Bridging the Gap Between Space-Filling and Optimal Designs

Design for Computer Experiments

by

Kathryn Kennedy

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2013 by the
Graduate Supervisory Committee:

Douglas C. Montgomery, Co-Chair
Rachel T. Johnson, Co-Chair
John W. Fowler
Connie M. Borror

ARIZONA STATE UNIVERSITY

August 2013

ABSTRACT

This dissertation explores different methodologies for combining two popular design paradigms in the field of computer experiments. Space-filling designs are commonly used in order to ensure that there is good coverage of the design space, but they may not result in good properties when it comes to model fitting. Optimal designs traditionally perform very well in terms of model fitting, particularly when a polynomial is intended, but can result in problematic replication in the case of insignificant factors. By bringing these two design types together, positive properties of each can be retained while mitigating potential weaknesses.

Hybrid space-filling designs, generated as Latin hypercubes augmented with I-optimal points, are compared to designs of each contributing component. A second design type called a bridge design is also evaluated, which further integrates the disparate design types. Bridge designs are the result of a Latin hypercube undergoing coordinate exchange to reach constrained D-optimality, ensuring that there is zero replication of factors in any one-dimensional projection. Lastly, bridge designs were augmented with I-optimal points with two goals in mind. Augmentation with candidate points generated assuming the same underlying analysis model serves to reduce the prediction variance without greatly compromising the space-filling property of the design, while augmentation with candidate points generated assuming a different underlying analysis model can greatly reduce the impact of model misspecification during the design phase.

Each of these composite designs are compared to pure space-filling and optimal designs. They typically out-perform pure space-filling designs in terms of prediction variance and alphabetic efficiency, while maintaining comparability with pure optimal designs at small sample size. This justifies them as excellent candidates for initial experimentation.

DEDICATION

I would like to dedicate this dissertation to my parents, whose unwavering support and tenacious confidence was invaluable over my years in the program. In particular, I'd like to thank my father for always leading by example, and championing my cause for time and fair treatment. And my mother, for her commitment to perfectionism, excellent literary taste, and welcomed penchant for feeding me. I would also like to thank my 'little' brother, always setting the proactive example I so wish I could emulate, and my friends for bearing with me as I prevaricated over the years.

ACKNOWLEDGMENTS

I would like to acknowledge my committee, and thank them for their continued support as I worked to find time to complete the research phase of the degree requirements. Dr. Douglas Montgomery has been an exemplary advisor, and I am honored he agreed to guide me as he did my father. Dr. Rachel Johnson's research provided an excellent starting point, and she has been instrumental in providing guidance as to its continuance. I would like to thank Dr. John Fowler, for without his assistance I could not have continued in the program. And last but never least, Dr. Connie Borror, who taught the class which originally inspired me to transition into the Industrial Engineering department back in my undergrad days. Many thanks as well to Dr. Bradley Jones, whose JMP script provided an important starting point for the latter phases of my research.

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES ix

CHAPTER

1 INTRODUCTION 1

2 LITERATURE REVIEW 3

Design of Computer Experiments 3

Traditional Response Surface Methodology Designs 3

Space Filling Designs 5

Analysis of Computer Experiments 20

Polynomial Models 21

Gaussian Process Models 23

Augmentation of Designs for Computer Experiments 25

3 HYBRID SPACE-FILLING DESIGNS FOR COMPUTER EXPERIMENTS 27

Methodology 27

Results 29

Theoretical Prediction Variance 29

Empirical Root Mean Squared Error 33

Design Variability 40

Theoretical Prediction Variance 41

Prediction Performance 47

Empirical Root Mean Squared Error 48

Conclusions 50

CHAPTER Page

4 BRIDGE DESIGN PROPERTIES 51

Methodology 51

Results 53

Second-Order 53

Third-Order 59

Fourth-Order 64

Fifth-Order 69

Theoretical Properties Summary 70

Empirical Model Fitting Results 71

Conclusions 79

5 AUGMENTED BRIDGE DESIGNS 80

Methodology 80

Results 83

Same-Order Augmentation 84

Augmentation With Higher Order Optimal Points 92

Discussion 102

Conclusions 103

6 CONCLUSIONS AND FUTURE WORK 105

Future Work 106

REFERENCES 109

LIST OF TABLES

Table Page

1. Minimum number of design points needed (n = p) 28

2. Ranges and fixed values for the experimental and fixed variables in Test Function 3 38

3. Summary statistics for two two-factor, 21-run Latin hypercube designs, in terms of a fifth-order polynomial model 42

4. Maximum prediction variance values observed across five two-factor, 21-run Latin hypercube designs, in terms of a fifth-order polynomial model 44

5. Summary statistics for two 21-run Latin hypercube designs, in terms of a GASP model 45

6. Number of runs necessary for each design combination (number of factors, underlying model order, and number of runs) 52

7. Prediction variance estimates for designs generated assuming a full quadratic model 55

8. Design efficiencies for designs generated intended to be fit with a second-order polynomial model 58

9. Prediction variance estimates for designs generated assuming a third-order polynomial model 61

10. Design efficiencies for designs generated intending to be fit with a third-order polynomial model 63

11. Prediction variance estimates for designs generated assuming a full fourth-order polynomial model 66

12. Design efficiencies for designs generated intending to be fit with a fourth-order polynomial model 68

Table Page

13. Prediction variance estimates for designs generated assuming a full fifth-order polynomial model 70

14. Percent of cases in which the bridge design SPE is smaller than the comparator design for Test Function 1 72

15. Percent of cases in which the bridge design SPE is smaller than the comparator design for Test Function 2 75

16. Percent of cases in which the bridge design SPE is smaller than the comparator design for Test Function 3 77

17. Design size for base bridge designs to be augmented with I-optimal points 81

18. Sample sizes for I-optimal designs generated as candidate sets for bridge design augmentation 81

19. Augmentation results for bridge designs generated with underlying second-order polynomial models 86

20. Augmentation results for bridge designs generated with underlying third-order polynomial models 88

21. Augmentation results for bridge designs generated with underlying fourth-order polynomial models 90

22. Two-factor, second-order bridge designs augmented with one, two, or three third-order I-optimal points 93

23. Three-factor, second-order bridge design augmented with one, two, or three third-order I-optimal points 95

24. Four-factor, second-order bridge design augmented with one, two, or three third-order I-optimal points 96

Table		Page

25. Five-factor, second-order bridge design augmented with one, ..two, or three third-order I-optimal points 97

26. Two-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points 98

27. Three-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points 100

28. Four-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points 101

29. Five-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points 102

LIST OF FIGURES

Figure Page

1. Example LHD(10, 2) 6

2. Example of an LHD(10, 2) with poor space-filling properties 7

3. Latin Hypercube Design Taxonomy 8

4. FDS plot for four-factor, second-order designs assuming a second-order polynomial 31

5. FDS plot for four-factor, second-order designs assuming a GASP model 33

6. Surface plot of Test Function 1 34

7. RMSE for two-factor designs 35

8. Surface plot of Test Function 2 (x_3 set to 0.2225) 36

9. RMSE for three-factor designs 37

10. RMSE for four-factor designs 38

11. RMSE for five-factor designs 39

12. Point placement for two two-factor, 21-run Latin hypercube designs 41

13. FDS plot for replicated two-factor, 21-run Latin hypercube designs, in terms of a fifth-order polynomial model 42

14. Theoretical prediction variance for Design 1, in terms of a fifth-order polynomial 43

15. Theoretical prediction variance for Design 2, in terms of a fifth-order polynomial 44

16. Theoretical prediction variance for Design 1, in terms of a GASP model (both θ_i= 3) 46

17. Theoretical prediction variance for Design 2, in terms of a GASP model (both θ_i = 3) 46

Figure Page

18. Predicted values for Test Function 1 for Designs 1 and 2, analyzed using fifth-order polynomials or GASP models ... 47

19. Deterministic RMSE for two-factor, fourth-order designs ... 49

20. Stochastic RMSE for two-factor, fourth-order designs ... 49

21. Two-factor, 12-run designs generated assuming a full quadratic model ... 54

22. Prediction Variance (minimum, median, and maximum) for designs generated intending to be fit with a second-order polynomial model ... 57

23. D-efficiencies for second-order designs ... 59

24. Two-factor, 20-run designs generated assuming a full cubic model ... 60

25. Prediction Variance (minimum, median, and maximum) for designs generated intending to be fit with a third-order polynomial model ... 62

26. D-efficiencies for third-order designs ... 64

27. Two-factor, 30-run designs generated assuming a full fourth-order model ... 65

28. Prediction Variance (minimum, median, and maximum) for designs generated intending to be fit with a fourth-order polynomial model ... 67

29. D-efficiencies for fourth-order designs ... 69

30. Test Function 1 surface plot ... 71

31. Squared prediction error results (mean, median, and maximum) for Test Function 1 ... 73

32. Test Function 2 surface plot ... 74

33. Squared prediction error results (mean, median, and maximum) for Test Function 2 ... 76

Figure Page

34. Squared prediction error results (mean, median, and maximum) for Test Function 3 78

35. Prediction variance for the original two-factor, second-order bridge design with six runs 83

36. Prediction variance for the two-factor, second-order bridge design with six runs, augmented with candidate point 4, (0, 0) 84

37. Mean and maximum prediction variance for original and augmented second-order designs 87

38. Mean and maximum prediction variance for original and augmented third-order designs 89

39. Mean and maximum prediction variance for original and augmented fourth-order designs 91

40. Prediction variance for a two-factor, second-order bridge design under a third-order model 94

41. Prediction variance for a two-factor, second-order bridge design under a third-order model, augmented with a single point at (-0.5, 0.5) 94

42. Prediction variance for a two-factor, third-order bridge design under a fourth-order model 99

43. Prediction variance for a two-factor, third-order bridge design under a fourth-order model, augmented with a single point at (-0.2, 0.1) 99

CHAPTER 1 – INTRODUCTION

Design and analysis of computer experiments (DACE) is a rapidly growing field. Computer simulation models are often used in place of or in conjunction with physical experiments. Computer simulation models are becoming increasingly prevalent with the increase in computing power, and the development of software geared towards handling complex simulations. There are many types of simulation – finite element analysis (FEA), computational fluid dynamics (CFD) and circuit simulation typically employ solutions to differential equations or complex numerical methods. Discrete-event simulation models are built to simulate the operation of a system as a chain of events. The applications are quite diverse. Regniere and Sharov (1999) simulated male gypsy moth phrenology over a large region including latitude, longitude, elevation and temperature in order to learn more about their prevalence and serve as a pest-management planning tool. Calise, Palombo, and Vanoli (2010) estimated the efficiency and costs associated with several large scale heating and cooling systems, while Ventriglia (2011) simulated a type of brain activity hypothesized to be important for learning and memory.

