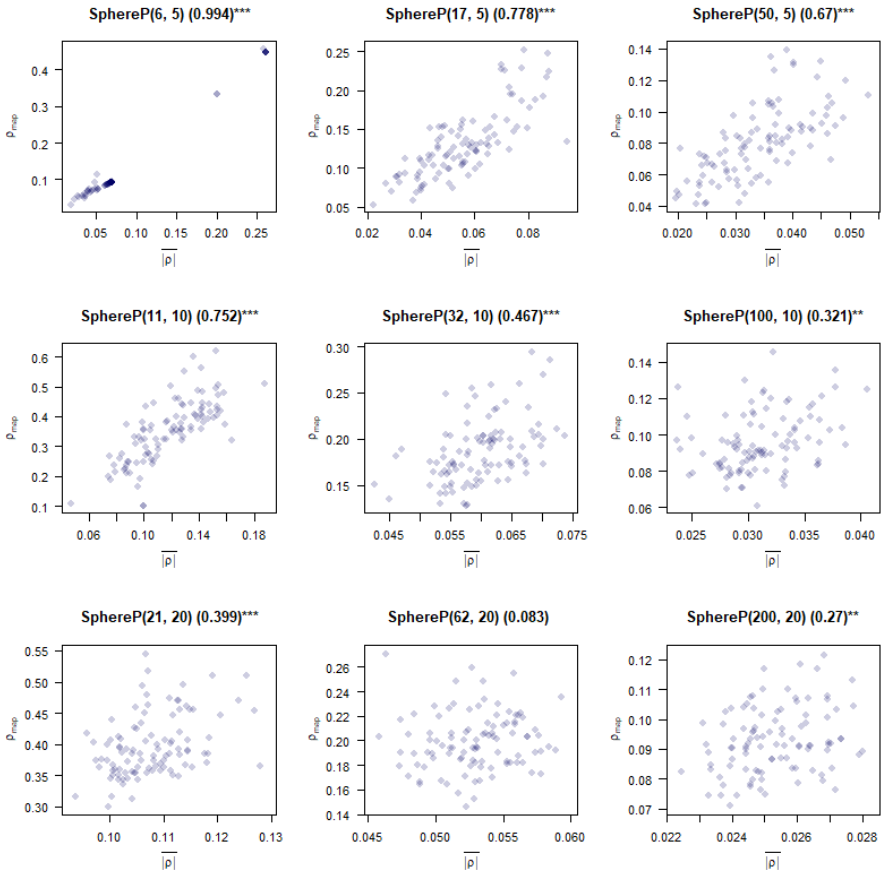


Figure 68. The combined nine-panel plot of SphereP(n, k) ($r(|\overline{\rho}|, \rho_{map})$) scatter plots. Rowwise, the top row is $k = 5$, middle is $k = 10$, and bottom $k = 20$. Columnwise (left, middle, and right), n is a function of k, i.e., $n = k + 1$, $3k + 2$, and $10k$.

Figure 69 shows LHD(n, k) ($r(|\overline{\rho}|, \rho_{map})$) for the nine design sizes. Here again, all the correlations are positive. They are all also statistically significant at the 0.05 level. The strongest correlations occur when $k = 5$.

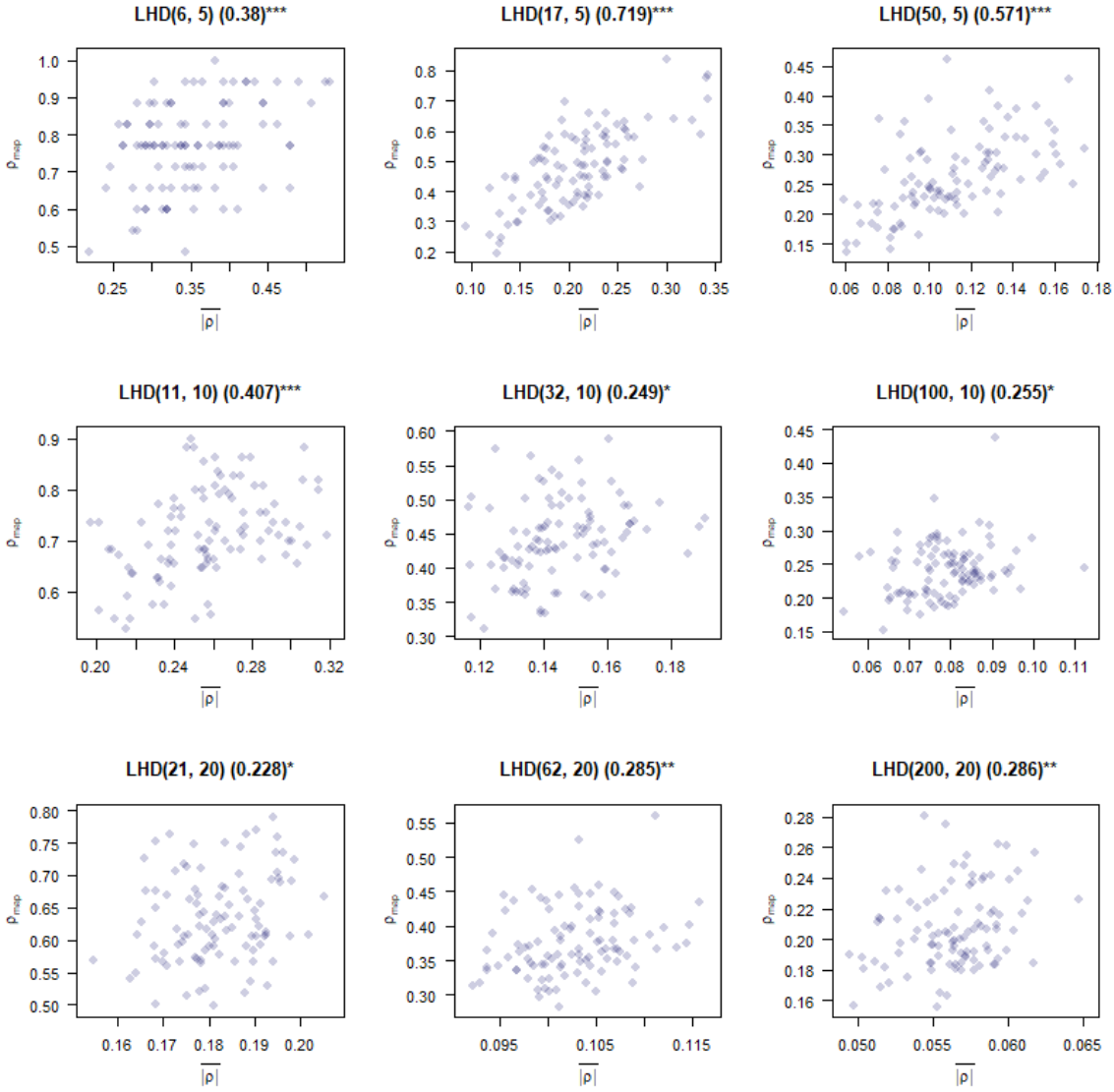


Figure 69. The combined nine-panel plot of LHD(n, k) ($r(|\overline{\rho}|, \rho_{map})$) scatter plots.

Correlations of discrepancy-based measures, $r((CL_2)^2, (ML_2)^2)$, are presented in Figure 70 for MmLHD(n, k)s. When k = 5 or 10, there is a significant correlation for all panels that increases with n. Interestingly, for designs with lots of factors; i.e., k = 20, none of the correlations are statistically significant, with two of them being negative. Thus, $(CL_2)^2$ and $(ML_2)^2$ result in very different orderings of design quality for large k.

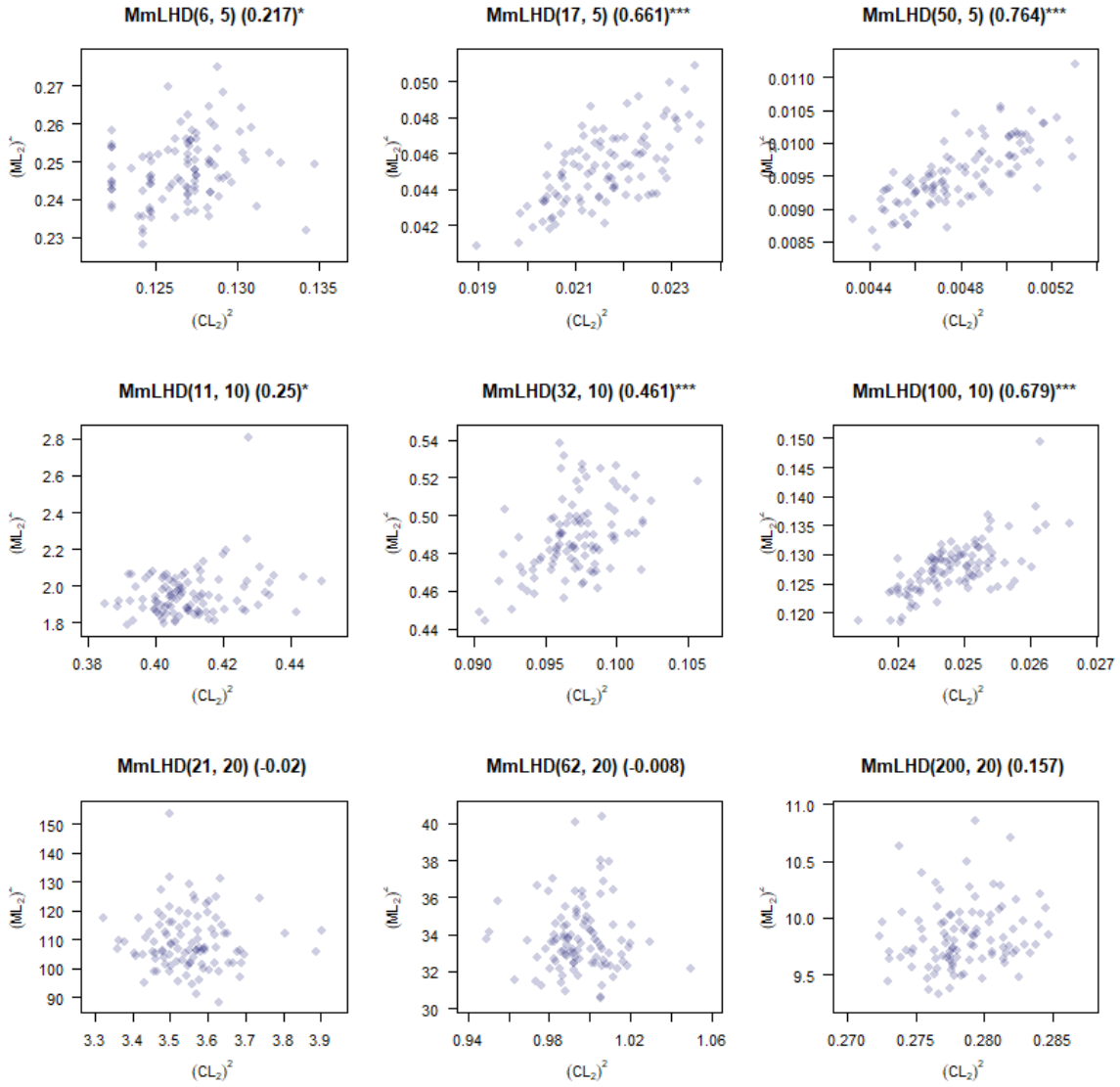


Figure 70. The combined nine-panel plot of MmLHD(n, k) ($r((CL_2)^2, (ML_2)^2)$) scatter plots.

Figure 71 presents the nine-panel plots of SphereP(n, k) ($r((CL_2)^2, (ML_2)^2)$). When k = 5 or 10, all correlations are positive and strong or very strong. When k = 20, the correlation is even negative when n = 21; i.e., SphereP(21, 20) (-0.018), but not statistically significant. As n increases, we see an increasingly positive correlation.

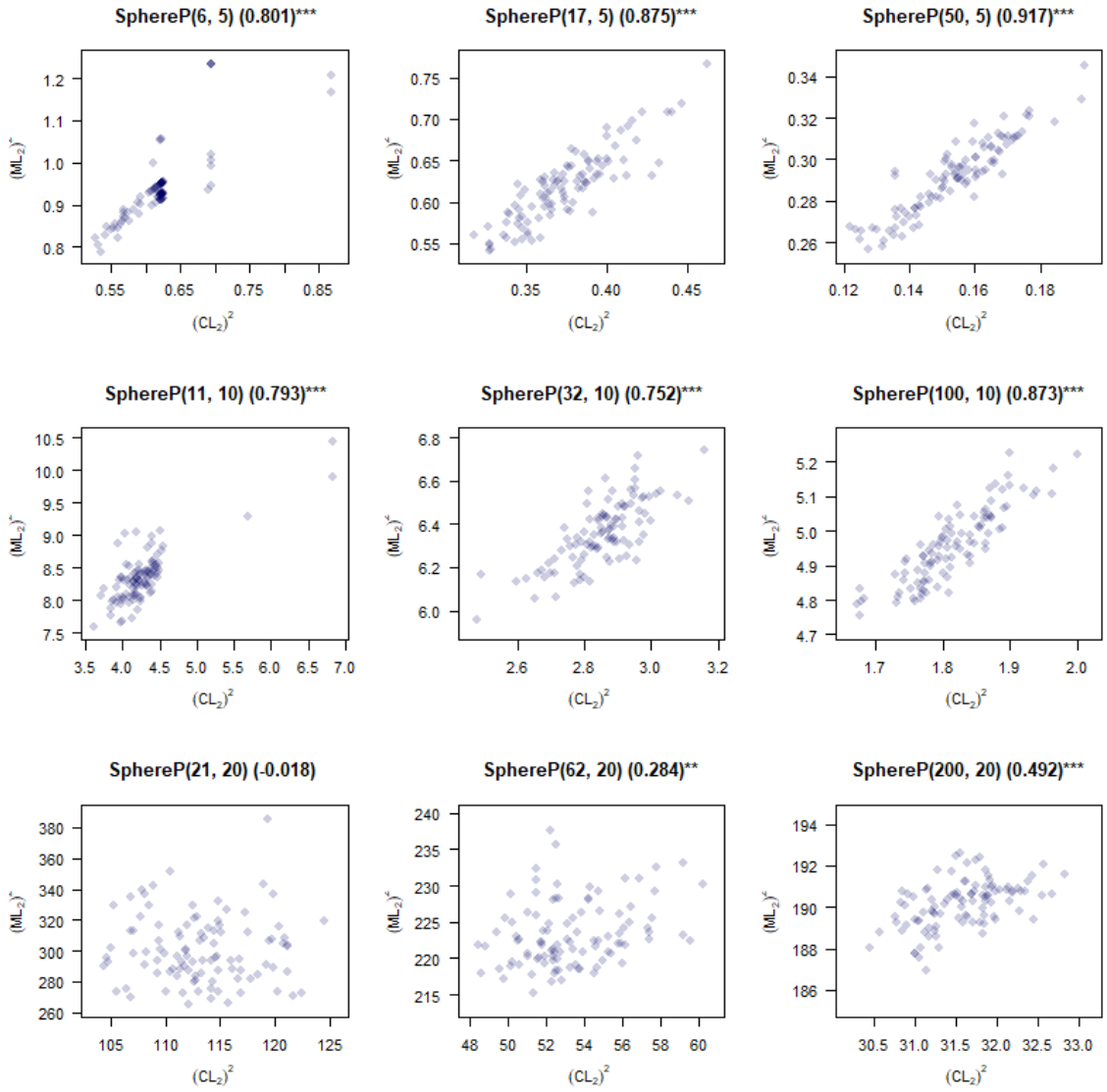


Figure 71. The combined nine-panel plot of SphereP(n, k) ($r((CL_2)^2, (ML_2)^2)$) scatter plots.

Figure 72 presents the nine-panel plots of LHD(n, k) ($r((CL_2)^2, (ML_2)^2)$). All correlations between these two discrepancy measures are positive, with the strongest being for $k = 5$ and $n = 50$ at 0.877. As n increases, the correlations also tend to increase for each row. For each column, as k increases, the correlations tend to decrease.

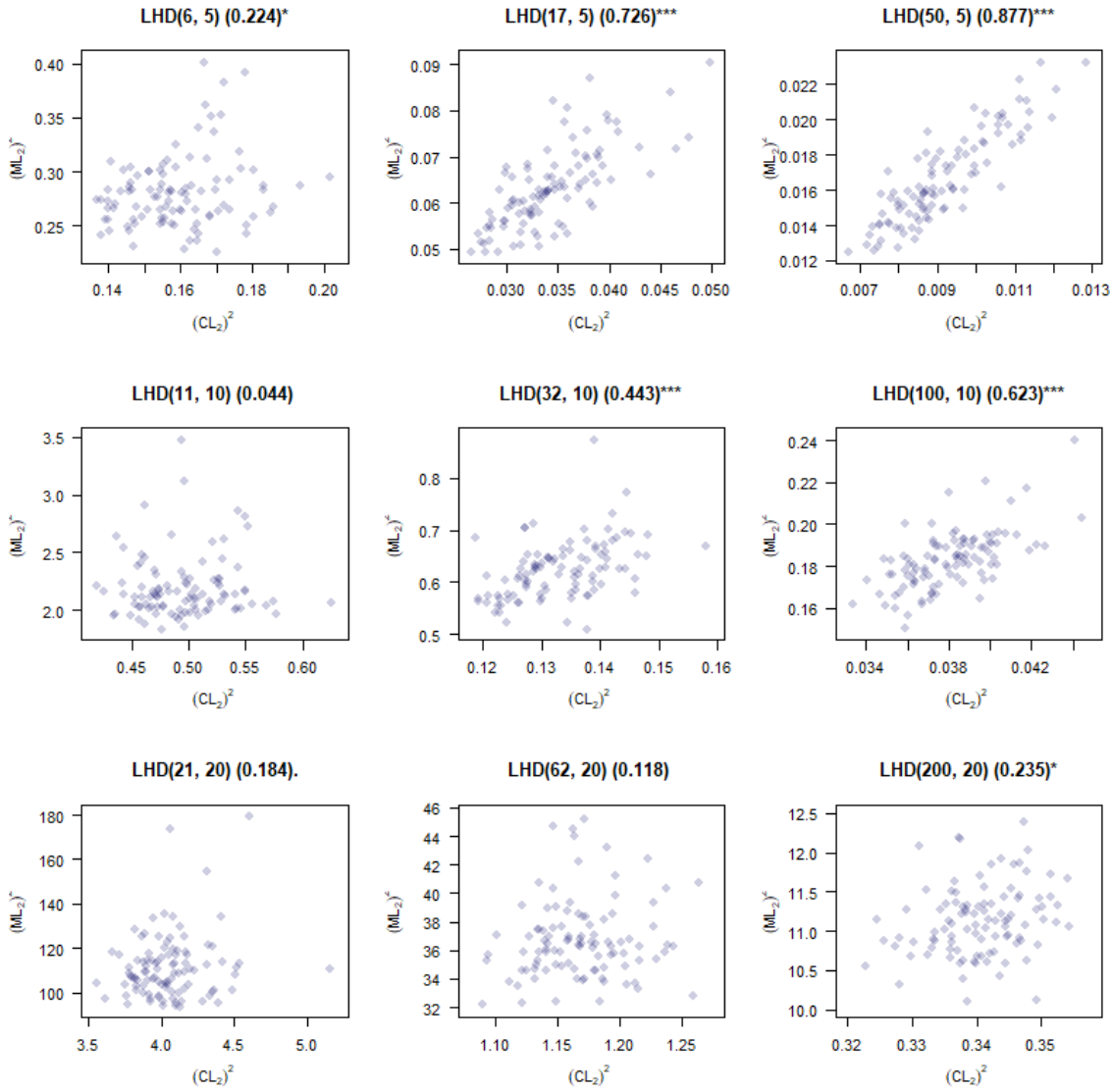


Figure 72. The combined nine-panel plot of LHD(n, k) ($r((CL_2)^2, (ML_2)^2)$) scatter plots.

We now compare the linear relationship of two distance measures; i.e., Euclidean Mm distance and MaxPro values. The sphere-packing design is omitted due to infinite values resulting from DP replication in subspaces. Concerning MmLHD(n, k) (r(Mm, MaxPro)) in Figure 73, none of the nine panels yield a significant correlation. Seven of the nine panels are negative, not surprising since larger Mm values and smaller MaxPro values are desired.

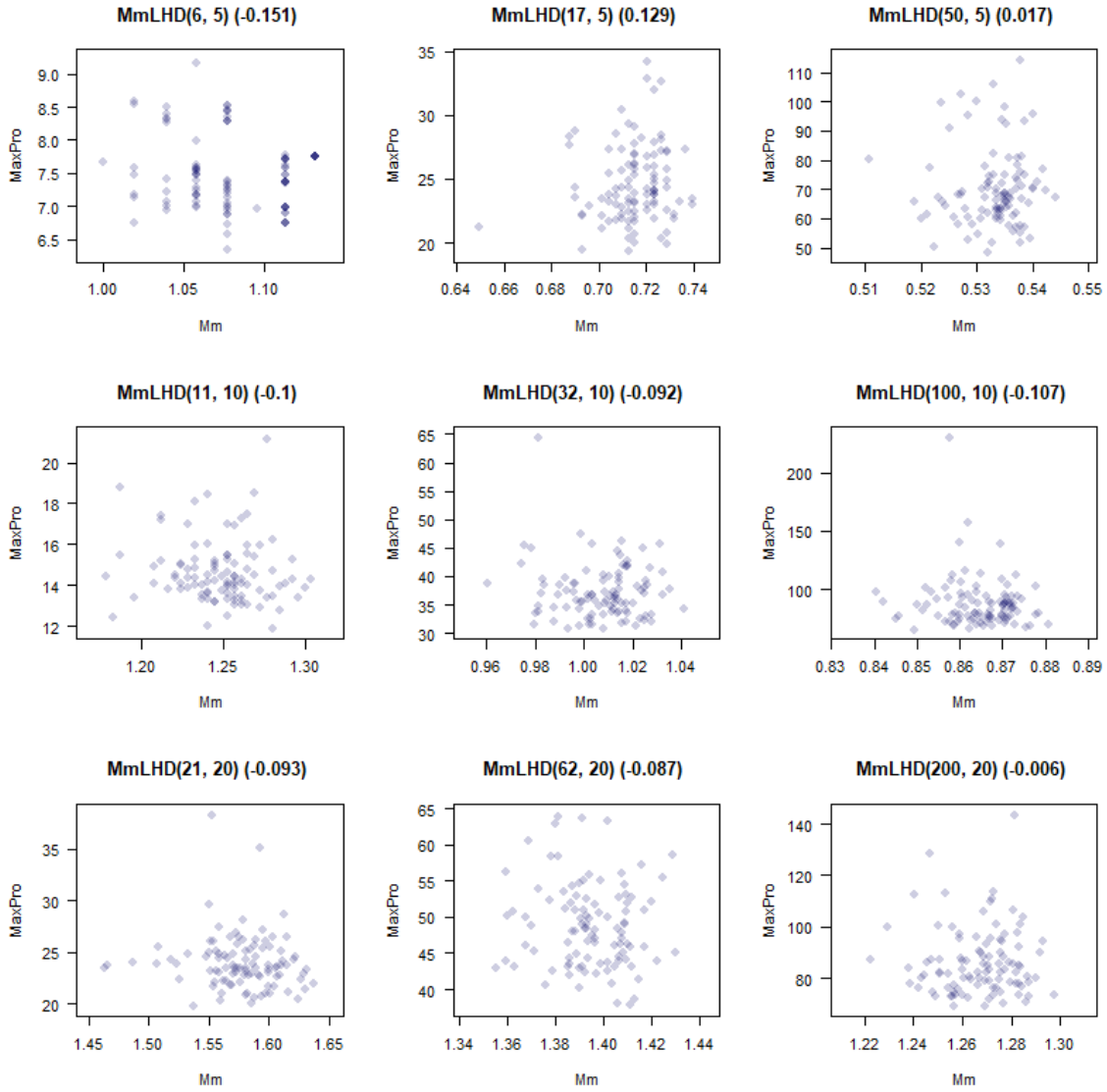


Figure 73. Nine MmLHD(n, k) (r(Mm, MaxPro)) scatter plots.

Figure 74 shows nine LHD(n, k) (r(Mm, MaxPro)) scatter plots. All of the design sizes yield a significant negative correlation. As noted above, higher Mm values and lower MaxPro values are preferred. The correlations are strongest for $k = 5$.

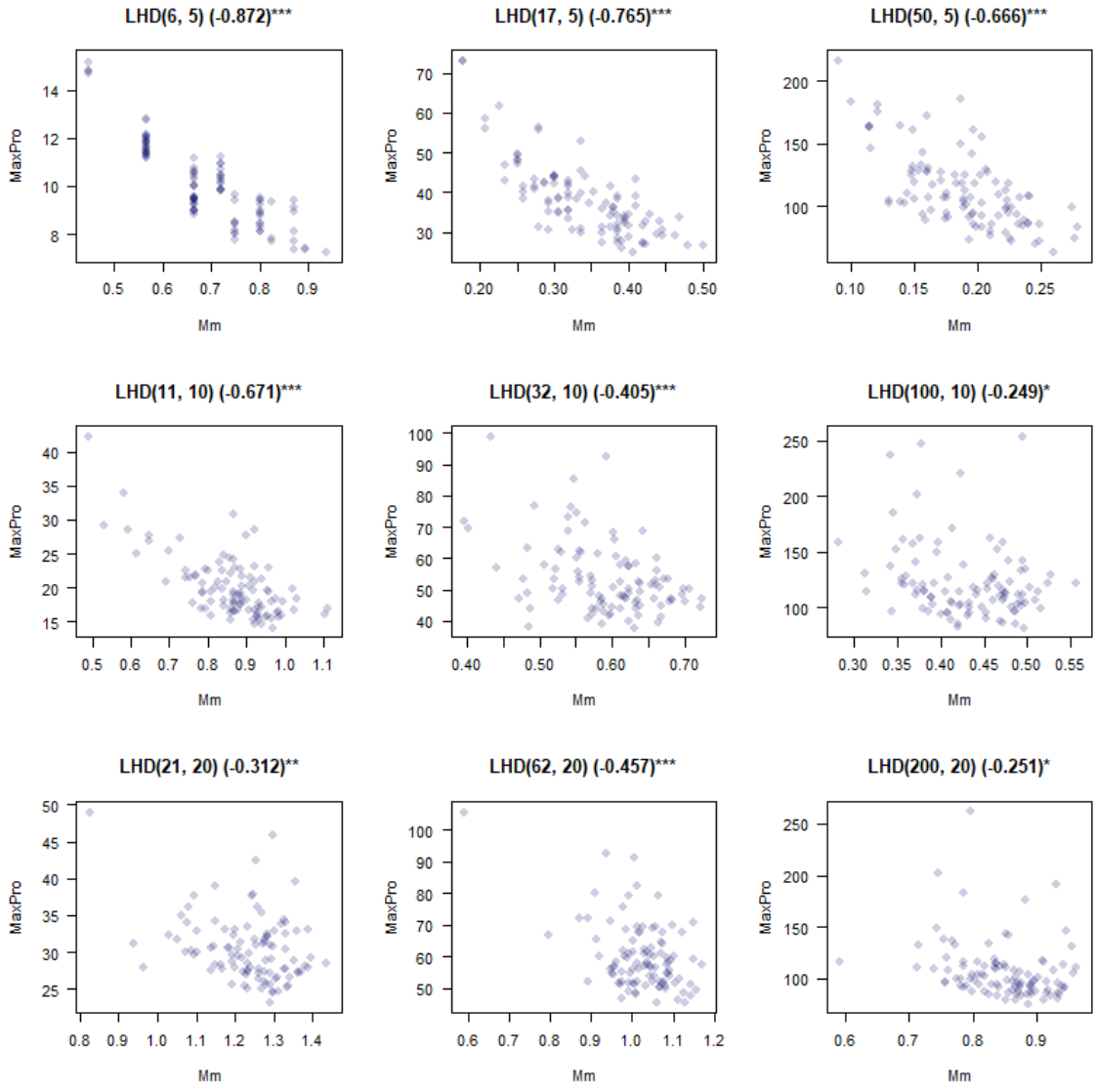


Figure 74. Nine scatter plots of LHD(n, k) r(Mm, MaxPro) correlations for the data.

Next, we examine the linear relationship between different types of design measures. Almost nothing in the literature compares design measures of different types over different design classes and sizes. Wang et al. (2020, p. 10) show six comparative scatter plots. Three show the linear relationship between discrepancy and orthogonality measures for 100 randomly generated LHDs of sizes 19×6, 19×12, and 19×18. The other three show uniform projectivity (Sun et al. 2019) versus orthogonality for the same LHDs. Our measures for each category are ρ_{map}, $(ML_2)^2$, and Mm distance. We continue with our three classes of design (MmLHD, sphere packing, and LHD) and nine design dimensions.

Figure 75 shows nine MmLHD(n, k) ($r((ML_2)^2, \rho_{map})$) scatter plots. When k = 5, all the design sizes have a significant but weak positive correlation. When k = 10, all correlations are positive, but only one is significant. For larger designs, when k = 20, none of the correlations are significant; in fact, two are slightly negative.

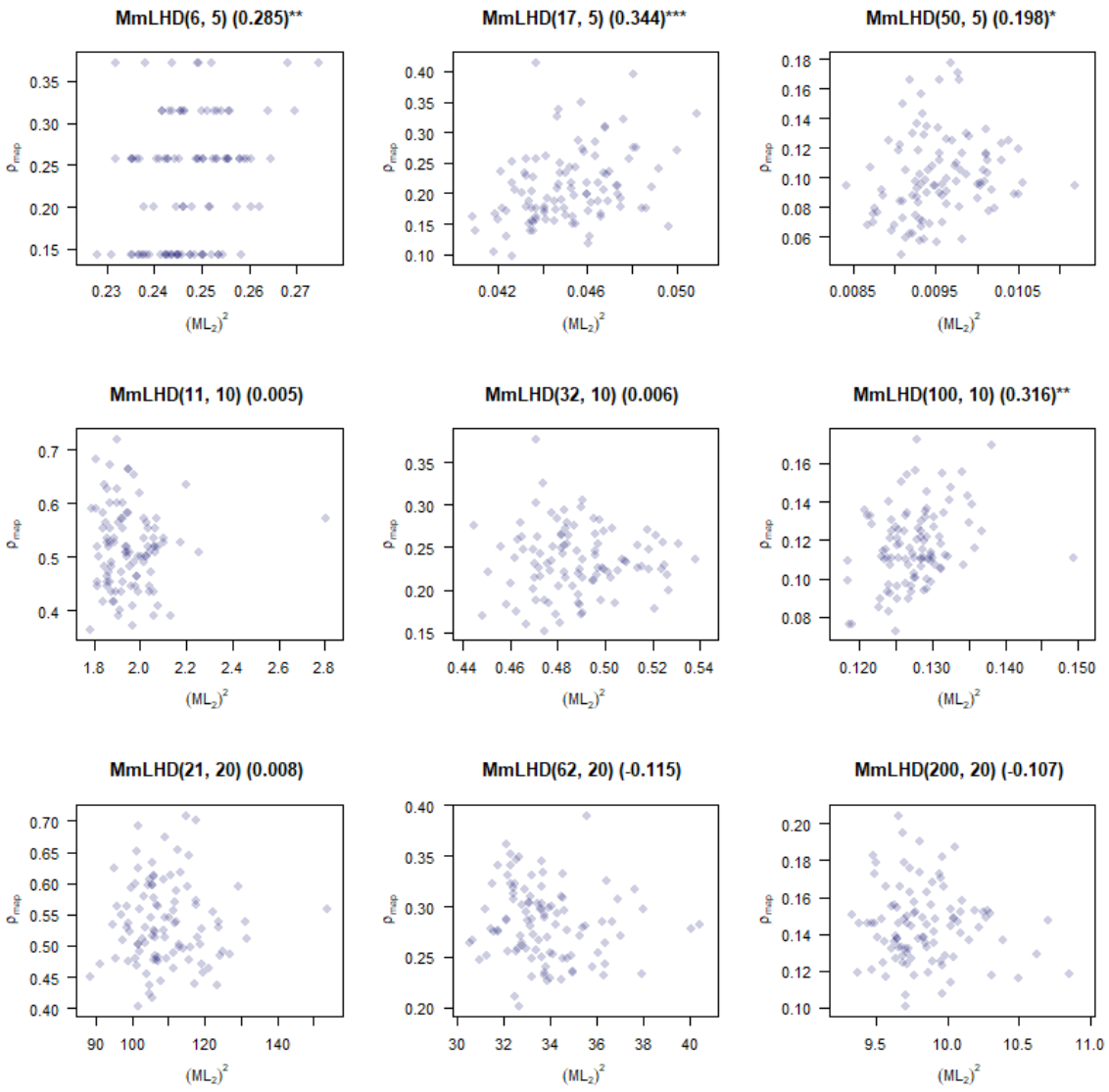


Figure 75. Nine MmLHD(n, k) ($r((ML2)^2, \rho_{map})$) scatter plots.

Figure 76 presents nine SphereP(n, k) ($r((ML2)^2, \rho_{map})$) scatter plots. Only the saturated, small Mm distance design shows a strong positive correlation; i.e., SphereP(6, 5) (0.720)***.

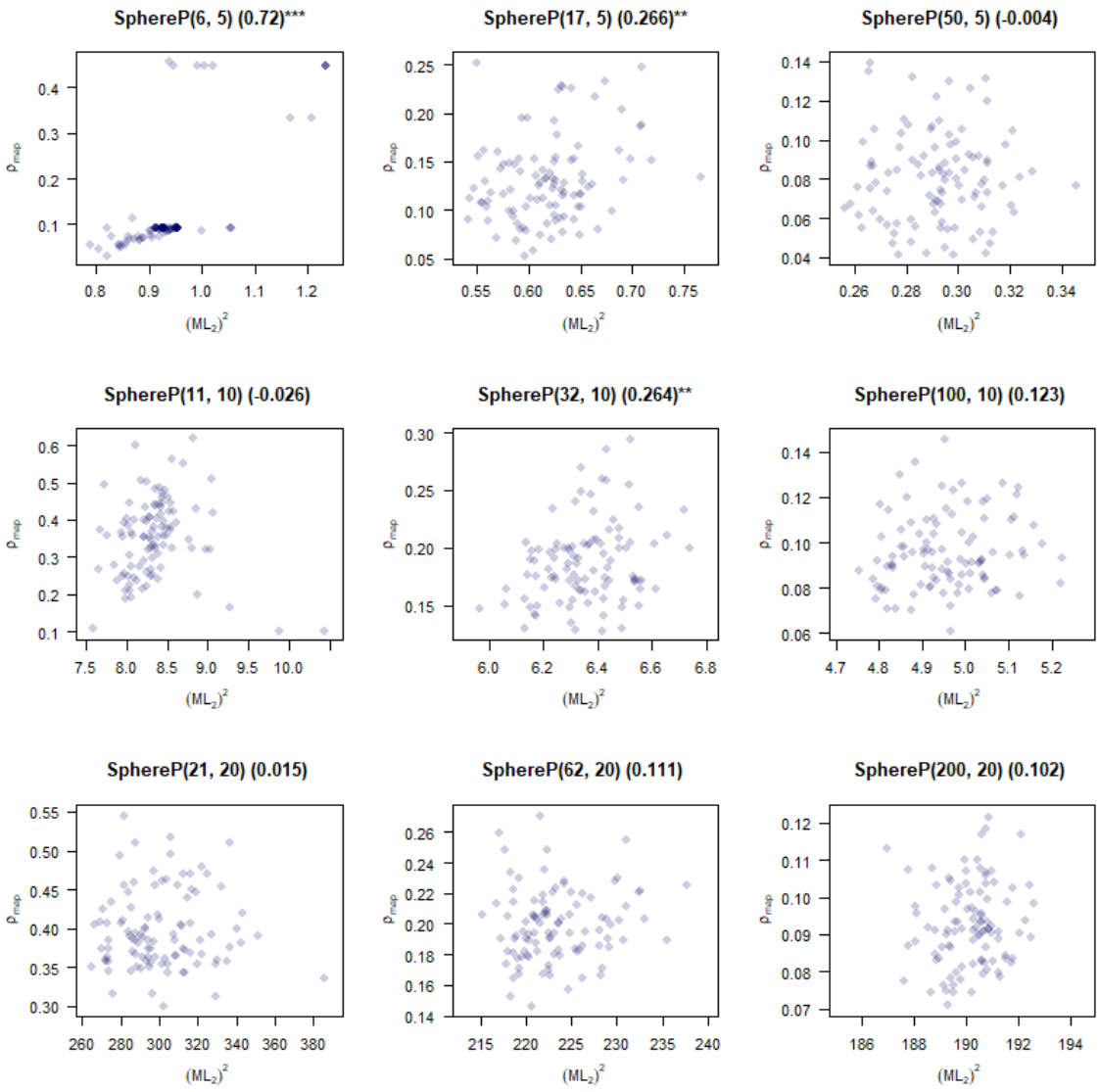


Figure 76. Nine SphereP(n, k) ($r((ML_2)^2, \rho_{map})$) scatter plots.

Figure 77 presents nine LHD(n, k) ($r((ML_2)^2, \rho_{map})$) scatter plots. All of the correlations are positive, but not all are significant. For designs with fewer factors; i.e., $k = 5$ (top row), highly significant moderate positive correlations are observed. Consider $k = 10$ (middle row), as n increases, the three LHD(n, 10) ($r((ML_2)^2, \rho_{map})$) scatter plots show an increasing weak correlation and increasing significance.