The design space covered by the simulation may be so large, or the time to run each simulation instance so long that efficient and intelligent experimental design can be just as important as in traditional physical experiments. There are several unique issues relating to computer experimentation however, that separate it into a branch of research of its own. Some types of simulation models (such as FEA or CFD) will provide a deterministic response, such that any run with the same levels of the input factors will provide the same answer. In the deterministic context, the presence of no natural error means that there is no additional information to be gained from replicates, and if factors are found to be insignificant the collapsing of the design may result in unintended replication of design points in terms of the remaining factors. Traditional statistical analysis methods that rely on error estimates are also inappropriate in such situations.

Even in the case of a stochastic response, the variance may be small. It is common to apply variance reduction techniques (such as common random numbers or importance sampling) to improve the efficiency of the simulation. In the presence of small variance, the same issues that impact experiments with deterministic results still apply.

This research looks at designs that seek to find a compromise between traditional designs from the design of experiments world and space-filling designs that are prevalent in the computer experiments world. Properties such as prediction variance and residual analysis will be used to evaluate the balance between good properties for modeling and good projection properties in the case of insignificant factors.

A literature review will be presented in the next chapter, in which topics relating to the design and analysis of computer experiments are discussed. Chapter 3 describes hybrid space-filling designs, Latin hypercube designs that have been augmented with I-optimal points prior to experimentation. Chapter 4 details properties of bridge designs, which are Latin hypercube designs that are D-optimal subject to a minimum distance between points in any one-dimensional projection, and compares their performance to other common designs. Chapter 5 assesses the prediction variance properties of bridge designs as they are augmented with I-optimal points, and in particular evaluates whether this type of augmentation may be used to mitigate problems associated with model misspecification. Finally, Chapter 6 presents conclusions and suggestions for future research.

CHAPTER 2 – LITERATURE REVIEW

The literature relating to computer experiments can be classified into two main subsets. One arm deals with the design of the experiments themselves. This work concentrates on two major design types and their combination, namely optimal designs and space-filling designs. The second subset focuses on the analysis of the results of the computer experiments. The modeling methods discussed here include polynomials and Gaussian process modeling. The chapter concludes with a review of augmentation strategies for computer experiments.

Design of Computer Experiments

Traditional Response Surface Methodology Designs

The field of design of experiments has a rich history, with many classical designs shown to be efficient for factor screening and optimization (Montgomery (2009)). Most commonly known are factorial and fractional factorial designs, where the factors are evaluated at all (or a fraction) of the combinations of the endpoints of their ranges. If there is curvature anticipated in the response surface, a design with a greater number of levels will be employed, such as a central composite design.

As described in Myers, Montgomery and Anderson-Cook (2009), optimal designs are computer-generated designs that are created to provide the best solution given a set of constraints imposed by the practitioner. The constraints may include (but are not limited to) the sample size, the intended analysis model, and the allowable ranges of the factors. Implicit in the intended analysis model is the number of factors, and the number of levels necessary for each factor. Typically there are one or more objectives to be optimized, chiefly dependent on the intended analysis model. There are several common criteria, with the letter-based naming convention giving rise to the term alphabetic optimality. The most commonly used criterion for screening designs is D-optimality, in which the goal is to maximize the determinant of the information matrix. The determinant is inversely proportional to the squared volume of the confidence region around the regression coefficients. Minimizing the volume of the confidence region increases the confidence in the coefficient estimates. In this way, maximizing the determinant of the information matrix leads to a design that will have good parameter estimation properties. I-optimality is another attractive criterion, seeking to place design points such that the average prediction variance is minimized over the design space with regard to the intended analysis model. Myers, Montgomery and Anderson-Cook (2009) discuss the I-optimal design with respect to the linear regression model in detail. I-optimality tends to be used in cases in which the form of the expected model is more understood, and greater numbers of terms or higher order terms are expected to be included. Other common optimality criteria include A-optimality and G-optimality. A-optimality attempts to minimize the average variance of the estimated regression coefficients by minimizing the trace of the inverse of the information matrix. G-optimality minimizes the maximum diagonal entry in the hat matrix, in order to minimize the maximum prediction variance across the design space.

Replicates are often included in optimal designs to ascertain estimates of random error inherent in the system. In the presence of insignificant factors, the designs often collapse to give greater numbers of replicates in the remaining factors, which help to further refine the estimates in experiments with noise present.

In the world of computer experiments however, the literature tends to divide into two camps, based on whether the simulation model provides deterministic or stochastic responses. There are many types of simulations that are deterministic in nature (e.g., FEA and CFD), where the response at a given set of input values will always return the same value. In the deterministic context, the presence of no natural error means that there is no additional information to be gained from replicates, and if factors are found to be insignificant the collapsing of the design results in the 'wasting' of the additional runs. Other types of simulations (e.g., discrete-event simulation) result in a stochastic response, where the response at a given set of inputs represents a sample from a random distribution. The methodology described in this work is generally more appropriate for application to cases with a stochastic response.

Space-Filling Designs

While traditional response surface designs are quite popular for physical experiments, other design methods have become prevalent in the world of design for computer experiments. Avoiding the problems with irrelevant replication and the impracticality of factorial designs at high dimensionality, space-filling designs such as sphere-packing designs or Latin hypercubes in particular have gained popularity. These designs attempt to fill the interior portion of the design space.

The Latin hypercube is one of the most popular designs for deterministic models. The Latin hypercube design was first proposed by McKay, Beckman, and Conover (1979), and has been widely used in the field of computer experiments ever since. It is defined in Fang, Li, and Sudjianto (2006) as, "A Latin hypercube design (LHD) with n runs and s input variables, denoted by LHD(n,S), is an n x s matrix, in which each column is a random permutation of {1, 2, ... , n}."

A Latin hypercube sample is an extension of stratified sampling, dividing each of the s variables into n partitions of width $1/n$ and ensuring that a single design point is placed in each of the n divisions for each of the s variables. In the case of a midpoint LHD, the design points are placed in the center of each of the partitions, and in the general case the location of the design points within the partitions is determined based on a random draw from a uniform distribution. An example of a Latin hypercube design with ten runs and two factors is presented in Figure 1.

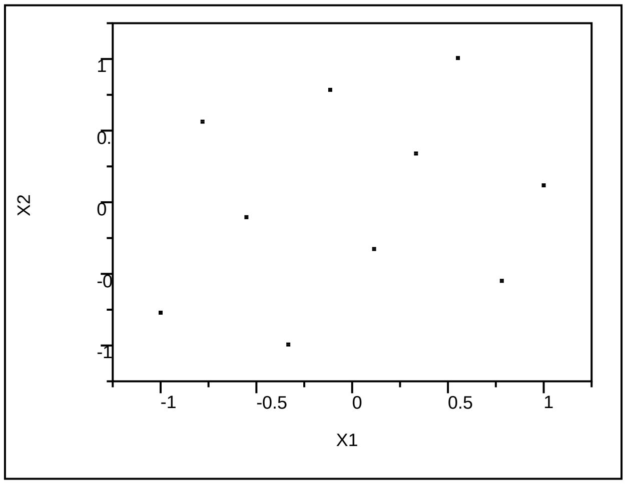

Figure 1 . Example LHD(10, 2)

In the original proposal, McKay, Beckman, and Conover (1979) investigated three methods for selecting the input values for a simulation study: random sampling, stratified sampling, and Latin hypercube sampling. Empirically, they found that all three sampling methods would yield unbiased estimates of the mean of the response, and while they did not have a direct means of comparing the variance of the mean from the LHD, they found that if the functions are monotonic in each of the arguments then the variance of the response mean as obtained with the LHD will be smaller than that of the random sample. The results of their experiments confirmed that the estimates of the mean response were comparable, and that while the variance of the sample mean determined from the stratified sampling was consistently lower than that of the simple random sample, the results based on the Latin hypercube designs consistently had the smallest variance.

McKay, Beckman, and Conover (1979) note that a great advantage of a LHD is that each of the components are fully stratified, which is beneficial in the case of a sparsity of effects. LHDs are easy to construct and ensure that the full range of each variable is explored. They showed that the mean response of a LHD will have a smaller variance than that of a simple random sample.

Without additional constraints, however, a randomly generated LHD can have poor coverage of the design space. In the most extreme case, a design could be generated in which the input variables were perfectly correlated, such as the LHD(10, 2) displayed in Figure 2.

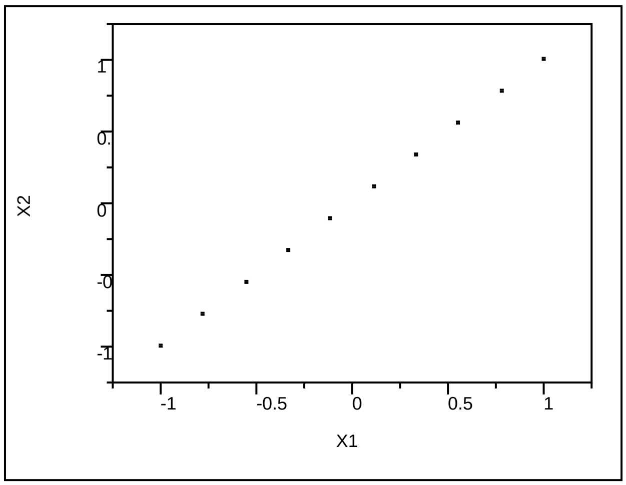

Figure 2. Example of an LHD(10, 2) with poor space-filling properties.

Fang, Li, and Sudjianto (2006) recommend examining bivariate scatter plots, and generating a new LHD in the case that the plots do not ‘look reasonably uniform.’ Due in part to the subjectivity associated with examining scatter plots and in part due to the fact that while the variance of the sample mean is comparatively small it is not minimized, many researchers have suggested modifications to the LHD. A taxonomy of the designs to be described is presented in Figure 3. The modified LHD are organized in a logical flow of their development, with other relevant designs or publications included where appropriate for chronology.

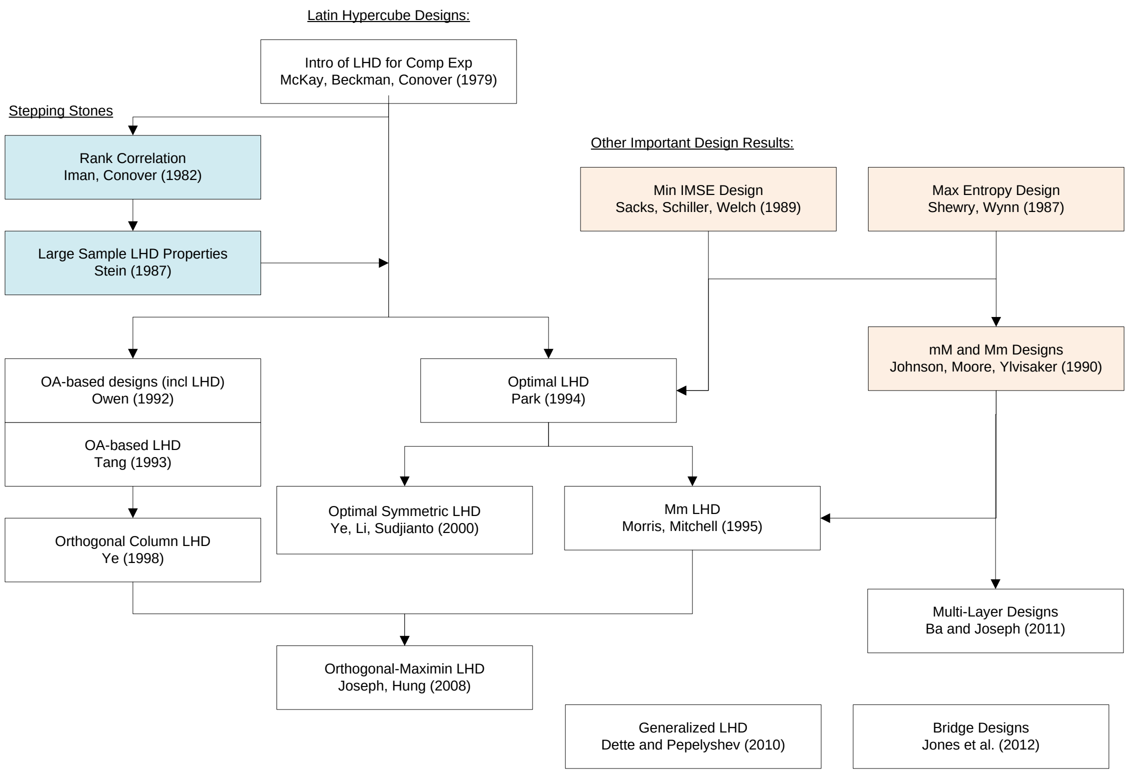

Figure 3. LHD Taxonomy.

Stepping Stones

Iman and Conover (1982) did not propose a new type of design, but they noted that although much effort was being expended in the development of designs and analysis methods for computer experiments, most of the research presented was based on the assumption of independent input variables. They introduced a method to induce dependencies amongst the input variables that can be applied to any sampling scheme, including LHD. Their method is distribution-free, simple to implement, and preserves the marginal distributions of the sampling scheme. The algorithm rearranges the values in each column of the n x s matrix such that the correlations between the input variables will approximate the desired correlation matrix supplied by the user. The authors note that their method can also be used to produce a sample rank correlation matrix that more closely resembles an orthogonal design matrix than a randomly generated design, if the inputs are intended to be independent.

Stein (1982) likewise did not propose a new design type, but rather investigated the large sample properties of LHDs. He confirmed the results of McKay, Beckman, and Conover (1979) that the variance of an estimator based on Latin hypercube sampling is less than that of a random sample, and that the size of the reduction is increased if the function being estimated is additive. He also noted that the results most individuals deal with depend on the assumption of independence of the input variables, and that the method of inducing correlation proposed by Iman and Conover (1982) does not hold up in the case of large N. He proposes a method that uses both the rank correlations and the joint distributions of the input variables to obtain a design that properly accounts for the dependencies amongst the variables at large sample size, but notes that additional simulations may be necessary to determine the necessary input information. In the application presented in the paper, 10 initial simulations were run to gain the information necessary for the variable transformations.

Orthogonal-Array Based Latin Hypercubes

The property of orthogonality has long been exploited in traditional design of experiments, meaning that the columns of the design matrix are un-correlated. However, many of the most common designs that exhibit orthogonality (such as factorials or fractional factorials) result in replicated points in the case of effect sparsity. Orthogonal array-based LHDs (OA-based LHD) were investigated by Owen (1992) and Tang (1993) contemporaneously but independently. Owen and Tang each sought to apply the orthogonality concept to LHD, hoping to achieve good balance not only amongst the variables themselves, but in the combinations of the variables. This results in designs enhanced to not only fill the space in a single dimension, but in higher dimensional space as well.

Tang (1993) states that "an n x m matrix **A**, with entries from a set of $s \geq 2$ symbols, is called an OA of strength r, size n, with m constraints and s levels if each n x r submatrix of **A** contains all possible 1 x r row vectors with the same frequency λ. The number λ is called the index of the array; clearly $n = \lambda s^r$. The array is denoted by OA(n, m, s, r)." By this definition, every LHD is already an OA of strength one (OA(n, m, n, 1)). Both Owen (1992) and Tang (1993) propose algorithms that begin with an existing orthogonal array, and permutes each of the entries within each column to create the OA-based LHD. The algorithm proposed by Tang (1993) will result in smaller variance of the mean response than that of Owen (1992) when the underlying function is additive.

Owen (1992) notes that arrays of strength more than 2 will require larger sample sizes than may be of practical use. Tang (1993) notes that OA-based LHD are not necessarily unique, and that two OA-based LHD are equivalent if one is a permutation of the rows, columns, and symbols of the other.

Orthogonal Latin Hypercubes

Orthogonal LHDs were proposed by Ye in 1998. In an orthogonal LHD, any pair of columns are orthogonal to one another (i.e., have zero correlation). Orthogonal LHDs ensure that the estimates for linear and quadratic effects are independent, and that the linear interactions are uncorrelated with the linear effects. The proportions of the design are strict – the number of runs must either be a power of 2 or a power of 2 plus one, and the number of columns s for a design of $n = 2^m$ or $2^m + 1$ runs is $s = 2m - 2$. The algorithm proposed by Ye (1998) constructs the first 2^{m-1} runs, and then reflects them to generate the second half of the design. In the case of $n = 2^m + 1$ runs, a center point is added to the design. Since the orthogonality of the columns does not necessarily guarantee good space-filling properties, it is recommended to generate permutations of the design (which can be easily done by reversing the signs of a subset of columns – analogous to reflecting the design over a hyperplane) and evaluating the properties of the permutations against a criterion of interest in order to select the best design. The advantage of the orthogonal LHD over an OA-based LHD is the ease of construction, but the inflexibility of the design proportions means that designs may grow too large for practicality in the case of large numbers of input variables.

Steinberg and Lin (2006) propose a construction method for orthogonal LHD that builds from factorial designs. Their algorithm rotates separate groups of factors and rescales them to a unit hypercube, which results in a LHD when they are all combined. Their construction method leads to larger orthogonal LHD than presented by Ye, but they are still severely limited as to sample size. The number of runs must be equal to 2^k, where k is also a power of 2.

Bingham, Sitter, and Tang (2009) do not restrict their scope to LHDs alone, but they present a construction algorithm for orthogonal designs that ameliorates the sample size restrictions of the other methods presented. They take the Kronecker product of an orthogonal array with two levels (±1) and another orthogonal array, which will then itself be an orthogonal array. This allows for the extension or expansion of existing orthogonal LHDs beyond their originally constrained sizes.

Optimal Latin Hypercubes

There is a long tradition of optimal designs in the design of experiments world, as described previously. Often, designs are optimized for a particular model, such that knowledge is required about the underlying model and variable relationships during the design process. Shewry and Wynn (1987) used entropy as an optimality criteria, in order to 'maximize the amount of information in an experiment.' In the case of a Kriging model, maximizing the entropy can be done by maximizing the log of the determinant of the design correlation matrix (similar to the method used to generate a D-optimal design in traditional designed experiment). Sacks, Schiller, and Welch (1989) propose the integrated mean squared error (IMSE) critiera, setting the design points so as to minimize the variance of prediction. Since the IMSE is dependent on the parameters of their model, however, they recommend either performing a robustness study to choose model parameters that will perform well over a wide range, or to design for asymptotic values of the parameters. They note that the design optimization process (using a quasi-Newtonian algorithm) is computationally intensive, particularly since the correlation matrix can be poorly conditioned in the case of small values of the correlation parameters. Currin, Mitchell, Morris, and Ylvisaker (1991) used the entropy criterion of Shewry and Wynn (1987) in their design generation. They used an excursion-based optimization algorithm to generate entropy-optimal designs, under the assumption that all the correlation parameters are equal.

Park (1994) applied the optimality concept to LHDs. He used a two-stage algorithm, first using an exchange-type algorithm to obtain an optimal midpoint LHD, and then relaxing the midpoint requirement to optimize the design with respect to criterion within the neighborhood using a Newtonian routine (which should not take long, given the narrow search region). Given the two-stage nature of the algorithm, there is the chance that it will not return the 'true' optimal design, but it expected that it will not be too far when the number of runs is large in comparison to the number of variables.

Park (1994) finds that in creating an optimal LHD for the maximum entropy criterion, the Latin hypercube structure prevents the replication of corner points at the edges of the design space typical of the criterion. An optimal LHD generated to minimize the IMSE helps to prevent the clustering of design points. He found that the designs are have good geometric properties (generally nearly symmetric), and are much more efficient than randomly generated LHDs. Since one of the greatest benefits of a LHD is that the generation is cheap, he notes the need for a faster algorithm to generate the optimal designs. Since the IMSE criterion requires the inversion of the correlation matrix as opposed to the entropy criterion which simply requires the determinant, it may be more efficient to use the entropy criterion.

Jin, Chen, and Sudjianto (2005) presented an enhanced stochastic evolutionary (ESE) algorithm in order to help more efficiently construct optimal designs. They also present more efficient ways of computing the optimality criteria, which further promotes the efficiency of the design process. They find that their algorithm is able to find good medium to large-sized designs within minutes (if not shorter), while retaining the desirable properties such as balance or orthogonality. For small-sized designs, there may not be time savings realized. They describe small designs as 12-24 runs in four factors, and medium to large designs as 50-100 runs in five to ten factors.

Maximin Latin Hypercubes

A special type of optimal LHD uses the maximin criteria in the optimization process. Johnson, Moore, and Ylvisaker (1990) originally investigated designs based on the minimax (mM) or maximin (Mm) criteria. A mM design is one in which the maximum distance from any design point to any point within the design space is minimized, similar to a covering problem. A Mm design is one in which the minimum distance between any two design points is maximized, as in a packing problem. The maximin distance criterion maximizes the minimum inter-site distance and is specified by

$$max_D \; min_{u,v \in D} \; d(u,v) = \; min_{u,v \in D_*} \; d(u,v),$$

where d(u,v) is a distance, which is greater than or equal to zero, and D represents the design points**.** The Mm criterion proves to be the same as the entropy criterion, achieved by minimizing the determinant (or maximizing the log of the determinant) of the covariance matrix as in D-optimal designs.

Morris and Mitchell (1995) applied the Mm criterion to LHDs. They extend the definition of a maximin design in order to allow for tiebreaking, and suggest a scalar-valued criterion φ_p so they can rank competing designs. Their algorithm uses simulated annealing, beginning with a randomly generated LHD, and 'perturbs' the design by interchanging two randomly chosen elements within a randomly chosen column of the design matrix. The results they present show that the Mm LHD is superior to both a Mm design and a randomly chosen LHD in terms of both mean squared error and maximum prediction error after modeling a response surface with a Gaussian process model.