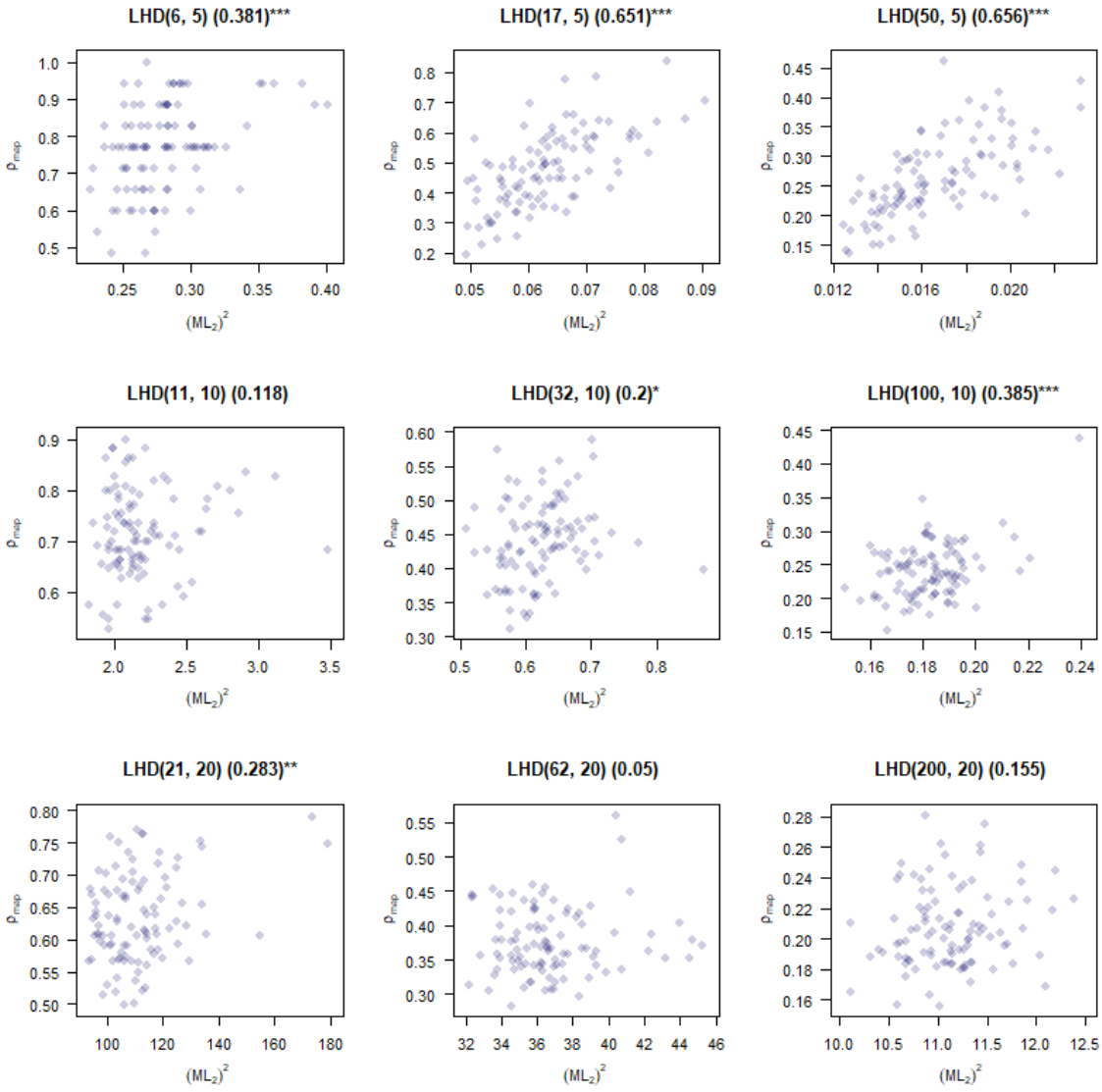


Figure 77. Nine scatter plots of LHD $r((ML_2)^2, \rho_{map})$ correlations for the data.

Figure 78 shows nine MmLHD(n, k) (r(Mm, ρ_{map})) scatter plots that illustrate the correlation between distance- and correlation-based measures for the data. Negative correlations are expected since higher Mm distances and lower ρ_{map} values are favorable. Negative correlation behavior is observed for the saturated design when $k = 5$; see the top-left panel. However, the other eight panels are either not significantly correlated or are weakly negatively correlated.

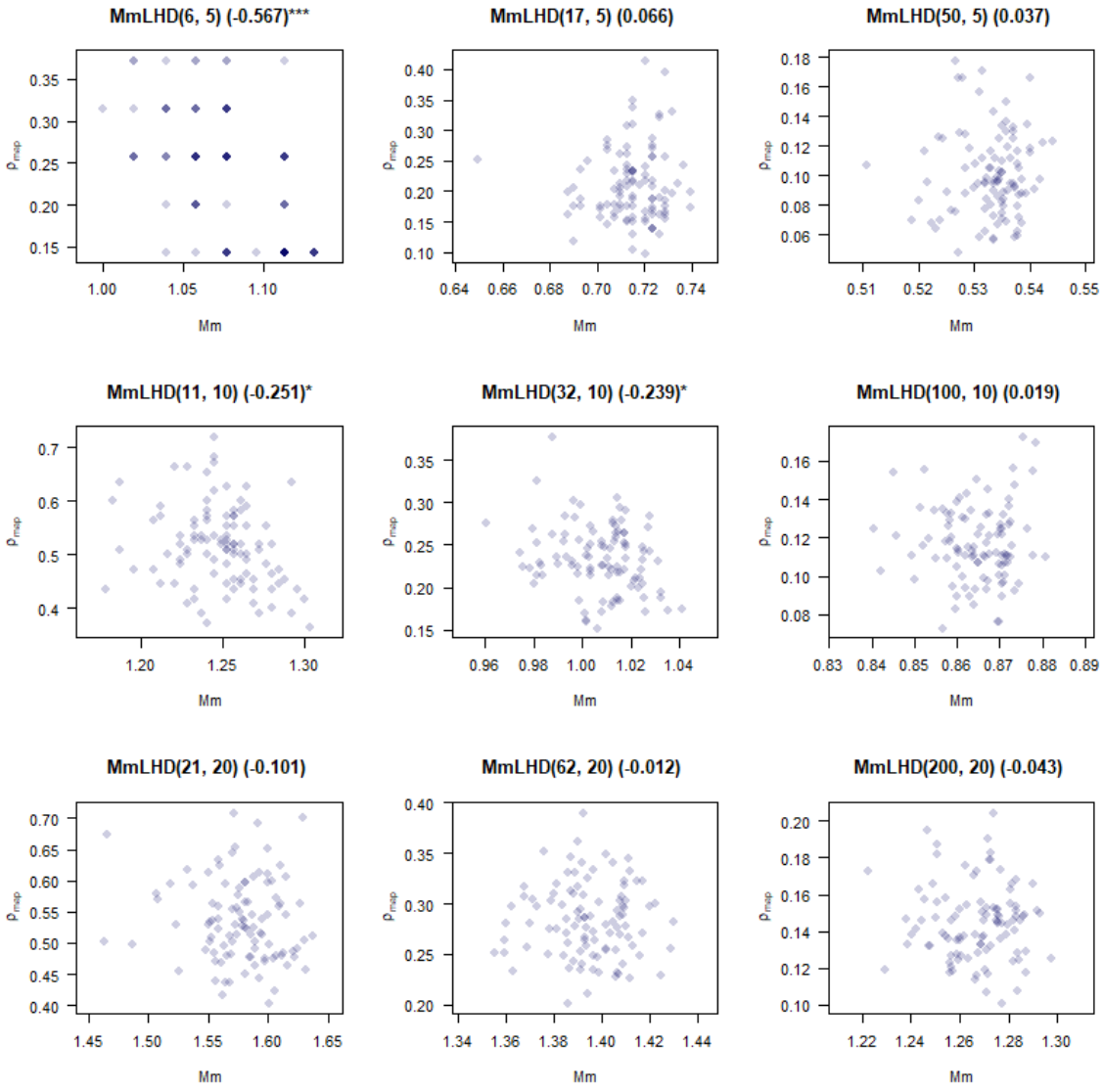


Figure 78. Nine MmLHD(n, k) (r(Mm, ρ_{map})) scatter plots.

Figure 79 shows nine SphereP(n, k) (r(Mm, ρ_{map})) scatter plots. For saturated designs, note the significant negative correlation that decreases with k. For all other design dimensions, there are no statistically significant correlations.

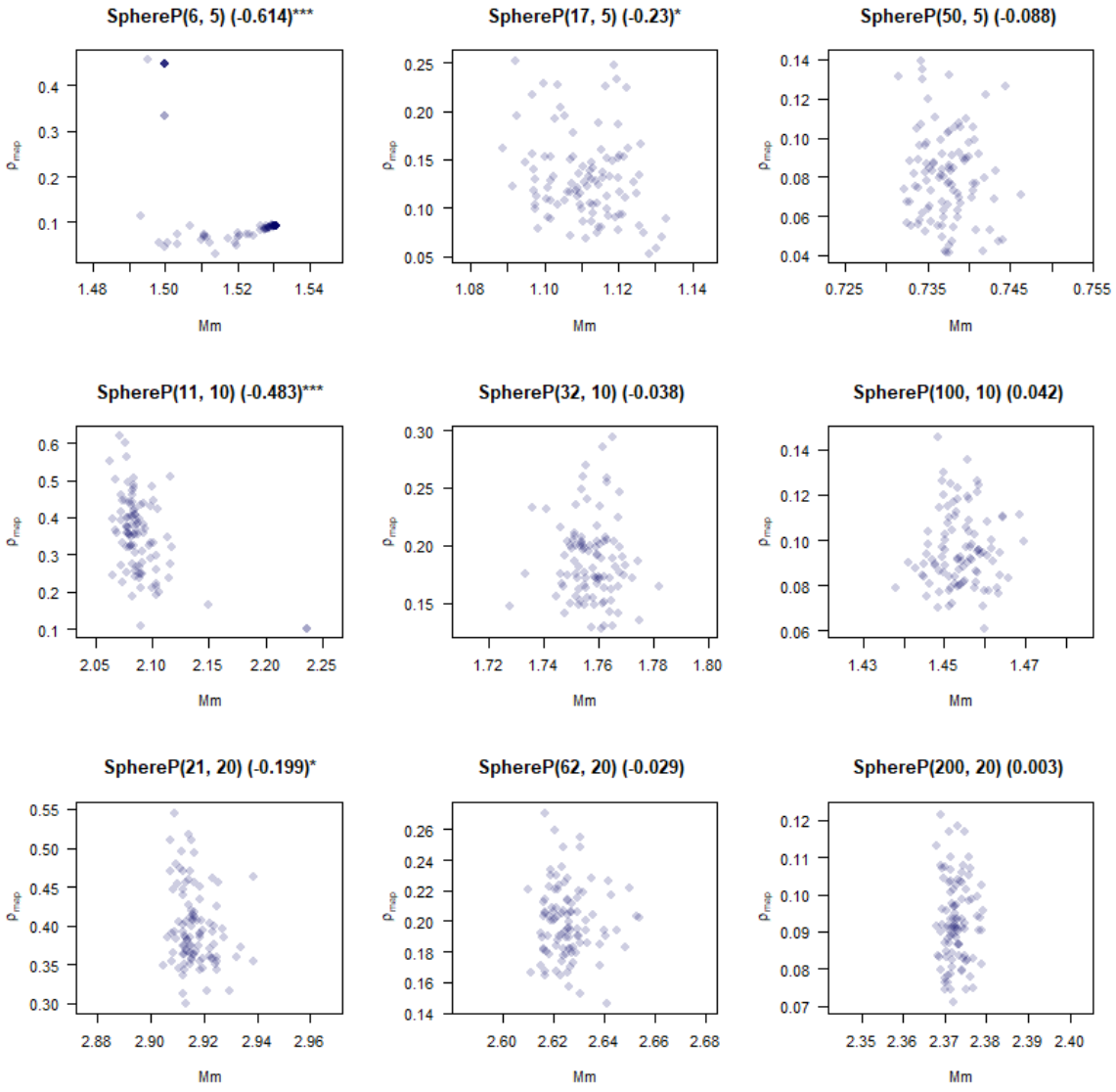


Figure 79. Nine SphereP(n, k) (r(Mm, ρ_{map})) scatter plots.

From Figure 80, we see little to no linear relationship between Mm and ρ_{map}; i.e., negligible correlation, which intuitively makes sense given neither of these measures is used when generating an LHD. The only significant correlations, which are weakly negative, occur in the 6×5 and 17×5 LHDs.

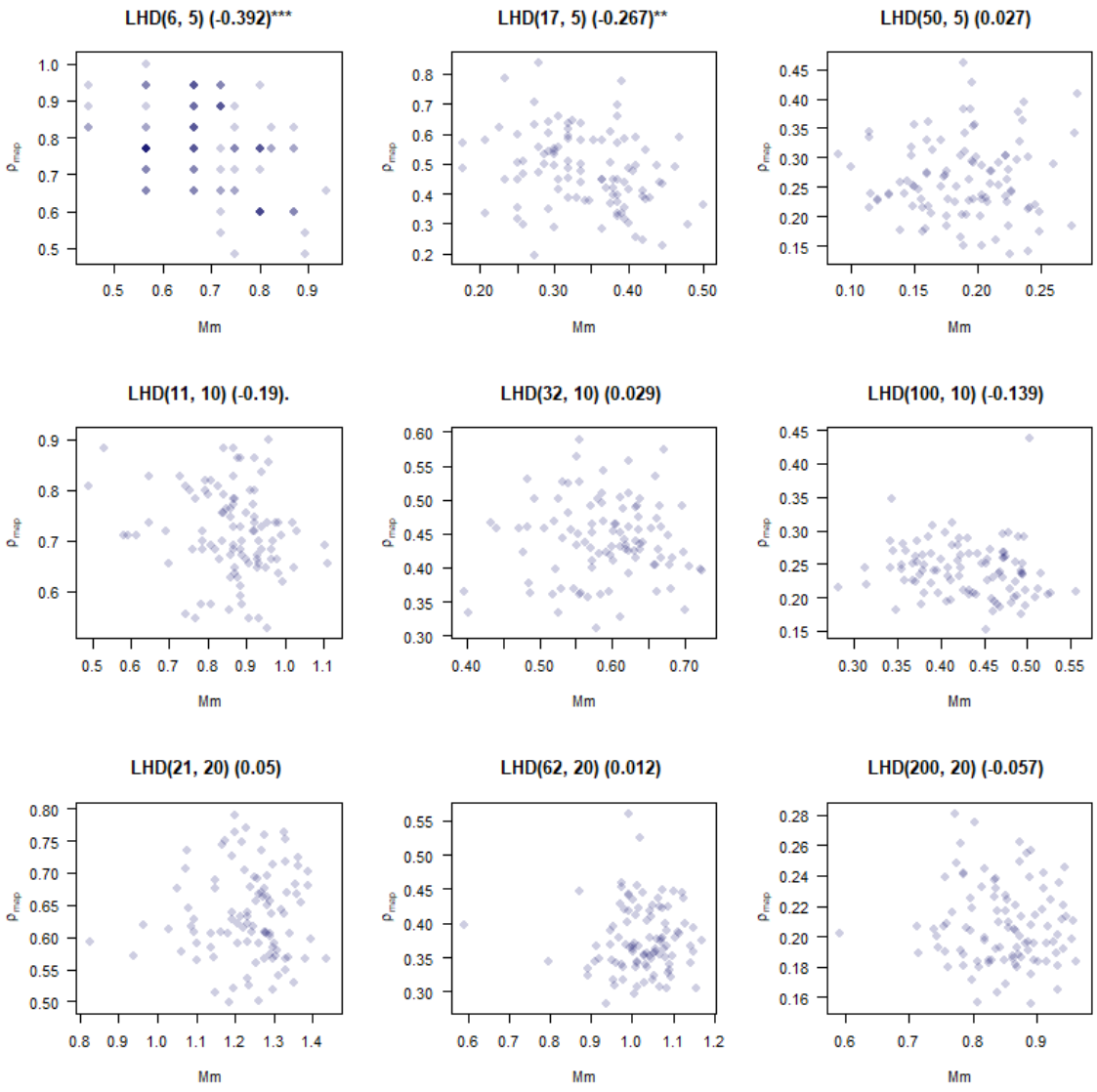


Figure 80. Nine LHD(n, k) (r(Mm, ρ_{map})) scatter plots.

Figure 81 shows the results of the experiments on the correlation between the discrepancy measure $(ML_2)^2$ and the distance measure MaxPro, as we present the MmLHD(n, k) (r($(ML_2)^2$, MaxPro)) case. We see weak positive correlations for k = 5 and 10, though not all are significant. For k = 20, none of the correlations are significant. In fact, two are negative. Several plots contain outliers, which illustrates the risk of creating a single of these stochastic designs.

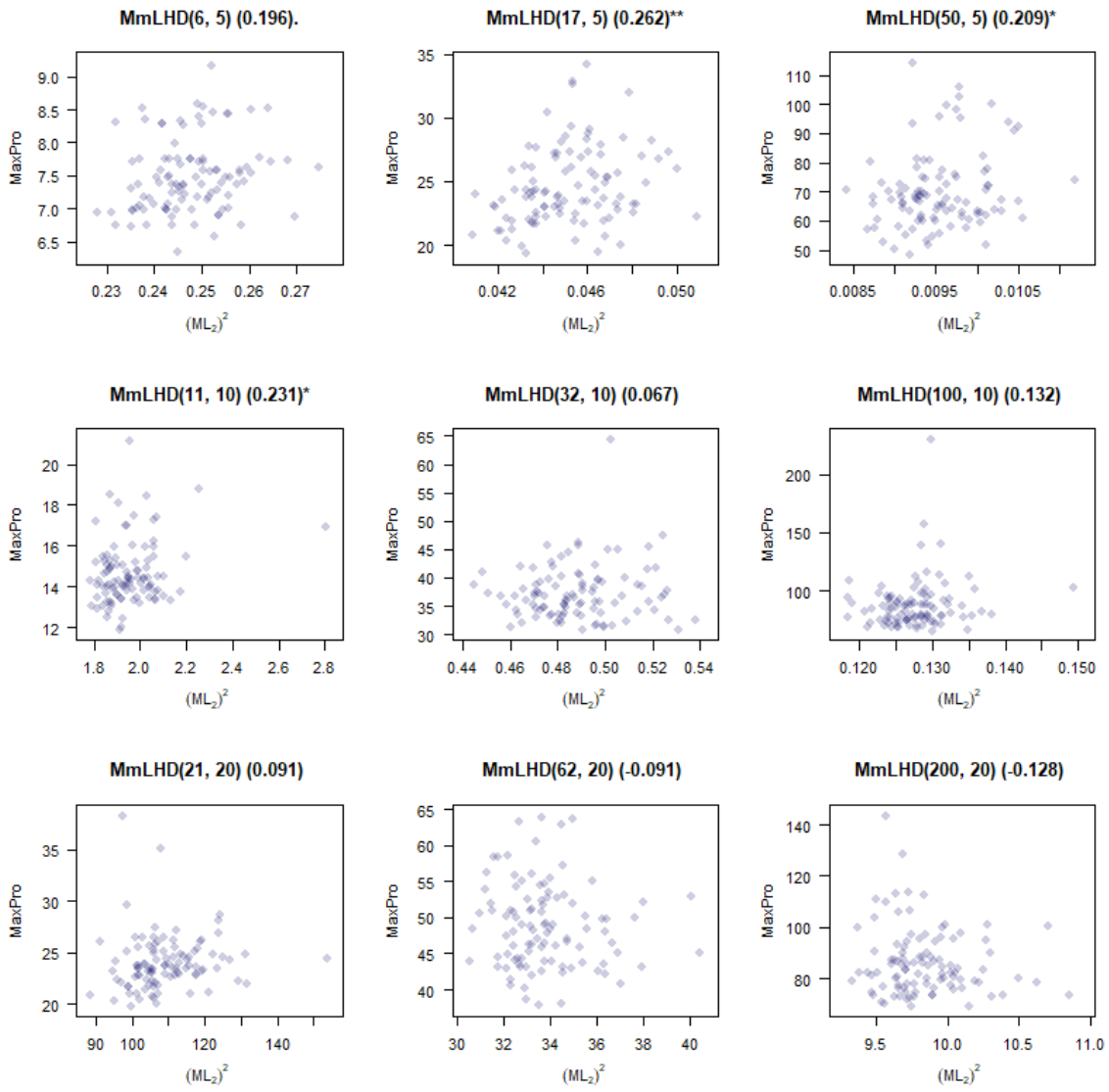


Figure 81. Nine MmLHD(n, k) ($r((ML_2)^2$, MaxPro)) scatter plots.

We omit the SphereP(n, k) ($r((ML_2)^2$, MaxPro)) case due to infinite MaxPro values resulting from replicated design points in subspace projections that occur in sphere-packing designs. Figure 82 presents nine LHD(n, k) ($r((ML_2)^2$, MaxPro)) scatter plots. The only significant correlations occur in the smaller saturated designs, which are weakly positive. We also see some outliers, as in the previous figure.

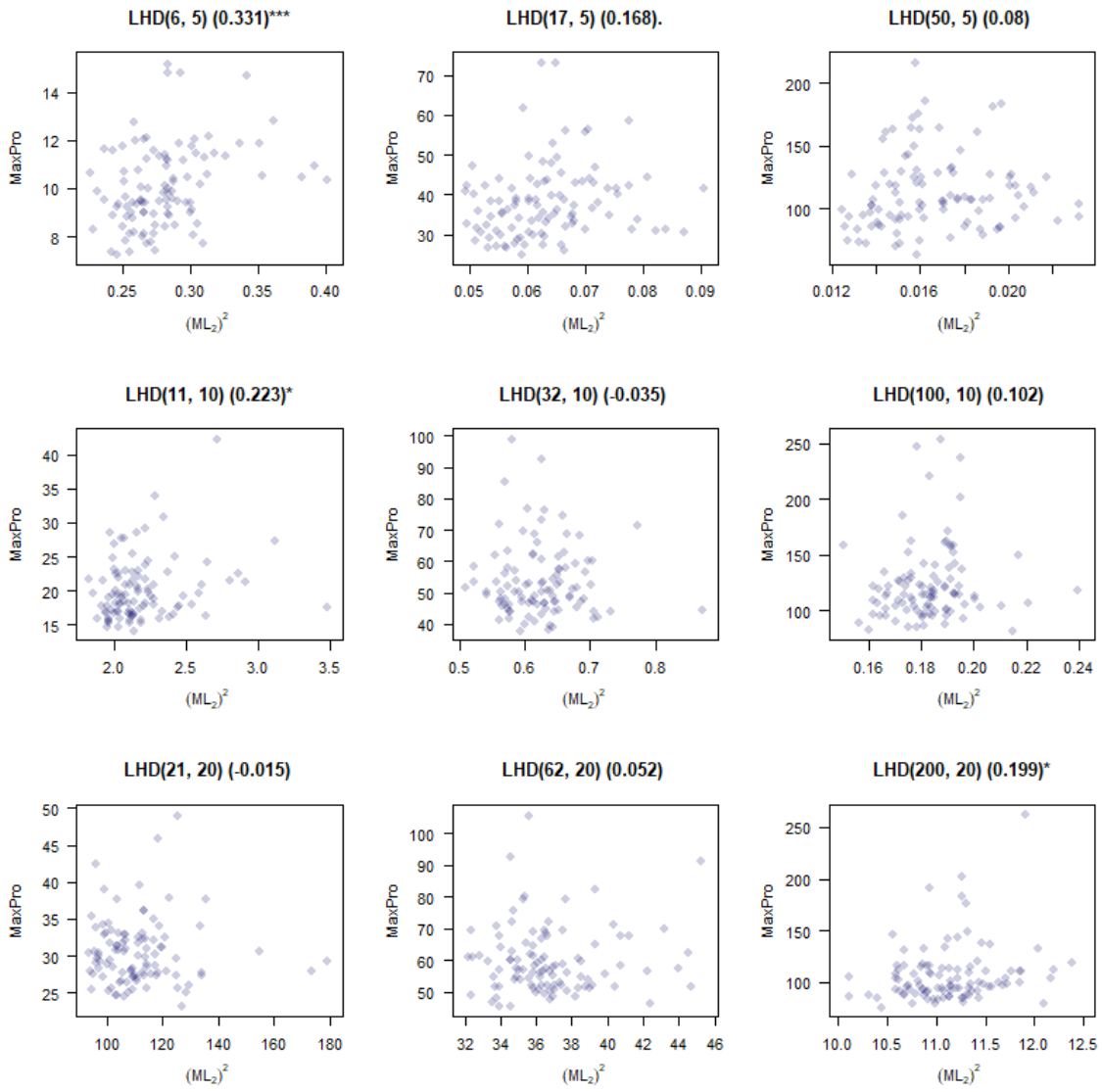


Figure 82. Nine LHD(n, k) ($r((ML_2)^2$, MaxPro)) scatter plots.

This section introduces correlation analysis between some of the most used correlation and space-filling measures for popular SFDs. Here, we have shown the results only for up to four measures in three design classes of nine sizes. In total, we ran experiments and collected data for ten measures of design quality (resulting in 45 pairwise correlations) for nine design sizes in each of six design classes. The observed correlations for all 45 combinations for the 54 design types and sizes are provided in Appendix G. For

illustrative purposes, Table 16 shows the empirical correlations between ρ_{map} and the other measures for six design classes of size 6×5.

Table 16. Correlation values, $r(x, \rho_{map})$, for ten SFMs and six SFDs size 6×5.

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
$k+1$	ρ_{map}	ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.808	0.921	0.994	0.997	0.38	0.476
		$(ML_2)^2$	0.285	-0.464	0.72	0.477	0.381	0.409
		$(CL_2)^2$	0.581	-0.732	0.725	0.997	0.517	0.426
		Cov	0.165	0.865	-0.214	0.918	0.214	0.116
		MaxPro	0.373	0.874	NA	0.001	0.388	0.228
		$\phi_{p=15}$	0.775	0.913	0.242	0.996	0.37	0.36
		$\phi_{p=50}$	0.632	0.908	0.552	0.992	0.36	0.343
		Mm	-0.567	-0.928	-0.614	-0.988	-0.392	-0.34
		$\overline{\gamma}$	-0.749	-0.955	-0.682	-0.996	-0.49	-0.54

F. CHAPTER SUMMARY

To address Jin et al.'s (2003, p. 554) call for a "thorough future investigation" of the "many optimality criteria available in the literature," this chapter reviewed, summarized, compared, and provided insights regarding quality measures of many popular design classes of multiple sizes. Several algorithms, formulas, and optimization tools were required to generate and assess these designs using R and JMP software packages.

A wide-ranging review has found nothing in the literature as comprehensive in quantifying and displaying the relationships (or lack thereof) for so many measures in multiple SFD classes. Across the design classes, for many measures, design preference can change with n and k. High variances and outliers in the plots reveal the risks associated with generating one "optimal" design with stochastic software. Researchers need to generate many SFDs to ensure they obtain a design with good properties for their application. Thus, practitioners may require departing from a single initial random-number-generator seed choice, such as the one provided by JMP. However, without JSL

(JMP's scripting language) this needs to be done manually. Further, Chapter V shows that, although measures may be of the same category and seem similar, such as ρ_{map} and $\overline{|\rho|}$, their linear association is less than what was expected and they can provide very different rankings on which designs are preferred. This is also true for the two discrepancy measures $(ML_2)^2$ and $(CL_2)^2$ explored. In addition, the strengths of linear associations can change dramatically, even disappear, with n and k.

An additional contribution of this work is the creation of tools that (1) automate creating many SFDs with cutting-edge software (e.g., JMP and R packages) and (2) facilitate assessing designs with multiple measures. Although we provide visualizations for only a few of the vast number of possibilities, researchers can use the tabled results in Appendix G for specific inquiries and the tools developed for more comprehensive evaluations of many design classes, sizes, and measures. Finally, this chapter provides evidence for including multiple measures together, e.g., a mixture of a discrepancy-, distance-, and correlation-based measures, when assessing the goodness of a design.

VI. A DISTRIBUTION MODEL OF THE MAXIMUM ABSOLUTE PAIRWISE CORRELATION FOR MMLHDS

> A defined probability model incorporating the variables n, k, and G becomes a powerful tool for planning better experiments.
>
> —Hernandez (2016), p. 1

Due to randomness in generating "optimum" space-filling designs (SFDs), many practitioners generate multiple designs from which they can select the best one according to some criterion. This chapter provides guidance on the minimum ρ_{map} an experimenter can expect to obtain when they generate multiple MmLHDs of various design dimensions. The organization of this chapter is as follows: introduction and approach, design-data generation, insights on minimum ρ_{map}, insights on G (number of designs generated), distribution modeling, and chapter summary of findings.

A. INTRODUCTION AND APPROACH

Not all arrived-at "optimal" SFDs possess the best properties when there is randomness in design generation. Given the randomness in the vast majority of algorithms that generate SFDs, how does an experimenter choose the design they will use? A common practice is to generate several designs and then select the one that optimizes some measure (e.g., Fang et al. [2000a], Hernandez et al. [2012b], Joseph and Hung [2008], MacCalman et al. [2017], Santner et al. [2018], Wang et al. [2020]). This simple practice improves the likelihood of obtaining a preferable SFD.

We have studied several design classes and ten measures of design quality. To focus this portion of the research, we limit the analysis to MmLHDs and ρ_{map}. MmLHDs are chosen since they are widely used. As Joseph (2016, p. 31) writes, MmLHDs seem "to be the most commonly used experimental design for computer experiments in practice." Similarly, Wang et al. (2021, p. 1) write that "[e]fficient Latin hypercube designs (LHDs), including maximin distance LHDs, maximum projection LHDs, and orthogonal LHDs, are widely used in computer experiments." Our measure of choice is ρ_{map}, as orthogonality

has been a foundational principle in experimental design from the beginning (Box 1978; Fisher 1926, 1971; Montgomery 2013) and ρ_{map} is a bounding measure of how close **X** is to being orthogonal. Sun and Tang (2017a, p. 683) write, "orthogonality is of importance when polynomial modeling is considered, and can also be viewed as a simple device to get closer to space-filling designs [SFDs]."

Let $\rho_{map}^{\min}$ be the minimum ρ_{map} obtained when an experimenter generates G independent designs. Since each design's ρ_{map} is the result of a stochastic process, so too is the $\rho_{map}^{\min}$ attained after creating G designs. To better understand how $\rho_{map}^{\min}$ depends on G for MmLHDs, we generated 10,000 MmLHDs in JMP for each of the following design sizes: k = 5, 6, …, 20 and $n = k+1$, $3k+2$, and $10k$. In sum, for each k, we have 10,000 saturated designs, a similar number with two more than three DPs per factor, and 10,000 designs with ten DPs per factor. This process yields a total of 480,000 MmLHDs. From each of these designs, we extracted ten measures of design quality.

Borrowing from Hernandez (2016), we study $\rho_{map}^{\min}$ as a function of the *design size environment* (DSE), which we denote as $\text{DSE}_{(n,\ k,\ G)}$. In this notation, n is the number of DPs, k is the number of factors, and G is the number of MmLHDs (of size $n \times k$) generated to obtain $\rho_{map}^{\min}$. Specifically, we investigate $\rho_{map}^{\min}$ as a function of $\text{DSE}_{(n,\ k,\ G)}$, with a primary interest in how $\rho_{map}^{\min}$ changes with G. We also study the asymptotic distribution of $\rho_{map}^{\min}$ as G increases.

B. GENERATING 480,000 MMLHDS OVER MULTIPLE DESIGN DIMENSIONS

Space-filling Maximin Euclidean distance Latin hypercube designs (MmLHDs) are generated using JMP's DOE (Design of Experiments) software (SAS 2021). JMP's scripting language (JSL) facilitates the creation of thousands of designs found under the following tabs: DOE > Special Purpose > Space-filling Design (Latin Hypercube with Optimal Spacing), which is JMP's MmLHD. The goal of the software is to generate the design with the largest possible maximin (Mm) Euclidean distance constrained to an $n \times k$

matrix using LHD structure; i.e., evenly spaced levels. Due to randomness in JMP's stochastic optimization algorithm, an optimal design is not guaranteed. To generate one 200×20 MmLHD, our largest design, on the author's MacBook Pro with a 3.1GHz processor, Intel Core i7, and a memory of 16GB requires (on average) approximately three minutes. The operating system used was Microsoft Windows 10 Pro.

Software scripts were systematically executed to generate 10,000 MmLHDs for each of $k = 5, 6, \ldots, 20$ and $n = k+1, 3k+2$, and $10k$ using JMP's default parameters. As the number of DPs per factor increases, so does the design's density (i.e., DPs per unit volume of χ). We refer to these three levels of design density as low, medium, and high—with low-density designs being fully saturated (i.e., no degrees of freedom remaining when a main effects linear regression with an intercept term is fit). Constructing multiple MmLHDs can be time-consuming. Generating 10,000 200×20 MmLHDs requires about three weeks on the author's MacBook Pro. Thus, the design construction was done in parallel across 23 computers. It took a total of fourteen days to create the 480,000 MmLHDs. Storing the resultant designs requires over 10.5 GB, spread across 1,024,003 files in 124 folders.

The author had to overcome challenges when conducting these large-scale computational experiments. Specifically,

- Commercial software can restrict the number of simultaneously open files; e.g., JMP (SAS 2021) limits the number of panels (or windows) open to 582, which impedes one's ability to make thousands of SFDs when each requires a new window for each design table.

- Edge computing platforms, e.g., computer labs connected to the cloud, can easily store a generated design locally; however, deploying statistical analysis software on the design when saved to the cloud can encounter complications with storing, opening, and analyzing—all employed in sequence. For example, the analysis software may halt prematurely when a locally generated design (i.e., stored locally) has not been saved fully to the cloud.

- Consolidating ten measures of design characteristics across 480,000 designs saved to the cloud has to be done carefully, using design labels and functions to ensure traceability.

Table 17 shows the last 12 rows of the combined CSV data file. We generated over 480,000 designs for this study and collected ten design measures for each MmLHD constructed, for a total of 4.8 million values.

Table 17. The combined data CSV file

ID	Class	n	k	rho_MAP	avg_absrho	ML_2	CL_2	Cov	MaxPro	Phip15	Phip50	Mm	avg_nn	Cat
9989	MmLH	9	8	0.5833	0.1732	0.8421	0.2510	0.0336	10.5480	0.9881	0.8739	1.1792	1.2112	Low
9990	MmLH	9	8	0.4833	0.1893	0.8028	0.2606	0.0280	10.4987	0.9950	0.8729	1.1792	1.2122	Low
9991	MmLH	9	8	0.5000	0.1625	0.8287	0.2471	0.0149	12.3324	0.9635	0.8345	1.2374	1.2561	Low
9992	MmLH	9	8	0.5167	0.1988	0.8732	0.2603	0.0215	13.1001	0.9816	0.8619	1.1792	1.2174	Low
9993	MmLH	9	8	0.3833	0.1875	0.9213	0.2508	0.0364	11.7567	0.9742	0.8702	1.1524	1.2317	Low
9994	MmLH	9	8	0.3667	0.1750	0.8661	0.2659	0.0196	13.1777	0.9794	0.8557	1.1990	1.2266	Low
9995	MmLH	9	8	0.4667	0.1649	0.8202	0.2493	0.0282	11.3445	0.9787	0.8575	1.1990	1.2334	Low
9996	MmLH	9	8	0.4167	0.1440	0.9117	0.2494	0.0110	11.7070	0.9657	0.8432	1.2183	1.2374	Low
9997	MmLH	9	8	0.4667	0.1506	0.8070	0.2555	0.0244	14.5893	0.9734	0.8520	1.1990	1.2392	Low
9998	MmLH	9	8	0.3167	0.1351	0.8725	0.2484	0.0541	10.9230	0.9891	0.8790	1.1592	1.2151	Low
9999	MmLH	9	8	0.3333	0.1315	0.8007	0.2472	0.0526	11.4970	0.9841	0.8753	1.1592	1.2195	Low
10000	MmLH	9	8	0.6333	0.1690	0.8997	0.2520	0.0354	11.0645	1.0005	0.8971	1.1250	1.1829	Low

C. THE DISTRIBUTION OF ρ_{MAP} IN MMLHDS

To see the variability in ρ_{map} from JMP's MmLHD generation algorithm, consider when k is 5, 6, and 7 and $n = k + 1$. Though the design measure we explore is ρ_{map}, any of the other nine measures could be chosen and will exhibit variability. Figure 83 shows box plots of 10,000 ρ_{map} values of MmLHDs for three design dimensions (6×5, 7×6, and 8×7). The ρ_{map} values range from under 0.1 to over 0.9. We also see that for these saturated designs, the ρ_{map} values tend to increase with k.