Optimal Symmetric Latin Hypercubes

Symmetry or near symmetry in optimal LHDs was observed by both Park (1994) and Morris and Mitchell (1995). Ye, Li, and Sudjianto (2000) suggest the symmetric LHD as a good compromise design, yielding good geometric properties with greater ease of construction. The design actually takes three criteria into account: the projection properties guaranteed by the Latin hypercube structure, the orthogonality properties imparted by the symmetry of the design, and the space-filling properties of the maximin criterion used in the search algorithm. They define a symmetric LHD (SLHD) as a LHD in which each row has a mirror image twin (reflected about the center). SLHD have some orthogonal properties (each main effect is uncorrelated with all two-factor interactions and quadratic terms), but the sample size is flexible. Ye, Li, and Sudjianto (2000) present a columnwise-pairwise (CP) exchange algorithm, where two simultaneous pair exchanges are made in each column in order to maintain symmetry (in the case of odd-numbered experiments, the centerpoint does not participate in the exchange). They compare the CP algorithm to the exchange algorithm of Park (1994), and find that CP consistently performs better. They also compare CP to the simulated annealing algorithm of Morris and Mitchell (1995), and find that while CP is more efficient in the case of smaller designs, the simulated annealing algorithm does work better in the case of large designs. In the example presented, the smaller designs consisted of 12 runs in two factors, while the larger designs were 25 runs in four factors.

Orthogonal-Maximin Latin Hypercubes

Joseph and Hung (2008) propose a multi-objective criterion for optimized LHDs. Orthogonal designs have attractive properties, as do maximin designs. They note that intuitively, one would expect that a design with low correlation amongst the columns would spread points out within the design space, and that concurrently, spreading points out within the design would reduce correlation. In truth however, they find that there is no definite relationship between the two, and that designs generated with the different criteria can vary greatly. Owen (1994) proposed a performance measure for LHDs that takes the root mean square of all pairwise correlations between columns of the design, ρ^2, which should be minimized to gain a design with the best orthogonality properties. Morris and Mitchell (1995) defined their criterion, φ_p, so that the design with the minimum value will identify the design with the best spread of design points. Joseph and Hung (2008) suggest that the two objectives be combined in a weighted sum, ψ_p, and propose an algorithm that minimizes ψ_p.

Their algorithm is a modification of the simulated annealing algorithm from Morris and Mitchell (1995), selecting promising exchanges by choosing a column that is highly correlated with others, and a row that is closest to other design points. They compare their orthogonal-maximin LHDs with the maximin LHDs proposed by Morris and Mitchell (1995) as well as orthogonal LHDs as proposed by Ye (1998) in terms of the three criteria (ρ^2, φ_p, and ψ_p), and find that their designs are a good compromise between the two comparators. The efficient selection of exchanges also enables their algorithm to converge quickly.

Generalized Latin Hypercubes

Dette and Pepelyshev (2010) note that in uniformly spaced Latin hypercube designs, the concentration of design points is greatest in the center of the design and decreases as the boundaries are approached. They argue that this reduction of available information around the boundaries serves as a motivation for shifting some of the design points towards the edges of the experimental region, similar to an optimal or factorial design in traditional design of experiments. They propose to ameliorate the problem by generalizing the Latin hypercube by taking a transformation of the points using the quantile function of a Beta density. The tuning parameter can then be used to specify the importance of the boundaries, such that low importance results in the original uniformly spaced design. The Integrated Mean Square Error associated with the generalized design was found to be much lower than that of a maximin Latin hypercube design in models with a stochastic term. The generalized design also evens the mean squared error throughout the design space, which is typically small in the center and large in the outer regions of a typical Latin hypercube.

Multi-Layer Designs

Ba and Joseph (2011) propose the conversion of optimal factorial designs into space-filling designs called multi-layer designs (MLDs) in order to make them appropriate for computer experiments. The geometric properties of the factorial designs help to reduce the time needed to produce the optimal space-filling design as compared to other methods.

The base design for the construction of the MLD should be selected carefully. A full factorial design with p factors two levels each would have $n = 2^p$ runs. If the sample size for a full factorial is prohibitive, a 2^{-k} fraction can be chosen, with $n = 2^{p-k}$ runs. The authors recommend selecting a fractional factorial design based on the minimum aberration criterion of Fries and Hunter (1980). The design with the minimum number of words of the shortest length in the defining relation is the design with minimum aberration, ensuring that there is the smallest number of aliased effects.

The base design is then split optimally into subgroups for allocation to different layers. The methodology the authors propose is essentially a reverse foldover, where instead of reversing the signs of one (or more columns) to double the experimental runs, it is done backwards to halve the design (such that one half is the foldover of the other). This procedure can be repeated multiple times to split the design into L layers.

The authors propose that $L = n/2$ layers will result in the most desirable projection properties. Since points are split into layers using the minimum aberration criteria, the majority of factors will have n levels, with only a few having $n/2$ levels. This will lead to one-dimensional projection properties nearly as good as LHDs, although perhaps not as uniform depending on the choice of spacing between the layers.

For spacing between the layers, the authors recommend keeping an empty area at the center of the design. Leaving a hypercube of size $(-s, s)^p$ within the $[-1, 1]^p$ hypercube of the design space since the volume of the layers increases as they get closer to the outer boundaries of the design space. Experimentation revealed that a value of $s = 0.45$ provides a nice balance between the maximin and minimax evaluation criteria.

One constraint of MLDs is their restrictive sample size. For MLDs constructed by splitting 2^{p-k} designs into layers, the sample sizes are limited to powers of 2. If there is budget for more points, the easiest way is to design the largest MLD allowable within n, and then add the additional $n - 2^{p-k-1}$ points. The authors recommend using the methodology for optimal foldover plans described by Li and Lin (2003), and subsetting the appropriate optimal foldover plan to the number of additional points that can be added. These extra points get arranged into their own layers, and added to the original design for scaling.

Bridge Designs

Similarly to the generalized Latin hypercubes proposed by Dette and Pepelyshev (2010), bridge designs are introduced by Jones, Johnson, Montgomery, and Steinberg (2012) as an attractive way to merge the properties of Latin hypercube designs and optimal designs. The designs bridge the gap between Latin hypercubes and D-optimal designs, guaranteeing a minimum distance between points in any projection, and D-optimality with respect to the specified analysis model subject to that constraint. Their algorithm begins with a Latin hypercube, and uses point exchange to maximize the determinant of the information matrix.

With *n* experimental points, each with a range of [-1, 1], the interpoint distance δ of any projection can take values between $0 \le \delta \le 2/(n-1)$. At $\delta = 0$, there is no constraint on the interpoint distance and a true D-optimal design can be returned. The larger the δ value becomes, the more the design resembles a Latin hypercube, until at $\delta = 2 / (n-1)$ it becomes a D-optimal Latin hypercube. The authors recommend a more flexible setting, letting $0 \le \delta \le 1 / (n-1)$.

The algorithm requires four inputs to be set: the number of factors, the number of runs, an *a priori* regression model intended for analysis, and the minimum spacing between points in any projection. Once the inputs are set, a grid of candidate design points is generated based on δ. The starting point for the design is a random Latin hypercube design, and a coordinate exchange algorithm is used to evaluate whether the determinant of the design can be improved upon by replacing one of the design points with a candidate grid point. Since the algorithm only considers grid points and not all points within the design space, there is some degree of approximation in the solution, but it is minimal.

In comparing the bridge designs to other types of LHDs such as multi-layer designs and maximin Latin hypercubes, the bridge designs have better D-efficiency as well as average variance. The algorithm does not attempt to maximize the distance between points, so the other designs do perform better in terms of minimum distances between design points. The bridge designs also have an advantage over the multi-layer designs in terms of fitting higher-order models. The bridge design is proposed for use in situations where the experimental error is small in comparison to the factor effects. In this case, the minimization of the variance incurred by the optimization of the design can be secondary, and the resulting increase in factor levels an appropriate trade. This allows for greater flexibility in modeling choices, in the addition of regression terms or in other models such as the Gaussian process model.

Comparison

An ordinary randomly generated LHD can be vastly improved upon, but there is no one best choice for all situations. There are several types of LHDs that are model-dependent, useful in cases in which the experimenter has good hypotheses as to the form of the model prior to experimentation. Non-model dependent optimality criterion are more generally applicable, since the models are not known in advance of the experimental design, however, can still be a good starting point if the class of models is known to be likely. The maximin criteria is the most commonly used, since it is fairly easy to implement with the various algorithms that have been developed and yields good space-filling properties. Orthogonality is also a nice property in terms of model fitting, and that if it can be achieved within reason, it is likely to be helpful. The original orthogonal designs proposed by Owen (1992), Tang (1993) and Ye (1998) carry difficulty of construction and are restrictive in terms of design size. Multi-layer designs as proposed by Ba and Joseph (2011) also provide nice properties, but are restrictive in terms of sample size and model fitting. Bridge designs as introduced by Jones, Johnson, Montgomery, and Steinberg (2012) are good options for situations in which a polynomial is deemed to be a good candidate model for analysis.

Analysis of Computer Experiments

As important as the design of a computer experiments is, the analysis of the output is equally significant. Computer experiments can be computationally expensive in terms of time required to run an experiment on the simulation model. As an example, each of Ventriglia's (2011) simulations took four days to complete. Therefore, surrogate models or metamodels are often used to mimic the input-output relationship in the form of a simpler mathematical expression that can be quickly computed. Surrogate models encompass a broad range of methods ranging from parametric to nonparametric analysis, and the type of experimental design chosen should be done in context of what type of surrogate model is intended to be employed. Both Santner (2003) and Fang, Li, and Sudjianto (2006) provide good review information on the issues of design and subsequent analysis of computer experiments.

There are many publications that evaluate experimental designs and analysis methods for computer simulations. Since the appearance of the seminal paper by Sacks, Schiller, and Welch (1989), the most prevalent methods studied are variants of Kriging. Hussain, Barton, and Joshi (2002) tested two different types of metamodels for their designs, a radial basis function originally developed to fit irregular topographic contours of geographical data, and quadratic polynomial models. Allen, Bernshteyn, and Kabiri-Bamoradian (2003) compared combinations of experimental design classes with respect to second-order response surfaces and Kriging modeling methods. Bursztyn and Steinberg (2004) develop a new method of design comparison based on a Bayesian interpretation of an alias matrix. Chen, Tsui, Barton, and Meckesheimer (2008) discuss various designs used for computer simulation models and seven surrogate modeling methods, including response surface methodology, spatial correlation models (Kriging), multivariate adaptive regression splines, regression trees, artificial neural networks, and least interpolating polynomials. Ankenman, Nelson, and Staum (2008) discusses the application stochastic Kriging for stochastic simulation output in particular, employing methodology to separate out the intrinsic and extrinsic uncertainty.

Kriging models are not the only appropriate models, however. High-order polynomials can approximate complex surfaces as well, and Bingham, Sitter, and Tang (2009) note that many researchers are increasingly interested in them. Polynomials are attractive due to the ease of fitting, and interpretability. They can also be built up gradually, adding terms to reduce prediction error.