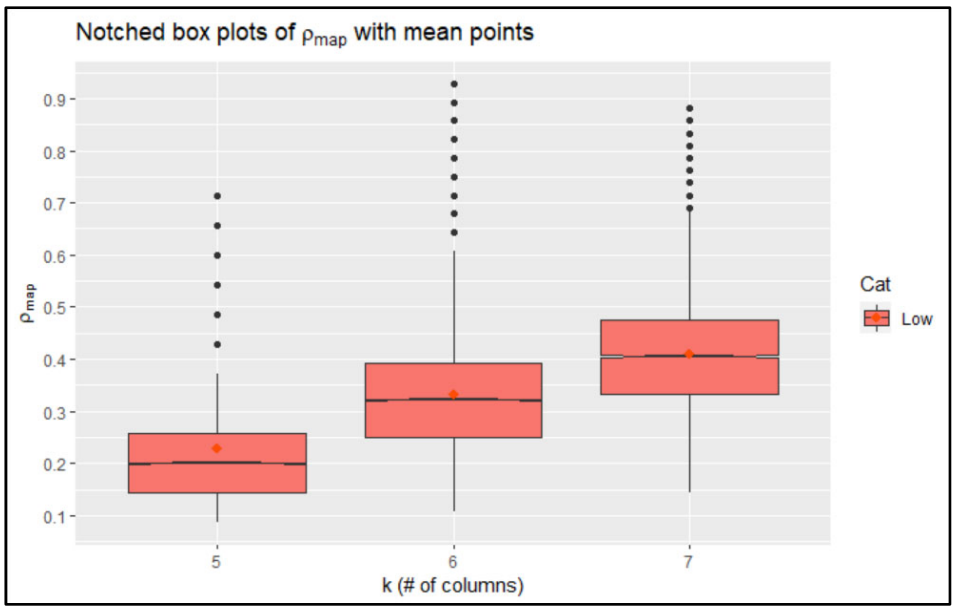


Figure 83. 10,000 ρ_{map} values for each of 6×5, 7×6, and 8×7 MmLHDs. Values range from less than 0.1 to greater 0.9.

Figure 84 displays the variability of ρ_{map} for the low-, medium-, and high-density designs as the number of factors (k) goes from five to 20. This figure contains 48 box plots for the 16 levels of k and the three design densities. Each box plot summarizes the results of 10,000 MmLHDs. The red diamonds show the mean ρ_{map} value for each of the 48 design sizes. For all k, higher-density designs tend to have lower ρ_{map} values. ρ_{map} values tend to increase with k, especially for low-density designs when k is towards the bottom of the range investigated. For all k, the variability in ρ_{map} values decreases with design density. Variability decreases with density because more DPs (i.e., large n) reduces the likelihood of any two columns having a high correlation (Hernandez et al. 2012b). As a reminder, design correlation is not considered when JMP constructs MmLHDs.

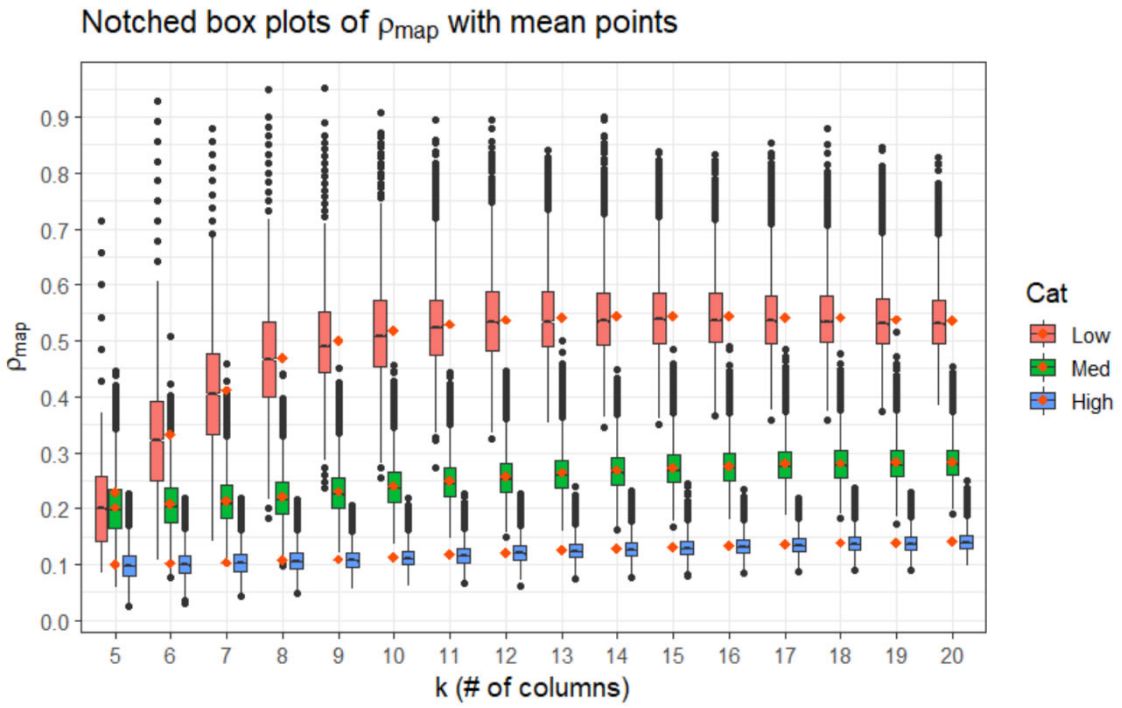


Figure 84. Box plots of 10,000 ρ_{map} values for each of 48 MmLHD sizes (e.g., 6×5, 17×5, 50×5,…, 21×20, 62×20, 200×20).

D. THE MINIMUM ρ_{MAP} OBTAINED BY GENERATING G MMLHDS

This section uses the 480,000 MmLHDs generated and their property data to investigate how $\rho_{map}^{\min}$ depends on DSE$_{(n, k, G)}$, with an emphasis on providing guidance on how G effects $\rho_{map}^{\min}$. The best performing design from the G generated ones is the one that minimizes ρ_{map}, though ties are possible. To assess the effect of G on $\rho_{map}^{\min}$, we use the 10,000 MmLHDs built for each n and k to get samples of $\rho_{map}^{\min}$ for multiple levels of G. For example, when G = 10, we can empirically obtain 1,000 independent $\rho_{map}^{\min}$ values. More generally, for any G that evenly divides 10,000, we can get m = 10,000/G independent $\rho_{map}^{\min}$ values. We did this for G = 10, 25, 50, and 100—i.e., for m = 1,000, 400, 200, and 100. With more time or processing power, we could generate many more designs and extend this initial investigation to more k, n, and G. Moreover, this approach can also be applied to the other nine measures and other design classes.

Let $\overline{\rho_{map}^{\min}}$ be the empirical mean of the m $\rho_{map}^{\min}$ values obtained in DSE$_{(n,\ k,\ G)}$. This is an estimate of the $\rho_{map}^{\min}$ an experimenter can expect to achieve by selecting the design with the lowest ρ_{map} from G independently generated $n \times k$ MmLHDs. $\rho_{map}^{\min}$ is the ρ_{map} of the design the researcher would use for computer experimentation if selecting based on the ρ_{map} criterion from G MmLHDs in DSE$_{(n,\ k,\ G=100)}$. Table 18 presents the 48 $\overline{\rho_{map}^{\min}}$ values for $G = 100$ and the estimated standard deviations based on the $m = 100$ $\rho_{map}^{\min}$ values for each design size. With $m = 100$, the estimated standard error of the mean is the estimated standard deviation divided by ten.

There are clear trends in this table. For each design density, as k increases, $\overline{\rho_{map}^{\min}}$ values tend to increase. For each k, $\overline{\rho_{map}^{\min}}$ decreases as the design density (i.e, n) increases. We also see that the estimated standard deviation decreases as k increases for each design density. Furthermore, for each k, the estimated standard deviation decreases as design density increases. The lowest $\overline{\rho_{map}^{\min}}$ value in Table 18 corresponds to DSE$_{(n=50,\ k=5,\ G=100)}$, which is a high-density design with few factors. MmLHDs of size 50×5 is the only DSE$_{(n,\ k,\ G=100)}$ explored that yields a nearly orthogonal design on average. The largest $\overline{\rho_{map}^{\min}}$ values are for DSE$_{(n=21,\ k=20,\ G=100)}$ and DSE$_{(n=20,\ k=19,\ G=100)}$, which are sparse, fully-saturated, low-density designs with many factors. Appendix H contains similar tables for G = 10, 25, and 50.

Table 18. Calculated $\overline{\rho_{map}^{\min}}$ value and standard deviation (SD) for the specified DSE dimensions.

$\overline{\rho_{map}^{min}}$ for a given $\mathrm{DSE}_{(n,k,G=100)}$

n	$k+1$		$3k+2$		$10k$	
k	mean	(SD)	mean	(SD)	mean	(SD)
5	0.1297	(0.0241)	0.0935	(0.0144)	0.0455	(0.0067)
6	0.1510	(0.0202)	0.1113	(0.0112)	0.0526	(0.0066)
7	0.2066	(0.0236)	0.1225	(0.0115)	0.0589	(0.0058)
8	0.2603	(0.0265)	0.1338	(0.0119)	0.0650	(0.0056)
9	0.3098	(0.0279)	0.1486	(0.0103)	0.0701	(0.0049)
10	0.3431	(0.0270)	0.1599	(0.0095)	0.0749	(0.0048)
11	0.3688	(0.0212)	0.1701	(0.0096)	0.0793	(0.0055)
12	0.3823	(0.0214)	0.1797	(0.0095)	0.0839	(0.0049)
13	0.3946	(0.0155)	0.1851	(0.0093)	0.0872	(0.0051)
14	0.4007	(0.0172)	0.1928	(0.0099)	0.0909	(0.0050)
15	0.4070	(0.0157)	0.1976	(0.0095)	0.0949	(0.0047)
16	0.4071	(0.0174)	0.2032	(0.0083)	0.0975	(0.0044)
17	0.4115	(0.0172)	0.2095	(0.0086)	0.1007	(0.0045)
18	0.4145	(0.0157)	0.2116	(0.0089)	0.1032	(0.0044)
19	0.4164	(0.0166)	0.2124	(0.0096)	0.1045	(0.0045)
20	0.4164	(0.0144)	0.2154	(0.0079)	0.1067	(0.0037)

This section's primary goal is to provide guidance to the experimenter on the expected value of $\rho_{map}^{\min}$ for JMP MmLHDs in $\mathrm{DSE}_{(n, k, G)}$, with an emphasis on G. To begin, let's look at the minimum ρ_{map} that was obtained from the 10,000 runs for each of the 16 saturated designs. This provides information on what may reasonably be attained, at least for the G values explored. Figure 85 presents the $\rho_{map}^{\min}$ value obtained for each $(k+1)\times k$ saturated-design dimension. There is a clear knee in the curve. We see that ρ_{map} increases rapidly with k until about $k = 13$, at which point the increasing trend substantially reduces.

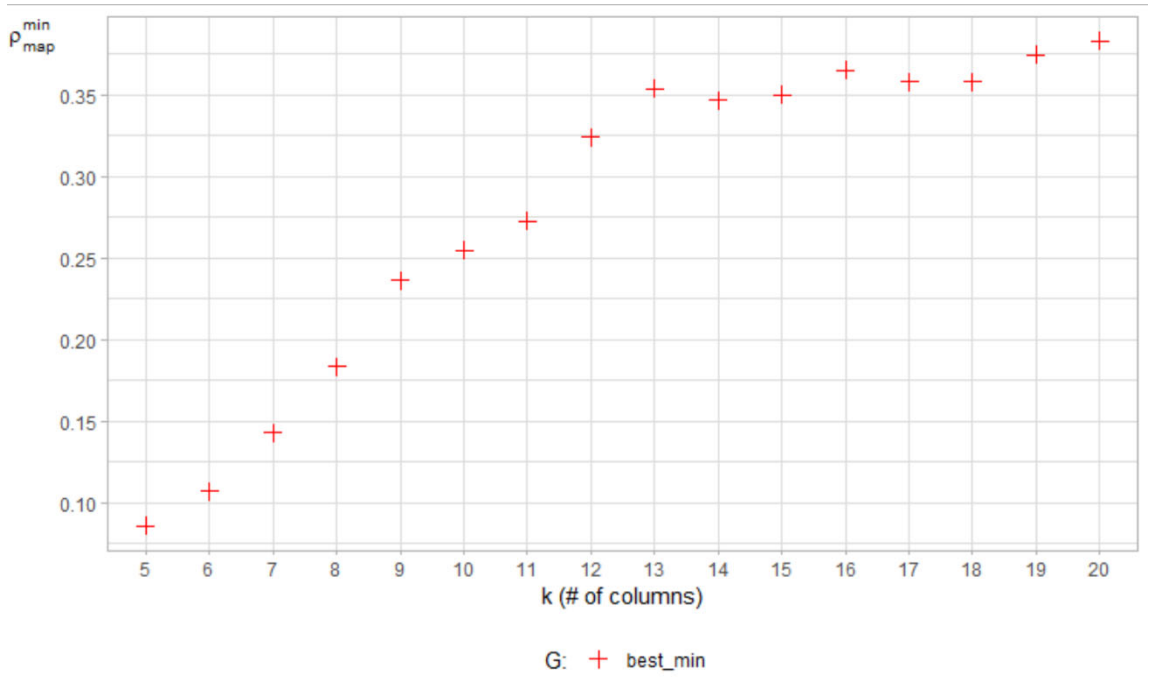


Figure 85. The "+" symbols show the best-saturated design ρ_{map} values in 10,000 MmLHDs for $k = 5, 6, \ldots, 20$.

Figure 86 builds on Figure 85 by adding a red square ("■") for the $\overline{\rho_{map}^{\min}}$ value when $G = 10$ for each k. This shows the $\rho_{map}^{\min}$ value the experimentalist can expect to attain when selecting the best of ten $(k+1)\times k$ MmLHDs, as well as a comparison to the best found after 10,000 generations. Here too, $\overline{\rho_{map}^{\min}}$ steadily increases with k, and there is a knee in the curve at about $k = 12$ or 13.

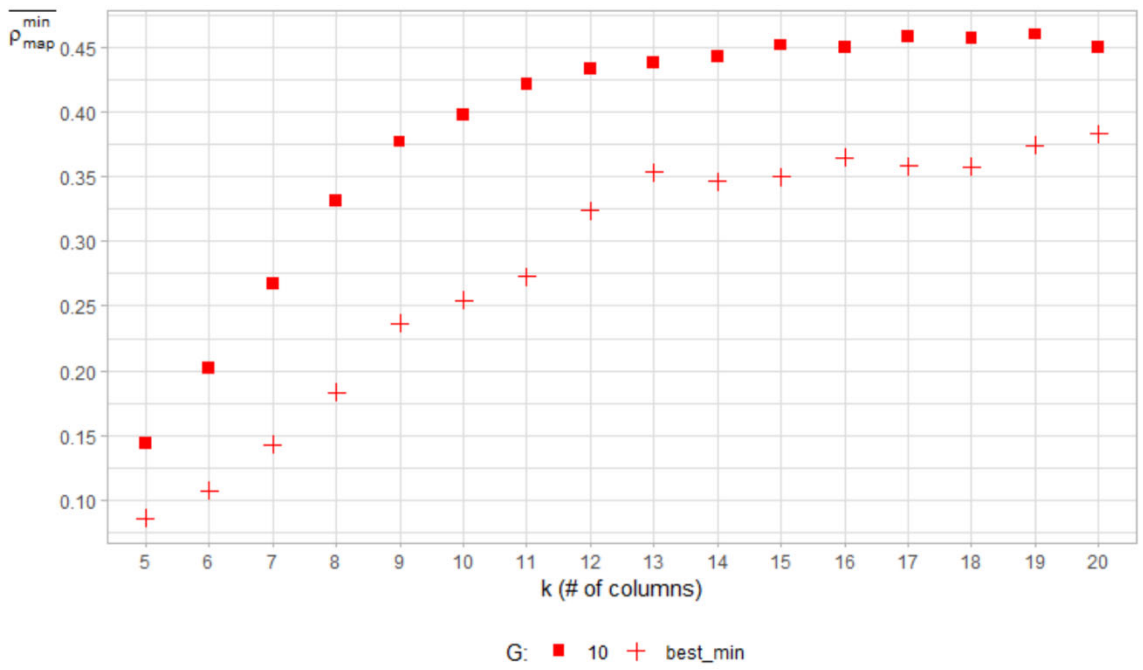


Figure 86. A comparative plot of the $\overline{\rho_{map}^{min}}$ values ("■") in DSE$_{(n=k+1,\ k,\ G=10)}$ and the best observed ρ_{map} values ("+") from the 10,000 saturated $(k+1)\times k$ MmLHDs over the k levels.

Figure 87 adds a red circle, triangle, and diamond ("●, ▲, and ♦") for the $\overline{\rho_{map}^{min}}$ values corresponding to G=25, 50, and 100 for each of our 16 saturated-design sizes. This gives an estimate of how much additional benefit, in terms of ρ_{map}, the experimenter can expect as G increases. For example, the $\overline{\rho_{map}^{min}}$ value for DSE$_{(n=8,\ k=7,\ G=100)}$ is approximately 0.207. This is 0.061 lower than what to expect for DSE$_{(n=8,\ k=7,\ G=10)}$, though still far higher than the lowest (0.143) of our 10,000 8×7 MmLHDs. There appears to be a stabilization or "knee" in the $\overline{\rho_{map}^{min}}$ values for each of the G curves (for G =10, 25, 50, and 100) near k = 12 or 13. Saturated MmLHDs present a challenge to software in obtaining a low ρ_{map} value, especially for large k.

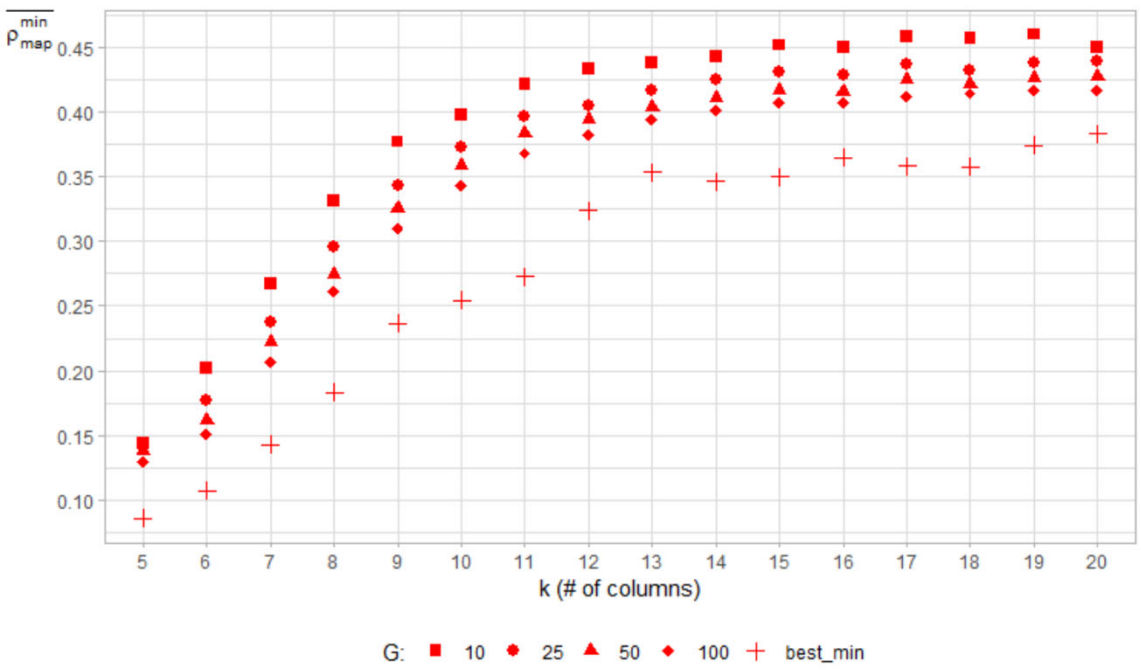


Figure 87. A comparative plot of the $\overline{\rho}^{min}_{map}$ values of DSE$_{(n=k+1,\ k,\ G)}$ when G = 10, 25, 50, and 100) and the best saturated k+1×k MmLHD (+) values achieved.

Figure 88 adds the medium- and high-density designs to Figure 87. The medium-density designs; i.e., $n = 3k+2$, are shown with blue symbols—a blue square, circle, triangle, and diamond ("■, ●, ▲, and ♦") for the $\overline{\rho}^{min}_{map}$ values of G=10, 25, 50, and 100, respectively. The high-density designs; i.e., $n = 10k$, are shown with dark green symbols—a dark green square, circle, triangle, and diamond ("■, ●, ▲, and ♦") for the $\overline{\rho}^{min}_{map}$ values of G=10, 25, 50, and 100, respectively. The higher-density designs have significantly lower $\overline{\rho}^{min}_{map}$ values than the medium-density designs, which perform much better than the low-density designs. As k increases, the $\overline{\rho}^{min}_{map}$ values tend to increase within all three density categories, though by a lesser amount as design density increases. For low design densities, there is a greater reduction in $\overline{\rho}^{min}_{map}$ as G increases than for medium- and high-design densities.

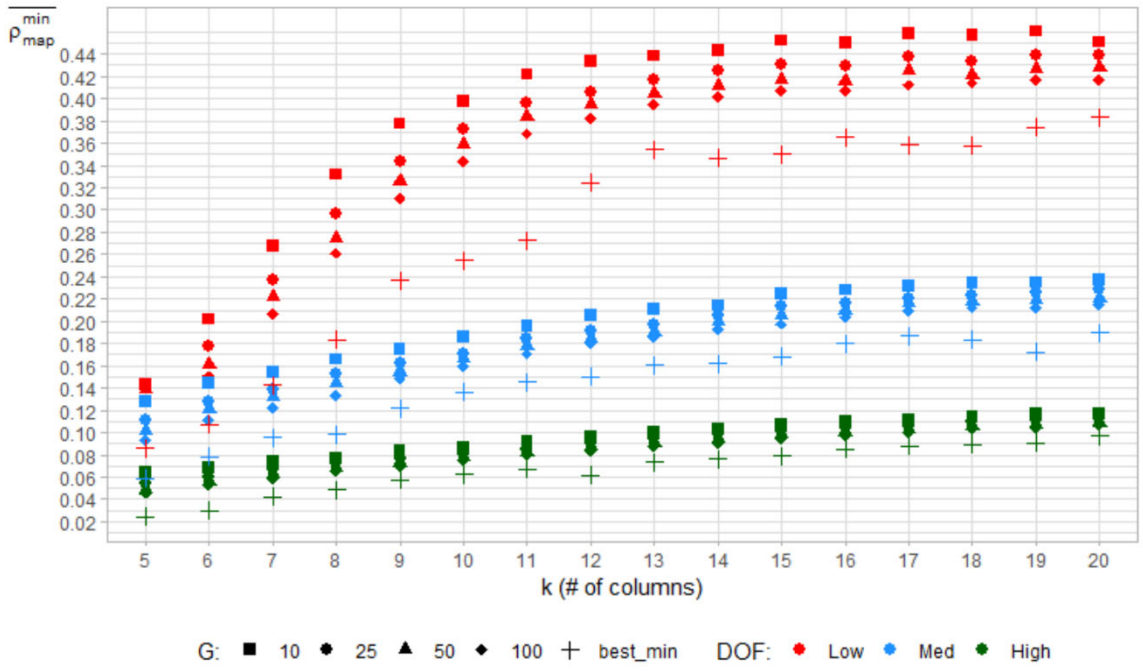


Figure 88. A comparative plot of the $\overline{\rho_{map}^{min}}$ values and best ρ_{map} values ("+") for each saturated-, medium-, and high-density case and all values of G.

This subsection provides the minimum ρ_{map} an experimenter can expect to attain using MmLHDs for various n, k, and G combinations. The following section explores the asymptotic probability distribution of ρ_{map}^{min} in 200×20 LHDs and 200×20 MmLHDs as G increases. Note that these are the largest designs explored in this chapter in terms of n and k.

E. DISTRIBUTION MODELS FOR ρ_{map}^{min} VALUES OF LHDS AND MMLHDS

The previous section explores how $\overline{\rho_{map}^{min}}$ depends on $DSE_{(n, k, G)}$. Can we say more about ρ_{map}^{min} beyond its expectation? Specifically, can we determine the limiting probability distribution of ρ_{map}^{min} as G increases? This section takes a first step toward answering that question by looking at the probability distributions of ρ_{map}^{min} in 200×20 LHDs and 200×20 MmLHDs as G increases. LHDs and MmLHDs are among the most widely used SFDs. They are most valuable to computer experimenters when n and k are large.

This research builds on Hernandez (2016), who found that for large G, $\rho_{map}^{\min}$ in LHDs is reasonably well fit by a Gumbel distribution (Gumbel 1935). We investigate the distributional fit within the broader generalized extreme value (GEV) distribution (Jenkinson 1955, Park and Sohn 2006) and attain a better fit with a Weibull distribution. The above process is then applied to 200×20 MmLHDs. The best fit for the distribution of $\rho_{map}^{\min}$ in 200×20 MmLHDs as G increases also appears to be a Weibull distribution.

To assess the limiting distribution of $\rho_{map}^{\min}$ in LHDs, we generate 10,000 200×20 LHDs and compute ρ_{map} for each. We use these results to find $\rho_{map}^{\min}$ for G = 10, 25, 50, and 100. The G values result in sample sizes of m = 1,000, 400, 200, and 100, respectively. Table 19 shows the first ten (of m = 200) rows of the $\text{DSE}_{(n=200,\ k=20,\ G=50)}$ data.

Table 19. The first ten of the m = 200 $\rho_{map}^{\min}$ values with G = 50 for 200×20 LHDs.

```
> RLHDs_to_Data(RLHDDf, G=50)
     id   n  k G_value min_rho_map
1     1 200 20      50   0.1690497
2     2 200 20      50   0.1570614
3     3 200 20      50   0.1531988
4     4 200 20      50   0.1603630
5     5 200 20      50   0.1607305
6     6 200 20      50   0.1611535
7     7 200 20      50   0.1580830
8     8 200 20      50   0.1480447
9     9 200 20      50   0.1355164
10   10 200 20      50   0.1642961
```

A single probability model known as the generalized extreme value (GEV) distribution can be used to broaden Hernandez's (2016) approach. The GEV distribution combines the three possible extreme value distributions for the minimum of G independent and identically distributed (iid) observations from a continuous distribution; see Jenkinson (1955) and Park and Sohn (2006) for details. Citing Park and Sohn (2006), "extreme value statistics (EVS) focus on modeling those extreme points near the tails without prior knowledge of the parent distributions. The extreme values can asymptotically follow one

of only three possible extreme value distributions: Gumbel, Weibull, or Frechet" (p. 376). For a comprehensive treatment of extreme value distributions, see David and Nagaraja (2004). The R software package *extRemes* (Gilleland and Katz 2016) implements the GEV.

Figure 89 shows three examples of the three-parameter GEV distribution (Jenkinson 1955, Park and Sohn 2006) in three separate charts, each with the same location and scale parameters (location = 0 and scale = 1). The shape parameter's value aids in determining whether the GEV distribution is a Gumbel, Fréchet, or Weibull. Specifically, a shape value of zero, positively valued, or negatively valued yields the Gumbel, Frechet, or Weibull distribution, respectively.

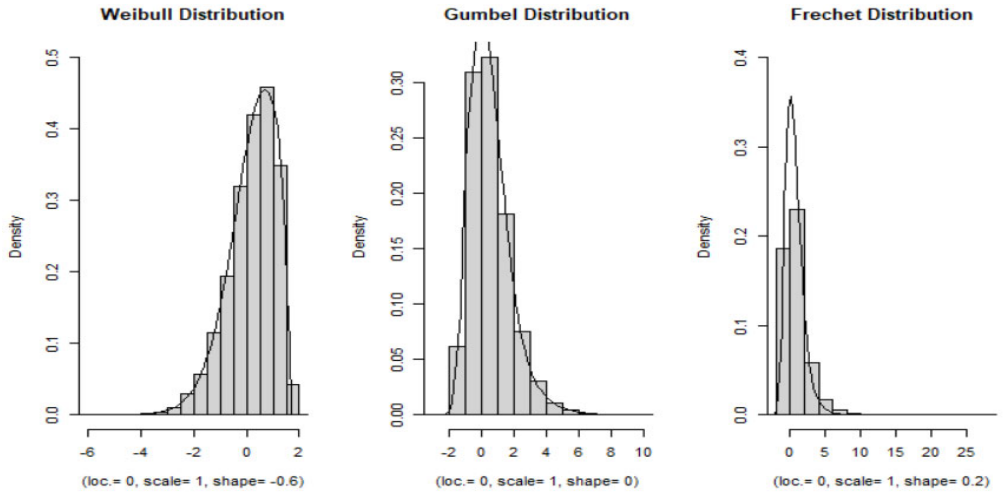


Figure 89. Three examples of the GEV (Jenkinson 1955, Park and Sohn 2006): Weibull (left), Gumbel (middle), and Fréchet (right), which all have the same location and scale parameters (location = 0 and scale = 1).

Figure 90 shows four empirical frequency histograms of $\rho_{map}^{\min}$ for LHDs in DSE(n=200, k=20, G=10, 25, 50, and 100). For example, the bottom left plot is for 200 observations of G = 50 data. As G increases, we expect the histograms to converge to one of the three limiting distributions in the GEV family. Beginning when G=25 (top right of Figure 90), the left tail becomes more prominent as G increases, suggestive of a Weibull distribution.

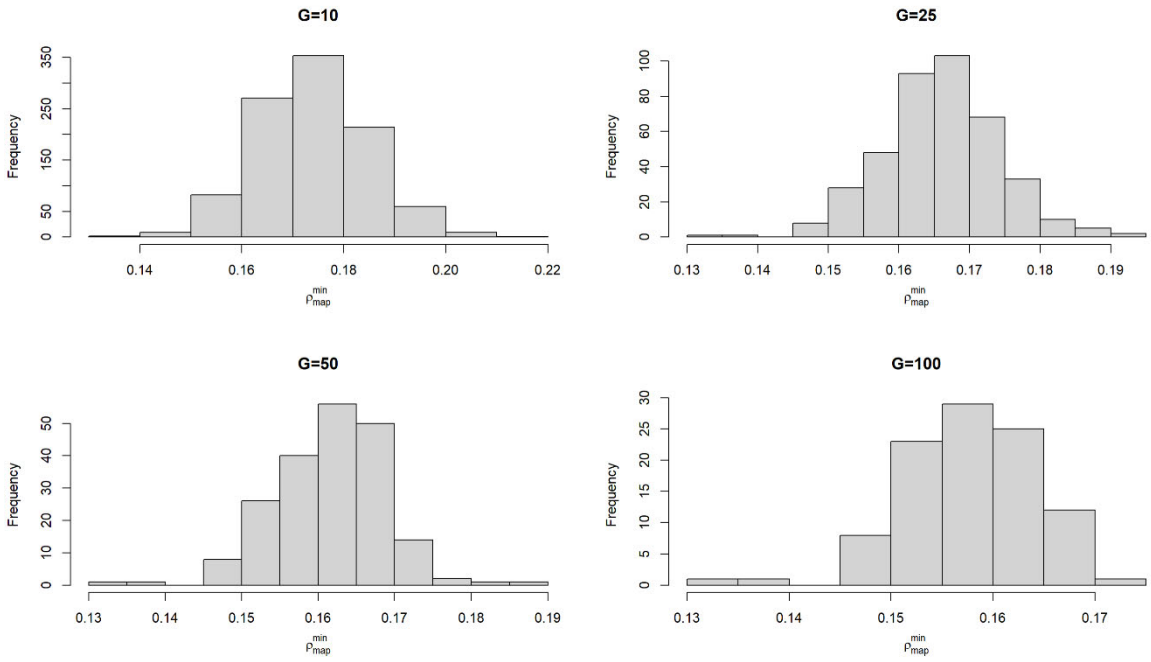


Figure 90. Four empirical frequency histograms for the various G and the LHD DSE$_{(n=200,\ k=20,\ G=10,\ 25,\ 50,\ \text{and}\ 100)}$ data.

A benefit of fitting the single GEV function to data, as implemented in the R software package *extRemes* (Gilleland and Katz 2016), is that it combines the three possible extreme value distributions (Jenkinson 1955, Park and Sohn 2006) to arrive at parameters for any of the distributions (Weibull, Gumbel, or Frechet) without selecting one *a priori*. Using numerical maximum likelihood estimation (MLE), the fitted distribution for LHDs in DSE$_{(n=200,\ k=20,\ G=50)}$ has location, scale, and shape parameters of 0.159024, 0.007740, and −0.272185, respectively. Notably, the shape parameter's value (−0.272185) suggests that, in this case, the GEV converges to a Weibull distribution (i.e., has a negative shape parameter). The blue line in Figure 91 is the fitted distribution using the GEV function in the *extRemes* (Gilleland and Katz 2016) R software package, which reasonably matches both the empirical density (histogram) and the cumulative distribution function (CDF).

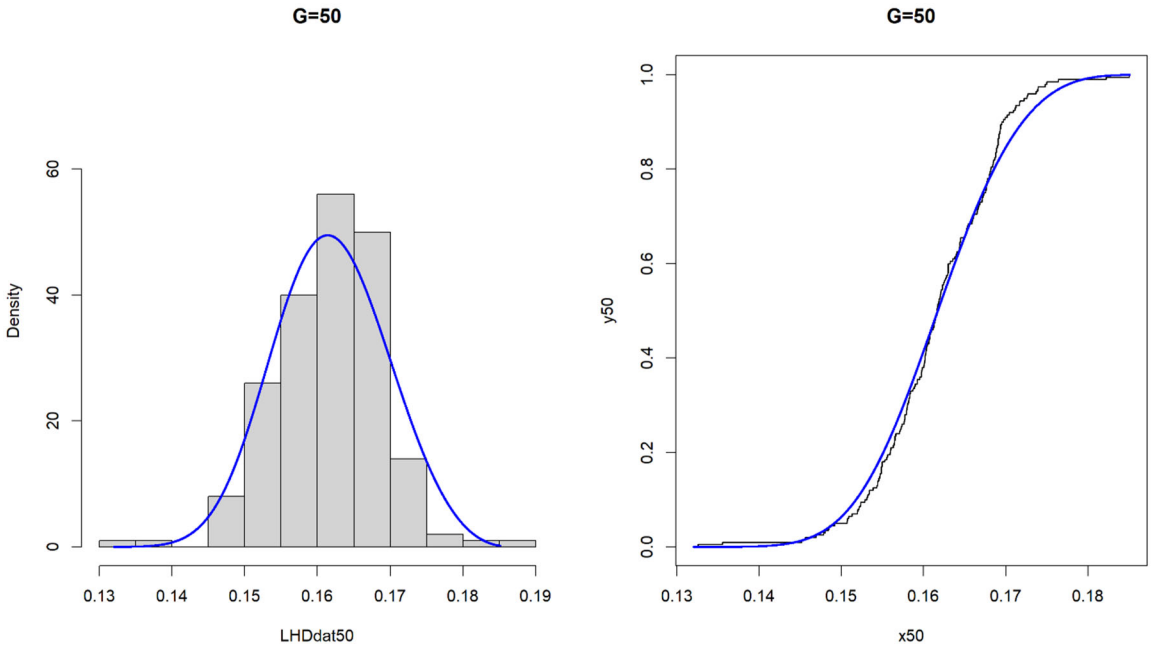


Figure 91. Side-by-side plots: Density histogram (left) and the empirical CDF (right) of 200 observations of $\rho_{map}^{\min}$ for LHDs in DSE$_{(n=200,\ k=20,\ G=50)}$. The blue line represents the GEV fit using MLE, which suggests a Weibull distribution.

To quantify the goodness of fit, we perform the one-sample Kolmogorov-Smirnov (K-S) test (Massey 1951). The K-S test statistic is $D = \sup_x \left|F_m(x) - F(x)\right|$, where $F_m(x)$ is the empirical CDF from the m $\rho_{map}^{\min}$ values, and $F(x)$ is the CDF of the null-hypothesis distribution. The K-S test statistic is the largest vertical gap between the data-generated empirical CDF and the null-hypothesis distribution's CDF, with lower D values indicating a better fit. The null hypothesis in our exploration is that $F(x)$ is a GEV with location, scale, and shape parameters of 0.159024, 0.007740, and −0.272185, respectively—i.e., the best fitting GEV, which is a Weibull distribution. The results of the K-S test are D = 0.06865, p-value = 0.3024, and we fail to reject (FTR) the null hypothesis at classic levels of significance.

In this exploration, we use an approximate test since the null hypothesis CDF was fit from data rather than being specified. In such cases, "[t]he distribution tested will be rejected far less than it should be" (Kotz et al. 1983, p. 400). That is, p-values obtained using the fitted distribution will tend to be higher than they would be when testing a given distribution. To assess the effect of using a fitted distribution rather than a specified one as the null hypothesis, we apply the K-S fitting approach to obtain 100,000 K-S test statistics when simulated samples are drawn from a known GEV distribution (of size $m = 200$ with the parameters from above), but the fit and test are to the MLE parameter estimates of the samples. When testing a fully-specified continuous distribution, when the null hypothesis is true, the p-value is distributed uniform [0, 1] (Casella and Berger 2002, p. 398).

Figure 92 displays a histogram of the simulated K-S test-statistic values when the approach is applied to a fitted GEV distribution rather than to a specified one. The black dashed and solid blue lines represent the p-values 0.05 and 0.3024—the first's being a classic significance level and the latter's being the p-value obtained above when testing the goodness of fit of the best-fitting GEV. As expected, our approach's distribution of empirical p-values generally has increasing probability in intervals moving up the domain than one would expect from a uniform. Also, fewer than five percent of the simulated K-S stats are below 0.05, showing that the test will fail to reject a bit more than its stated significance level. More precisely, of the 100,000 simulated p-values, 4,808 are less than 0.05. The 95% CI on the true proportion less than 0.05 for our approach is [0.0468, 0.0494]. This, and the nearly uniform shape of the histogram in Figure 92, show that the method works reasonably well at estimating the significance of the test statistics—though it will slightly understate the significance level.

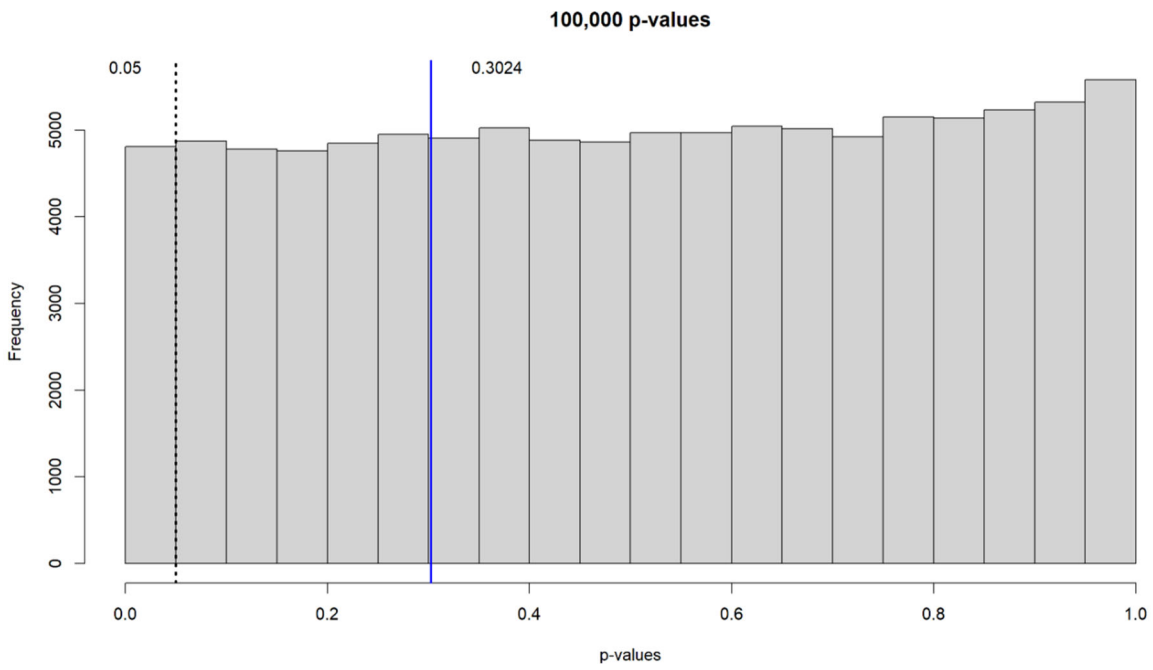


Figure 92. Histogram of 100,000 K-S statistic p-values using fitted parameters (location, scale, and shape) of the GEV function as the null hypothesis and the p-value we observed.

To see if the shape parameter is significantly less than zero, we bootstrap the shape parameter of the GEV distribution function using the R software package *boot* (Canty and Ripley 2021 and Davison and Hinkley 1997). That is, we resample 100,000 times, with replacement, the data series (i.e., the $m = 200$ $\rho_{map}^{\min}$ values for LHDs in DSE$_{(n=200,\, k=20,\, G=50)}$) to obtain 100,000 bootstrap estimates of the shape parameter. The resultant bootstrap 95% percentile bootstrap confidence interval is [−0.4703, −0.1848], which does not contain zero; i.e., the shape-parameter value that suggests that the limiting distribution tends to the Gumbel. The negative interval is strong evidence that the shape-parameter value is negative (e.g., −0.2606), and the data are converging to the Weibull distribution.

The approach above on LHDs is now applied to MmLHDs. Figure 93 shows the fitted theoretical GEV (blue) distributions for MmLHDs in DSE$_{(n=200,\, k=20,\, G)}$ when $G = 10$,

25, 50, and 100 overlaid on histograms of the empirical data. The fitted GEV distributions (blue) seem to be reasonably good fits.

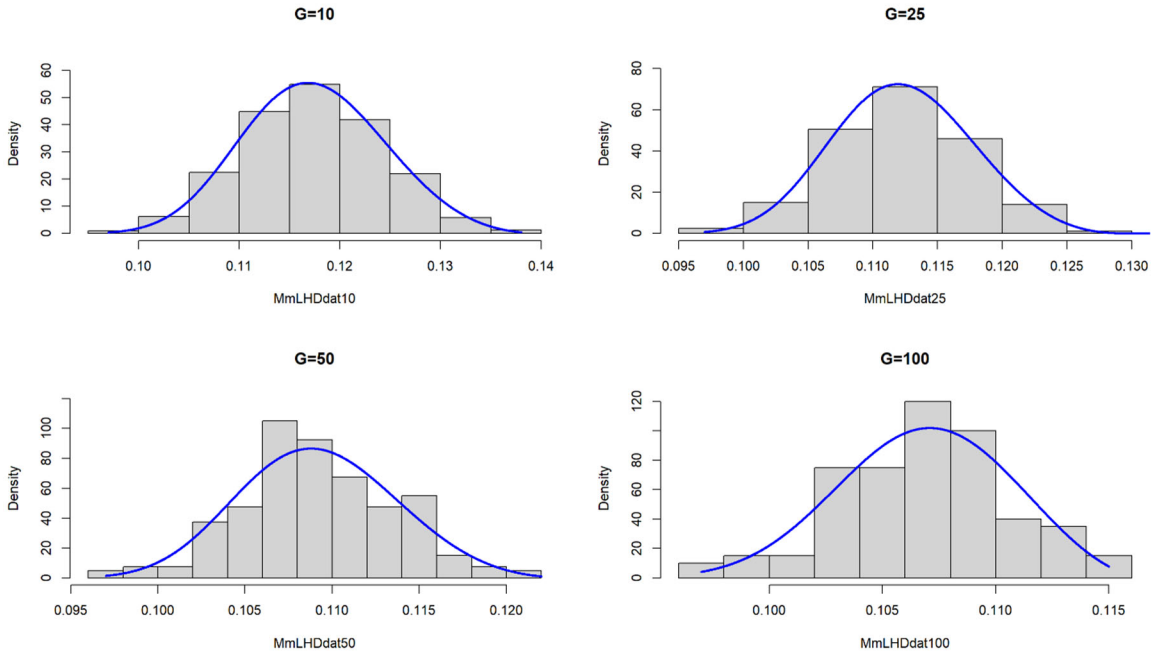


Figure 93. The MLE fitted GEV distributions (in blue) for MmLHDs in $DSE_{(n=200,\ k=20,\ G)}$ when G = 10, 25, 50, and 100.

Figure 94 shows the empirical CDF plots (in black) of $\rho_{map}^{\min}$ values for MmLHDs in $DSE_{(n=200,\ k=20,\ G)}$ for G = 10, 25, 50, and 100. The figure also contains the best fitting GEV CDF (in blue) for each G. The empirical CDF and the best-fitting CDF closely match, showing an excellent GEV fit.

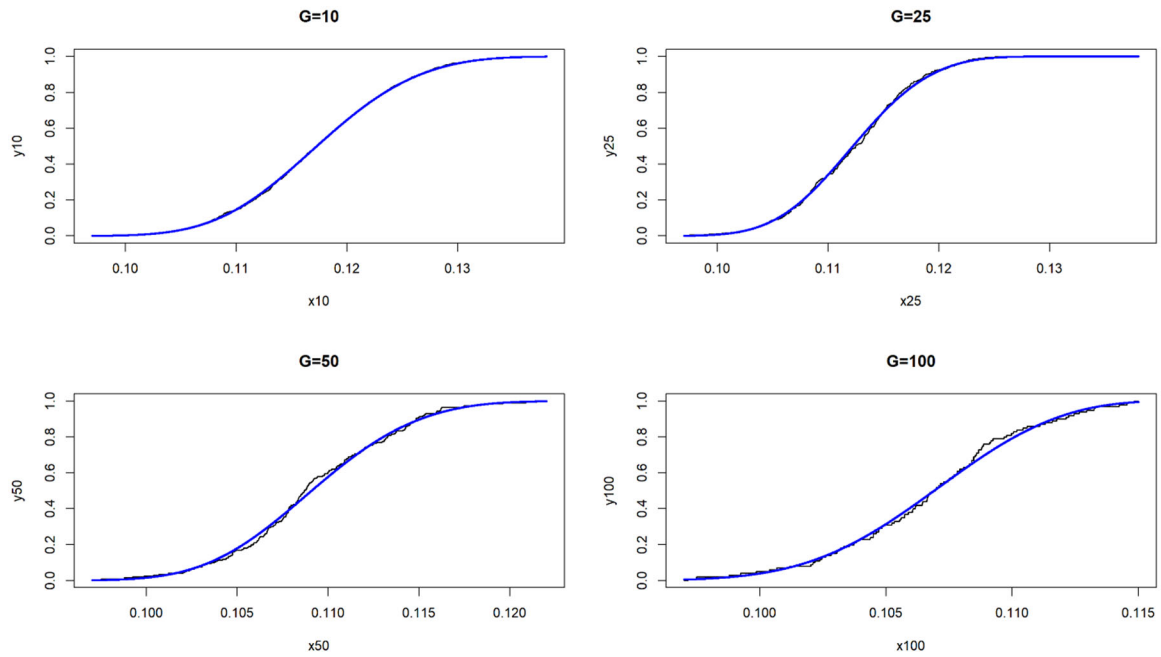


Figure 94. Empirical CDFs of 1,000 (top left), 400 (top right), 200 (bottom left), and 100 (bottom right) observations of $\rho_{map}^{\min}$ values of MmLHDs in DSE$_{(n=200,\ k=20,\ G)}$. The GEV fits (blue) show a good match.

Table 20 summarizes the results of the GEV fits for G = 10, 25, 50, and 100 for LHDs and MmLHDs in DSE$_{(n=200,\ k=20,\ G)}$. In all eight cases, the estimated shape parameter is significantly negative, suggesting that the limiting distribution on $\rho_{map}^{\min}$ is the Weibull. The result is stronger for the MmLHDs.

Table 20. Read columnwise: the G value, location, scale, and shape parameter estimates using the GEV function, 95% percentile bootstrap confidence interval of the shape parameter, K-S statistic, p-value, and fail to reject (FTR), if appropriate.

$DSE_{n=200,k=20,G=\mathbf{G}}$

	G	location	scale	shape	shape Bootstrap 95%CI	D	p-value	
LHD	10	0.169791	0.010690	-0.235116	[-0.2776, -0.1999]	0.03050	0.3098	FTR
	25	0.162976	0.008490	-0.260605	[-0.3243, -0.2027]	0.05532	0.1728	FTR
	50	0.159024	0.007740	-0.272185	[-0.4703, -0.1848]	0.06865	0.3024	FTR
	100	0.155537	0.007185	-0.358878	[-0.5586, -0.1883]	0.09209	0.3645	FTR
MmLHD	10	0.114857	0.006872	-0.255875	[-0.2902, -0.2236]	0.01651	0.9479	FTR
	25	0.110379	0.005277	-0.265336	[-0.3595, -0.2196]	0.03865	0.5885	FTR
	50	0.107527	0.004403	-0.256185	[-0.3530, -0.1805]	0.05598	0.5576	FTR
	100	0.105569	0.003856	-0.343591	[-0.4500, -0.2367]	0.06552	0.7838	FTR

F. SUMMARY OF FINDINGS

This chapter provides guidance on the $\rho_{map}^{\min}$ value experimenters can expect to obtain when they generate G MmLHDs for k = 5, 6, …, 20 and $n = k+1$, $3k+2$, and $10k$. We also explore the limiting distribution of $\rho_{map}^{\min}$ in LHDs and MmLHDs for our largest designs; i.e., k = 20 and n = 200. The results suggest that asymptotically, as G gets larger, $\rho_{map}^{\min}$ converges to a Weibull distribution. The research in this section only scratches the surface of what is possible. The analysis above can be applied to other design sizes in LHDs and MmLHDs. Moreover, the approach can also be applied to any of our ten measures for the other classes of SFDs. Finally, with more experimentation, it might be useful to fit a regression of $\overline{\rho_{map}^{\min}}$ in MmLHDs to n, k, and G as Hernandez et al. (2012b) did for LHDs. If so, more G and design density levels (i.e., n) would be valuable.

The ultimate goal of this research is to provide experimenters with good designs. The experiments within this chapter are only the beginning of what is possible in terms of understanding how the distributions of design measures depend on the class of design, n, k, and G—with an initial investigation of the distribution of $\rho_{map}^{\min}$ for LHDs and MmLHDs. With more experimentation, in addition to utilizing asymptotic fits, experimenters can use the empirical distributions of $\rho_{map}^{\min}$ (and other measures) from the experiments to better

understand the quality of design they can obtain as functions of the class of design, n, k, and G.

VII. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Large-scale computer-based experimentation underpins much of the analysis the DOD (Department of Defense) uses to keep the nation safe. This research develops methods that increase the inventory of space-filling designs (SFDs) available for those experiments. The new `permute_and_stack` method uses a quadratically constrained mixed integer program to enable researchers to efficiently add sequential blocks of design points to existing SFDs that minimize ρ_{map}. SFDs extended in this work include both cataloged designs and computationally expensive software-generated designs.

To evaluate `permute_and_stack`'s potential, this research explored how `permute_and_stack` reduces ρ_{map} when successively adding batches (or stacks) of n design points (DPs) for several state-of-the-art classes of cataloged and computer-generated SFDs. The reductions in ρ_{map} are contrasted with values obtained by applying shift-and-stack, a current approach used by practitioners, to base designs using several correlation-based column-reordering heuristics as well as random permutations of the columns. We also evaluate `permute_and_stack` against newly generated SFDs in the extended design space. In all cases, `permute_and_stack` yields better designs in terms of ρ_{map} than do the competing methods. However, new designs generated in the extended space may produce better designs with respect to some of the measures of space-fillingness. For every class and design dimension explored, the set of multi-objective (i.e., ρ_{map} and $(ML_2)^2$) non-dominated designs includes `permute_and_stack` designs.

There are many classes of SFDs and measures to assess them. However, finding an "optimal" SFD is no easy task, with stochastic search algorithms almost always used in their construction. Moreover, the arrived-at designs' properties can vary significantly based on random-number-generator seeds, starting points, or search-algorithm parameter settings. To better understand the variability in design characteristic measures and their relationships, nearly a million SFDs of several types and sizes were created using JMP and R software packages. Across the design classes, for many measures, design preference can

change with the number of DPs (n) and factors (k). High variances and outliers in the plots reveal the risks of generating just one "optimal" design with stochastic software. Researchers need to generate many SFDs to ensure they obtain a design with good properties for their application.

With the goal of helping experimenters obtain good designs, this research provides guidance on the minimum ρ_{map} (denoted $\rho_{map}^{\min}$) value they can expect to obtain when generating G independent maximin Latin hypercube designs (MmLHDs) for multiple n and k. We also explore the limiting distribution of $\rho_{map}^{\min}$ in Latin hypercube designs (LHDs) and MmLHDs for our largest designs; i.e., $k = 20$ and $n = 200$. The results suggest that as G increases, $\rho_{map}^{\min}$ converges to a Weibull distribution.

The research in this dissertation only scratches the surface of what is possible. Investigators can extend the analysis approaches and can apply them to any measure of design characteristics for any class of SFD. Exciting research areas that would benefit from further study include:

- The `permute_and_stack` algorithm optimizes one measure of design quality, ρ_{map}. Can sequential optimization approaches simultaneously improve a design's correlation, balance, and higher-dimensional space-fillingness properties? To improve correlation, balance (Bathke 2004, Vieira et al. 2011), and space-fillingness for mixes of factor types (continuous and discrete), some initial analysis has been done. Row Enhanced eXperiments (REX) is an algorithm that expands upon the ideas of `permute_and_stack` to sequentially improve a design's correlation, balance, and two-dimensional space-fillingness properties. Initial thoughts and scatter plots are provided in Appendix I.

- To better understand the many criteria in the literature assessing SFDs, we ran nearly a million experiments. From them, we collected data for ten measures of design quality for nine design sizes in each of six SFD

classes. At present, only a small fraction of them have been investigated in detail.

- This investigation examined the asymptotic behavior of one design measure ($\rho_{map}^{\min}$) as G increases in only two design types of one size. However, we have the designs generated, measurement data, and software scripts to apply the methods of this dissertation to many measures of design characteristics, classes, and sizes.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. EXTENDING CATALOGED DESIGNS

Extending cataloged nearly orthogonal Latin hypercubes (NOLHs)

Figure 95 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to ten stacks for all the cataloged NOLH designs. This is the motivation that originated Chapters III and IV of this research. We significantly improve with `permute_and_stack` in terms of ρ_{map} at each stage and observe that the $(ML_2)^2$ values are similar.

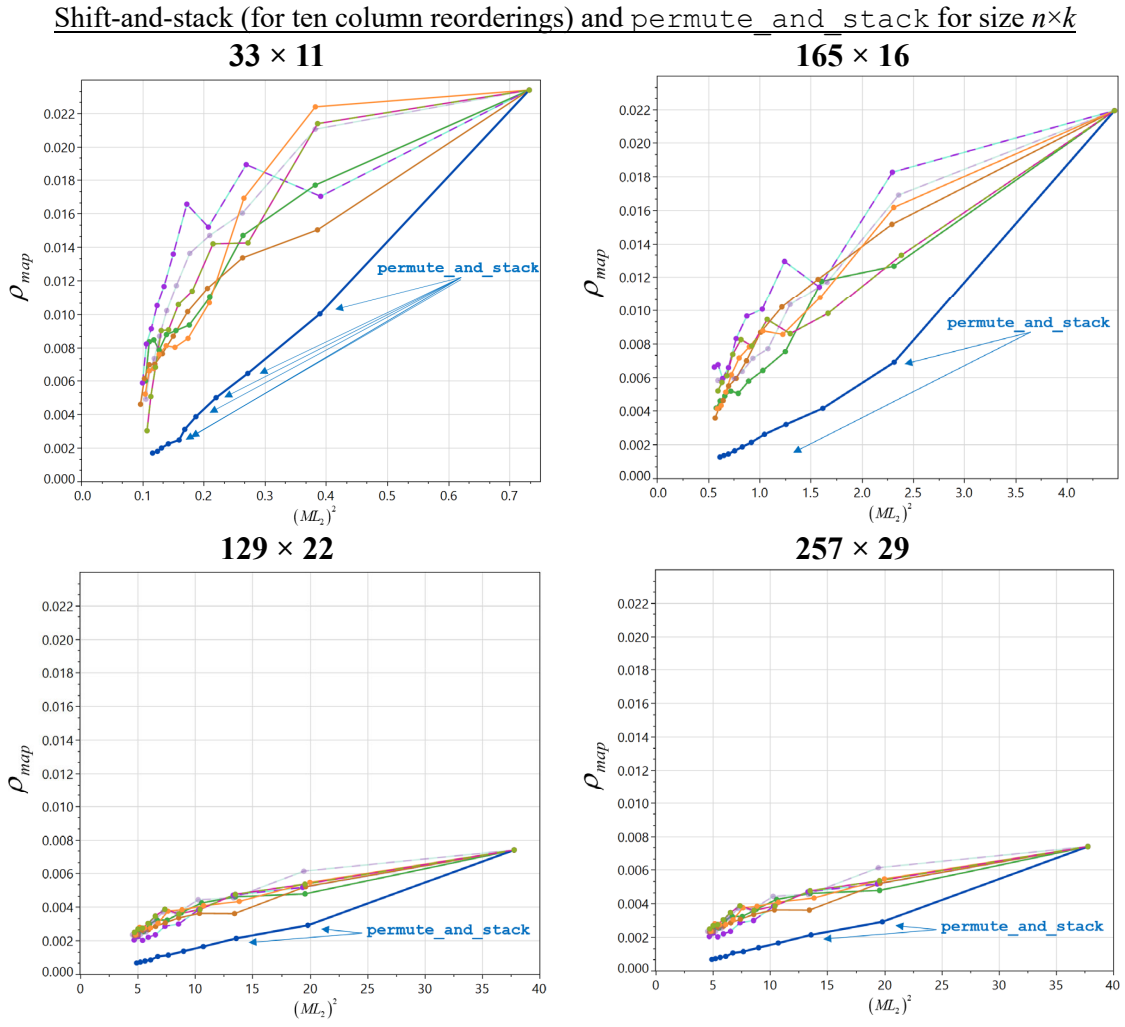


Figure 95. Multi-objective ρ_{map} and $(ML_2)^2$ comparison for extended NOLH designs up to ten applications of shift-and-stack and `permute_and_stack`.

Extending cataloged uniform design (UDs)

Figure 96 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to ten stacks for select cataloged uniform designs (UDs). We significantly improve with `permute_and_stack` in terms of ρ_{map} at each stage and observe that the $(ML_2)^2$ values are similar.

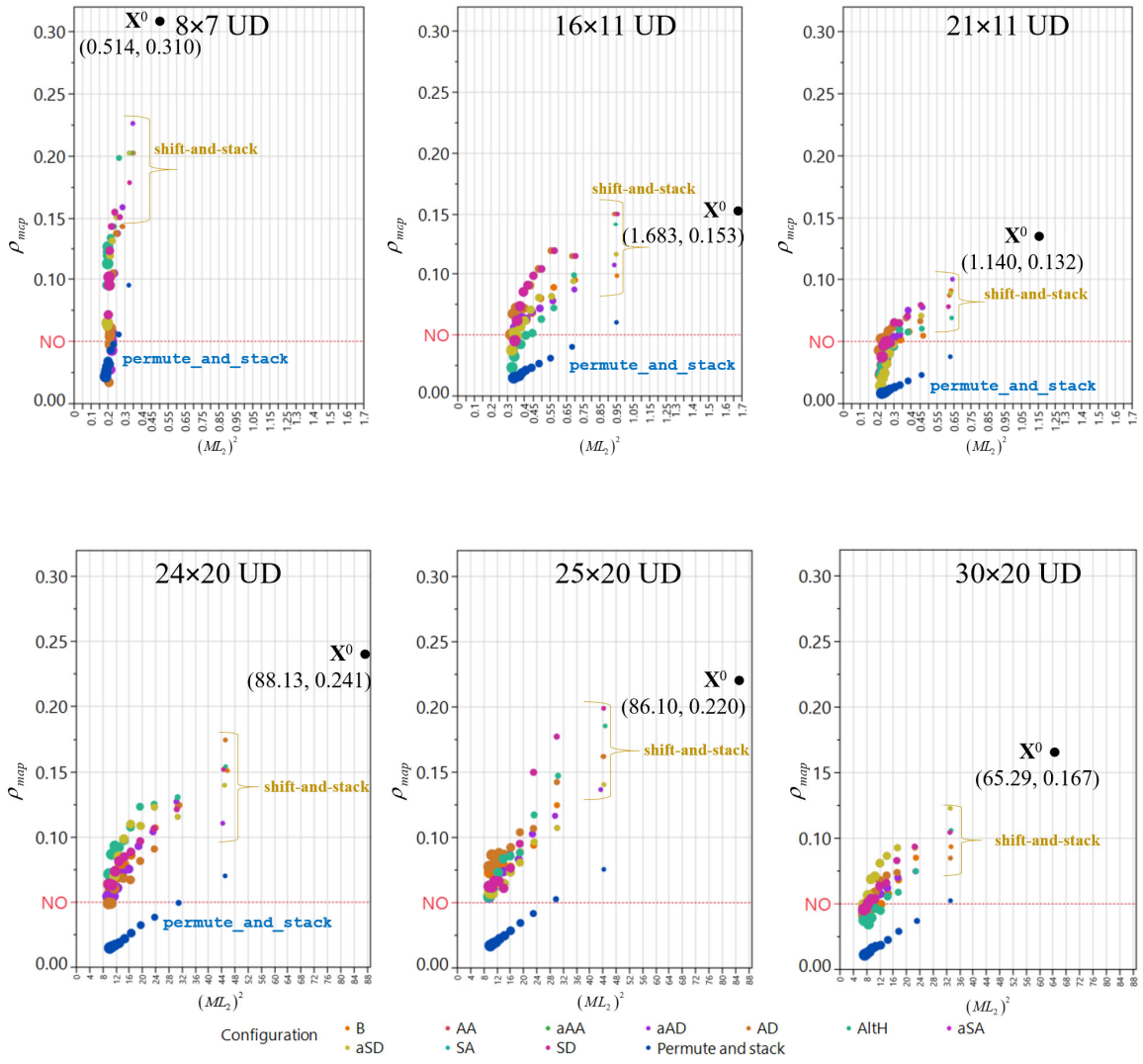


Figure 96. Multi-objective ρ_{map} and $(ML_2)^2$ comparison for extended uniform designs up to ten applications of shift-and-stack and `permute_and_stack`.

APPENDIX B. EXTENDING STOCHASTICALLY GENERATED SPACE-FILLING DESIGNS USING JMP AND R SOFTWARE PACKAGES

Extending JMP's MmLHDs

Figure 97 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to four stacks of stochastically generated MmLHDs constructed using JMP.

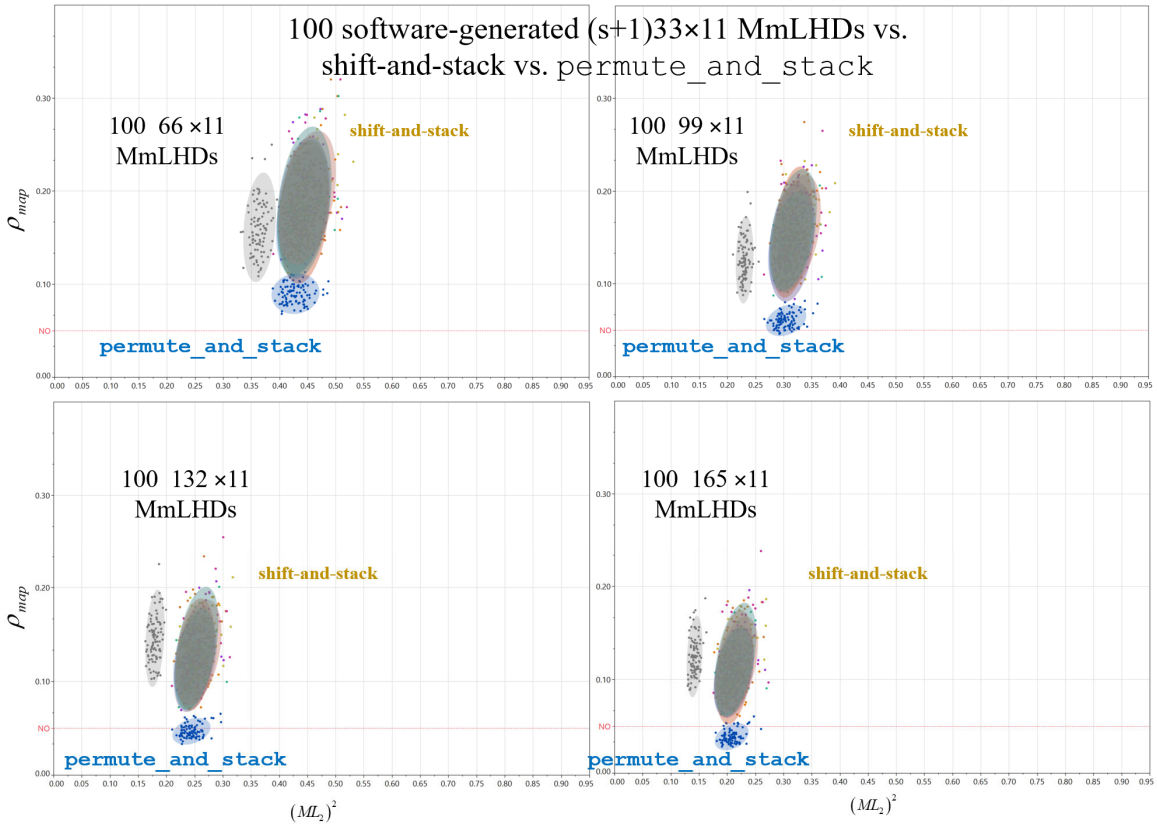


Figure 97. 100 33×11 MmLHDs after four applications (s = 1, 2, 3, and 4) of forward shift-and-stack using the ten column reordering heuristics (light colors) and `permute_and_stack` (blue). 100 new (s+1)×11 MmLHD constructions are plotted individually (grey).

Extending R's maximum projection (MaxPro) designs

Figure 98 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to four stacks of stochastically generated MaxPro designs.

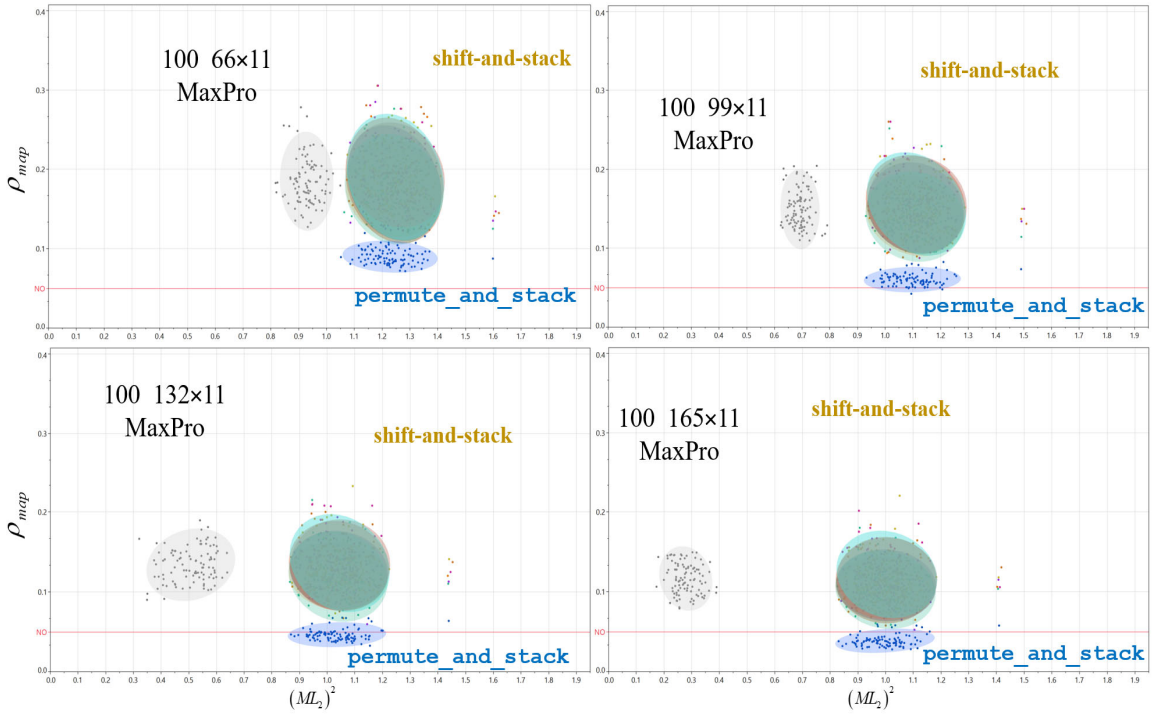


Figure 98. 100 33×11 MaxPro designs after four applications (s = 1, 2, 3, and 4) of forward shift-and-stack using the ten column reordering heuristics (light colors) and `permute_and_stack` (blue). 100 new (s+1)×11 MaxPro design constructions are plotted individually (grey).

Extending JMP's sphere-packing (Mm distance) designs

Figure 99 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to four stacks of stochastically generated sphere-packing (Mm distance) designs constructed using JMP.

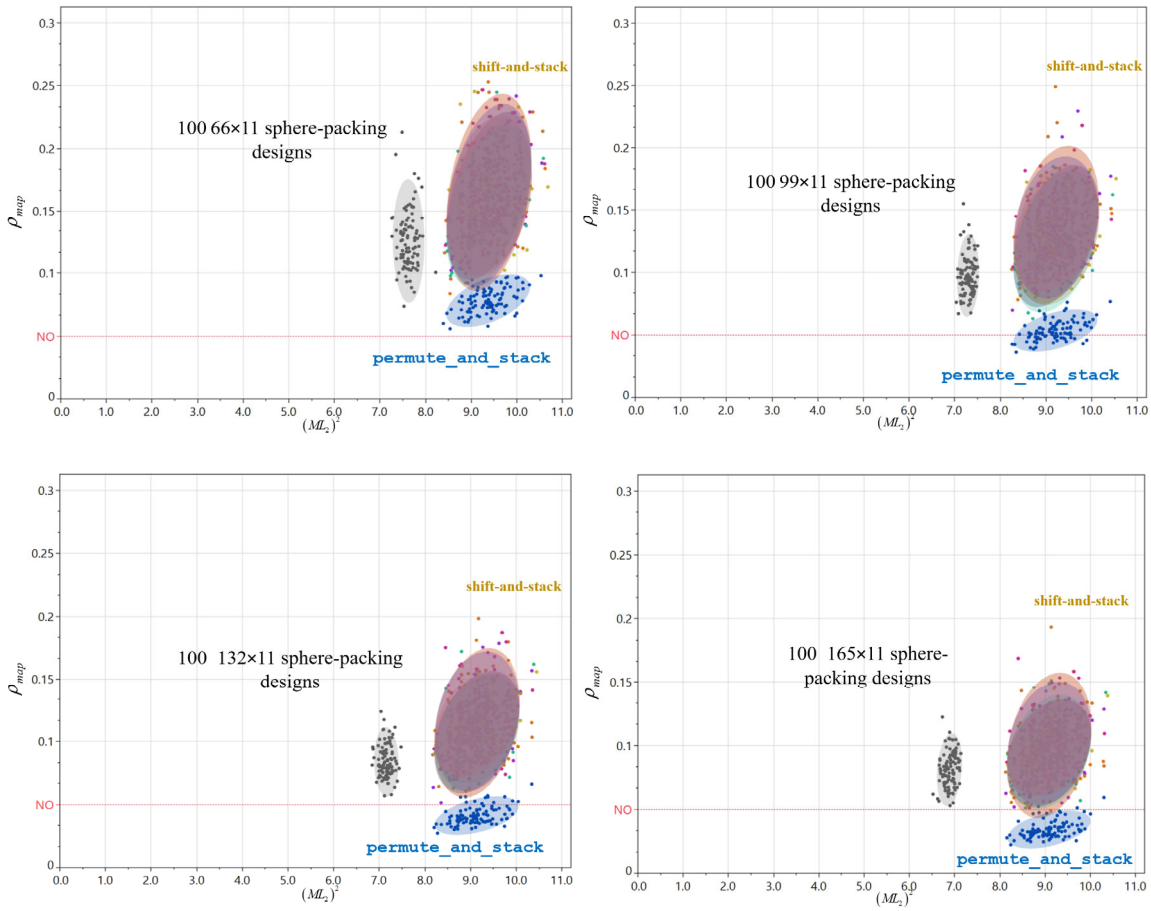


Figure 99. 100 33×11 sphere-packing designs after four applications (s = 1, 2, 3, and 4) of forward shift-and-stack using the ten column reordering heuristics (light colors) and `permute_and_stack` (blue). 100 new (s+1)×11 sphere-packing designs constructions are plotted individually (grey).

Extending R's UniDOE uniform designs

Figure 100 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to four stacks of stochastically generated UniDOE designs constructed using the *UniDOE* R software package.

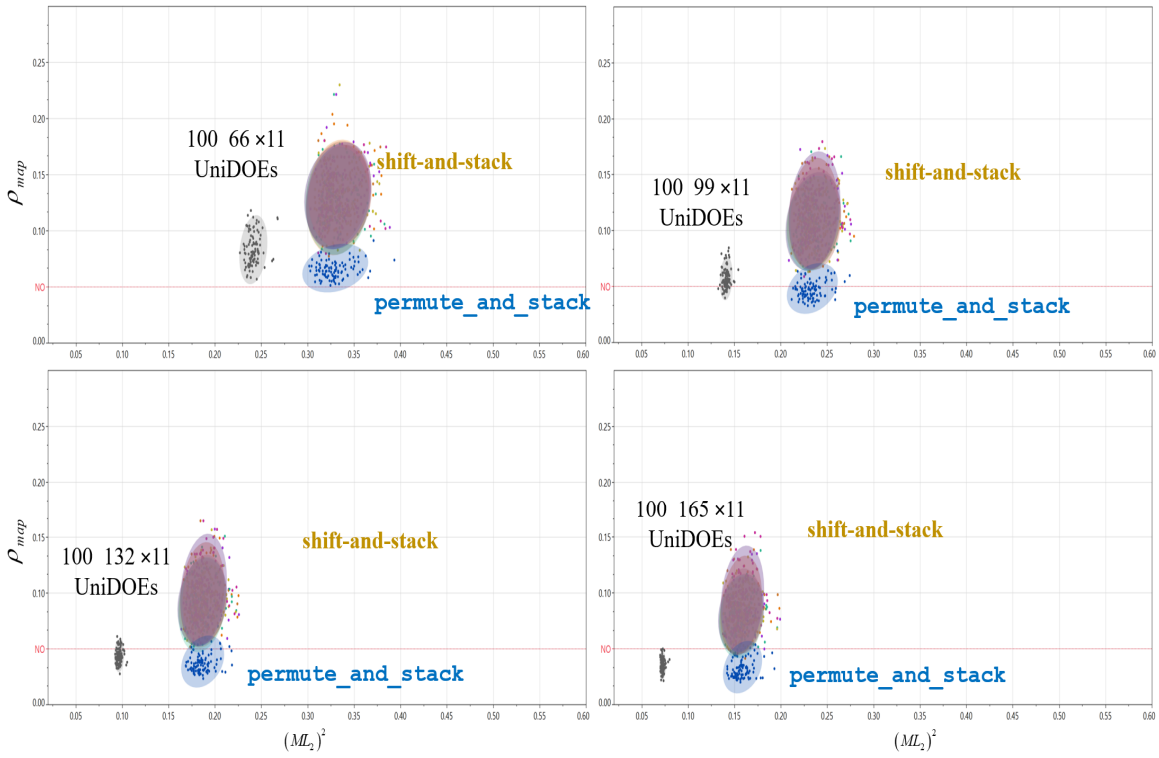


Figure 100. 100 33×11 UniDOE designs after four applications (s = 1, 2, 3, and 4) of forward shift-and-stack using the ten column reordering heuristics (light colors) and `permute_and_stack` (blue). 100 new (s+1)×11 UniDOE design constructions are plotted individually (grey).