Barton and Meckesheimer (2006) note that high-order polynomials can be non-robust or over-fit in some situations where higher order terms added to fit certain areas of the response can cause the response to be overshot in other areas. They note that regression splines make a good compromise for that problem, fitting lower order polynomials to local areas and requiring continuity at the edges.

Other modeling methods have been proposed, such as radial basis functions or neural networks. Radial basis functions are quite sensitive to scaling and experimental design, however. Neural networks are difficult to fit, and very difficult to interpret.

The two modeling methods employed for this work are traditional linear regression models (polynomials) and Gaussian Process (GASP) models (described in the next sections).

Polynomial Models

Linear regression modeling as described in Montgomery, Peck, and Vining (2012) can be used to fit the polynomials that would approximate the response surface, describing the relationship between the independent variables (or factors) and the dependent variable (or response). For example, a full second-order model in two factors would have six parameters (or β's) to estimate, and take the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \varepsilon$$

while if a fifth-order model was to be fit with two factors, it would have 21 parameters:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \beta_6 x_1^2 x_2 + \beta_7 x_1 x_2^2 + \beta_8 x_1^3 + \beta_9 x_2^3 + \beta_{10} x_1^3 x_2 + \beta_{11} x_1 x_2^3 + \beta_{12} x_1^2 x_2^2 + \beta_{13} x_1^4 + \beta_{14} x_2^4 + \beta_{15} x_1^4 x_2 + \beta_{16} x_1 x_2^4 + \beta_{17} x_1^3 x_2^2 + \beta_{18} x_1^2 x_2^3 + \beta_{19} x_1^5 + \beta_{20} x_2^5 + \varepsilon$$

The method of least squares is commonly used to estimate the β's, solving the set of n least squares normal equations to minimize the sum of the squares of the errors, ε_i. Typically, the model can be written in terms of matrices, as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

If n represents the number of observations, k the number of independent factors (including one factor for each of the x's, plus one for each interaction and higher order term), and p the number of parameters to be estimated, $\mathbf{y}$ is an n x 1 vector of the response observations, $\mathbf{X}$ is an n x p matrix with the levels of the independent variable expanded to model form, $\boldsymbol{\beta}$ is a p x 1 vector of the fitted coefficients, and $\boldsymbol{\varepsilon}$ is an n x 1 vector of random errors associated with each of the observations.

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}, \text{ and } \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

The solution to the least squares normal equations can be written as:

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

Various hypothesis tests can then be used to determine the significance of the model, or the individual regression coefficients, and so on. In addition, residuals (the difference between the observed values and the predicted values obtained from the model) can be examined for patterns that may indicate lack of fit.

Gaussian Process Models

Kriging is an interpolation technique named for D.G. Krige, a mining engineer who began developing the technique in order to predict locations of ore deposits. It works as an approximation method that attempts to predict unknown locations based on a random process, assuming that the closer the input values the more positively correlated the prediction errors will be. It is particularly attractive in the world of deterministic computer experiments, because as an interpolator it guarantees that the model will match the observed output at each of the observed points. It has been applied successfully to stochastic cases as well, both in the same form and with other mathematical modifications. One example of modified Kriging for stochastic models is presented by van Beers and Kleijnen (2003), who applied what they call 'detrended Kriging,' where the data go through a preprocessing step to prepare it for ordinary Kriging. Linear regression is used in an attempt to separate the signal from the noise.

The most common form of Kriging applied in the field of computer experiments is usually known as Gaussian process modeling. It assumes that the form of the underlying random process follows a multivariate normal distribution. The response is represented as an n x 1 vector, $\boldsymbol{y}(\boldsymbol{x}) \sim N\left(\mu \mathbf{1}_n, \sigma^2 \boldsymbol{R}(\boldsymbol{X}, \boldsymbol{\theta})\right)$. With **R** representing the correlation matrix of the observed points (dimension *n* x *n*), and **r** as the vector representing the correlation of the point to be estimated with the design matrix, the fitted model form is:

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}'\left(\boldsymbol{x}, \hat{\boldsymbol{\theta}}\right) \boldsymbol{R}^{-\mathbf{1}}\left(\boldsymbol{X}, \hat{\boldsymbol{\theta}}\right)\left(\boldsymbol{y} - \hat{\mu} \mathbf{1}_{\boldsymbol{n}}\right)$$

The correlation matrix can take on several forms. One commonly used form is called the product exponential, given by:

$$R_{ij}(X, \theta) = exp\left(-\sum_{k=1}^{p} \theta_k \left(x_{ik} - x_{jk}\right)^2\right)$$

while the estimated correlations of the unobserved $y(x)$ at a new value of explanatory variables becomes:

$$r_i(x, \hat{\theta}) = exp\left(-\sum_{k=1}^{p} \theta_k (x_k - x_{jk})^2\right)$$

While not technically correlations, the θ_k values do describe the correlation in each of the *k* directions. They must be greater than or equal to zero. When θ_k is equal to zero, there is no effect of the k^{th} factor and the fitted surface will be flat in that direction. As θ_k grows larger, correlation becomes smaller in the *k* direction and the fitted surface will become rougher in that direction. The models are usually fit using maximum likelihood estimation (MLE).

The idea that the data collected is a realization of a stochastic process makes sense in most situations, and the idea of fitting a surface through the observed points is intuitive. There are also several nice properties associated with the Gaussian process that make it attractive for modeling. It is infinitely differentiable with probability 1, given the proper correlation structure. The optimal predictor and the optimal linear predictor are identical in terms of squared-error loss, whereas non-Gaussian assumptions typically have non-linear optimal predictors.

Different variants of the GASP model may be more appropriate in different situations. Typically the type of GASP model that is applied is ordinary kriging, where the response is modeled by a constant mean term added to a stochastic process. Universal kriging or blind kriging (Joseph, Hung, and Sudjianto (2008)) could be good extensions, where a linear model is substituted for the constant mean. In universal kriging the terms of the model are set, and in blind kriging model fitting procedures are used to determine which terms should be included. Each extends the capability of the ordinary kriging model, at the expense of complexity of model fitting.

Augmentation of Designs for Computer Experiments

Building up information sequentially through design augmentation is efficient and economical. Montgomery (2009) points out that it is almost always preferable to run a small pilot design, analyze the results, and then decide on the best set of runs to perform next based on the results. Design augmentation can also be used in computer simulation modeling to great benefit to the resulting models. Johnson, Montgomery, and Jones (2010) demonstrate that augmenting a space-filling design with optimal points can be effective in improving the prediction variance across the design region.

Other researchers have evaluated augmentation strategies with differing results. Kleijnen and van Beers (2004) augmented their initial pilot designs one point at a time, based on the assumption that simulation experiments must proceed sequentially anyway (unless parallel computers were used). After analyzing the pilot runs, they create a set of candidate points and choose to simulate at the location with the highest estimated variance. Ranjan, Bingham, and Michailidis (2008) also propose sequential augmentation in their algorithms meant to optimize the estimation of response surface contours, adding a single point in each cycle where the expected improvement in the contour estimation is maximized. Loeppky, Moore, and Williams (2010) point out that one-at-a-time augmentation is impractical, since typically the reason for building the emulator is that the run time of the simulation is prohibitive. They also note that it can tend to cluster points together, particularly when the minimization of the integrated mean squared error is used as the criteria for selecting the next design point. This clustering can lead to the degradation of model performance overall. In consequence, they introduce an algorithm which adds points in batches, choosing a value of 8 at a time. They included a fixed maximin Latin hypercube design at the full sample size as a comparator to the sequential strategy designs. In general, they found that the fixed design performed as well as the sequential designs for RMSE (highlighting the importance of filling the design space), although the sequential designs performed better in terms of maximum error.

Another important aspect of the augmentation problem is how the runs should be budgeted. Ranjan, Bingham, and Michailidis (2008) have empirical results that show that a good rule of thumb is to allocate approximately 25-35% of the runs to the initial design. Loeppky, Moore, and Williams (2010) acknowledge that the issue of allocation bears further study, but state in general terms that a minimum of a quarter of the full budget should be used for the initial design, with larger proportions dedicated to the initial design in the case of high dimensionality.

There are many different methods proposed for evaluation of augmented designs in the literature, and what is appropriate is somewhat dependent on the intent of the augmentation. Williams, Santner, and Notz (2000) used one-at-a-time augmentation to either maximize or minimize the response, and simply compared their designs' performance to the known test function. Kleijnen and van Beers (2004) compare their sequential design to a Latin hypercube sample in terms of the empirical integrated mean squared error and the $L\infty$ norm. Ranjan, Bingham, and Michailidis (2008) were interested in optimizing the ability to estimate a contour. They created three distance-based measures to compare their different design strategies in terms of how well they matched the true contour, and ultimately compared the performance of all the designs to that of a simple random Latin hypercube design (LHD). Loeppky, Moore, and Wiliams (2010) compared their batch sequential designs to fully sequential designs and a maximin LHD in terms of the RMSE and maximum error in fitting the Gaussian process model.

CHAPTER 3 – HYBRID SPACE-FILLING DESIGNS FOR COMPUTER EXPERIMENTS

Johnson, Montgomery, Jones, and Parker (2010) evaluate space-filling designs and optimal designs with respect to their performance when used to fit linear regression models. They compare designs based on their prediction variance. Their conclusions indicate that: 1) space-filling designs do not perform as well as optimal designs with respect to a linear regression model, 2) of the space-filling designs sphere packing designs generally have the lowest prediction variance followed closely by the Latin hypercube designs, and 3) augmentation of space-filling designs with I-optimal points is suggested whenever initial modeling indicates that the computer simulation model can be adequately approximated by a polynomial. Their last point suggests that hybrid designs, which combine both optimal points and space-filling points, have the potential to be powerful designs. We compare hybrid space-filling experimental designs based on their prediction variance with respect to linear regression models and the Gaussian process model, both theoretically and empirically.

Methodology

The hybrid space-filling designs are created by generating a space-filling maximin Latin hypercube design in n points and then augmenting that design with m I-optimal points. Latin hypercubes are commonly used in design for computer experiments, and are selected as the base space-filling design. They can be generated fairly easily for large sample sizes and large numbers of factors, and do not result in replicated runs if factors drop out due to lack of effect. The I-optimal points are determined assuming that the model the experimenter plans to fit is a polynomial of a specified order. To maintain comparability between different design compositions, sample size was kept constant within each set to be compared. For this chapter, designs were created to be saturated designs in terms of the full form of the polynomial when fully completed, with compositions ranging from a full LHD to a full I-optimal design.

To generate a space-filling design, no model specification is necessary, only the required number of points (sample size) is needed. To generate an optimal design (or to augment a design with optimal points), the intended form of the analysis model must be specified as well as the number of design points required. In order to test the predictive capabilities of space-filling designs and optimal designs when fitting a linear regression model we generated designs ranging from two to five factors and used second-order to fifth-order polynomials to generate the **X** model matrix. Table 1 illustrates the minimum number of design points needed to fit a given polynomial with 2 to 5 factors.

Table 1. Minimum number of design points needed (n = p).

	Order of Polynomial			
Factors	2	3	4	5
2	6	10	15	21
3	10	20	35	56
4	15	35	70	126
5	21	56	126	252

One hundred six (106) designs were generated, with several designs at each of the sixteen combinations of number of factors and polynomial order, holding sample size constant at the minimum number of design points needed to fit the full polynomial so that designs could be compared directly. Within each combination of number of factors and polynomial order, designs ranged from a full Latin hypercube to a full I-optimal design with two to five smaller Latin hypercubes augmented with I-optimal points. The notation used to identify designs throughout this work is LxIy_aF_bO, where *x* is the number of initial Latin hypercube points, *y* is the number of I-optimal points the Latin hypercube was augmented with, *a* represents the number of factors (x's), and *b* represents the order of the intended polynomial specified for the augmentation.

All results presented for the polynomial analysis, both theoretical and empirical, represent a worst-case scenario. The principle of sparsity of effects holds in most cases, in general meaning that systems are typically dominated by only a few main effects and lower-order interactions. In these cases however, no attempt to reduce the models have been made, and hence all results are presented for the full model. If fewer terms were included in the model the results for the polynomial analysis would be more favorable.

Results

Theoretical Prediction Variance

Here the hybrid space-filling designs are compared with strictly space-filling designs and strictly optimal designs using prediction variance over the design region.

The prediction variance is a standard criterion for comparing designs when modeling physical systems as well as computer simulations. If the intent is to analyze the data using a polynomial model, the scaled prediction variance (SPV) normalizes the prediction variance over the design region and is computed as

$$\frac{N\,V[\hat{y}(x_o)]}{\sigma^2} = N\boldsymbol{x}_o'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{x}_o$$

where $\mathbf{X}$ is the model matrix and $\mathbf{x}_o$ is the point being evaluated. In order to penalize designs with larger sample size (which are more ‘expensive’ to run), the scaled variance is multiplied by the number of runs (N), but this is unnecessary when comparing designs of the same sample size. Since deterministic computer experiments have no stochastic component it is necessary to justify the use of scaled prediction variance as a performance criterion. Suppose that a given computer experiment is adequately modeled using a polynomial fit. The difference between the observed and fitted values in a deterministic computer model, however, is not stochastic error, but rather is model bias. If the polynomial model adequately describes the response surface of the true underlying function, the model bias of the fitted betas is negligible. The model bias of an individual prediction is also fairly small because the fit is adequate. Assuming that the source of this bias is due to multiple high order terms, and deviations between the observed and predicted values will then behave like the sum of a number of independent small quantities. Appealing to the central limit theorem, as the number of these bias quantities gets large, these deviations will converge to the normal distribution. We then justify the prediction variance criterion as a measure of the sum of a large number of small biases.

In order to compare the various designs, test spaces with 10,000 uniformly distributed points were generated. The prediction variance was then calculated over the entire design space for each design, assuming the full form of the polynomial model. Comparing the designs based on summary statistics can be problematic, since designs with the same mean prediction variance could have very different profiles. Hence, rather than trying to balance comparisons of the mean or maximum prediction variances, they were sorted from smallest to largest, and plots similar to Fraction of Design Space (FDS) plots were generated. FDS plots graph the empirical distribution function of the prediction variance over the design space (Zaharan, Anderson-Cook, and Myers (2003)). They efficiently present a large amount of information, and allow for comparisons of design performance over the whole space rather than simply comparing designs based on summary statistics such as the mean or maximum. Because the plot does not address location within the design space, it is only appropriate in cases where the entire design space is of equal importance. If some regions are more interesting than others, a weighting scheme or partitioning should be applied. The FDS plot for the four-factor, second-order polynomial case is presented in Figure 4 as an example.

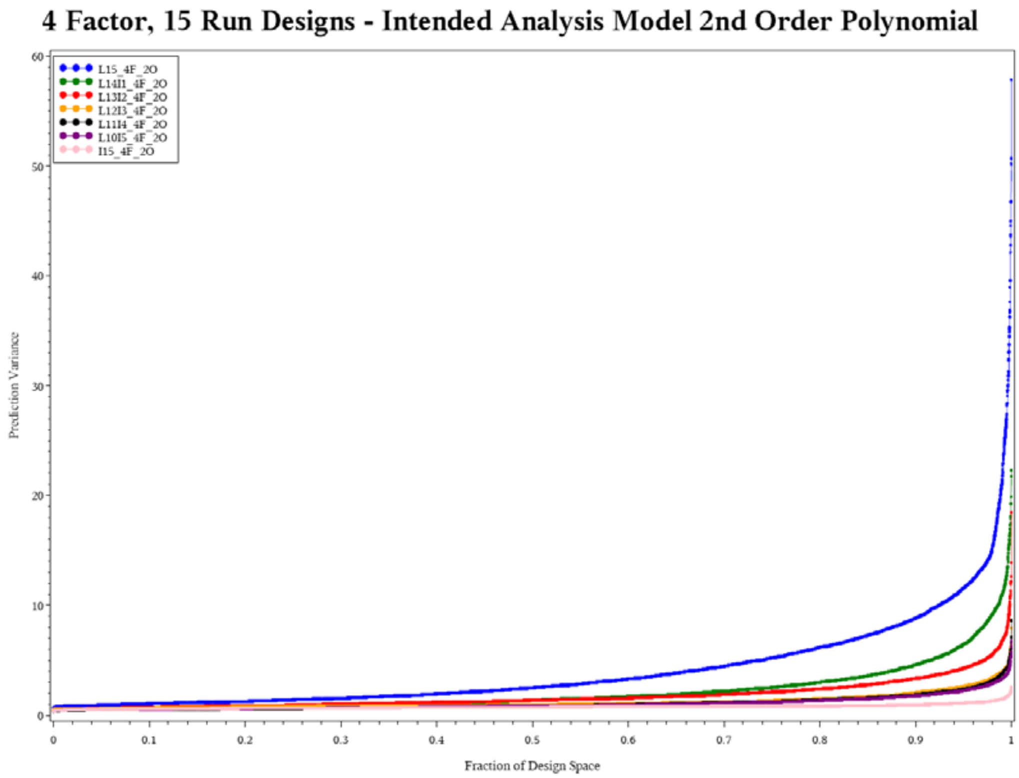

Figure 4. FDS plot for four-factor, second-order designs assuming a second-order polynomial.

As expected, the full I-optimal designs performed best in terms of prediction variance, since the I-optimal criteria minimizes the average variance of prediction over the design region (with respect to the hypothesized model form). The full Latin hypercube has the highest prediction variance across the whole design space. The prediction variance is visibly reduced as a single I-optimal point is included, and again as a second I-optimal point is included. The designs in which 3, 4, and 5 I-optimal points are included all perform similarly, and the minimum prediction variance over the design space is observed for the full I-optimal design. In general, the hybrid designs in the other design categories perform similarly.

It was noted across the different design categories that there were several outliers in terms of maximum prediction variance, particularly in the full Latin hypercube designs. For each of the two-factor designs, five different designs were created to attempt to assess design-related variability (only four designs were generated for the case with 4 Latin hypercube points augmented with 2 I-optimal points, since only four 4-run Latin hypercube designs are possible). The impact of design variability will be addressed in a following section.

For the GASP model, the relative prediction variance is dependent on the design points (and hence implicitly, the sample size and number of factors) and the unknown thetas:

$$\frac{var\left(\hat{y}(x)\right)}{\sigma^2} = 1 - r'\left(x, \widehat{\boldsymbol{\theta}}\right)R^{-1}\left(\boldsymbol{X}, \widehat{\boldsymbol{\theta}}\right)r\left(x, \widehat{\boldsymbol{\theta}}\right) + \frac{\left(1 - \mathbf{1}'R^{-1}\left(\boldsymbol{X}, \widehat{\boldsymbol{\theta}}\right)r\left(x, \widehat{\boldsymbol{\theta}}\right)\right)^2}{\mathbf{1}'R^{-1}\left(\boldsymbol{X}, \widehat{\boldsymbol{\theta}}\right)\mathbf{1}}$$

Since the unknown thetas are not known a priori, it is more difficult to evaluate designs in advance. As described in Loeppky, Sacks, and Welch (2008), the worst case for prediction occurs when the thetas are equal. For illustration, the designs were evaluated under the assumption that all thetas are equal at a value of 3.

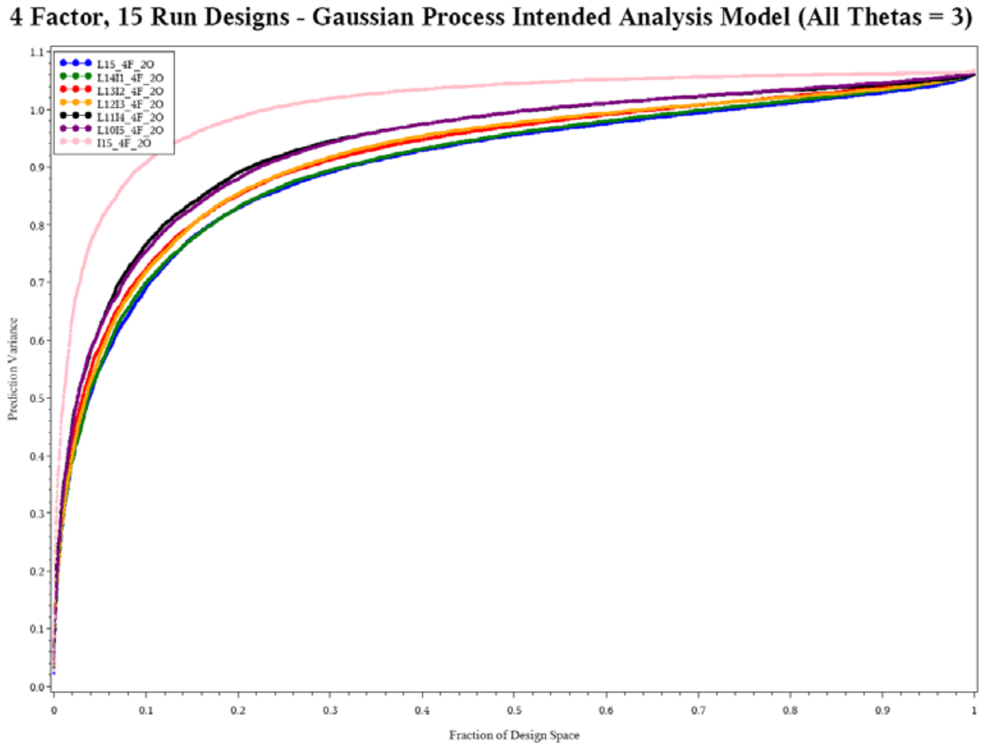

Figure 5. FDS plot for four-factor, second-order designs assuming a GASP model.