Extending JMP's uniform designs (UDs)

Figure 101 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to four stacks of stochastically generated uniform designs constructed using JMP.

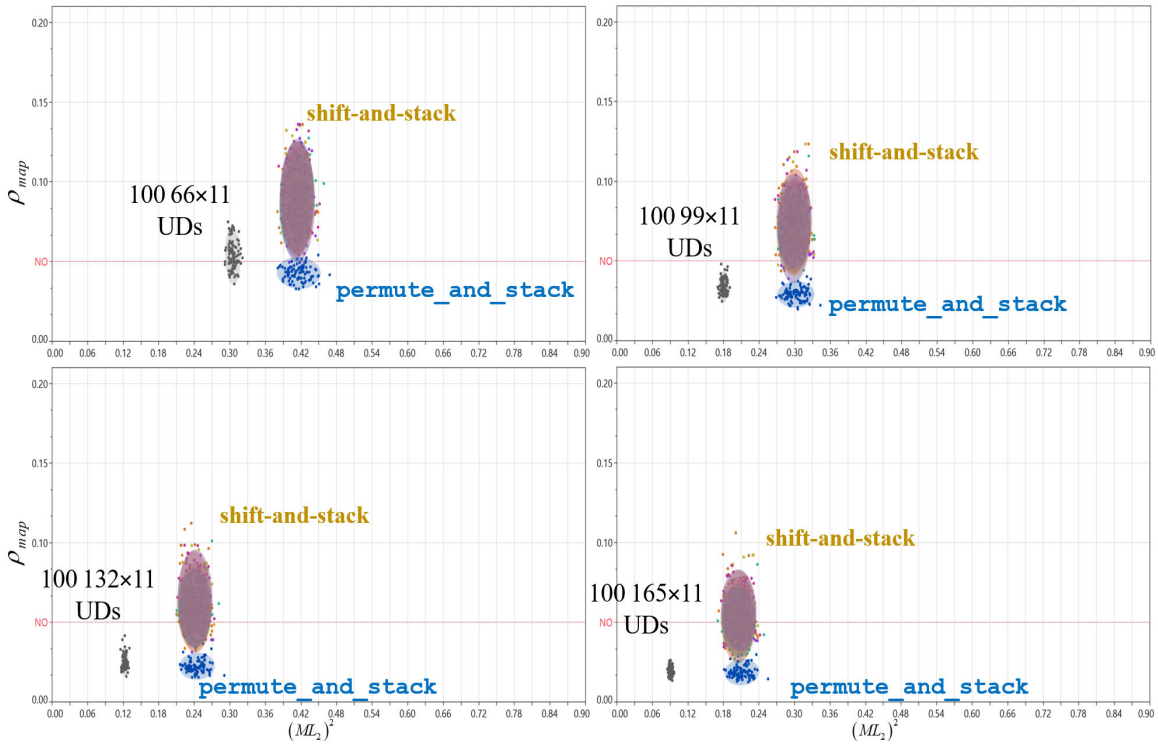


Figure 101. 100 33×11 UDs generated using JMP, four application (s = 1, 2, 3, and 4) of forward shift-and-stack using the ten column reordering heuristics (light colors), and `permute_and_stack` (blue). 100 new (s+1)×11 UDs for the extended space are plotted individually (grey).

Extending random Latin hypercube designs (LHDs)

Figure 102 shows ρ_{map} and $(ML_2)^2$ discrepancy measures for up to four stacks of stochastically generated random Latin hypercube designs (LHDs) constructed using R.

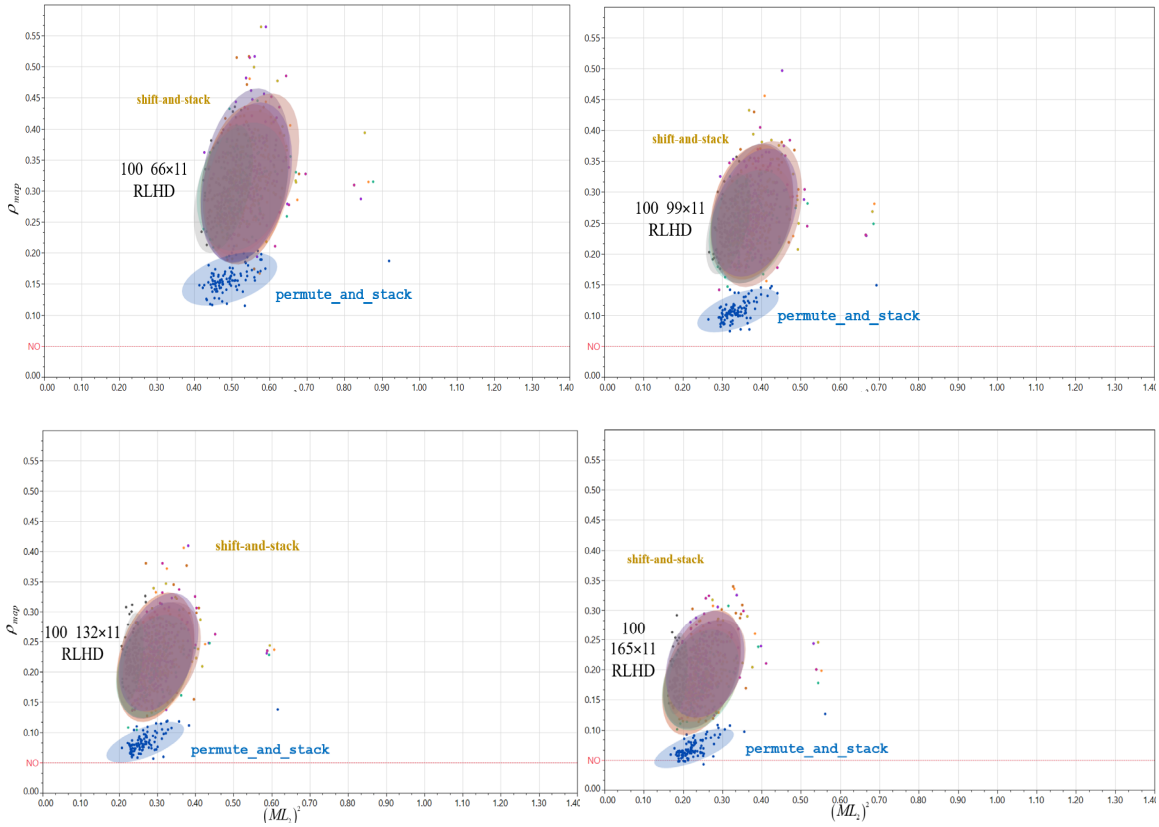


Figure 102. 100 33×11 LHDs generated using R, four applications (s = 1, 2, 3, and 4) of forward shift-and-stack using the ten column reordering heuristics (light colors), and `permute_and_stack` (blue). 100 new (s+1)×11 LHDs for the extended space are plotted individually (grey). We observe that most of the grey observations are plotted within the shaded ellipses of the shift-and-stack observations.

APPENDIX C. PERMUTATIONS FOR THE NOLH DESIGN USING OPTIMIZATION

Table 21. Permutations for the 65×16 NOLH through 10 stacks.

Optimal `permute_and_stack` permutations $p^1, \ldots, p^{10}$:
1: [1,0,4,12,2,10,7,3,9,5,8,13,11,6,15,14], 2: [6,1,2,11,13,12,10,8,7,14,15,4,3,5,0,9], 3: [14,4,12,10,9,13,15,2,0,8,1,3,7,5,11,6], 4: [12,1,11,9,5,13,10,7,3,15,8,0,6,2,4,14], 5: [15,1,9,7,12,2,8,14,11,13,5,6,0,4,3,10], 6: [12,11,4,1,7,0,9,2,8,5,13,3,10,15,6,14], 7: [13,4,7,8,12,14,11,9,2,1,15,0,10,6,5,3], 8: [14,3,15,0,12,11,1,9,10,8,6,13,2,4,7,5], 9: [3,5,1,8,9,10,2,4,6,7,11,0,13,15,12,14], 10:[0,14,7,3,10,2,12,6,9,11,1,5,13,8,4,15].
ρ_{map}:
1: [0.0069], 2: [0.0041], 3: [0.0032], 4: [0.0026], 5: [0.0021], 6: [0.0018], 7: [0.0016], 8: [0.0014], 9: [0.0013], 10: [0.0012].

Table 22. Permutations for the 129×22 NOLH through 10 stacks.

Optimal `permute_and_stack` permutations $p^1, \ldots, p^{10}$:
1: [10,12,19,7,13,6,20,15,18,9,11,3,16,14,21,8,0,5,17,2,4,1], 2: [5,15,4,9,21,10,1,16,6,0,18,11,3,20,7,17,12,14,2,13,8,19], 3: [3,2,12,15,11,0,6,10,20,18,1,14,13,19,7,17,5,21,16,9,4,8], 4: [12,3,7,1,18,9,2,5,10,15,6,20,17,4,0,13,21,16,19,11,8,14], 5: [12,7,4,11,3,16,9,1,20,10,13,5,6,0,21,18,2,17,14,15,19,8], 6: [6,21,12,11,17,1,20,5,13,3,15,7,18,19,8,4,0,14,10,16,9,2], 7: [4,21,13,15,3,11,5,6,18,19,9,0,8,16,17,10,2,12,20,1,14,7], 8: [9,8,6,19,17,5,2,4,16,12,15,7,14,0,11,3,13,1,20,21,10,18], 9: [10,5,12,21,20,0,18,4,17,15,8,1,16,14,13,19,9,7,3,11,6,2], 10:[11,1,16,15,9,18,10,5,19,17,4,6,3,14,21,0,8,2,20,12,13,7].
ρ_{map}:
1: [0.0033], 2: [0.0023], 3: [0.0019], 4: [0.0017], 5: [0.0015], 6: [0.0013], 7: [0.0012], 8: [0.0010], 9: [0.0010], 10: [0.0009].

Table 23. Permutations for the 257×29 NOLH through 10 stacks.

Optimal `permute_and_stack` permutations $p^1, \ldots, p^{10}$:
1:[10,5,28,12,0,3,9,1,4,14,2,8,19,7,20,17,6,15,21,13,22,11,27,26,24,23,16,25,18], 2:[13,5,4,23,0,11,14,6,3,25,27,1,17,7,2,8,15,16,22,21,10,20,12,18,24,26,19,9,28], 3:[13,20,24,12,2,17,4,1,3,6,18,8,14,9,7,16,10,11,28,21,22,26,19,0,25,27,5,15,23], 4:[10,23,9,18,0,4,5,12,19,6,2,28,15,16,20,13,14,1,21,26,22,8,7,3,25,11,17,24,27], 5:[12,7,2,19,5,4,13,3,20,9,1,18,25,6,28,15,14,10,27,23,21,0,11,26,22,16,8,24,17], 6:[10,27,5,12,0,13,11,1,4,8,17,7,6,18,25,22,14,3,21,9,2,24,16,26,15,23,28,20,19], 7:[20,27,5,23,1,9,12,4,6,22,7,18,15,17,0,16,14,3,19,13,2,11,28,26,24,10,25,8,21], 8:[13,7,19,12,0,3,2,1,4,6,28,18,5,9,20,17,14,8,15,22,26,11,27,25,24,16,10,21,23], 9:[13,5,8,19,2,12,11,0,14,16,1,24,15,6,26,4,20,3,21,9,22,10,23,17,28,25,27,18,7], 10:[27,5,7,12,6,1,2,28,4,14,20,18,13,9,10,0,15,3,19,17,22,11,16,24,25,23,21,26,8].
ρ_{map}:
1: [0.0019], 2: [0.0016], 3: [0.0013], 4: [0.0011], 5: [0.0009], 6: [0.0009], 7: [0.0009], 8: [0.0010], 9: [0.0008], 10:[0.0007].

APPENDIX D. SCRIPTS AND FUNCTIONS

This appendix gives a sampling of the many scripts underpinning this research. For example, generating many designs in JMP, R (e.g., MaxPro) or computing measures (e.g., $(ML_2)^2$ and $(CL_2)^2$).

```
//                          Sphere Packing                          //
For( k=1, k<=100, k++,
//-----------------------------------------------------------------
dim = 5;
dpts= 50;
      temp = Random Integer(1000000);
//-----------------------------------------------------------------
      path = "C:\Users\jdparker8\Box\00 Dissertation\07
100_SFSP\5\50\";
      path2 = "C:\Users\jdparker8\Box\00 Dissertation\07
100_SFSP\5\50\SEEDs\";
//-----------------------------------------------------------------
      d = DOE(
            Space Filling Design,
            {Add Response( Maximize, "Y", ., ., . ), Add Factor( Contin-
uous, 0, 1, "X1", 0 ),Add Factor( Continuous, 0, 1, "X2", 0 ), Add
Factor( Continuous, 0, 1, "X3", 0 ),Add Factor( Continuous, 0, 1,
"X4", 0 ), Add Factor( Continuous, 0, 1, "X5", 0 ),
Set Random Seed( temp ),
Space Filling Design Type( Sphere Packing, dpts ), Simulate Responses(
0 ),Set Run Order( Randomize )}
 );
      design = d << Make Table;
      design2 = d << Make Table;
//-----------------------------------------------------------------
      design:Y << Set Selected;
      design2:Y << Set Selected;
      Wait(2);
      design << Delete Columns();
      design2:Y << Set Selected;
//-----------------------------------------------------------------
      final = design << Make Table;
      final2= design << Make Table;
      final << Save( path ||
"SF_SphereP"||"_"||char(dpts)||"_"||char(dim)||"_"||char(k)||".csv");
            final2<< Save( path2||
"SF_SphereP"||"_"||char(dpts)||"_"||char(dim)||"_"||char(k)||"_"||char
(temp)||".csv");
);
```

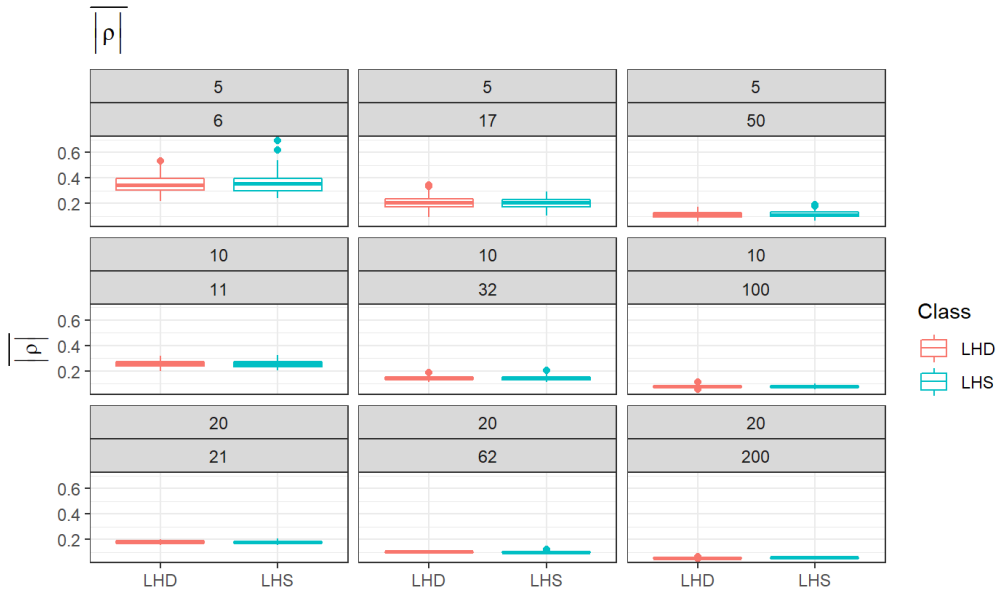
```
# ------------------------   MaxPro   ------------------------#
install.packages("MaxPro")
library(MaxPro)
num_facs = 5; num_dps = 6
#size_list = c(6,99,132,165,198,231,264,297,330,363) # DPs
size_list = c(num_dps)
for (size in size_list) {
  dp=size
  k=num_facs
  num_calls = 5
  for (i in 1:num_calls)
  {
    # The initial design matrix, which we recommend
    # to be a MaxPro Latin hypercube design
    # generated by the MaxProLHD function
    temp_init_des <- MaxProLHD(n = dp, p = k)$Design     #design
    temp <- MaxPro(temp_init_des)$Design
    tt <- data.frame(temp)
    write.csv(tt, file =
                paste("MaxPro_",size,"_",k,"_",i,"_SEED",
                      x,".csv", sep = ""),
              row.names=FALSE)
 }
}
```

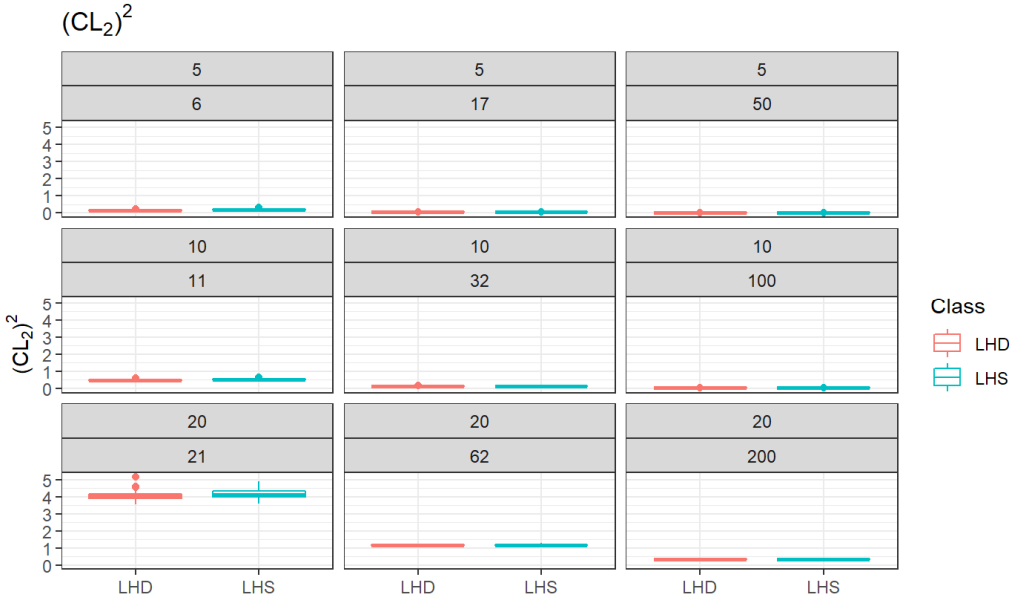
```
# ------------------    ML_2_function   ----------------------#

ML_2_function <- function(df) {
  k = ncol(df)       # cols
  n = nrow(df)       # rows
  sum_2 <- 0
  sum_3 <- 0
  for(d in 1:n){
    prod_2 <- 1               # a product
    sum_32 <- 0
    for( j in 1:n){
      prod_3 <- 1
      for(i in 1:k){
        if( j == 1){
          num1 = (3 - (df[d, i])**2)
          prod_2 <- num1*prod_2
        }
        num_3 <- (2.0 - max(df[d,i] , df[j,i]))
        prod_3<- num_3*prod_3
      }
      sum_32 <- sum_32 + prod_3
    }
    sum_2 <- sum_2 + prod_2
    sum_3 <- sum_3 + sum_32
  }
  term_1 <- (4/3)**k
  term_2 <- ((2**(1-k))/n)*sum_2
  term_3 <- (1/(n**2))*(sum_3)
  ml2_disc <- term_1 - term_2 + term_3
  return(as.numeric(ml2_disc))    # max function X16 and a value
}
# Eg.,
# ML_2_function( apply(read.csv("nolhcioppa_n33k11.csv") ,
MARGIN = 2, FUN = function(X) (X - min(X))/diff(range(X)))) #[1]
0.7318222
# or... w/ scaler function
# ML_2_function( scaler_func(read.csv("nolhcioppa_n33k11.csv"
))) #[1] 0.7318222
```

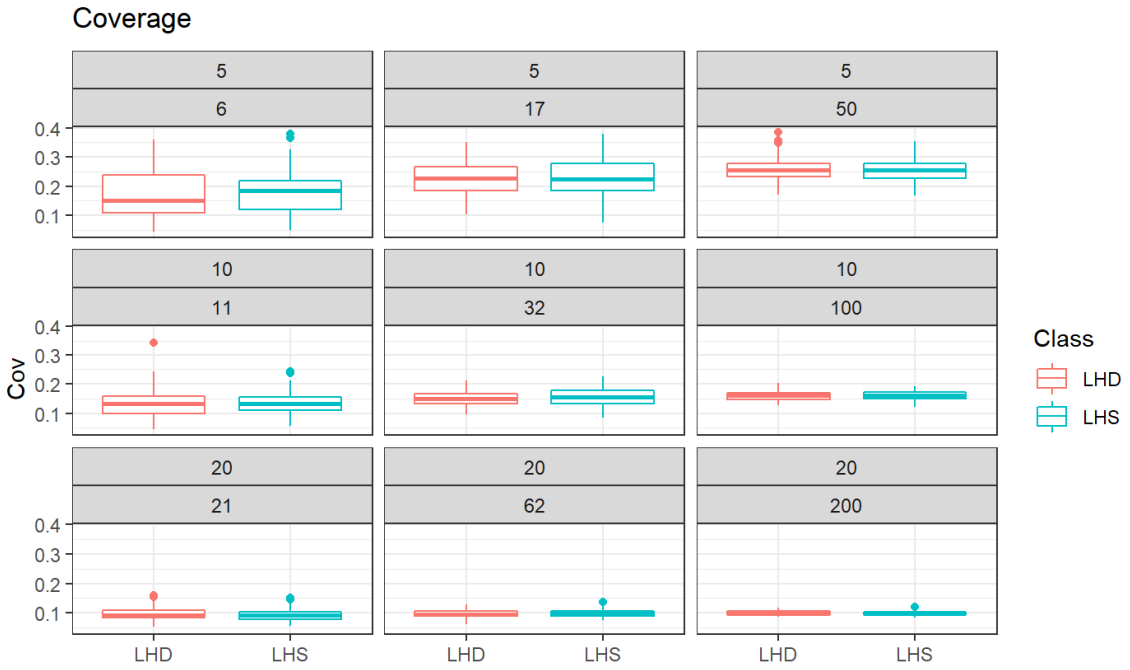
```
# ---------------------  CL_2_function  ----------------------#

CL_2_function <- function(df) {
  # CL_2 Function
  k = ncol(df)       # cols
  n = nrow(df)       # rows
  sum_2 <- 0
  sum_3 <- 0
  for(d in 1:n){
    prod_2 <- 1
    sum_32 <- 0
    for(j in 1:n){
      prod_3 <- 1
      for(i in 1:k){
        if(j == 1){
          num1 = (1.0 + abs(df[d, i] - 0.5)/2.0 - abs(df[d, i] -
0.5)**2/2.0)
          prod_2 <- prod_2*num1
        }
        num_3 = (1.0 + abs(df[d,i]-0.5)/2.0 + abs(df[j,i]-
0.5)/2.0 - abs(df[d,i] - df[j,i])/2.0)
        prod_3 <- num_3*prod_3}
      sum_32 <- sum_32 + prod_3
    }
    sum_2  <- sum_2 + prod_2
    sum_3  <- sum_3 + sum_32
  }
  term_1 <- (13/12)**k
  term_2 <- (2/n)*(sum_2)
  term_3 <- (1/(n**2))*(sum_3)
  cl2_disc <- term_1 - term_2 + term_3
  return(as.numeric(cl2_disc))
}
# Eg.,
# CL_2_function(read.csv("uniform_n33k11_09508.csv"))
# CL_2_function(apply(read.csv("nolhcioppa_n33k11.csv") , MARGIN
= 2, FUN = function(X) (X - min(X))/diff(range(X))))
# CL_2_function(df)  # [1] 0.09058806  *  <-----matches JMP! }
}
```

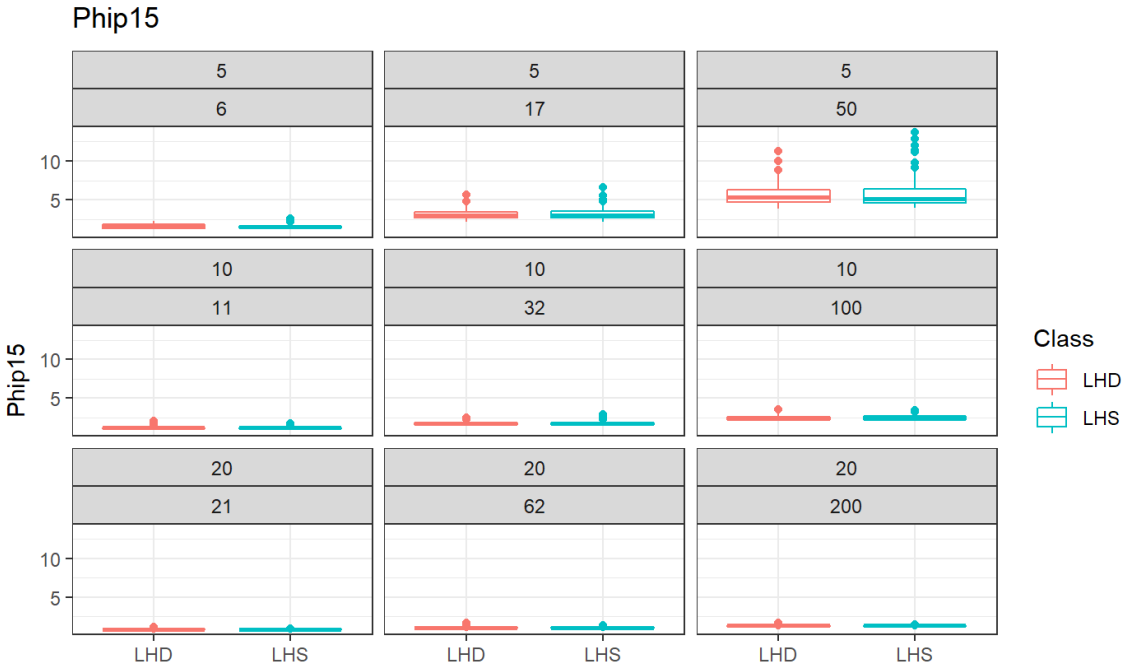
APPENDIX E. COMPLETE LHD AND LHS COMPARISON

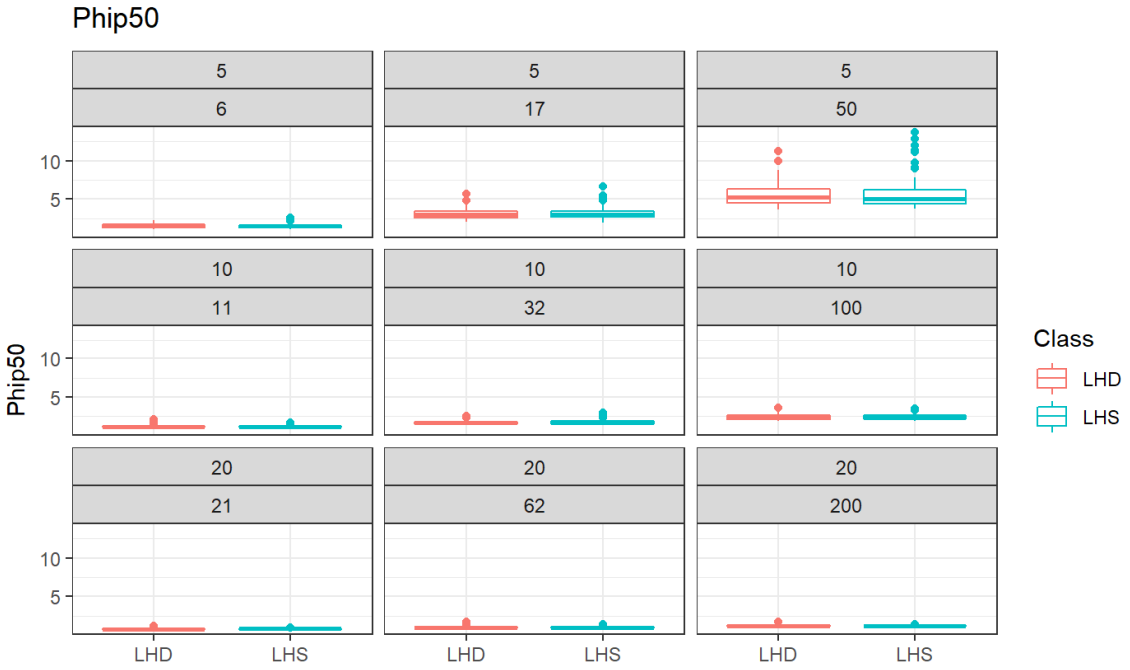
We compare Latin hypercube sampling (LHS) (McKay et al. 1979) and Latin hypercube designs (LHDs) on a lattice i.e., natural numbers 1,…, *n*.

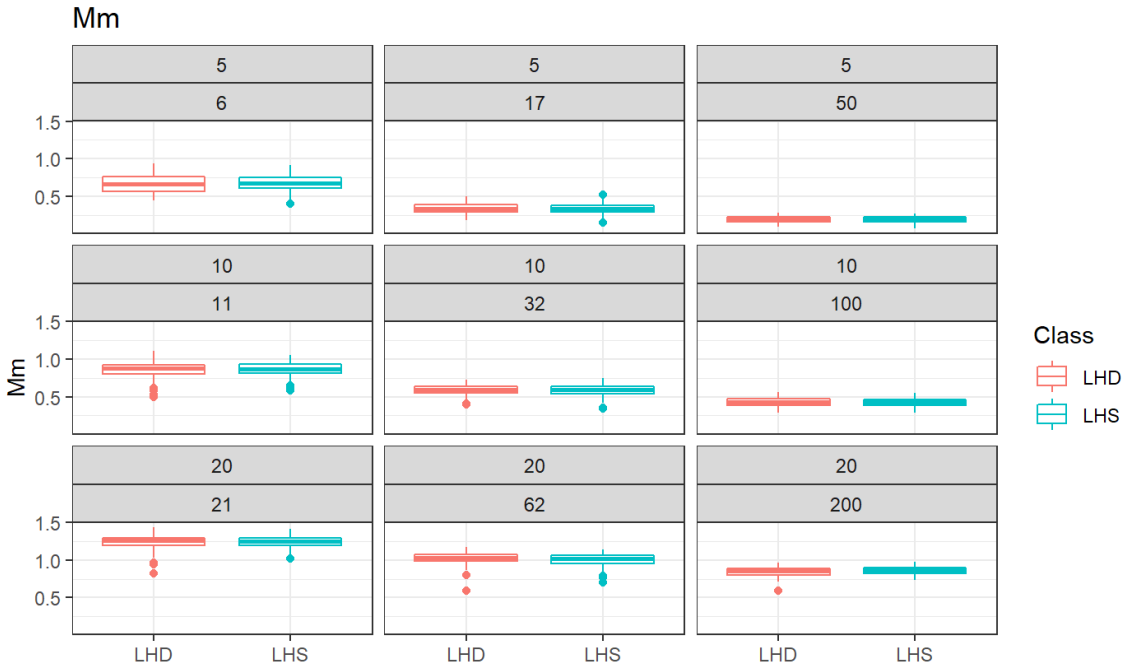


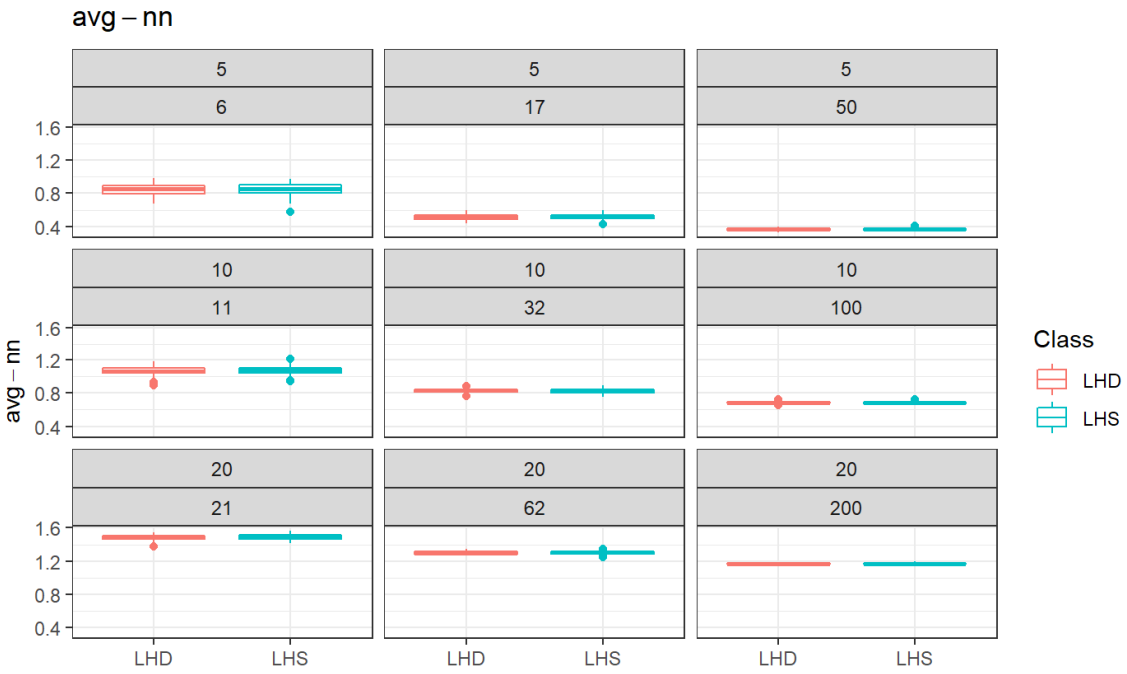



Coverage
5
6
5
17
5
50
10
11
10
32
10
100
20
21
20
62
20
200
Cov
0.4
0.3
0.2
0.1
LHD
LHS
Class
LHD
LHS


Phip15
5
6
5
17
5
50
10
11
10
32
10
100
20
21
20
62
20
200
10
5
LHD
LHS
Class
LHD
LHS


Phip50
5
6
5
17
5
50
10
11
10
32
10
100
20
21
20
62
20
200
10
5
Phip50
LHD
LHS
Class
LHD
LHS


Mm
5
6
5
17
5
50
10
11
10
32
10
100
20
21
20
62
20
200
1.5
1.0
0.5
Mm
LHD
LHS
Class
LHD
LHS


avg – nn
5
6
5
17
5
50
10
11
10
32
10
100
20
21
20
62
20
200
1.6
1.2
0.8
0.4
avg – nn
LHD
LHS
Class
LHD
LHS

APPENDIX F. NUMERICAL RESULTS ON MEASURES

We compare the mean and standard deviation (SD) values for a partial list of SFMs for six design classes—each size k = 5, 10, and 20 and $n = k+1$, $3k+2$, and $10k$. The tables that follow present an SFM for correlation, discrepancy, and distance criterion (i.e., ρ_{map}, $(ML_2)^2$, Mm distance). We *italicize* the best and **bold** the worst values for each design size—for the sphere packing, MmLHD, MaxPro, and UniDOE designs. An asterisk (*) is placed next to nearly orthogonal averages.

Table 24. The mean ρ_{map} and standard deviation (SD) for 100 generations.

		ρ_{map} Mean(SD)		
n	k	5	10	20
	SphereP	*0.122*(0.109)	*0.351*(**0.103**)	*0.396*(*0.049*)
	MmLHD	0.232(*0.075*)	**0.515**(0.074)	**0.535**(**0.065**)
$k+1$	MaxPro	0.268(0.150)	0.473(0.068)	0.472(0.059)
	UniDOE	**0.314**(**0.243**)	0.453(*0.063*)	0.417(0.050)
	LHD	0.771(0.117)	0.715(0.0863)	0.633(0.068)
	LHS	0.776(0.116)	0.713(0.0885)	0.626(0.069)
	SphereP	*0.130*(0.043)	0.186(0.034)	0.199(0.023)
	MmLHD	**0.210**(**0.059**)	0.233(0.039)	**0.282**(**0.036**)
$3k+2$	MaxPro	0.199(0.049)	**0.241**(**0.044**)	0.255(0.030)
	UniDOE	0.179(*0.040*)	*0.167*(*0.027*)	*0.148*(*0.017*)
	LHD	0.478(0.127)	0.441(0.058)	0.376(0.049)
	LHS	0.466(0.105)	0.431(0.068)	0.373(0.041)
	SphereP	0.080(0.022)	0.095(0.016)	0.092(0.011)
	MmLHD	0.100(0.027)	0.116(0.019)	0.144(**0.020**)
$10k$	MaxPro	**0.123**(**0.032**)	**0.138**(**0.022**)	0.144(0.018)
	UniDOE	*0.048*(*0.012*)*	*0.052*(*0.009*)	*0.052*(*0.006*)
	LHD	0.263(0.066)	0.241(0.040)	0.207(0.026)
	LHS	0.269(0.069)	0.238(0.039)	0.208(0.022)

Table 25. The mean $(ML_2)^2$ and SD for 100 generations.

		$(ML_2)^2$ Mean(SD)		
n	k	5	10	20
$k+1$	SphereP	**0.9395(0.0852)**	**8.3473(0.4077)**	**300.6357(21.6563)**
	MmLHD	*0.2476(0.0089)*	1.9561(0.1279)	109.3001(9.9046)
	MaxPro	0.3513(0.0151)	2.9921(0.1576)	150.0854(14.5567)
	UniDOE	0.2484(0.0113)	*1.8418(0.082)*	*99.2175(8.1533)*
	LHD	0.2807(0.0334)	2.1941(0.271)	111.2679(14.5126)
	LHS	0.3455(0.0507)	2.3325(0.2738)	116.6352(17.1141)
$3k+2$	SphereP	**0.6187(0.0435)**	**6.3572(0.1488)**	**223.3541(4.4844)**
	MmLHD	0.0452(0.002)	0.4884(0.0197)	33.7993(1.8507)
	MaxPro	0.0883(0.0056)	1.0132(0.0447)	57.0677(3.8417)
	UniDOE	*0.0357(0.0015)*	*0.3627(0.0152)*	*27.3118(1.3974)*
	LHD	0.0632(0.0087)	0.6238(0.0553)	36.7359(2.7603)
	LHS	0.0701(0.0094)	0.64 15(0.0648)	36. 8082(3.2555)
$10k$	SphereP	**0.2918(0.0178)**	**4.9562(0.1066)**	**190.1633(1.1378)**
	MmLHD	0.0095(5e-04)	0.1277(0.0045)	9.8368(0.2871)
	MaxPro	0.0309(0.0018)	0.4142(0.0267)	9.3869(0.3309)
	UniDOE	*0.0055(2e-04)*	*0.0749(0.0015)*	*7.0692(0.1609)*
	LHD	0.0166(0.0025)	0.1828(0.0139)	11.1295(0.4487)
	LHS	0.0183(0.0031)	0.1865(0.0181)	11.2464(0.5526)

JMP sphere packing (Mm distance) designs are dominated by all design types (in terms of $(ML_2)^2$), which includes randomly sampled LHDs.

Table 26. The mean Mm distance and SD for 100 generations of six design types for nine different sizes $n \times k$.

		Mm Mean(SD)		
n	k	5	10	20
$k+1$	SphereP	*1.522*(0.011)	*2.090*(0.024)	*2.916*(0.006)
	MmLHD	1.078(0.033)	1.248(0.025)	**1.577**(0.032)
	MaxPro	1.084(0.129)	1.311(0.061)	1.785(0.067)
	UniDOE	**1.012**(0.104)	**1.197**(0.034)	1.611(0.036)
	LHD	0.690(0.110)	0.862(0.109)	1.235(0.104)
	LHS	0.680(0.105)	0.866(0.097)	1.245(0.085)
$3k+2$	SphereP	*1.111*(0.009)	*1.757*(0.008)	*2.625*(0.008)
	MmLHD	0.715(0.013)	1.007(0.015)	1.394(0.016)
	MaxPro	0.599(0.034)	0.904(0.047)	1.398(0.057)
	UniDOE	**0.493**(0.034)	**0.764**(0.035)	**1.217**(0.039)
	LHD	0.341(0.069)	0.591(0.069)	1.025(0.080)
	LHS	0.334(0.072)	0.581(0.075)	1.000(0.086)
$10k$	SphereP	*0.737*(0.003)	*1.454*(0.006)	*2.372*(0.002)
	MmLHD	0.533(0.005)	0.864(0.008)	1.267(0.014)
	MaxPro	0.371(0.016)	0.652(0.030)	1.035(0.042)
	UniDOE	**0.308**(0.025)	**0.570**(0.037)	**1.014**(0.039)
	LHD	0.188(0.040)	0.430(0.056)	0.848(0.065)
	LHS	0.189(0.046)	0.421(0.057)	0.857(0.056)

R UniDOE uniform designs are dominated (in terms of Mm distance), which changes for $k = 20$ and $n = k+1$ to the JMP MmLHD.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. CORRELATION TABLES

Correlation data for each pair of ten measures for six design classes and nine sizes.

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

k			5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.808	0.921	0.994	0.997	0.38	0.476
		$(ML_2)^2$	0.285	-0.464	0.72	0.477	0.381	0.409
		$(CL_2)^2$	0.581	-0.732	0.725	0.997	0.517	0.426
	ρ_{map}	Cov	0.165	0.865	-0.214	0.918	0.214	0.116
		MaxPro	0.373	0.874	NA	0.001	0.388	0.228
		$\phi_{p=15}$	0.775	0.913	0.242	0.996	0.37	0.36
		$\phi_{p=50}$	0.632	0.908	0.552	0.992	0.36	0.343
		Mm	-0.567	-0.928	-0.614	-0.988	-0.392	-0.34
		$\overline{\gamma}$	-0.749	-0.955	-0.682	-0.996	-0.49	-0.54
		ρ_{map}	0.808	0.921	0.994	0.997	0.38	0.476
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	0.249	-0.479	0.754	0.478	0.354	0.333
		$(CL_2)^2$	0.68	-0.867	0.767	0.997	0.757	0.539
	$\overline{\lvert\rho\rvert}$	Cov	0.209	0.916	-0.297	0.895	0.055	0.167
		MaxPro	0.468	0.927	NA	-0.027	0.449	0.355
		$\phi_{p=15}$	0.825	0.96	0.145	0.991	0.358	0.593
		$\phi_{p=50}$	0.676	0.957	0.466	0.983	0.34	0.578
		Mm	-0.617	-0.956	-0.532	-0.978	-0.373	-0.547
		$\overline{\gamma}$	-0.768	-0.977	-0.606	-0.994	-0.707	-0.764
		ρ_{map}	0.285	-0.464	0.72	0.477	0.381	0.409
$k+1$		$\overline{\lvert\rho\rvert}$	0.249	-0.479	0.754	0.478	0.354	0.333
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	0.217	0.518	0.801	0.481	0.224	0.561
	$(ML_2)^2$	Cov	-0.187	-0.441	-0.412	0.441	0.028	0.013
		MaxPro	0.196	-0.524	NA	0.022	0.331	0.238
		$\phi_{p=15}$	0.068	-0.496	-0.217	0.471	0.267	0.177
		$\phi_{p=50}$	0.003	-0.48	0.115	0.47	0.25	0.17
		Mm	0.001	0.475	-0.189	-0.471	-0.258	-0.157
		$\overline{\gamma}$	-0.067	0.504	-0.256	-0.472	-0.454	-0.23
		ρ_{map}	0.581	-0.732	0.725	0.997	0.517	0.426
		$\overline{\lvert\rho\rvert}$	0.68	-0.867	0.767	0.997	0.757	0.539
		$(ML_2)^2$	0.217	0.518	0.801	0.481	0.224	0.561
		$(CL_2)^2$	1	1	1	1	1	1
	$(CL_2)^2$	Cov	0.138	-0.692	-0.492	0.918	0.352	0.116
		MaxPro	0.803	-0.926	NA	0.026	0.63	0.437
		$\phi_{p=15}$	0.571	-0.844	-0.354	0.989	0.525	0.253
		$\phi_{p=50}$	0.418	-0.812	0.079	0.986	0.503	0.238
		Mm	-0.361	0.766	-0.163	-0.985	-0.52	-0.217
		$\overline{\gamma}$	-0.444	0.835	-0.24	-0.99	-0.632	-0.349

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	0.165	0.865	-0.214	0.918	0.214	0.116
		$\overline{\lvert\rho\rvert}$	0.209	0.916	-0.297	0.895	0.055	0.167
		$(ML_2)^2$	-0.187	-0.441	-0.412	0.441	0.028	0.013
		$(CL_2)^2$	0.138	-0.692	-0.492	0.918	0.352	0.116
	Cov	Cov	1	1	1	1	1	1
		MaxPro	-0.019	0.765	NA	0.329	0.748	0.387
		$\phi_{p=15}$	0.551	0.947	0.661	0.921	0.859	0.76
		$\phi_{p=50}$	0.738	0.969	0.509	0.948	0.863	0.766
		Mm	-0.784	-0.981	-0.516	-0.966	-0.852	-0.789
		$\overline{\gamma}$	-0.512	-0.935	-0.41	-0.9	-0.399	-0.37
		ρ_{map}	0.373	0.874	NA	0.001	0.388	0.228
		$\overline{\lvert\rho\rvert}$	0.468	0.927	NA	-0.027	0.449	0.355
		$(ML_2)^2$	0.196	-0.524	NA	0.022	0.331	0.238
		$(CL_2)^2$	0.803	-0.926	NA	0.026	0.63	0.437
	MaxPro	Cov	-0.019	0.765	NA	0.329	0.748	0.387
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	0.298	0.871	NA	-0.03	0.918	0.486
		$\phi_{p=50}$	0.179	0.852	NA	0.038	0.907	0.467
		Mm	-0.151	-0.846	NA	-0.112	-0.872	-0.406
$k+1$		$\overline{\gamma}$	-0.219	-0.916	NA	0.073	-0.738	-0.409
		ρ_{map}	0.775	0.913	0.242	0.996	0.37	0.36
		$\overline{\lvert\rho\rvert}$	0.825	0.96	0.145	0.991	0.358	0.593
		$(ML_2)^2$	0.068	-0.496	-0.217	0.471	0.267	0.177
		$(CL_2)^2$	0.571	-0.844	-0.354	0.989	0.525	0.253
	$\phi_{p=15}$	Cov	0.551	0.947	0.661	0.921	0.859	0.76
		MaxPro	0.298	0.871	NA	-0.03	0.918	0.486
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.938	0.996	0.9	0.997	0.997	0.998
		Mm	-0.862	-0.978	-0.854	-0.99	-0.967	-0.963
		$\overline{\gamma}$	-0.949	-0.983	-0.818	-0.998	-0.732	-0.806
		ρ_{map}	0.632	0.908	0.552	0.992	0.36	0.343
		$\overline{\lvert\rho\rvert}$	0.676	0.957	0.466	0.983	0.34	0.578
		$(ML_2)^2$	0.003	-0.48	0.115	0.47	0.25	0.17
		$(CL_2)^2$	0.418	-0.812	0.079	0.986	0.503	0.238
	$\phi_{p=50}$	Cov	0.738	0.969	0.509	0.948	0.863	0.766
		MaxPro	0.179	0.852	NA	0.038	0.907	0.467
		$\phi_{p=15}$	0.938	0.996	0.9	0.997	0.997	0.998
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.971	-0.99	-0.99	-0.997	-0.977	-0.973
		$\overline{\gamma}$	-0.938	-0.981	-0.983	-0.991	-0.722	-0.793

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.567	-0.928	-0.614	-0.988	-0.392	-0.34
		$\overline{\lvert\rho\rvert}$	-0.617	-0.956	-0.532	-0.978	-0.373	-0.547
		$(ML_2)^2$	0.001	0.475	-0.189	-0.471	-0.258	-0.157
		$(CL_2)^2$	-0.361	0.766	-0.163	-0.985	-0.52	-0.217
	Mm	Cov	-0.784	-0.981	-0.516	-0.966	-0.852	-0.789
		MaxPro	-0.151	-0.846	NA	-0.112	-0.872	-0.406
		$\phi_{p=15}$	-0.862	-0.978	-0.854	-0.99	-0.967	-0.963
		$\phi_{p=50}$	-0.971	-0.99	-0.99	-0.997	-0.977	-0.973
		Mm	1	1	1	1	1	1
k | 1		$\overline{\gamma}$	0.9	0.982	0.993	0.981	0.756	0.784
		ρ_{map}	-0.749	-0.955	-0.682	-0.996	-0.49	-0.54
		$\overline{\lvert\rho\rvert}$	-0.768	-0.977	-0.606	-0.994	-0.707	-0.764
		$(ML_2)^2$	-0.067	0.504	-0.256	-0.472	-0.454	-0.23
		$(CL_2)^2$	-0.444	0.835	-0.24	-0.99	-0.632	-0.349
	$\overline{\gamma}$	Cov	-0.512	-0.935	-0.41	-0.9	-0.399	-0.37
		MaxPro	-0.219	-0.916	NA	0.073	-0.738	-0.409
		$\phi_{p=15}$	-0.949	-0.983	-0.818	-0.998	-0.732	-0.806
		$\phi_{p=50}$	-0.938	-0.981	-0.983	-0.991	-0.722	-0.793
		Mm	0.9	0.982	0.993	0.981	0.756	0.784
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
$3k+2$	ρ_{map}	ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.584	0.607	0.778	0.532	0.719	0.471
		$(ML_2)^2$	0.344	0.102	0.266	0.21	0.651	0.313
		$(CL_2)^2$	0.468	0.154	0.387	0.383	0.783	0.586
		Cov	-0.045	-0.06	-0.191	0.045	0.106	0.013
		MaxPro	-0.126	0.074	NA	0.043	0.167	-0.042
		$\phi_{p=15}$	0.289	-0.009	0.169	-0.009	0.232	0.054
		$\phi_{p=50}$	0.017	0.03	0.257	-0.017	0.228	0.048
		Mm	0.066	-0.049	-0.23	0.008	-0.267	-0.047
		$\overline{\gamma}$	-0.098	0.022	-0.29	-0.016	-0.491	-0.243
	$\overline{\lvert\rho\rvert}$	ρ_{map}	0.584	0.607	0.778	0.532	0.719	0.471
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	0.441	0.042	0.385	0.233	0.694	0.458
		$(CL_2)^2$	0.516	-0.001	0.487	0.631	0.858	0.639
		Cov	0.018	0.01	-0.132	0.196	0.027	0.083
		MaxPro	0.07	0.323	NA	0.05	0.15	-0.023
		$\phi_{p=15}$	0.331	0.098	0.127	0.117	0.2	0.107
		$\phi_{p=50}$	0.077	0.111	0.238	0.106	0.198	0.104
		Mm	0.013	-0.125	-0.21	-0.123	-0.222	-0.131
		$\overline{\gamma}$	-0.086	-0.081	-0.262	-0.052	-0.49	-0.36
	$(ML_2)^2$	ρ_{map}	0.344	0.102	0.266	0.21	0.651	0.313
		$\overline{\lvert\rho\rvert}$	0.441	0.042	0.385	0.233	0.694	0.458
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	0.661	0.48	0.875	0.119	0.726	0.585
		Cov	-0.09	-0.053	0.089	-0.092	0.107	0.118
		MaxPro	0.262	-0.016	NA	0.062	0.168	0.031
		$\phi_{p=15}$	0.213	-0.173	-0.693	-0.014	0.25	0.143
		$\phi_{p=50}$	-0.045	-0.177	-0.542	0.004	0.247	0.152
		Mm	0.125	0.186	0.505	-0.022	-0.288	-0.19
		$\overline{\gamma}$	-0.005	0.171	0.51	-0.076	-0.606	-0.315
	$(CL_2)^2$	ρ_{map}	0.468	0.154	0.387	0.383	0.783	0.586
		$\overline{\lvert\rho\rvert}$	0.516	-0.001	0.487	0.631	0.858	0.639
		$(ML_2)^2$	0.661	0.48	0.875	0.119	0.726	0.585
		$(CL_2)^2$	1	1	1	1	1	1
		Cov	-0.076	0.197	0.119	0.211	0.229	0.048
		MaxPro	0.344	-0.251	NA	0.117	0.285	-0.038
		$\phi_{p=15}$	0.355	0.016	-0.525	0.198	0.317	0.069
		$\phi_{p=50}$	0.075	0.04	-0.35	0.191	0.31	0.076
		Mm	0.022	-0.021	0.317	-0.225	-0.353	-0.105
		$\overline{\gamma}$	-0.116	0.166	0.332	-0.222	-0.658	-0.263

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.045	-0.06	-0.191	0.045	0.106	0.013
		$\overline{\lvert\rho\rvert}$	0.018	0.01	-0.132	0.196	0.027	0.083
		$(ML_2)^2$	-0.09	-0.053	0.089	-0.092	0.107	0.118
		$(CL_2)^2$	-0.076	0.197	0.119	0.211	0.229	0.048
	Cov	Cov	1	1	1	1	1	1
		MaxPro	-0.216	-0.052	NA	0.119	0.63	0.312
		$\phi_{p=15}$	0.123	0.686	-0.183	0.745	0.773	0.762
		$\phi_{p=50}$	0.648	0.683	-0.06	0.711	0.769	0.753
		Mm	-0.713	-0.666	-0.079	-0.702	-0.817	-0.805
		$\overline{\gamma}$	-0.009	-0.282	0.186	-0.277	-0.453	-0.45
		ρ_{map}	-0.126	0.074	NA	0.043	0.167	-0.042
		$\overline{\lvert\rho\rvert}$	0.07	0.323	NA	0.05	0.15	-0.023
		$(ML_2)^2$	0.262	-0.016	NA	0.062	0.168	0.031
		$(CL_2)^2$	0.344	-0.251	NA	0.117	0.285	-0.038
	MaxPro	Cov	-0.216	-0.052	NA	0.119	0.63	0.312
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	0.03	0.045	NA	0.371	0.835	0.425
		$\phi_{p=50}$	-0.098	0.03	NA	0.356	0.828	0.423
		Mm	0.129	-0.054	NA	-0.348	-0.765	-0.374
$3k+2$		$\overline{\gamma}$	-0.085	-0.124	NA	-0.237	-0.47	-0.114
		ρ_{map}	0.289	-0.009	0.169	-0.009	0.232	0.054
		$\overline{\lvert\rho\rvert}$	0.331	0.098	0.127	0.117	0.2	0.107
		$(ML_2)^2$	0.213	-0.173	-0.693	-0.014	0.25	0.143
		$(CL_2)^2$	0.355	0.016	-0.525	0.198	0.317	0.069
	$\phi_{p=15}$	Cov	0.123	0.686	-0.183	0.745	0.773	0.762
		MaxPro	0.03	0.045	NA	0.371	0.835	0.425
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.659	0.958	0.955	0.963	0.998	0.998
		Mm	-0.439	-0.938	-0.898	-0.94	-0.95	-0.932
		$\overline{\gamma}$	-0.836	-0.829	-0.941	-0.608	-0.502	-0.463
		ρ_{map}	0.017	0.03	0.257	-0.017	0.228	0.048
		$\overline{\lvert\rho\rvert}$	0.077	0.111	0.238	0.106	0.198	0.104
		$(ML_2)^2$	-0.045	-0.177	-0.542	0.004	0.247	0.152
		$(CL_2)^2$	0.075	0.04	-0.35	0.191	0.31	0.076
	$\phi_{p=50}$	Cov	0.648	0.683	-0.06	0.711	0.769	0.753
		MaxPro	-0.098	0.03	NA	0.356	0.828	0.423
		$\phi_{p=15}$	0.659	0.958	0.955	0.963	0.998	0.998
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.918	-0.994	-0.972	-0.992	-0.96	-0.944
		$\overline{\gamma}$	-0.617	-0.699	-0.985	-0.463	-0.49	-0.454

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	0.066	-0.049	-0.23	0.008	-0.267	-0.047
		$\overline{\lvert\rho\rvert}$	0.013	-0.125	-0.21	-0.123	-0.222	-0.131
		$(ML_2)^2$	0.125	0.186	0.505	-0.022	-0.288	-0.19
		$(CL_2)^2$	0.022	-0.021	0.317	-0.225	-0.353	-0.105
	Mm	Cov	-0.713	-0.666	-0.079	-0.702	-0.817	-0.805
		MaxPro	0.129	-0.054	NA	-0.348	-0.765	-0.374
		$\phi_{p=15}$	-0.439	-0.938	-0.898	-0.94	-0.95	-0.932
		$\phi_{p=50}$	-0.918	-0.994	-0.972	-0.992	-0.96	-0.944
		Mm	1	1	1	1	1	1
$3k+2$		$\overline{\gamma}$	0.455	0.68	0.957	0.46	0.575	0.553
		ρ_{map}	-0.098	0.022	-0.29	-0.016	-0.491	-0.243
		$\overline{\lvert\rho\rvert}$	-0.086	-0.081	-0.262	-0.052	-0.49	-0.36
		$(ML_2)^2$	-0.005	0.171	0.51	-0.076	-0.606	-0.315
		$(CL_2)^2$	-0.116	0.166	0.332	-0.222	-0.658	-0.263
	$\overline{\gamma}$	Cov	-0.009	-0.282	0.186	-0.277	-0.453	-0.45
		MaxPro	-0.085	-0.124	NA	-0.237	-0.47	-0.114
		$\phi_{p=15}$	-0.836	-0.829	-0.941	-0.608	-0.502	-0.463
		$\phi_{p=50}$	-0.617	-0.699	-0.985	-0.463	-0.49	-0.454
		Mm	0.455	0.68	0.957	0.46	0.575	0.553
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.574	0.619	0.67	0.691	0.571	0.59
		$(ML_2)^2$	0.198	-0.018	-0.004	0.238	0.656	0.465
		$(CL_2)^2$	0.467	0.081	0.029	0.372	0.7	0.602
	ρ_{map}	Cov	0.006	0.155	-0.049	-0.025	0.025	-0.232
		MaxPro	0.028	0.258	NA	0.15	0.033	-0.01
		$\phi_{p=15}$	0.188	0.256	0.077	-0.001	0.006	-0.182
		$\phi_{p=50}$	0.075	0.099	0.066	-0.005	-0.003	-0.186
		Mm	0.037	-0.08	-0.088	0.01	0.027	0.259
		$\overline{\gamma}$	-0.07	-0.36	-0.078	0.021	-0.219	0.081
		ρ_{map}	0.574	0.619	0.67	0.691	0.571	0.59
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	0.408	-0.21	-0.058	0.182	0.757	0.716
		$(CL_2)^2$	0.593	0.02	-0.001	0.408	0.847	0.806
	$\overline{\lvert\rho\rvert}$	Cov	-0.073	0.108	0.059	-0.084	0.118	-0.089
		MaxPro	0.069	0.209	NA	0.181	0.014	-0.023
		$\phi_{p=15}$	0.135	0.148	0.092	0.021	0.055	-0.058
		$\phi_{p=50}$	-0.015	0.068	0.056	0.016	0.048	-0.063
$10k$		Mm	0.068	-0.057	-0.063	-0.02	-0.05	0.093
		$\overline{\gamma}$	-0.013	-0.15	-0.015	-0.077	-0.361	-0.017
		ρ_{map}	0.198	-0.018	-0.004	0.238	0.656	0.465
		$\overline{\lvert\rho\rvert}$	0.408	-0.21	-0.058	0.182	0.757	0.716
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	0.764	0.543	0.917	0.495	0.877	0.741
	$(ML_2)^2$	Cov	-0.08	0.136	0.298	-0.059	0.165	0.02
		MaxPro	0.209	-0.041	NA	0.129	0.08	0.039
		$\phi_{p=15}$	0.033	-0.005	-0.77	0.126	0.054	0.033
		$\phi_{p=50}$	-0.108	-0.061	-0.529	0.121	0.05	0.026
		Mm	0.187	0.061	0.33	-0.11	-0.072	-0.003
		$\overline{\gamma}$	0.028	-0.065	0.474	-0.188	-0.477	-0.209
		ρ_{map}	0.467	0.081	0.029	0.372	0.7	0.602
		$\overline{\lvert\rho\rvert}$	0.593	0.02	-0.001	0.408	0.847	0.806
		$(ML_2)^2$	0.764	0.543	0.917	0.495	0.877	0.741
		$(CL_2)^2$	1	1	1	1	1	1
	$(CL_2)^2$	Cov	0.029	0.269	0.388	0.015	0.183	0.023
		MaxPro	0.182	0.019	NA	0.185	0.068	0.026
		$\phi_{p=15}$	0.162	0.035	-0.775	0.118	0.083	0.019
		$\phi_{p=50}$	0.028	-0.028	-0.508	0.075	0.076	0.01
		Mm	0.07	0.058	0.299	-0.068	-0.095	0.026
		$\overline{\gamma}$	-0.043	0.062	0.49	-0.294	-0.556	-0.187

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	0.006	0.155	-0.049	-0.025	0.025	-0.232
		$\overline{\lvert\rho\rvert}$	-0.073	0.108	0.059	-0.084	0.118	-0.089
		$(ML_2)^2$	-0.08	0.136	0.298	-0.059	0.165	0.02
		$(CL_2)^2$	0.029	0.269	0.388	0.015	0.183	0.023
	Cov	Cov	1	1	1	1	1	1
		MaxPro	0.073	0.083	NA	0.191	0.37	0.325
		$\phi_{p=15}$	0.059	0.626	-0.346	0.625	0.521	0.535
		$\phi_{p=50}$	0.499	0.458	-0.178	0.558	0.511	0.522
		Mm	-0.531	-0.419	0.024	-0.561	-0.573	-0.635
		$\overline{\gamma}$	0.069	-0.199	0.434	-0.228	-0.43	-0.491
		ρ_{map}	0.028	0.258	NA	0.15	0.033	-0.01
		$\overline{\lvert\rho\rvert}$	0.069	0.209	NA	0.181	0.014	-0.023
		$(ML_2)^2$	0.209	-0.041	NA	0.129	0.08	0.039
		$(CL_2)^2$	0.182	0.019	NA	0.185	0.068	0.026
	MaxPro	Cov	0.073	0.083	NA	0.191	0.37	0.325
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	-0.005	0.222	NA	0.365	0.712	0.454
		$\phi_{p=50}$	-0.063	0.144	NA	0.361	0.705	0.451
		Mm	0.017	-0.132	NA	-0.351	-0.666	-0.381
$10k$		$\overline{\gamma}$	0.074	-0.173	NA	-0.299	-0.201	-0.156
		ρ_{map}	0.188	0.256	0.077	-0.001	0.006	-0.182
		$\overline{\lvert\rho\rvert}$	0.135	0.148	0.092	0.021	0.055	-0.058
		$(ML_2)^2$	0.033	-0.005	-0.77	0.126	0.054	0.033
		$(CL_2)^2$	0.162	0.035	-0.775	0.118	0.083	0.019
	$\phi_{p=15}$	Cov	0.059	0.626	-0.346	0.625	0.521	0.535
		MaxPro	-0.005	0.222	NA	0.365	0.712	0.454
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.763	0.867	0.906	0.978	0.999	0.999
		Mm	-0.252	-0.822	-0.736	-0.962	-0.945	-0.923
		$\overline{\gamma}$	-0.846	-0.631	-0.863	-0.509	-0.223	-0.448
		ρ_{map}	0.075	0.099	0.066	-0.005	-0.003	-0.186
		$\overline{\lvert\rho\rvert}$	-0.015	0.068	0.056	0.016	0.048	-0.063
		$(ML_2)^2$	-0.108	-0.061	-0.529	0.121	0.05	0.026
		$(CL_2)^2$	0.028	-0.028	-0.508	0.075	0.076	0.01
	$\phi_{p=50}$	Cov	0.499	0.458	-0.178	0.558	0.511	0.522
		MaxPro	-0.063	0.144	NA	0.361	0.705	0.451
		$\phi_{p=15}$	0.763	0.867	0.906	0.978	0.999	0.999
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.695	-0.992	-0.923	-0.993	-0.953	-0.93
		$\overline{\gamma}$	-0.755	-0.29	-0.946	-0.423	-0.217	-0.438

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		5					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	0.037	-0.08	-0.088	0.01	0.027	0.259
		$\overline{\lvert\rho\rvert}$	0.068	-0.057	-0.063	-0.02	-0.05	0.093
		$(ML_2)^2$	0.187	0.061	0.33	-0.11	-0.072	-0.003
		$(CL_2)^2$	0.07	0.058	0.299	-0.068	-0.095	0.026
	Mm	Cov	-0.531	-0.419	0.024	-0.561	-0.573	-0.635
		MaxPro	0.017	-0.132	NA	-0.351	-0.666	-0.381
		$\phi_{p=15}$	-0.252	-0.822	-0.736	-0.962	-0.945	-0.923
		$\phi_{p=50}$	-0.695	-0.992	-0.923	-0.993	-0.953	-0.93
		Mm	1	1	1	1	1	1
$10k$		$\overline{\gamma}$	0.323	0.256	0.869	0.425	0.309	0.48
		ρ_{map}	-0.07	-0.36	-0.078	0.021	-0.219	0.081
		$\overline{\lvert\rho\rvert}$	-0.013	-0.15	-0.015	-0.077	-0.361	-0.017
		$(ML_2)^2$	0.028	-0.065	0.474	-0.188	-0.477	-0.209
		$(CL_2)^2$	-0.043	0.062	0.49	-0.294	-0.556	-0.187
	$\overline{\gamma}$	Cov	0.069	-0.199	0.434	-0.228	-0.43	-0.491
		MaxPro	0.074	-0.173	NA	-0.299	-0.201	-0.156
		$\phi_{p=15}$	-0.846	-0.631	-0.863	-0.509	-0.223	-0.448
		$\phi_{p=50}$	-0.755	-0.29	-0.946	-0.423	-0.217	-0.438
		Mm	0.323	0.256	0.869	0.425	0.309	0.48
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.295	0.052	0.752	0.178	0.407	0.414
		$(ML_2)^2$	0.005	0.172	-0.026	-0.012	0.118	0.202
		$(CL_2)^2$	0.28	-0.169	-0.144	0.274	0.386	0.3
	ρ_{map}	Cov	0.009	0.108	-0.055	-0.093	0.088	0.045
		MaxPro	0	0.138	NA	-0.037	0.3	-0.032
		$\phi_{p=15}$	0.357	0.218	0.438	0.277	0.222	0.204
		$\phi_{p=50}$	0.268	0.168	0.49	0.111	0.2	0.2
		Mm	-0.251	-0.171	-0.483	-0.05	-0.19	-0.205
		$\overline{\gamma}$	-0.365	-0.234	-0.485	-0.279	-0.395	-0.361
		ρ_{map}	0.295	0.052	0.752	0.178	0.407	0.414
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	0.258	-0.066	0.28	-0.132	0.182	0.118
		$(CL_2)^2$	0.595	-0.399	0.086	0.413	0.676	0.455
	$\overline{\lvert\rho\rvert}$	Cov	-0.238	0.083	-0.03	-0.231	0.227	0.262
		MaxPro	0.127	0.645	NA	0.191	0.325	0.183
		$\phi_{p=15}$	0.648	0.471	0.202	0.441	0.456	0.453
		$\phi_{p=50}$	0.38	0.327	0.243	0.125	0.443	0.429
$k+1$		Mm	-0.271	-0.33	-0.22	-0.056	-0.455	-0.451
		$\overline{\gamma}$	-0.617	-0.502	-0.219	-0.511	-0.691	-0.649
		ρ_{map}	0.005	0.172	-0.026	-0.012	0.118	0.202
		$\overline{\lvert\rho\rvert}$	0.258	-0.066	0.28	-0.132	0.182	0.118
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	0.25	0.096	0.793	-0.186	0.044	0.016
	$(ML_2)^2$	Cov	0.005	-0.006	0.103	-0.097	0.061	0.113
		MaxPro	0.231	-0.117	NA	-0.072	0.223	0.072
		$\phi_{p=15}$	0.247	-0.085	-0.689	-0.14	0.161	0.168
		$\phi_{p=50}$	0.228	-0.055	-0.647	-0.127	0.171	0.165
		Mm	-0.171	0.042	0.629	0.102	-0.157	-0.164
		$\overline{\gamma}$	-0.289	0.074	0.636	0.176	-0.153	-0.15
		ρ_{map}	0.28	-0.169	-0.144	0.274	0.386	0.3
		$\overline{\lvert\rho\rvert}$	0.595	-0.399	0.086	0.413	0.676	0.455
		$(ML_2)^2$	0.25	0.096	0.793	-0.186	0.044	0.016
		$(CL_2)^2$	1	1	1	1	1	1
	$(CL_2)^2$	Cov	0.127	0.128	0.037	0.113	0.296	0.152
		MaxPro	0.221	-0.417	NA	0.504	0.341	0.152
		$\phi_{p=15}$	0.744	-0.133	-0.838	0.591	0.392	0.154
		$\phi_{p=50}$	0.473	-0.031	-0.768	0.363	0.361	0.143
		Mm	-0.329	0.026	0.718	-0.318	-0.372	-0.15
		$\overline{\gamma}$	-0.602	0.261	0.718	-0.491	-0.611	-0.169

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	0.009	0.108	-0.055	-0.093	0.088	0.045
		$\overline{\lvert\rho\rvert}$	-0.238	0.083	-0.03	-0.231	0.227	0.262
		$(ML_2)^2$	0.005	-0.006	0.103	-0.097	0.061	0.113
		$(CL_2)^2$	0.127	0.128	0.037	0.113	0.296	0.152
	Cov	Cov	1	1	1	1	1	1
		MaxPro	0.048	-0.052	NA	0.101	0.645	0.26
		$\phi_{p=15}$	0.316	0.768	-0.02	0.478	0.871	0.861
		$\phi_{p=50}$	0.621	0.876	-0.012	0.739	0.87	0.867
		Mm	-0.643	-0.88	-0.067	-0.767	-0.879	-0.874
		$\overline{\gamma}$	-0.197	-0.473	0.004	-0.167	-0.58	-0.488
		ρ_{map}	0	0.138	NA	-0.037	0.3	-0.032
		$\overline{\lvert\rho\rvert}$	0.127	0.645	NA	0.191	0.325	0.183
		$(ML_2)^2$	0.231	-0.117	NA	-0.072	0.223	0.072
		$(CL_2)^2$	0.221	-0.417	NA	0.504	0.341	0.152
	MaxPro	Cov	0.048	-0.052	NA	0.101	0.645	0.26
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	0.079	0.329	NA	0.443	0.753	0.352
		$\phi_{p=50}$	0.078	0.15	NA	0.25	0.729	0.341
		Mm	-0.1	-0.131	NA	-0.227	-0.671	-0.334
$k+1$		$\overline{\gamma}$	-0.059	-0.353	NA	-0.272	-0.616	-0.277
		ρ_{map}	0.357	0.218	0.438	0.277	0.222	0.204
		$\overline{\lvert\rho\rvert}$	0.648	0.471	0.202	0.441	0.456	0.453
		$(ML_2)^2$	0.247	-0.085	-0.689	-0.14	0.161	0.168
		$(CL_2)^2$	0.744	-0.133	-0.838	0.591	0.392	0.154
	$\phi_{p=15}$	Cov	0.316	0.768	-0.02	0.478	0.871	0.861
		MaxPro	0.079	0.329	NA	0.443	0.753	0.352
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.825	0.936	0.98	0.861	0.994	0.991
		Mm	-0.663	-0.901	-0.943	-0.786	-0.948	-0.959
		$\overline{\gamma}$	-0.886	-0.84	-0.945	-0.871	-0.76	-0.672
		ρ_{map}	0.268	0.168	0.49	0.111	0.2	0.2
		$\overline{\lvert\rho\rvert}$	0.38	0.327	0.243	0.125	0.443	0.429
		$(ML_2)^2$	0.228	-0.055	-0.647	-0.127	0.171	0.165
		$(CL_2)^2$	0.473	-0.031	-0.768	0.363	0.361	0.143
	$\phi_{p=50}$	Cov	0.621	0.876	-0.012	0.739	0.87	0.867
		MaxPro	0.078	0.15	NA	0.25	0.729	0.341
		$\phi_{p=15}$	0.825	0.936	0.98	0.861	0.994	0.991
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.944	-0.989	-0.985	-0.979	-0.971	-0.982
		$\overline{\gamma}$	-0.829	-0.696	-0.988	-0.689	-0.741	-0.638