It can be seen in Figure 5 that the full Latin hypercube performs best, while the full I-optimal design performs worst, but there is little difference between the designs across the majority of the design space.

Empirical Root Mean Squared Error

To evaluate the prediction properties of the GASP model and polynomials for the hybrid designs, a hypothetical response variable was created for each of the designs using a test function. The designs were then "analyzed" using both a GASP model and a polynomial. To assess their performance, the resulting models were then used to predict the response values for 10,000 randomly generated uniformly distributed test points, and the residual error calculated as the difference from the values determined by the test function. For each of the test functions used, the function and its source is described, a response surface varying two of the input factors is shown, and results pertaining to root mean squared error (RMSE) for the linear regression models (polynomials) and GASP models is provided. Descriptions of the results are also included.

Test Function 1: The first test function was used in Santner, Williams, and Notz (2003), and first appeared in Brainin (1972). The function is

$$y = \left(x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + 10$$

$$x_1 \in (-5,10), x_2 \in (0,15)$$

The resulting surface (with x_1 and x_2 scaled from -1 to 1) is presented in Figure 6.

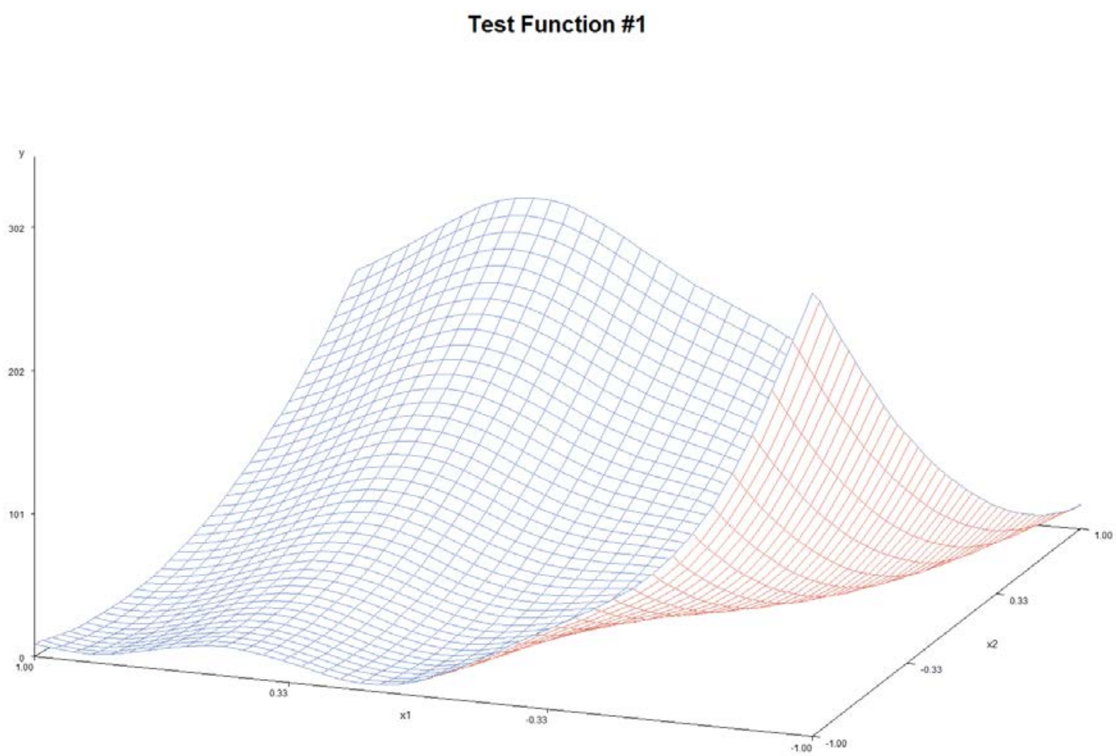

Figure 6. Surface plot of Test Function 1.

As the polynomial order increases, the number of terms in the linear regression model increases (the number of terms in the model is equivalent to the number of design points). The GASP model interpolates the design points, and hence is also dependent on the sample size. In Figure 7, it can be seen that the RMSE for both models is reduced as the number of design points increases.

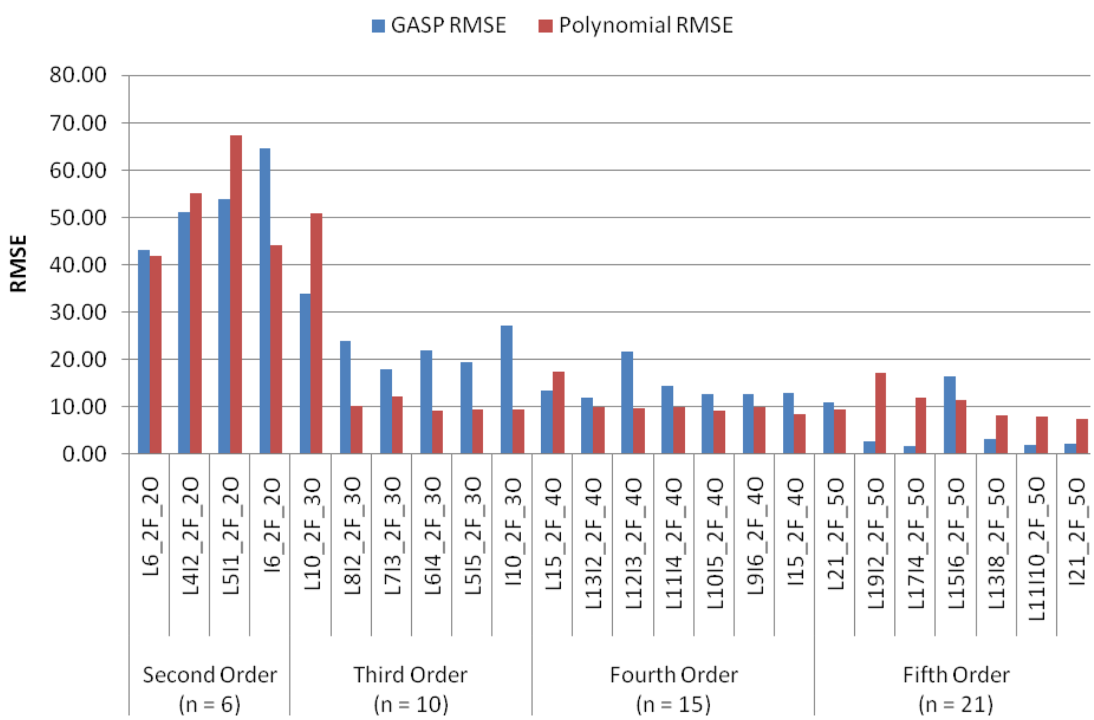

Figure 7. RMSE for two-factor designs.

There does not seem to be a tractable pattern of how the RMSE varies depending on the design composition (ratio of space-filling to I-optimal points). Because the location of the design points is a factor in both models, the lack of a defined pattern may be related to the fact that only one design was generated for each composition.

Test Function 2: The second test function is found in Allen, Bernshteyn, and Kabiri-Bamoradian (2003) and is designed to act as a surrogate model for a plastic seal design. The approximate analytical function is given as

$$y(x_1, x_2, x_3) = (105[0.58(x_2 + x_3 - 0.85) + \ 3.0]^3 x_3) \times \left(\frac{\sin\left[\frac{1.5x_3}{x_1 - 2.0}\right]}{(x_1 - 2.0)^3} \right)$$

where x_1, x_2, and x_3 represent input parameter dimensions on the plastic seal. The bounds for the parameters are (in millimeters): $4 \le x_1 \le 7$, $0.7 \le x_2 \le 1.7$, and $0.055 \le x_3 \le 0.500$. A surface plot of Test Function 2 is shown in Figure 8 for variables x_1 and x_2 at a fixed value of $x_3 = 0.2225$.

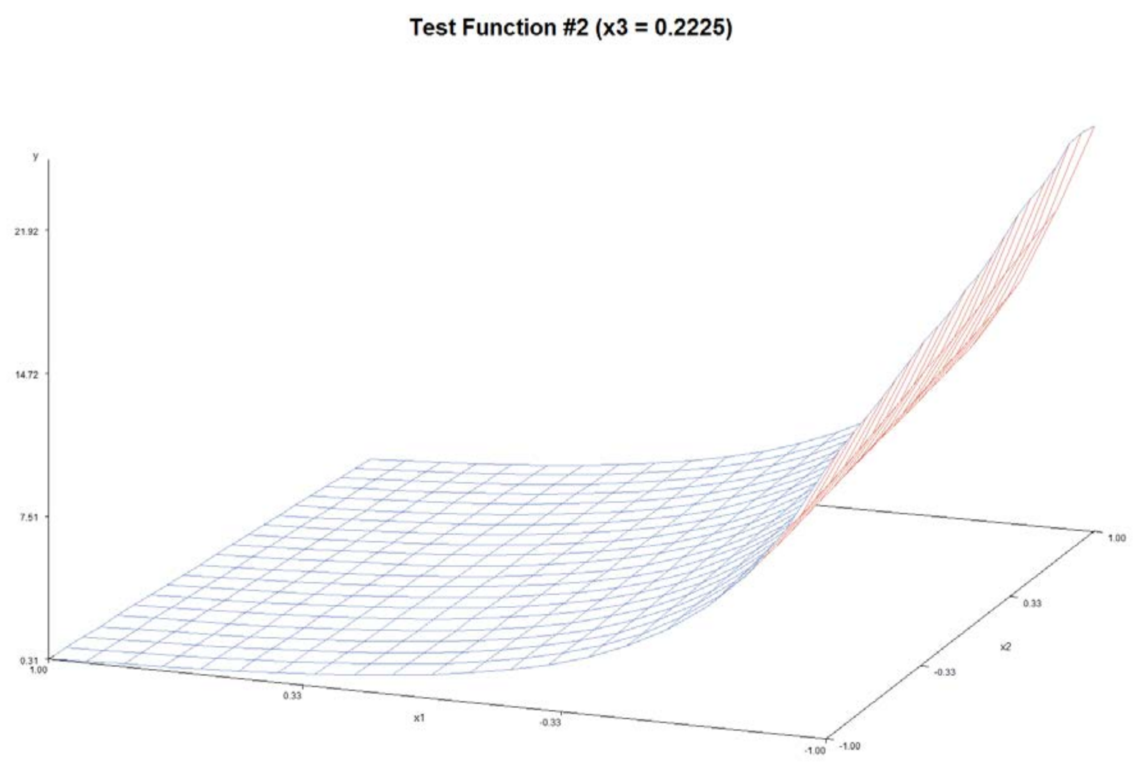

Figure 8. Surface plot of Test Function 2 (x_3 set to 0.2225).

As with the two-factor designs, Figure 9 shows that in the three-factor case both model types' RMSE improves as the sample size increases. The models perform comparably, and there does not seem to be a tractable pattern of how the RMSE varies depending on the design composition (ratio of space-filling to I-optimal points).