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.251	-0.171	-0.483	-0.05	-0.19	-0.205
		$\overline{\lvert\rho\rvert}$	-0.271	-0.33	-0.22	-0.056	-0.455	-0.451
		$(ML_2)^2$	-0.171	0.042	0.629	0.102	-0.157	-0.164
		$(CL_2)^2$	-0.329	0.026	0.718	-0.318	-0.372	-0.15
	Mm	Cov	-0.643	-0.88	-0.067	-0.767	-0.879	-0.874
		MaxPro	-0.1	-0.131	NA	-0.227	-0.671	-0.334
		$\phi_{p=15}$	-0.663	-0.901	-0.943	-0.786	-0.948	-0.959
		$\phi_{p=50}$	-0.944	-0.989	-0.985	-0.979	-0.971	-0.982
		Mm	1	1	1	1	1	1
k | 1		$\overline{\gamma}$	0.723	0.673	0.997	0.608	0.762	0.672
		ρ_{map}	-0.365	-0.234	-0.485	-0.279	-0.395	-0.361
		$\overline{\lvert\rho\rvert}$	-0.617	-0.502	-0.219	-0.511	-0.691	-0.649
		$(ML_2)^2$	-0.289	0.074	0.636	0.176	-0.153	-0.15
		$(CL_2)^2$	-0.602	0.261	0.718	-0.491	-0.611	-0.169
	$\overline{\gamma}$	Cov	-0.197	-0.473	0.004	-0.167	-0.58	-0.488
		MaxPro	-0.059	-0.353	NA	-0.272	-0.616	-0.277
		$\phi_{p=15}$	-0.886	-0.84	-0.945	-0.871	-0.76	-0.672
		$\phi_{p=50}$	-0.829	-0.696	-0.988	-0.689	-0.741	-0.638
		Mm	0.723	0.673	0.997	0.608	0.762	0.672
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
$3k+2$	ρ_{map}	ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.312	0.35	0.467	0.465	0.249	0.467
		$(ML_2)^2$	0.006	-0.025	0.264	0.131	0.2	0.293
		$(CL_2)^2$	0.296	-0.077	0.318	0.255	0.284	0.458
		Cov	0.092	-0.028	-0.011	0.13	-0.019	-0.098
		MaxPro	-0.031	0.333	NA	-0.002	0.081	-0.071
		$\phi_{p=15}$	0.1	0.119	-0.077	0.079	-0.075	-0.001
		$\phi_{p=50}$	0.157	0.017	0.016	0.085	-0.066	0.015
		Mm	-0.239	0.026	-0.038	-0.083	0.029	-0.053
		$\overline{\gamma}$	-0.121	-0.18	-0.048	0.031	-0.205	-0.244
	$\overline{\lvert\rho\rvert}$	ρ_{map}	0.312	0.35	0.467	0.465	0.249	0.467
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	0.286	-0.077	0.235	0.124	0.47	0.445
		$(CL_2)^2$	0.596	0.012	0.326	0.456	0.695	0.717
		Cov	0.061	0.072	-0.074	0.156	0.16	-0.035
		MaxPro	-0.056	0.377	NA	-0.013	-0.005	-0.039
		$\phi_{p=15}$	0.389	0.165	0.177	0.115	0.1	0.116
		$\phi_{p=50}$	0.139	0.094	0.233	0.13	0.084	0.118
		Mm	-0.074	-0.08	-0.215	-0.121	-0.107	-0.158
		$\overline{\gamma}$	-0.271	-0.172	-0.231	0.013	-0.346	-0.48
	$(ML_2)^2$	ρ_{map}	0.006	-0.025	0.264	0.131	0.2	0.293
		$\overline{\lvert\rho\rvert}$	0.286	-0.077	0.235	0.124	0.47	0.445
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	0.461	0.18	0.752	0.099	0.443	0.339
		Cov	-0.043	-0.108	0.117	-0.007	0.094	0.121
		MaxPro	0.067	-0.274	NA	0.059	-0.035	0.135
		$\phi_{p=15}$	0.253	-0.194	-0.529	0.031	-0.019	0.112
		$\phi_{p=50}$	0.12	-0.196	-0.272	0.063	-0.032	0.118
		Mm	-0.082	0.194	0.168	-0.066	0.008	-0.166
		$\overline{\gamma}$	-0.25	0.186	0.177	0.022	-0.241	-0.393
	$(CL_2)^2$	ρ_{map}	0.296	-0.077	0.318	0.255	0.284	0.458
		$\overline{\lvert\rho\rvert}$	0.596	0.012	0.326	0.456	0.695	0.717
		$(ML_2)^2$	0.461	0.18	0.752	0.099	0.443	0.339
		$(CL_2)^2$	1	1	1	1	1	1
		Cov	0.169	-0.173	0.011	0.225	0.299	0.103
		MaxPro	0.173	-0.172	NA	0.009	0.075	0.087
		$\phi_{p=15}$	0.53	-0.308	-0.544	0.032	0.191	0.151
		$\phi_{p=50}$	0.293	-0.274	-0.269	0.041	0.17	0.151
		Mm	-0.127	0.251	0.181	-0.04	-0.183	-0.211
		$\overline{\gamma}$	-0.393	0.283	0.173	-0.07	-0.307	-0.462

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
$3k+2$	Cov	ρ_{map}	0.092	-0.028	-0.011	0.13	-0.019	-0.098
		$\overline{\lvert\rho\rvert}$	0.061	0.072	-0.074	0.156	0.16	-0.035
		$(ML_2)^2$	-0.043	-0.108	0.117	-0.007	0.094	0.121
		$(CL_2)^2$	0.169	-0.173	0.011	0.225	0.299	0.103
		Cov	1	1	1	1	1	1
		MaxPro	-0.009	0.274	NA	-0.047	0.253	0.32
		$\phi_{p=15}$	0.333	0.802	-0.083	0.721	0.698	0.768
		$\phi_{p=50}$	0.681	0.793	-0.017	0.654	0.654	0.752
		Mm	-0.595	-0.776	-0.101	-0.613	-0.683	-0.787
		$\overline{\gamma}$	-0.17	-0.53	0.029	-0.165	-0.405	-0.489
	MaxPro	ρ_{map}	-0.031	0.333	NA	-0.002	0.081	-0.071
		$\overline{\lvert\rho\rvert}$	-0.056	0.377	NA	-0.013	-0.005	-0.039
		$(ML_2)^2$	0.067	-0.274	NA	0.059	-0.035	0.135
		$(CL_2)^2$	0.173	-0.172	NA	0.009	0.075	0.087
		Cov	-0.009	0.274	NA	-0.047	0.253	0.32
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	-0.04	0.321	NA	0.058	0.416	0.395
		$\phi_{p=50}$	0.069	0.237	NA	0.016	0.417	0.376
		Mm	-0.092	-0.214	NA	-0.028	-0.405	-0.339
		$\overline{\gamma}$	-0.018	-0.262	NA	-0.202	-0.235	-0.264
	$\phi_{p=15}$	ρ_{map}	0.1	0.119	-0.077	0.079	-0.075	-0.001
		$\overline{\lvert\rho\rvert}$	0.389	0.165	0.177	0.115	0.1	0.116
		$(ML_2)^2$	0.253	-0.194	-0.529	0.031	-0.019	0.112
		$(CL_2)^2$	0.53	-0.308	-0.544	0.032	0.191	0.151
		Cov	0.333	0.802	-0.083	0.721	0.698	0.768
		MaxPro	-0.04	0.321	NA	0.058	0.416	0.395
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.696	0.924	0.892	0.921	0.988	0.992
		Mm	-0.268	-0.884	-0.792	-0.881	-0.953	-0.947
		$\overline{\gamma}$	-0.805	-0.852	-0.801	-0.549	-0.283	-0.494
	$\phi_{p=50}$	ρ_{map}	0.157	0.017	0.016	0.085	-0.066	0.015
		$\overline{\lvert\rho\rvert}$	0.139	0.094	0.233	0.13	0.084	0.118
		$(ML_2)^2$	0.12	-0.196	-0.272	0.063	-0.032	0.118
		$(CL_2)^2$	0.293	-0.274	-0.269	0.041	0.17	0.151
		Cov	0.681	0.793	-0.017	0.654	0.654	0.752
		MaxPro	0.069	0.237	NA	0.016	0.417	0.376
		$\phi_{p=15}$	0.696	0.924	0.892	0.921	0.988	0.992
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.798	-0.99	-0.971	-0.99	-0.983	-0.972
		$\overline{\gamma}$	-0.704	-0.691	-0.98	-0.356	-0.228	-0.469

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.239	0.026	-0.038	-0.083	0.029	-0.053
		$\overline{\lvert\rho\rvert}$	-0.074	-0.08	-0.215	-0.121	-0.107	-0.158
		$(ML_2)^2$	-0.082	0.194	0.168	-0.066	0.008	-0.166
		$(CL_2)^2$	-0.127	0.251	0.181	-0.04	-0.183	-0.211
	Mm	Cov	-0.595	-0.776	-0.101	-0.613	-0.683	-0.787
		MaxPro	-0.092	-0.214	NA	-0.028	-0.405	-0.339
		$\phi_{p=15}$	-0.268	-0.884	-0.792	-0.881	-0.953	-0.947
		$\phi_{p=50}$	-0.798	-0.99	-0.971	-0.99	-0.983	-0.972
		Mm	1	1	1	1	1	1
$3k+2$		$\overline{\gamma}$	0.342	0.656	0.989	0.357	0.244	0.53
		ρ_{map}	-0.121	-0.18	-0.048	0.031	-0.205	-0.244
		$\overline{\lvert\rho\rvert}$	-0.271	-0.172	-0.231	0.013	-0.346	-0.48
		$(ML_2)^2$	-0.25	0.186	0.177	0.022	-0.241	-0.393
		$(CL_2)^2$	-0.393	0.283	0.173	-0.07	-0.307	-0.462
	$\overline{\gamma}$	Cov	-0.17	-0.53	0.029	-0.165	-0.405	-0.489
		MaxPro	-0.018	-0.262	NA	-0.202	-0.235	-0.264
		$\phi_{p=15}$	-0.805	-0.852	-0.801	-0.549	-0.283	-0.494
		$\phi_{p=50}$	-0.704	-0.691	-0.98	-0.356	-0.228	-0.469
		Mm	0.342	0.656	0.989	0.357	0.244	0.53
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.378	0.326	0.321	0.413	0.255	0.394
		$(ML_2)^2$	0.316	0.038	0.123	0.078	0.385	0.471
		$(CL_2)^2$	0.386	0.065	0.196	0.151	0.477	0.412
	ρ_{map}	Cov	-0.095	-0.115	0.022	0.052	-0.042	-0.056
		MaxPro	-0.14	0.07	NA	0.128	-0.04	0.224
		$\phi_{p=15}$	0.168	-0.068	-0.09	0.011	0.079	-0.019
		$\phi_{p=50}$	0.022	-0.082	-0.06	0.019	0.118	-0.022
		Mm	0.019	0.08	0.042	-0.011	-0.139	0.017
		$\overline{\gamma}$	-0.1	0.032	0.049	0.091	-0.21	-0.263
		ρ_{map}	0.378	0.326	0.321	0.413	0.255	0.394
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	0.473	0.143	0.075	0.199	0.464	0.68
		$(CL_2)^2$	0.593	0.148	0.128	0.4	0.798	0.851
	$\overline{\lvert\rho\rvert}$	Cov	0.048	-0.082	0.081	0.158	-0.094	-0.081
		MaxPro	0.018	0.265	NA	0.089	0.105	0.146
		$\phi_{p=15}$	0.334	-0.052	0.074	0.161	-0.04	-0.034
		$\phi_{p=50}$	0.21	-0.117	0.101	0.159	-0.024	-0.039
$10k$		Mm	-0.132	0.116	-0.112	-0.165	0.004	0.032
		$\overline{\gamma}$	-0.214	-0.078	-0.099	-0.034	-0.203	-0.401
		ρ_{map}	0.316	0.038	0.123	0.078	0.385	0.471
		$\overline{\lvert\rho\rvert}$	0.473	0.143	0.075	0.199	0.464	0.68
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	0.679	0.415	0.873	0.298	0.623	0.765
	$(ML_2)^2$	Cov	-0.073	-0.156	0.169	0.19	-0.096	0.144
		MaxPro	0.132	0.097	NA	-0.013	0.102	0.142
		$\phi_{p=15}$	0.258	-0.289	-0.784	0.205	-0.04	0.109
		$\phi_{p=50}$	0.132	-0.259	-0.651	0.176	-0.022	0.086
		Mm	-0.037	0.243	0.586	-0.182	-0.001	-0.074
		$\overline{\gamma}$	-0.218	0.347	0.615	-0.187	-0.171	-0.391
		ρ_{map}	0.386	0.065	0.196	0.151	0.477	0.412
		$\overline{\lvert\rho\rvert}$	0.593	0.148	0.128	0.4	0.798	0.851
		$(ML_2)^2$	0.679	0.415	0.873	0.298	0.623	0.765
		$(CL_2)^2$	1	1	1	1	1	1
	$(CL_2)^2$	Cov	-0.001	-0.071	0.094	0.245	0.036	0.069
		MaxPro	0.076	0.17	NA	0.111	0.124	0.221
		$\phi_{p=15}$	0.366	-0.24	-0.763	0.159	0.085	0.051
		$\phi_{p=50}$	0.21	-0.224	-0.613	0.117	0.099	0.046
		Mm	-0.049	0.195	0.553	-0.107	-0.123	-0.066
		$\overline{\gamma}$	-0.254	0.169	0.573	-0.238	-0.35	-0.504

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.095	-0.115	0.022	0.052	-0.042	-0.056
		$\overline{\lvert\rho\rvert}$	0.048	-0.082	0.081	0.158	-0.094	-0.081
		$(ML_2)^2$	-0.073	-0.156	0.169	0.19	-0.096	0.144
		$(CL_2)^2$	-0.001	-0.071	0.094	0.245	0.036	0.069
	Cov	Cov	1	1	1	1	1	1
		MaxPro	0.078	0.075	NA	0.145	0.192	0.095
		$\phi_{p=15}$	0.299	0.829	0.006	0.631	0.689	0.678
		$\phi_{p=50}$	0.581	0.662	0.094	0.521	0.662	0.654
		Mm	-0.44	-0.625	-0.191	-0.527	-0.692	-0.678
		$\overline{\gamma}$	-0.098	-0.489	-0.094	-0.313	-0.388	-0.45
		ρ_{map}	-0.14	0.07	NA	0.128	-0.04	0.224
		$\overline{\lvert\rho\rvert}$	0.018	0.265	NA	0.089	0.105	0.146
		$(ML_2)^2$	0.132	0.097	NA	-0.013	0.102	0.142
		$(CL_2)^2$	0.076	0.17	NA	0.111	0.124	0.221
	MaxPro	Cov	0.078	0.075	NA	0.145	0.192	0.095
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	0.096	0.034	NA	0.28	0.281	0.227
		$\phi_{p=50}$	0.144	-0.007	NA	0.287	0.269	0.219
		Mm	-0.107	-0.009	NA	-0.294	-0.249	-0.21
$10k$		$\overline{\gamma}$	-0.122	-0.102	NA	-0.203	-0.149	-0.245
		ρ_{map}	0.168	-0.068	-0.09	0.011	0.079	-0.019
		$\overline{\lvert\rho\rvert}$	0.334	-0.052	0.074	0.161	-0.04	-0.034
		$(ML_2)^2$	0.258	-0.289	-0.784	0.205	-0.04	0.109
		$(CL_2)^2$	0.366	-0.24	-0.763	0.159	0.085	0.051
	$\phi_{p=15}$	Cov	0.299	0.829	0.006	0.631	0.689	0.678
		MaxPro	0.096	0.034	NA	0.28	0.281	0.227
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.846	0.895	0.962	0.954	0.991	0.992
		Mm	-0.342	-0.853	-0.921	-0.923	-0.962	-0.957
		$\overline{\gamma}$	-0.795	-0.668	-0.936	-0.469	-0.251	-0.285
		ρ_{map}	0.022	-0.082	-0.06	0.019	0.118	-0.022
		$\overline{\lvert\rho\rvert}$	0.21	-0.117	0.101	0.159	-0.024	-0.039
		$(ML_2)^2$	0.132	-0.259	-0.651	0.176	-0.022	0.086
		$(CL_2)^2$	0.21	-0.224	-0.613	0.117	0.099	0.046
	$\phi_{p=50}$	Cov	0.581	0.662	0.094	0.521	0.662	0.654
		MaxPro	0.144	-0.007	NA	0.287	0.269	0.219
		$\phi_{p=15}$	0.846	0.895	0.962	0.954	0.991	0.992
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.588	-0.991	-0.987	-0.989	-0.985	-0.981
		$\overline{\gamma}$	-0.797	-0.403	-0.994	-0.371	-0.227	-0.254

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		10					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	0.019	0.08	0.042	-0.011	-0.139	0.017
		$\overline{\lvert\rho\rvert}$	-0.132	0.116	-0.112	-0.165	0.004	0.032
		$(ML_2)^2$	-0.037	0.243	0.586	-0.182	-0.001	-0.074
		$(CL_2)^2$	-0.049	0.195	0.553	-0.107	-0.123	-0.066
	Mm	Cov	-0.44	-0.625	-0.191	-0.527	-0.692	-0.678
		MaxPro	-0.107	-0.009	NA	-0.294	-0.249	-0.21
		$\phi_{p=15}$	-0.342	-0.853	-0.921	-0.923	-0.962	-0.957
		$\phi_{p=50}$	-0.588	-0.991	-0.987	-0.989	-0.985	-0.981
		Mm	1	1	1	1	1	1
$10k$		$\overline{\gamma}$	0.285	0.365	0.994	0.398	0.259	0.279
		ρ_{map}	-0.1	0.032	0.049	0.091	-0.21	-0.263
		$\overline{\lvert\rho\rvert}$	-0.214	-0.078	-0.099	-0.034	-0.203	-0.401
		$(ML_2)^2$	-0.218	0.347	0.615	-0.187	-0.171	-0.391
		$(CL_2)^2$	-0.254	0.169	0.573	-0.238	-0.35	-0.504
	$\overline{\gamma}$	Cov	-0.098	-0.489	-0.094	-0.313	-0.388	-0.45
		MaxPro	-0.122	-0.102	NA	-0.203	-0.149	-0.245
		$\phi_{p=15}$	-0.795	-0.668	-0.936	-0.469	-0.251	-0.285
		$\phi_{p=50}$	-0.797	-0.403	-0.994	-0.371	-0.227	-0.254
		Mm	0.285	0.365	0.994	0.398	0.259	0.279
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
$k+1$	ρ_{map}	ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.071	0.07	0.399	0.019	0.228	0.254
		$(ML_2)^2$	0.008	-0.079	0.015	-0.173	0.283	-0.138
		$(CL_2)^2$	0.167	-0.048	0.1	0.2	0.27	0.059
		Cov	-0.067	0.048	-0.067	-0.109	0.005	0.121
		MaxPro	-0.06	0.291	NA	-0.02	0.035	0.095
		$\phi_{p=15}$	0.209	0.154	0.077	0.176	-0.033	0.183
		$\phi_{p=50}$	0.092	0.118	0.208	0.037	-0.059	0.156
		Mm	-0.101	-0.122	-0.199	0.041	0.05	-0.166
		$\overline{\gamma}$	-0.191	-0.082	-0.227	-0.218	-0.069	-0.248
	$\overline{\lvert\rho\rvert}$	ρ_{map}	0.071	0.07	0.399	0.019	0.228	0.254
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	-0.057	0.026	0.071	-0.041	0.204	0.244
		$(CL_2)^2$	0.296	-0.319	0.192	0.638	0.303	0.424
		Cov	-0.168	0.206	-0.118	-0.05	0.034	0.148
		MaxPro	0.116	0.43	NA	0.062	0.017	0.23
		$\phi_{p=15}$	0.584	0.547	0.218	0.49	0.175	0.342
		$\phi_{p=50}$	0.216	0.347	0.422	0.245	0.087	0.248
		Mm	-0.102	-0.326	-0.401	-0.125	-0.064	-0.244
		$\overline{\gamma}$	-0.458	-0.481	-0.434	-0.483	-0.442	-0.61
	$(ML_2)^2$	ρ_{map}	0.008	-0.079	0.015	-0.173	0.283	-0.138
		$\overline{\lvert\rho\rvert}$	-0.057	0.026	0.071	-0.041	0.204	0.244
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	-0.02	-0.125	-0.018	-0.062	0.184	0.096
		Cov	0.01	0.003	-0.171	-0.034	0.139	-0.109
		MaxPro	0.091	-0.019	NA	-0.019	-0.015	-0.023
		$\phi_{p=15}$	0.101	-0.011	0.071	-0.081	0.124	-0.038
		$\phi_{p=50}$	0.027	-0.041	0.083	-0.053	0.106	-0.079
		Mm	0.028	0.054	-0.083	0.033	-0.105	0.088
		$\overline{\gamma}$	-0.051	0.001	-0.122	0.078	-0.085	-0.051
	$(CL_2)^2$	ρ_{map}	0.167	-0.048	0.1	0.2	0.27	0.059
		$\overline{\lvert\rho\rvert}$	0.296	-0.319	0.192	0.638	0.303	0.424
		$(ML_2)^2$	-0.02	-0.125	-0.018	-0.062	0.184	0.096
		$(CL_2)^2$	1	1	1	1	1	1
		Cov	0.074	0.056	0.067	-0.018	0.258	0.251
		MaxPro	0.036	-0.353	NA	0.073	-0.017	0.11
		$\phi_{p=15}$	0.564	-0.16	-0.453	0.443	0.193	0.343
		$\phi_{p=50}$	0.215	-0.072	-0.284	0.297	0.103	0.264
		Mm	-0.091	0.046	0.183	-0.192	-0.106	-0.261
		$\overline{\gamma}$	-0.383	0.201	0.188	-0.515	-0.317	-0.441

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.067	0.048	-0.067	-0.109	0.005	0.121
		$\overline{\lvert\rho\rvert}$	-0.168	0.206	-0.118	-0.05	0.034	0.148
		$(ML_2)^2$	0.01	0.003	-0.171	-0.034	0.139	-0.109
		$(CL_2)^2$	0.074	0.056	0.067	-0.018	0.258	0.251
	Cov	Cov	1	1	1	1	1	1
		MaxPro	0.119	0.11	NA	0.158	0.352	0.15
		$\phi_{p=15}$	0.288	0.788	-0.159	0.44	0.831	0.871
		$\phi_{p=50}$	0.724	0.891	-0.121	0.755	0.838	0.871
		Mm	-0.713	-0.876	-0.118	-0.752	-0.85	-0.867
		$\overline{\gamma}$	-0.065	-0.61	0.105	-0.349	-0.575	-0.556
		ρ_{map}	-0.06	0.291	NA	-0.02	0.035	0.095
		$\overline{\lvert\rho\rvert}$	0.116	0.43	NA	0.062	0.017	0.23
		$(ML_2)^2$	0.091	-0.019	NA	-0.019	-0.015	-0.023
		$(CL_2)^2$	0.036	-0.353	NA	0.073	-0.017	0.11
	MaxPro	Cov	0.119	0.11	NA	0.158	0.352	0.15
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	0.145	0.353	NA	0.42	0.395	0.23
		$\phi_{p=50}$	0.105	0.215	NA	0.21	0.357	0.239
		Mm	-0.093	-0.185	NA	-0.171	-0.312	-0.243
$k+1$		$\overline{\gamma}$	-0.153	-0.247	NA	-0.256	-0.131	-0.276
		ρ_{map}	0.209	0.154	0.077	0.176	-0.033	0.183
		$\overline{\lvert\rho\rvert}$	0.584	0.547	0.218	0.49	0.175	0.342
		$(ML_2)^2$	0.101	-0.011	0.071	-0.081	0.124	-0.038
		$(CL_2)^2$	0.564	-0.16	-0.453	0.443	0.193	0.343
	$\phi_{p=15}$	Cov	0.288	0.788	-0.159	0.44	0.831	0.871
		MaxPro	0.145	0.353	NA	0.42	0.395	0.23
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.673	0.88	0.814	0.749	0.967	0.955
		Mm	-0.483	-0.814	-0.553	-0.57	-0.913	-0.923
		$\overline{\gamma}$	-0.789	-0.874	-0.586	-0.868	-0.599	-0.739
		ρ_{map}	0.092	0.118	0.208	0.037	-0.059	0.156
		$\overline{\lvert\rho\rvert}$	0.216	0.347	0.422	0.245	0.087	0.248
		$(ML_2)^2$	0.027	-0.041	0.083	-0.053	0.106	-0.079
		$(CL_2)^2$	0.215	-0.072	-0.284	0.297	0.103	0.264
	$\phi_{p=50}$	Cov	0.724	0.891	-0.121	0.755	0.838	0.871
		MaxPro	0.105	0.215	NA	0.21	0.357	0.239
		$\phi_{p=15}$	0.673	0.88	0.814	0.749	0.967	0.955
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.931	-0.98	-0.896	-0.945	-0.982	-0.991
		$\overline{\gamma}$	-0.562	-0.7	-0.926	-0.76	-0.534	-0.63

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.101	-0.122	-0.199	0.041	0.05	-0.166
		$\overline{\lvert\rho\rvert}$	-0.102	-0.326	-0.401	-0.125	-0.064	-0.244
		$(ML_2)^2$	0.028	0.054	-0.083	0.033	-0.105	0.088
		$(CL_2)^2$	-0.091	0.046	0.183	-0.192	-0.106	-0.261
	Mm	Cov	-0.713	-0.876	-0.118	-0.752	-0.85	-0.867
		MaxPro	-0.093	-0.185	NA	-0.171	-0.312	-0.243
		$\phi_{p=15}$	-0.483	-0.814	-0.553	-0.57	-0.913	-0.923
		$\phi_{p=50}$	-0.931	-0.98	-0.896	-0.945	-0.982	-0.991
		Mm	1	1	1	1	1	1
$k \mid 1$		$\overline{\gamma}$	0.447	0.653	0.973	0.61	0.547	0.628
		ρ_{map}	-0.191	-0.082	-0.227	-0.218	-0.069	-0.248
		$\overline{\lvert\rho\rvert}$	-0.458	-0.481	-0.434	-0.483	-0.442	-0.61
		$(ML_2)^2$	-0.051	0.001	-0.122	0.078	-0.085	-0.051
		$(CL_2)^2$	-0.383	0.201	0.188	-0.515	-0.317	-0.441
	$\overline{\gamma}$	Cov	-0.065	-0.61	0.105	-0.349	-0.575	-0.556
		MaxPro	-0.153	-0.247	NA	-0.256	-0.131	-0.276
		$\phi_{p=15}$	-0.789	-0.874	-0.586	-0.868	-0.599	-0.739
		$\phi_{p=50}$	-0.562	-0.7	-0.926	-0.76	-0.534	-0.63
		Mm	0.447	0.653	0.973	0.61	0.547	0.628
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
$3k+2$	ρ_{map}	ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.175	0.211	0.083	0.246	0.285	0.34
		$(ML_2)^2$	-0.115	0.126	0.111	-0.182	0.05	0.163
		$(CL_2)^2$	0.258	-0.016	0.093	0.01	0.144	0.16
		Cov	-0.13	-0.087	-0.053	-0.08	0.024	-0.02
		MaxPro	-0.031	0.023	NA	-0.126	0.013	-0.1
		$\phi_{p=15}$	0.09	0.022	-0.065	0.045	0.028	-0.097
		$\phi_{p=50}$	-0.016	0.017	-0.001	0.077	-0.001	-0.089
		Mm	-0.012	-0.034	-0.029	-0.081	0.012	0.072
		$\overline{\gamma}$	-0.104	-0.016	-0.04	-0.134	-0.205	-0.046
	$\overline{\lvert\rho\rvert}$	ρ_{map}	0.175	0.211	0.083	0.246	0.285	0.34
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	0.024	-0.324	-0.22	0.026	0.129	0.212
		$(CL_2)^2$	0.4	-0.28	0.01	0.271	0.423	0.467
		Cov	-0.094	-0.167	-0.201	0.019	-0.099	0.024
		MaxPro	-0.208	0.506	NA	0.094	0.034	0.021
		$\phi_{p=15}$	0.543	0.389	0.218	-0.142	0.135	0.118
		$\phi_{p=50}$	0.204	0.104	0.247	-0.098	0.06	0.07
		Mm	-0.067	-0.07	-0.222	0.073	0.006	-0.057
		$\overline{\gamma}$	-0.308	-0.358	-0.242	0.173	-0.199	-0.37
	$(ML_2)^2$	ρ_{map}	-0.115	0.126	0.111	-0.182	0.05	0.163
		$\overline{\lvert\rho\rvert}$	0.024	-0.324	-0.22	0.026	0.129	0.212
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	-0.008	0.771	0.284	0.056	0.118	0.119
		Cov	-0.028	0.187	-0.086	-0.086	0.021	-0.068
		MaxPro	-0.091	-0.465	NA	-0.02	0.052	0.106
		$\phi_{p=15}$	0.063	-0.662	-0.376	-0.148	-0.047	-0.046
		$\phi_{p=50}$	-0.01	-0.108	-0.323	-0.146	-0.092	-0.048
		Mm	0.101	0.065	0.269	0.124	0.097	0.053
		$\overline{\gamma}$	-0.068	0.607	0.265	0.136	-0.206	-0.066
	$(CL_2)^2$	ρ_{map}	0.258	-0.016	0.093	0.01	0.144	0.16
		$\overline{\lvert\rho\rvert}$	0.4	-0.28	0.01	0.271	0.423	0.467
		$(ML_2)^2$	-0.008	0.771	0.284	0.056	0.118	0.119
		$(CL_2)^2$	1	1	1	1	1	1
		Cov	0.041	0.135	-0.298	0.093	0.284	0.17
		MaxPro	-0.138	-0.507	NA	0.058	0.057	0.098
		$\phi_{p=15}$	0.676	-0.798	-0.891	-0.249	0.152	0.113
		$\phi_{p=50}$	0.288	-0.17	-0.812	-0.182	0.13	0.082
		Mm	-0.011	0.134	0.769	0.156	-0.143	-0.091
		$\overline{\gamma}$	-0.446	0.735	0.759	0.166	-0.329	-0.313

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.13	-0.087	-0.053	-0.08	0.024	-0.02
		$\overline{\lvert\rho\rvert}$	-0.094	-0.167	-0.201	0.019	-0.099	0.024
		$(ML_2)^2$	-0.028	0.187	-0.086	-0.086	0.021	-0.068
		$(CL_2)^2$	0.041	0.135	-0.298	0.093	0.284	0.17
	Cov	Cov	1	1	1	1	1	1
		MaxPro	0.007	0.022	NA	0.082	0.32	0.249
		$\phi_{p=15}$	0.295	0.271	0.2	0.64	0.544	0.716
		$\phi_{p=50}$	0.643	0.692	0.181	0.617	0.61	0.708
		Mm	-0.383	-0.684	-0.236	-0.58	-0.694	-0.734
		$\overline{\gamma}$	-0.141	-0.161	-0.168	-0.299	-0.478	-0.63
		ρ_{map}	-0.031	0.023	NA	-0.126	0.013	-0.1
		$\overline{\lvert\rho\rvert}$	-0.208	0.506	NA	0.094	0.034	0.021
		$(ML_2)^2$	-0.091	-0.465	NA	-0.02	0.052	0.106
		$(CL_2)^2$	-0.138	-0.507	NA	0.058	0.057	0.098
	MaxPro	Cov	0.007	0.022	NA	0.082	0.32	0.249
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	-0.158	0.529	NA	-0.149	0.511	0.259
		$\phi_{p=50}$	-0.066	0.138	NA	-0.08	0.502	0.264
		Mm	-0.087	-0.115	NA	0.039	-0.457	-0.258
$3k+2$		$\overline{\gamma}$	0.141	-0.488	NA	0.124	-0.186	-0.147
		ρ_{map}	0.09	0.022	-0.065	0.045	0.028	-0.097
		$\overline{\lvert\rho\rvert}$	0.543	0.389	0.218	-0.142	0.135	0.118
		$(ML_2)^2$	0.063	-0.662	-0.376	-0.148	-0.047	-0.046
		$(CL_2)^2$	0.676	-0.798	-0.891	-0.249	0.152	0.113
	$\phi_{p=15}$	Cov	0.295	0.271	0.2	0.64	0.544	0.716
		MaxPro	-0.158	0.529	NA	-0.149	0.511	0.259
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.679	0.599	0.965	0.699	0.955	0.968
		Mm	-0.216	-0.545	-0.917	-0.619	-0.839	-0.923
		$\overline{\gamma}$	-0.702	-0.92	-0.917	-0.813	-0.374	-0.487
		ρ_{map}	-0.016	0.017	-0.001	0.077	-0.001	-0.089
		$\overline{\lvert\rho\rvert}$	0.204	0.104	0.247	-0.098	0.06	0.07
		$(ML_2)^2$	-0.01	-0.108	-0.323	-0.146	-0.092	-0.048
		$(CL_2)^2$	0.288	-0.17	-0.812	-0.182	0.13	0.082
	$\phi_{p=50}$	Cov	0.643	0.692	0.181	0.617	0.61	0.708
		MaxPro	-0.066	0.138	NA	-0.08	0.502	0.264
		$\phi_{p=15}$	0.679	0.599	0.965	0.699	0.955	0.968
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.714	-0.985	-0.983	-0.978	-0.958	-0.986
		$\overline{\gamma}$	-0.732	-0.416	-0.985	-0.315	-0.369	-0.429

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.012	-0.034	-0.029	-0.081	0.012	0.072
		$\overline{\lvert\rho\rvert}$	-0.067	-0.07	-0.222	0.073	0.006	-0.057
		$(ML_2)^2$	0.101	0.065	0.269	0.124	0.097	0.053
		$(CL_2)^2$	-0.011	0.134	0.769	0.156	-0.143	-0.091
	Mm	Cov	-0.383	-0.684	-0.236	-0.58	-0.694	-0.734
		MaxPro	-0.087	-0.115	NA	0.039	-0.457	-0.258
		$\phi_{p=15}$	-0.216	-0.545	-0.917	-0.619	-0.839	-0.923
		$\phi_{p=50}$	-0.714	-0.985	-0.983	-0.978	-0.958	-0.986
		Mm	1	1	1	1	1	1
$3k+2$		$\overline{\gamma}$	0.409	0.37	0.997	0.235	0.412	0.445
		ρ_{map}	-0.104	-0.016	-0.04	-0.134	-0.205	-0.046
		$\overline{\lvert\rho\rvert}$	-0.308	-0.358	-0.242	0.173	-0.199	-0.37
		$(ML_2)^2$	-0.068	0.607	0.265	0.136	-0.206	-0.066
		$(CL_2)^2$	-0.446	0.735	0.759	0.166	-0.329	-0.313
	$\overline{\gamma}$	Cov	-0.141	-0.161	-0.168	-0.299	-0.478	-0.63
		MaxPro	0.141	-0.488	NA	0.124	-0.186	-0.147
		$\phi_{p=15}$	-0.702	-0.92	-0.917	-0.813	-0.374	-0.487
		$\phi_{p=50}$	-0.732	-0.416	-0.985	-0.315	-0.369	-0.429
		Mm	0.409	0.37	0.997	0.235	0.412	0.445
		$\overline{\gamma}$	1	1	1	1	1	1

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	1	1	1	1	1	1
		$\overline{\lvert\rho\rvert}$	0.202	0.303	0.27	0.107	0.286	0.41
		$(ML_2)^2$	-0.107	0.266	0.102	0.012	0.155	0.244
		$(CL_2)^2$	0.227	0.241	0.172	0.1	0.099	0.291
	ρ_{map}	Cov	0.072	-0.05	0.01	0.007	0.011	0.21
		MaxPro	0.142	0.187	NA	0.069	0.022	-0.027
		$\phi_{p=15}$	0.046	0.111	-0.071	-0.083	0.03	0.193
		$\phi_{p=50}$	0.051	0.109	-0.027	-0.068	0.043	0.157
		Mm	-0.043	-0.108	0.003	0.064	-0.057	-0.147
		$\overline{\gamma}$	-0.013	-0.013	0.006	0.139	-0.248	-0.248
		ρ_{map}	0.202	0.303	0.27	0.107	0.286	0.41
		$\overline{\lvert\rho\rvert}$	1	1	1	1	1	1
		$(ML_2)^2$	-0.091	0.238	-0.092	0.116	0.291	0.225
		$(CL_2)^2$	0.465	0.373	0.04	0.399	0.451	0.608
	$\overline{\lvert\rho\rvert}$	Cov	-0.19	-0.107	0.015	-0.147	0.039	0.305
		MaxPro	0.084	0.333	NA	0.006	0.133	0.249
		$\phi_{p=15}$	0.259	0.08	0.201	-0.261	0.005	0.406
		$\phi_{p=50}$	0.002	-0.063	0.222	-0.155	0.005	0.325
$10k$		Mm	0.032	0.045	-0.222	0.14	-0.013	-0.285
		$\overline{\gamma}$	-0.073	-0.111	-0.216	0.22	-0.242	-0.389
		ρ_{map}	-0.107	0.266	0.102	0.012	0.155	0.244
		$\overline{\lvert\rho\rvert}$	-0.091	0.238	-0.092	0.116	0.291	0.225
		$(ML_2)^2$	1	1	1	1	1	1
		$(CL_2)^2$	0.157	0.472	0.492	0.145	0.235	0.129
	$(ML_2)^2$	Cov	-0.118	-0.091	0.178	0.134	0.216	0.198
		MaxPro	-0.128	-0.083	NA	-0.092	0.199	0.086
		$\phi_{p=15}$	0.048	-0.072	-0.602	0.063	0.199	0.165
		$\phi_{p=50}$	-0.018	-0.053	-0.479	0.05	0.255	0.087
		Mm	0.044	0.036	0.372	-0.054	-0.286	-0.067
		$\overline{\gamma}$	-0.115	0.037	0.387	-0.085	-0.286	-0.214
		ρ_{map}	0.227	0.241	0.172	0.1	0.099	0.291
		$\overline{\lvert\rho\rvert}$	0.465	0.373	0.04	0.399	0.451	0.608
		$(ML_2)^2$	0.157	0.472	0.492	0.145	0.235	0.129
		$(CL_2)^2$	1	1	1	1	1	1
	$(CL_2)^2$	Cov	0.022	0.053	0.08	-0.064	0.295	0.339
		MaxPro	-0.038	-0.014	NA	-0.087	0.119	0.106
		$\phi_{p=15}$	0.429	0.046	-0.702	-0.176	0.134	0.321
		$\phi_{p=50}$	0.16	0.036	-0.476	-0.121	0.138	0.206
		Mm	-0.085	-0.055	0.336	0.143	-0.175	-0.167
		$\overline{\gamma}$	-0.188	-0.014	0.348	-0.001	-0.387	-0.4

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	0.072	-0.05	0.01	0.007	0.011	0.21
		$\overline{\lvert\rho\rvert}$	-0.19	-0.107	0.015	-0.147	0.039	0.305
		$(ML_2)^2$	-0.118	-0.091	0.178	0.134	0.216	0.198
		$(CL_2)^2$	0.022	0.053	0.08	-0.064	0.295	0.339
	Cov	Cov	1	1	1	1	1	1
		MaxPro	-0.01	-0.184	NA	0.072	0.075	0.063
		$\phi_{p=15}$	0.295	0.645	-0.008	0.82	0.602	0.798
		$\phi_{p=50}$	0.724	0.569	0.054	0.635	0.591	0.667
		Mm	-0.479	-0.535	-0.146	-0.575	-0.601	-0.647
		$\overline{\gamma}$	-0.19	-0.413	-0.063	-0.473	-0.619	-0.539
		ρ_{map}	0.142	0.187	NA	0.069	0.022	-0.027
		$\overline{\lvert\rho\rvert}$	0.084	0.333	NA	0.006	0.133	0.249
		$(ML_2)^2$	-0.128	-0.083	NA	-0.092	0.199	0.086
		$(CL_2)^2$	-0.038	-0.014	NA	-0.087	0.119	0.106
	MaxPro	Cov	-0.01	-0.184	NA	0.072	0.075	0.063
		MaxPro	1	1	NA	1	1	1
		$\phi_{p=15}$	-0.055	0.188	NA	0.011	0.205	0.073
		$\phi_{p=50}$	0.013	-0.09	NA	-0.016	0.257	0.034
		Mm	-0.006	0.079	NA	0.025	-0.251	-0.021
$10k$		$\overline{\gamma}$	0.02	-0.147	NA	0.059	-0.108	-0.096
		ρ_{map}	0.046	0.111	-0.071	-0.083	0.03	0.193
		$\overline{\lvert\rho\rvert}$	0.259	0.08	0.201	-0.261	0.005	0.406
		$(ML_2)^2$	0.048	-0.072	-0.602	0.063	0.199	0.165
		$(CL_2)^2$	0.429	0.046	-0.702	-0.176	0.134	0.321
	$\phi_{p=15}$	Cov	0.295	0.645	-0.008	0.82	0.602	0.798
		MaxPro	-0.055	0.188	NA	0.011	0.205	0.073
		$\phi_{p=15}$	1	1	1	1	1	1
		$\phi_{p=50}$	0.714	0.663	0.885	0.797	0.924	0.935
		Mm	-0.319	-0.617	-0.739	-0.742	-0.847	-0.896
		$\overline{\gamma}$	-0.666	-0.78	-0.746	-0.734	-0.407	-0.549
		ρ_{map}	0.051	0.109	-0.027	-0.068	0.043	0.157
		$\overline{\lvert\rho\rvert}$	0.002	-0.063	0.222	-0.155	0.005	0.325
		$(ML_2)^2$	-0.018	-0.053	-0.479	0.05	0.255	0.087
		$(CL_2)^2$	0.16	0.036	-0.476	-0.121	0.138	0.206
	$\phi_{p=50}$	Cov	0.724	0.569	0.054	0.635	0.591	0.667
		MaxPro	0.013	-0.09	NA	-0.016	0.257	0.034
		$\phi_{p=15}$	0.714	0.663	0.885	0.797	0.924	0.935
		$\phi_{p=50}$	1	1	1	1	1	1
		Mm	-0.653	-0.982	-0.956	-0.984	-0.982	-0.99
		$\overline{\gamma}$	-0.715	-0.306	-0.961	-0.418	-0.353	-0.421

Correlations $\mathbf{r}(\mathbf{x}, \mathbf{y})$

	k		20					
n	y	x	MmLHD	MaxPro	SphereP	UniDOE	LHD	LHS
		ρ_{map}	-0.043	-0.108	0.003	0.064	-0.057	-0.147
		$\overline{\lvert\rho\rvert}$	0.032	0.045	-0.222	0.14	-0.013	-0.285
		$(ML_2)^2$	0.044	0.036	0.372	-0.054	-0.286	-0.067
		$(CL_2)^2$	-0.085	-0.055	0.336	0.143	-0.175	-0.167
	Mm	Cov	-0.479	-0.535	-0.146	-0.575	-0.601	-0.647
		MaxPro	-0.006	0.079	NA	0.025	-0.251	-0.021
		$\phi_{p=15}$	-0.319	-0.617	-0.739	-0.742	-0.847	-0.896
		$\phi_{p=50}$	-0.653	-0.982	-0.956	-0.984	-0.982	-0.99
		Mm	1	1	1	1	1	1
$10k$		$\overline{\gamma}$	0.3	0.282	0.995	0.384	0.363	0.419
		ρ_{map}	-0.013	-0.013	0.006	0.139	-0.248	-0.248
		$\overline{\lvert\rho\rvert}$	-0.073	-0.111	-0.216	0.22	-0.242	-0.389
		$(ML_2)^2$	-0.115	0.037	0.387	-0.085	-0.286	-0.214
		$(CL_2)^2$	-0.188	-0.014	0.348	-0.001	-0.387	-0.4
	$\overline{\gamma}$	Cov	-0.19	-0.413	-0.063	-0.473	-0.619	-0.539
		MaxPro	0.02	-0.147	NA	0.059	-0.108	-0.096
		$\phi_{p=15}$	-0.666	-0.78	-0.746	-0.734	-0.407	-0.549
		$\phi_{p=50}$	-0.715	-0.306	-0.961	-0.418	-0.353	-0.421
		Mm	0.3	0.282	0.995	0.384	0.363	0.419
		$\overline{\gamma}$	1	1	1	1	1	1

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX H. CALCULATED $\overline{\rho_{map}^{\min}}$ VALUE AND STANDARD DEVIATION (SD) FOR THE SPECIFIED DSE DIMENSIONS AND G = 10, 25, AND 50

These are additional tables resulting from the experiments of Chapter VI.

Table 27. Calculated $\overline{\rho_{map}^{\min}}$ value and standard deviation (SD) for the specified DSE dimensions for G =10.

$\overline{\rho_{map}^{min}}$ for a given $\mathrm{DSE}_{(n,k,G=10)}$

$\boldsymbol{n}$	$k+1$		$3k+2$		$10k$	
$\boldsymbol{k}$	mean	(SD)	mean	(SD)	mean	(SD)
5	0.1429	(0.0126)	0.1284	0.0223	0.0629	0.0114
6	0.2015	(0.0365)	0.1428	0.0202	0.0686	0.0101
7	0.2676	(0.0412)	0.1527	0.0190	0.0735	0.0096
8	0.3294	(0.0460)	0.1629	0.0184	0.0785	0.0087
9	0.3751	(0.0402)	0.1746	0.0170	0.0826	0.0079
10	0.4013	(0.0367)	0.1860	0.0167	0.0868	0.0076
11	0.4221	(0.0334)	0.1959	0.0164	0.0914	0.0078
12	0.4329	(0.0311)	0.2040	0.0158	0.0958	0.0075
13	0.4411	(0.0287)	0.2107	0.0158	0.1000	0.0078
14	0.4468	(0.0296)	0.2172	0.0152	0.1032	0.0074
15	0.4512	(0.0284)	0.2230	0.0157	0.1065	0.0075
16	0.4528	(0.0289)	0.2268	0.0152	0.1096	0.0078
17	0.4552	(0.0273)	0.2311	0.0142	0.1120	0.0073
18	0.4556	(0.0260)	0.2333	0.0144	0.1143	0.0070
19	0.4560	(0.0257)	0.2345	0.0142	0.1159	0.0072
20	0.4556	(0.0246)	0.2366	0.0136	0.1173	0.0069

Table 28. Calculated $\overline{\rho_{map}^{\min}}$ value and standard deviation (SD) for the specified DSE dimensions for $G = 25$.

$\overline{\rho_{map}^{min}}$ for a given $\text{DSE}_{(n,k,G=25)}$

	n	$k+1$		$3k+2$		$10k$	
k		mean	(SD)	mean	(SD)	mean	(SD)
5		0.1395	0.0133	0.1123	0.0185	0.0547	0.0087
6		0.1766	0.0275	0.1279	0.0153	0.0612	0.0080
7		0.2384	0.0316	0.1391	0.0154	0.0666	0.0073
8		0.2953	0.0342	0.1502	0.0150	0.0723	0.0070
9		0.3465	0.0345	0.1626	0.0136	0.0769	0.0063
10		0.3748	0.0306	0.1734	0.0124	0.0814	0.0061
11		0.3970	0.0259	0.1842	0.0131	0.0857	0.0063
12		0.4105	0.0263	0.1927	0.0127	0.0903	0.0060
13		0.4195	0.0231	0.1988	0.0124	0.0942	0.0066
14		0.4256	0.0236	0.2064	0.0126	0.0978	0.0064
15		0.4298	0.0218	0.2117	0.0132	0.1012	0.0060
16		0.4319	0.0235	0.2159	0.0118	0.1040	0.0060
17		0.4356	0.0222	0.2208	0.0111	0.1066	0.0057
18		0.4366	0.0205	0.2223	0.0111	0.1092	0.0055
19		0.4373	0.0212	0.2244	0.0119	0.1108	0.0058
20		0.4373	0.0190	0.2268	0.0107	0.1122	0.0052

Table 29. Calculated $\overline{\rho_{map}^{\min}}$ value and standard deviation (SD) for the specified DSE dimensions for $G = 50$.

$\overline{\rho_{map}^{min}}$ for a given $\text{DSE}_{(n,k,G=50)}$

	n	$k+1$		$3k+2$		$10k$	
k		mean	(SD)	mean	(SD)	mean	(SD)
5		0.1362	0.0182	0.1015	0.0159	0.0499	0.0077
6		0.1612	0.0220	0.1190	0.0130	0.0567	0.0069
7		0.2219	0.0268	0.1299	0.0133	0.0626	0.0066
8		0.2766	0.0298	0.1418	0.0134	0.0685	0.0063
9		0.3270	0.0304	0.1545	0.0111	0.0732	0.0053
10		0.3581	0.0284	0.1664	0.0112	0.0779	0.0054
11		0.3824	0.0234	0.1769	0.0115	0.0823	0.0057
12		0.3953	0.0235	0.1859	0.0112	0.0870	0.0054
13		0.4061	0.0196	0.1922	0.0111	0.0905	0.0058
14		0.4126	0.0208	0.1992	0.0111	0.0942	0.0057
15		0.4175	0.0181	0.2043	0.0114	0.0978	0.0053
16		0.4179	0.0196	0.2091	0.0100	0.1004	0.0052
17		0.4231	0.0193	0.2148	0.0097	0.1034	0.0049
18		0.4248	0.0178	0.2166	0.0096	0.1061	0.0049
19		0.4263	0.0189	0.2178	0.0106	0.1074	0.0051
20		0.4259	0.0165	0.2206	0.0091	0.1091	0.0044

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I. REX

This section introduces some research ideas on sequentially using mathematical programming to construct designs with good balance, correlation, and space-filling properties. Towards that end, we are developing and experimenting with the Row Enhanced eXperiments (REX) algorithm. REX builds on previous approaches using mixed integer programming (MIP) to construct nearly orthogonal or balanced designs; see Hernandez et al. (2012a), Vieira et al. (2013), and Little et al. (2019). REX applies to designs with continuous and discrete level columns.

The idea behind REX is to incorporate design characteristics (e.g., correlation, balance, and space-filling measures) in the objective function or constraints and iteratively optimize the values in one column at a time until no more improvement is possible or a specified standard has been met. This process can be repeated with an adjusted objective function and/or updated constraints to iterate towards a design that performs well in multiple criteria. Figure 103 shows one possible iterative approach that focuses on one design measure at a time.

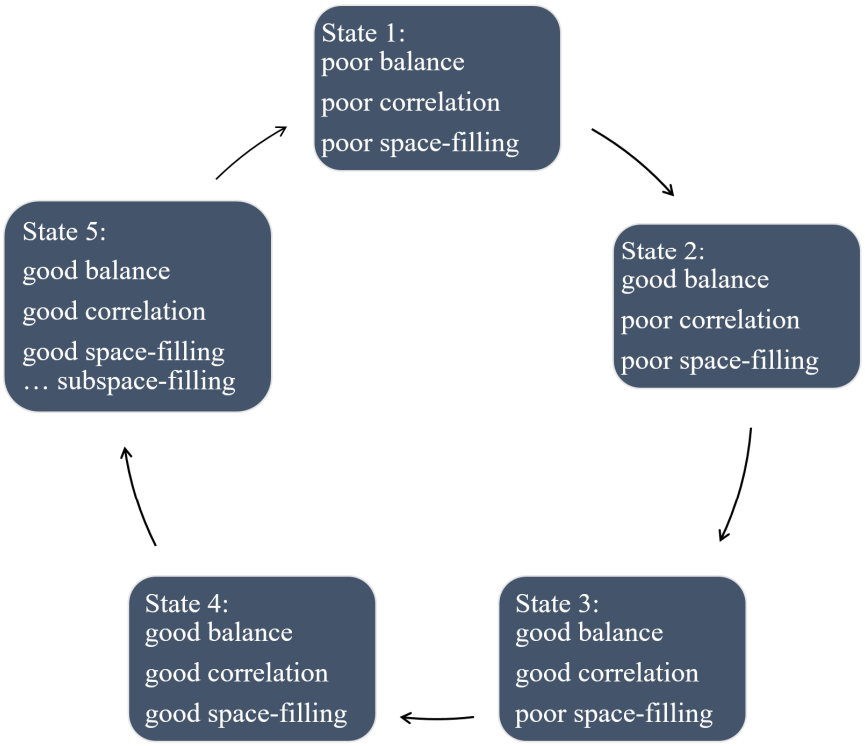


Figure 103. Sequentially improving the correlation, imbalance, space-filling, and subspace-filling properties of a design using a column-based MIP at each step.

We show the potential of REX with a small example. Figure 104 shows an initial design with two continuous factors, one two-level discrete factor and one four-level discrete factor. This initial design has very poor correlation (ρ_{map} = 1.0), balance (a maximum imbalance of 0.875), and space-filling properties. We first iterated through a series of single-column mixed integer programs to construct designs with minimal imbalance. Next, we iterated through the discrete columns to improve the correlations. Finally, we minimize correlations associated with the continuous columns. The final design of Figure 105 is perfectly balanced and orthogonal. Perhaps additional iterations could improve upon the space-filling properties.

	Initial Design			
n	X_1	X_2	X_3	X_4
1	1	1	1	1
2	2	2	2	1
3	3	3	2	1
4	4	4	2	1
5	5	5	2	1
6	6	6	2	1
7	7	7	2	1
8	8	8	2	1
9	9	9	2	1
10	10	10	2	1
11	11	11	2	1
12	12	12	2	1
13	13	13	2	1
14	14	14	2	1
15	15	15	2	1
16	16	16	2	4

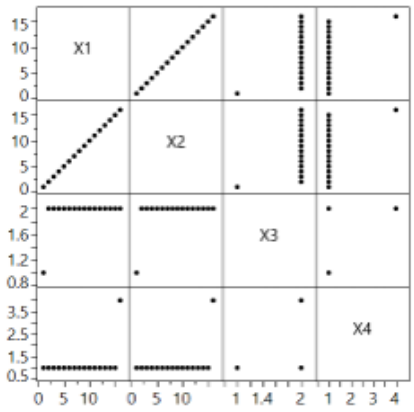


Figure 104. A very poor initial design with $\rho_{map} = 1$ and a maximal imbalance of 0.875.

	REX Design			
n	X_1	X_2	X_3	X_4
1	1	9	2	1
2	2	8	1	3
3	3	2	1	3
4	4	11	1	3
5	5	16	2	4
6	6	10	2	1
7	7	7	2	2
8	8	6	2	2
9	9	5	1	3
10	10	12	2	4
11	11	15	1	4
12	12	14	1	1
13	13	13	1	1
14	14	4	1	2
15	15	3	2	4
16	16	9	2	2

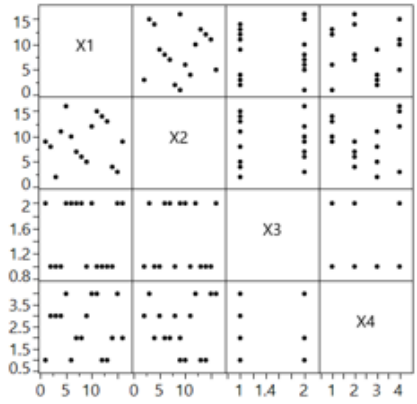


Figure 105. The resultant design after applying REX is orthogonal and perfectly balanced.

An extension of this research can explore the ability of various objective functions, constraints, and sequencing to construct designs with good column-correlation, balance, and space-filling measures.

LIST OF REFERENCES

Atkinson A, Donev A, Tobias R (2007) *Optimum Experimental Designs, with SAS* (Oxford University Press, Incorporated, Oxford, UK).

Audze P, Eglais V (1977) New approach to planning out of experiments. *Probl. Dyn. Strengths* 35:104–107.

Ba S (2015) SLHD: Maximin-distance (sliced) Latin hypercube designs (CRAN). Accessed May 10, 2022, https:// CRAN.R-project.org/package=SLHD.

Ba S, Joseph VR (2018) MaxPro: Maximum projection designs (CRAN). Accessed May 9, 2022, https://CRAN.R-project.org/package=MaxPro.

Ba S, Myers WR, Brenneman WA (2015) Optimal sliced Latin hypercube designs. *Technometrics* 57(4):479–487.

Bathke A (2004) The ANOVA F test can still be used in some balanced designs with unequal variances and nonnormal data. *J. Stat. Plan. Inference* 126:413–422.

Bazaraa M, Sherali H, Shetty C (2004) *Nonlinear Programming: Theory and Algorithms.* (John Wiley & Sons).

Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization* (Athena Scientific Belmont, MA).

Bingham D, Sitter RR, Tang B (2009) Orthogonal and nearly orthogonal designs for computer experiments. *Biometrika* 96(1):51–65.

Bohachevsky IO, Johnson ME, Stein ML (1986) Generalized simulated annealing for function optimization. *Technometrics* 28(3):209–217.

Box GEP (1978) *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building* (Wiley, New York).

Box, GEP (1993) George's column. *Qual. Eng.* 5(2), 321–330.

Box GEP, Wilson KB (1951) On the experimental attainment of optimum conditions. *J. R. Stat. Soc. Ser. B Methodol.* 13(1):1–45.

Butler NA (2001) Optimal and orthogonal Latin hypercube designs for computer experiments. *Biometrika* 88(3):847–857.

Bynum ML, Hackebeil GA, Hart WE, Laird CD, Nicholson BL, Siirola JD, Watson JP, Woodruff DL (2021) *Pyomo–optimization Modeling in Python*, 3rd ed. (Springer Science & Business Media).

Canty A, Ripley BD (2021) Boot: Bootstrap functions (CRAN). Accessed May 8, 2022, https://CRAN.R-project.org/package=boot.

Casella G, Berger R (2002) *Statistical Inference*, 2nd ed. (Belmont, CA, Duxbury).

Chen RB, Hsieh DN, Hung Y, Wang W (2013) Optimizing Latin hypercube designs by particle swarm. *Stat. Comput.* 23(5):663–676.

Cioppa TM (2002) Efficient nearly orthogonal and space-filling experimental designs for high-dimensional complex models. Doctoral dissertation. (Naval Postgraduate School, Monterey, CA).

Cioppa TM, Lucas TW (2007) Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* 49(1):45–55.

COIN-OR (2016). Accessed May 12, 2022, http://www.coin-or.org/.

CPLEX (2009) V12. 1: User's Manual for CPLEX. *Int. Bus. Mach. Corp.* 46(53):157.

van Dam ER, Rennen G, Husslage B (2009) Bounds for maximin Latin hypercube designs. *Oper. Res.* 57(3):595–608.

David HA, Nagaraja HN (2004) *Order Statistics* (John Wiley & Sons).

Davison AC, Hinkley DV (1997) *Bootstrap Methods and Their Applications* (Cambridge University Press, Cambridge).

Dean A, Morris M, Stufken J, Bingham D (2015) *Handbook of Design and Analysis of Experiments* (Boca Raton, FL, CRC Press).

DoD (2021) The Department of Defense Releases the President's Fiscal Year 2022 Defense Budget. *US Dep. Def.* Accessed August 12, 2021, https://www.defense.gov/Newsroom/Releases/Release/Article/2638711/the-department-of-defense-releases-the-presidents-fiscal-year-2022-defense-budg/.

Dupuy D, Helbert C, Franco J (2015) DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments. *J. Stat. Softw.* 65(11):1–38.

Efron B (1992) Bootstrap methods: another look at the jackknife. *Breakthr. Stat.* 569–593.

Efron B, Tibshirani R (1986) Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Stat. Sci.* 1(1):54–75.

Fang KT (1980) Uniform design: application of number-theoretic methods in experimental design. *Acta Math Appl Sin* 3:363–372.

Fang KT, Lin DKJ, Winker P, Zhang Y (2000a) Uniform design: Theory and application. *Technometrics* 42(3):237–248.

Fang KT, Ma CX, Winker P (2000b) The Uniform Design - Hong Kong Baptist University Mathematics Department. Accessed January 29, 2021, http://www.math.hkbu.edu.hk/UniformDesign/.

Fang KT, Runze L, Sudjianto A (2006) *Design and Modeling for Computer Experiments* (Chapman and Hall CRC).

Fisher RA (1925) *Statistical Methods for Research Workers, 10th* ed. (Oliver and Boyd, Edinburgh).

Fisher RA (1926) On the capillary forces in an ideal soil; correction of formulae given by WB Haines. *J. Agric. Sci.* 16(3):492–505.

Fisher RA (1971) *The Design of Experiments*, 8th ed. (Hafner).

Florian A (1992) An efficient sampling scheme: Updated Latin hypercube sampling. *Probabilistic Eng. Mech.* 7(2):123–130.

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (WH Freeman and Company).

Georgiou SD (2009) Orthogonal Latin hypercube designs from generalized orthogonal designs. *J. Stat. Plan. Inference* 139(4):1530–1540.

Georgiou SD, Efthimiou I (2014) Some classes of orthogonal Latin hypercube designs. *Stat. Sin.* 24(1):101–120.

Gilleland E, Katz RW (2016) ExtRemes 2.0: An extreme value analysis package in R. *J. Stat. Softw.* 72(8):1–39.

Goldberg DE, Holland JH (1988) *Genetic Algorithms and Machine Learning* (Reading, MA: Addison-Wesley).

Goldfarb HB, Borror CM, Montgomery DC, Anderson-Cook CM (2005) Using genetic algorithms to generate mixture-process experimental designs involving control and noise variables. *J. Qual. Technol.* 37(1):60–74.

Gumbel EJ (1935) Les valeurs extrêmes des distributions statistiques. *Ann. Inst. Henri Poincaré*. 115–158.

Gurobi Optimization, LLC (2022) Gurobi optimizer reference manual, Accessed May 12, 2022, https://www.gurobi.com.

Guttorp P, Lindgren G (2009) Karl Pearson and the Scandinavian school of statistics. *Int. Stat. Rev.* 77(1):64–71.

Hart WE, Watson JP, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in Python. *Math. Program. Comput*. 3(3):219–260.

Helton JC, Davis FJ (2003) Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliab. Eng. Syst. Saf.* 81(1):23–69.

Heredia-Langner A, Carlyle WM, Montgomery DC, Borror CM, Runger GC (2003) Genetic algorithms for the construction of D-optimal designs. *J. Qual. Technol.* 35(1):28–46.

Heredia-Langner A, Montgomery DC, Carlyle WM, Borror CM (2004) Model-robust optimal designs: A genetic algorithm approach. *J. Qual. Technol.* 36(3):263–279.

Hernandez AS (2008) Breaking barriers to design dimensions in nearly orthogonal Latin hypercubes. Doctoral dissertation. (Naval Postgraduate School, Monterey, CA).

Hernandez AS (2016) A Gumbel distribution model to describe correlation values in random Latin hypercube experimental designs for model and simulation based systems engineering. *J. Eng. Archit.* 4(2).

Hernandez AS, Lucas TW, Carlyle M (2012a) Constructing nearly orthogonal Latin hypercubes for any nonsaturated run-variable combination. *ACM Trans. Model. Comput. Simul.* 22(4):1–17.

Hernandez AS, Lucas TW, Sanchez PJ (2012b) Selecting random Latin hypercube dimensions and designs through estimation of maximum absolute pairwise correlation. *Proc. 2012 Winter Simul. Conf. WSC*. 1–12.

Hey GB (1938) A new method of experimental sampling illustrated on certain non-normal populations. *Biometrika* 30(1/2):68–80.

Hickernell FJ (1998) A generalized discrepancy and quadrature error bound. *Math. Comput. Am. Math. Soc.* 67(221):299–322.

Hinkelmann K, Kempthorne O (2008) *Introduction to Experimental Design* (Wiley-Interscience).

Holland JH (1992) Genetic algorithms. *Sci. Am.* 267(1):66–73.

Hou R, Lu L (2018) MOLHD: multiple objective Latin hypercube design. Accessed May 20, 2022, https://cran.r-project.org/web/packages/MOLHD/MOLHD.pdf.

Iman RL, Conover WJ (1982) A distribution-free approach to inducing rank correlation among input variables. *Commun. Stat. - Simul. Comput.* 11(3):311–334.

Jenkinson AF (1955) The frequency distribution of the annual maximum (or minimum) values of meteorological elements. *Q. J. R. Meteorol. Soc.* 81(348):158–171.

Jin R, Chen W, Sudjianto A (2003) An efficient algorithm for constructing optimal design of computer experiments. *Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.* 545–554.

Johnson ME, Moore LM, Ylvisaker D (1990) Minimax and maximin distance designs. *J. Stat. Plan. Inference* 26(2):131–148.

Joseph VR (2016) Space-filling designs for computer experiments: A review. *Qual. Eng.* 28(1):28–35.

Joseph VR, Gul E, Ba S (2015) Maximum projection designs for computer experiments. *Biometrika* 102(2):371–380.

Joseph VR, Hung Y (2008) Orthogonal-maximin Latin hypercube designs. *Stat. Sin.*:171–186.

Ke X, Zhang R, Ye HJ (2015) Two-and three-level lower bounds for mixture L2-discrepancy and construction of uniform designs by threshold accepting. *J. Complex.* 31(5):741–753.

Kesler G, Lucas TW, Sanchez PJ (2019) A data farming analysis of a simulation of Armstrong's stochastic salvo model. *Proc. 2019 Winter Simul. Conf. WSC.* 2443–2454.

Kiefer J (1959) Optimum experimental designs. *J. R. Stat. Soc. Ser. B Methodol.* 21(2):272–319.

Kim H, Loh WY (2003) Classification trees with bivariate linear discriminant node models. *J. Comput. Graph. Stat.* 12(3):512–530.

Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1983) Optimization by simulated annealing. *science* 220(4598):671–680.

Kleijnen JPC (2015) *Design and Analysis of Simulation Experiments*, 2nd ed. (Springer International Publishing, Cham).

Kleijnen JPC, Sanchez SM, Lucas TW, Cioppa TM (2005) State-of-the-art review: A user's guide to the brave new world of designing simulation experiments. *Inf. J. Comput.* 17(3):263–289.

Kotz S, Balakrishnan N, Read CB, Vidakovic B (1983) *Encyclopedia of Statistical Sciences,* vol. 4 (John Wiley & Sons).

Kowalski CJ (1972) On the effects of non-normality on the distribution of the sample product-moment correlation coefficient. *J. R. Stat. Soc. Ser. C Appl. Stat.* 21(1):1–12.

Law AM, Kelton WD (2007) *Simulation Modeling and Analysis*, 3rd ed. (McGraw-Hill, New York).

Lawler EL (1963) The quadratic assignment problem. *Manag. Sci.* 9(4):586–599.

Leary S, Bhaskar A, Keane A (2003) Optimal orthogonal-array-based Latin hypercubes. *J. Appl. Stat.* 30(5):585–598.

Liefvendahl M, Stocki R (2006) A study on algorithms for optimization of Latin hypercubes. *J. Stat. Plan. Inference* 136(9):3231–3247.

Lin CD, Tang B (2015) Latin hypercubes and space-filling designs. *Handb. Des. Anal. Exp.*:593–625.

Lin CD, Mukerjee R, Tang B (2009) Construction of orthogonal and nearly orthogonal Latin hypercubes. *Biometrika* 96(1):243–247.

Lin T (2018) Unmanned surface logistics concept of support. Master's thesis. Naval Postgraduate School, Monterey, CA.

Little ZC, Weir JD, Hill RR, Stone BB, Freels JK (2019) Second-order extensions to nearly orthogonal-and-balanced (NOAB) mixed-factor experimental designs. *J. Simul.* 13(3):226–237.

Lucas TW, Kelton WD, Sánchez PJ, Sanchez SM, Anderson BL (2015) Changing the paradigm: Simulation, now a method of first resort. *Nav. Res. Logist.* 62(4):293–303.

Lu L, Anderson-Cook CM, Robinson TJ (2011) Optimization of designed experiments based on multiple criteria utilizing a Pareto frontier. *Technometrics* 53(4):353–365.

Lukemire J, Xiao Q, Mandal A, Wong WK (2021) Statistical analysis of complex computer models in astronomy. *Eur. Phys. J. Spec. Top. ArXiv210207179, https://doi.org/10.1140/epjs/s11734-021-00204-y.*

MacCalman AD (2013) Flexible space-filling designs for complex system simulations. Doctoral dissertation. Naval Postgraduate School, Monterey, CA.

MacCalman AD, Vieira H, Lucas T (2017) Second-order nearly orthogonal Latin hypercubes for exploring stochastic simulations. *J. Simul.* 11(2):137–150.

Mak S (2021) Minimaxdesign: Minimax and minimax projection designs, Accessed May 20, 2022, https://cran.r-project.org/web/packages/minimaxdesign/minimaxdesign.pdf.

Massey FJ (1951) The Kolmogorov-Smirnov test for goodness of fit. *J. Am. Stat. Assoc.* 46(253):68–78.

McCool B, Lyman J, Ferguson LJ (1995) Evolution of the model-test-model concept for use in operational testing & advanced warfighting experiments. *US Army Conf. Appl. Stat. 18-20 Oct. 1995*. 93.

McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245.

Meyer RK, Nachtsheim CJ (1995) The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics* 37(1):60–69.

Montgomery D (2013) *Design and Analysis of Experiments*, 8th ed. (John Wiley & Sons).

Moon H, Dean AM, Santner TJ (2012) Two-stage sensitivity-based group screening in computer experiments. *Technometrics* 54(4):376–387.

Morgan BL, Schramm HC, Smith JR, Lucas TW, McDonald ML, Sánchez PJ, Sanchez SM, Upton SC (2018) Improving u.s. navy campaign analyses with big data. *Interfaces* 48(2):130–146.

Morris MD, Mitchell TJ (1995) Exploratory designs for computational experiments. *J. Stat. Plan. Inference* 43(3):381–402.

Owen AB (1994) Controlling Correlations in Latin Hypercube Samples. *J. Am. Stat. Assoc.* 89(428):1517–1522.

Pang F, Liu MQ, Lin DKJ (2009) A construction method for orthogonal Latin hypercube designs with prime power levels. *Stat. Sin.* 19(4):1721–1728.

Pareto V (1906) Manuale di economica politica, societa editrice libraria. *Man. Polit. Econ.* 1971.

Park HW, Sohn H (2006) Parameter estimation of the generalized extreme value distribution for structural health monitoring. *Probabilistic Eng. Mech.* 21(4):366–376.

Pearson K (1895) VII. Note on regression and inheritance in the case of two parents. *Proc. R. Soc. Lond.* 58(347–352):240–242.

Pearson K, Filon LNG (1898) Mathematical contributions to the theory of evolution. iv. on the probable errors of frequency constants and on the influence of random selection on variation and correlation. *Philos. Trans. R. Soc. Lond. Ser. Contain. Pap. Math. Phys. Character* 191:229–311.

Pukelsheim F (2006) *Optimal design of experiments* (Philadelphia, PA, SIAM).

Qian PZG (2012) Sliced Latin hypercube designs. *J. Am. Stat. Assoc.* 107(497):393–399.

R Core Team (2021) *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna, Austria).

Ryan TP (2007) *Modern Experimental Design*, 1st ed. (John Wiley & Sons).

Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat. Sci.* 4(4):409–423.

Saeger KJ, Hinch JH (2001) Understanding instability in a complex deterministic combat simulation. *Mil. Oper. Res.* 6(4):43–55.

Sanchez PJ, Sanchez SM (2019) Orthogonal second-order space-filling designs with insights from simulation experiments to support test planning. *Qual. Reliab. Eng. Int.* 35(3):854–867.

Sanchez SM (2005) NOLHdesigns spreadsheet. Softw. Accessed May 20, 2022, https://nps.edu/web/seed/software-downloads.

Sanchez SM, Lucas TW, Sanchez PJ, Nannini CJ, Wan H (2012) Designs for large-scale simulation experiments, with applications to defense and homeland security. *Design and Analysis of Experiments*, vol. 3. Hinkelmann K, ed. (John Wiley & Sons, Hoboken, NJ), 413–441.

Sanchez SM, Sanchez PJ, Wan H (2020) Work smarter, not harder: a tutorial on designing and conducting simulation experiments. *Proc. 2020 Winter Simul. Conf. WSC*. 1128–1142.

Santner TJ, Williams BJ, Notz WI (2003) *The Design and Analysis of Computer Experiments*, 1st ed. (Springer, New York).

Santner TJ, Williams BJ, Notz WI (2018) *The Design and Analysis of Computer Experiments*, 2nd ed. (Springer, New York).

SAS (2018) *Design of Experiments Guide* (SAS Publishing, Cary, NC).

SAS (2021) JMP Pro 15. Accessed May 20, 2022, https://www.jmp.com/en_us/home.html.

Satterthwaite FE (1959) Random balance experimentation. *Technometrics* 1(2):111–137.

Savage S (2002) The flaw of averages. *Harv. Bus. Rev.* 80(11):20–21.

Schober P, Boer C, Schwarte LA (2018) Correlation coefficients: appropriate use and interpretation. *Anesth. Analg.* 126(5):1763–1768.

Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). *Biometrika* 52(3/4):591–611.

Shields MD, Zhang J (2016) The generalization of Latin hypercube sampling. *Reliab. Eng. Syst. Saf.* 148:96–108.

Steinberg DM, Lin D (2015) Construction of orthogonal nearly Latin hypercubes. *Qual. Reliab. Eng. Int.* 31(8):1397–1406.

Steinberg DM, Lin DKJ (2006) A construction method for orthogonal Latin hypercube designs. *Biometrika* 93(2):279–288.

Sun F, Gramacy RB (2021) Maximin: Space-filling design under maximin distance (CRAN). Accessed May 20, 2022, https://CRAN.R-project.org/package=maximin.

Sun F, Gramacy RB, Haaland B, Lu S, Hwang Y (2019) Synthesizing simulation and field data of solar irradiance. *Stat. Anal. Data Min. ASA Data Sci. J.* 12(4):311–324.

Sun F, Liu MQ, Lin DKJ (2009) Construction of orthogonal Latin hypercube designs. *Biometrika* 96(4):971–974.

Sun F, Tang B (2017a) A method of constructing space-filling orthogonal designs. *J. Am. Stat. Assoc.* 112(518):683–689.

Sun F, Tang B (2017b) A general rotation method for orthogonal Latin hypercubes. *Biometrika* 104(2):465–472.

Sun F, Wang Y, Xu H (2019) Uniform projection designs. *Ann. Stat.* 47(1).

Vieira H, Sanchez SM, Kienitz KH, Belderrain MCN (2011) Improved efficient, nearly orthogonal, nearly balanced mixed designs. *Proc. 2011 Winter Simul. Conf. WSC*. (IEEE, Phoenix, AZ), 3600–3611.

Vieira H, Sanchez SM, Kienitz KH, Belderrain MCN (2013) Efficient, nearly orthogonal-and-balanced, mixed designs: an effective way to conduct trade-off analyses via simulation. *J. Simul.* 7(4):264–275.

Wang H, Xiao Q, Mandal A (2021) Musings about constructions of efficient Latin hypercube designs with flexible run-sizes. *ArXiv Prepr. ArXiv201009154*. Accessed May 20, 2022, http://arxiv.org/abs/2010.09154.

Wang Y, Sun F, Xu H (2020) On design orthogonality, maximin distance, and projection uniformity for computer experiments. *J. Am. Stat. Assoc.*:1–11.

Wang Y, Yang J, Xu H (2018) On the connection between maximin distance designs and orthogonal designs. *Biometrika* 105(2):471–477.

Wheeler B (2019) AlgDesign: Algorithmic experimental design. Accessed May 20, 2022, https://cran.r-project.org/web/packages/AlgDesign/AlgDesign.pdf

Winker P, Fang KT (1998) Optimal U-type designs. *Monte Carlo Quasi-Monte Carlo Methods 1996*. (Springer), 436–448.

Xiao Q, Xu H (2018) Construction of maximin distance designs via level permutation and expansion. *Stat. Sin.* 28(3):1395–1414.

Ye KQ (1998) Orthogonal column Latin hypercubes and their application in computer experiments. *J. Am. Stat. Assoc.* 93(444):1430–1439.

Ye KQ (2021) Orthogonal Latin hypercubes. Accessed May 21, 2021, http://www.ams.sunysb.edu/~kye/olh.html.

Zhang A, Li H, Quan S, Yang Z (2018) UniDOE: Uniform design of experiments (CRAN). Accessed May 21, 2021, https://CRAN.R-project.org/package=UniDOE.

Zhou X, Lin DKJ, Hu X, Ouyang L (2019) Sequential Latin hypercube design with both space-filling and projective properties. *Qual. Reliab. Eng. Int.* 35(6):1941–1951.

Zhou YD, Fang KT, Ning JH (2013) Mixture discrepancy for quasi-random point sets. *J. Complex.* 29(3–4):283–301.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California