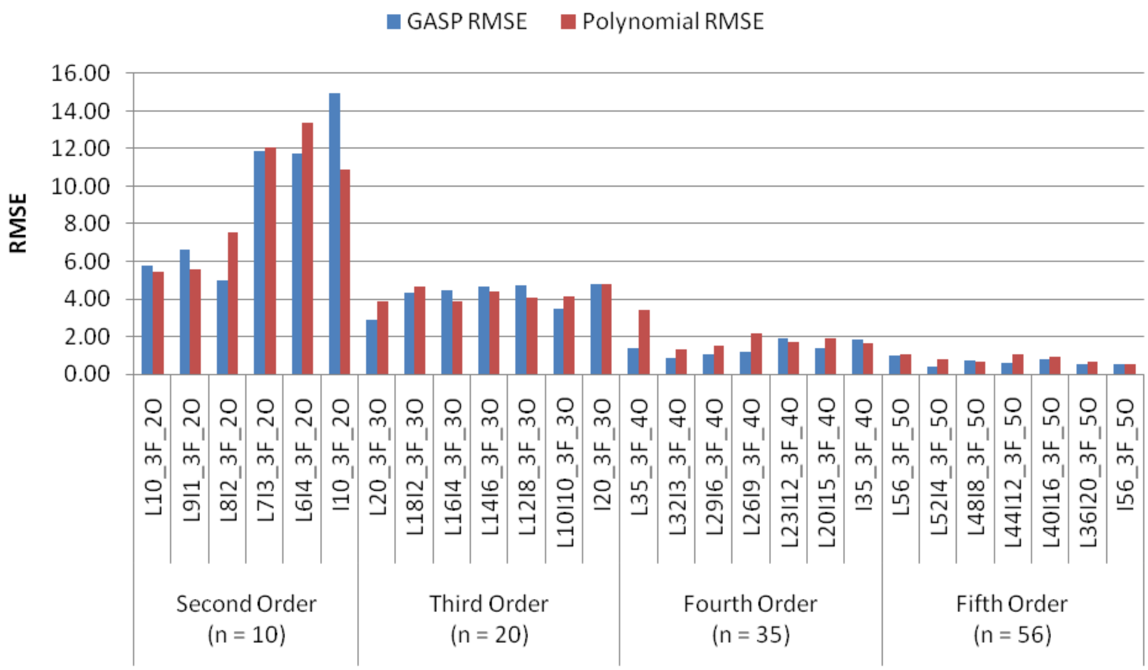

Figure 9. RMSE for three-factor designs.

Test Function 3: Our final test function was first published in Morris, Mitchell, and Ylvisaker (1993) and subsequently used for comparing metamodels in Allen, Bernshteyn, and Kabiri-Bamoradian (2003). The function is

$$y(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) = \frac{2\pi x_3 (x_4 - x_6)}{\ln\left(\frac{x_2}{x_1}\right)\left[1 + \frac{2 x_7 x_3}{\ln\left(\frac{x_2}{x_1}\right) x_1^{2} x_8} + \frac{x_3}{x_5}\right]}$$

where y predicts water flow – in cubic meters per year – as a function of eight design dimensions. As in Allen, Bernshteyn, and Kabiri-Bamoradian (2003), we only vary x_1, x_4, x_6, and x_7 and set the other four variables at their midpoint of the specified ranges from the experiment demonstrated in Morris, Mitchell, and Ylvisaker (1993). The ranges and fixed values for each of the variables are presented in Table 2.

Table 2. Ranges and fixed values for the experimental and fixed variables in Test Function 3.

Experimental Variables		
Variable	Low	High
x_1	0.05	0.15
x_4	990	1110
x_6	700	820
x_7	1120	1680

Fixed Variables	
Variable	Fixed Value
x_2	25050
x_3	89335
x_5	89.6
x_8	9855

For the four-factor designs, the polynomials' performance improves demonstrably as the number of design points and correspondingly the number of terms in the model increase (Figure 10).

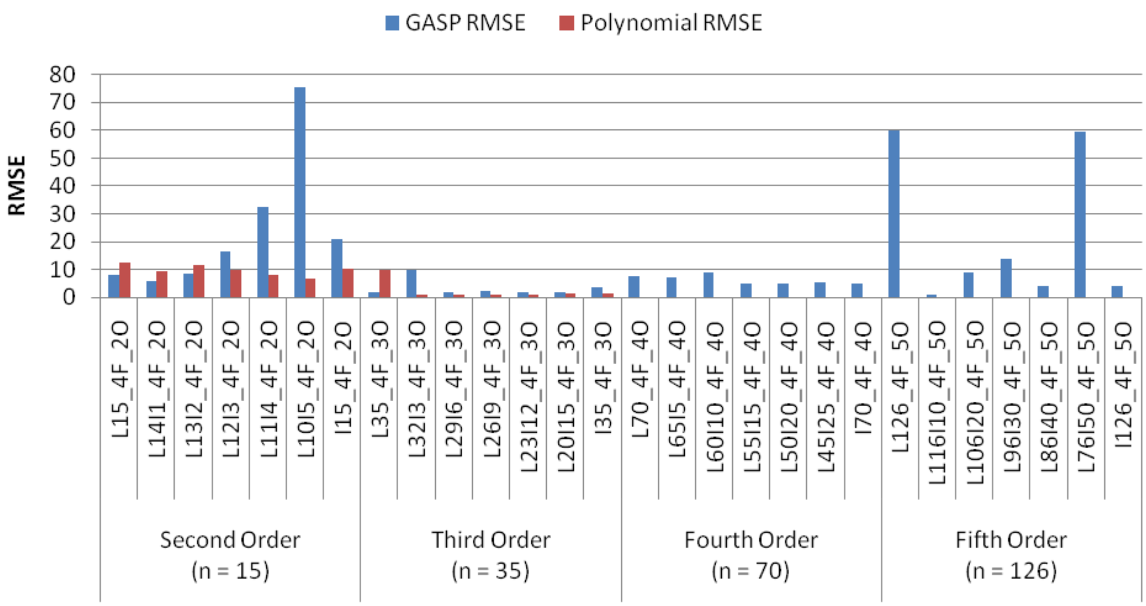

Figure 10. RMSE for four-factor designs.

The GASP models exhibit the lowest RMSE values when n = 35, which corresponds well to work by Loeppky, Sacks, and Welch (2008) that indicates that the GASP model works well given 10 times the number of factors' worth of runs. In the designs with 126 runs, it is likely that near singularity of the correlation matrix contributes to the increased error estimates. In the case of near-singular matrices, the model fitting algorithm within JMP includes the addition of a ridge parameter to the matrix to ensure it is invertible. Some of the designs with the ridge parameter have relatively low error, while others in the same design class have error an order of magnitude higher. One example being the disparity between the RMSE for the four-factor, fifth-order design with 86 Latin hypercube points augmented with 40 I-optimal points and the design with 76 Latin hypercube points and 50 I-optimal points, with RMSE for the GASP model of 3.78 and 59.64, respectively.

Test Function 3 was also used to evaluate the five-factor designs. Factor x_2, previously held fixed at 25,050, was added to the factors that were varied, ranging from 24,950 to 25,150. Factors x_3, x_5, and x_8 were all held constant at the same levels. Similar results were seen in the five-factor designs and models as were evidenced in four factors (Figure 11).

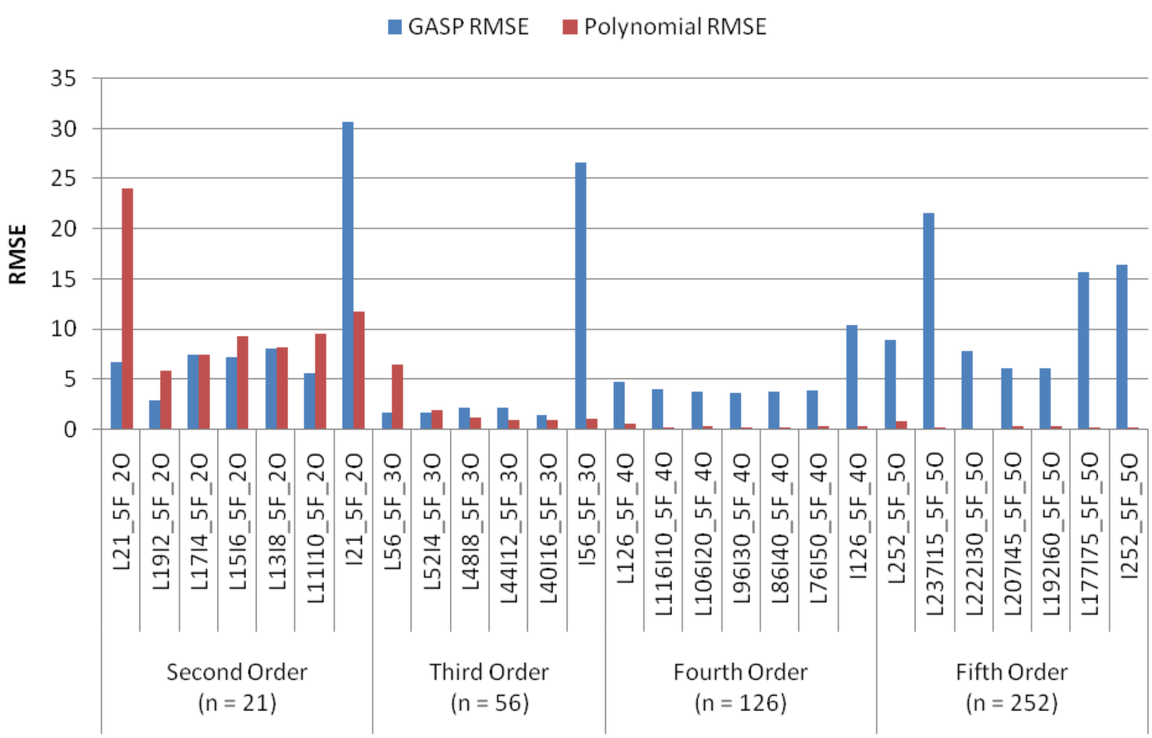

Figure 11. RMSE for five-factor designs.

Polynomial model performance improves with the addition of more design points and model terms, while the GASP models exhibit the lowest RMSE values when n is approximately 10 times the number of factors (n = 56). As in the larger designs with four factors, as the number of runs increases the correlation matrices edge closer to singularity and additional complexity is added to the model estimation.

The Predicted Residual Sum of Squares (PRESS) statistic may also be a useful statistic to aid in comparing designs in future research.

Design Variability

As noted earlier, there is also variability imparted on these summary statistics based on the exact design employed. In order to begin to evaluate the effect of the design itself, each of the two-factor design combinations was replicated such that there are five designs of each type. The theoretical prediction variance was evaluated for each design and modeling type, as well as the predictive capability of each as tested by Test Function 1.

To illustrate the effect of design variability, two of the five different 21-run Latin hypercube designs place points as follows in Figure 12.

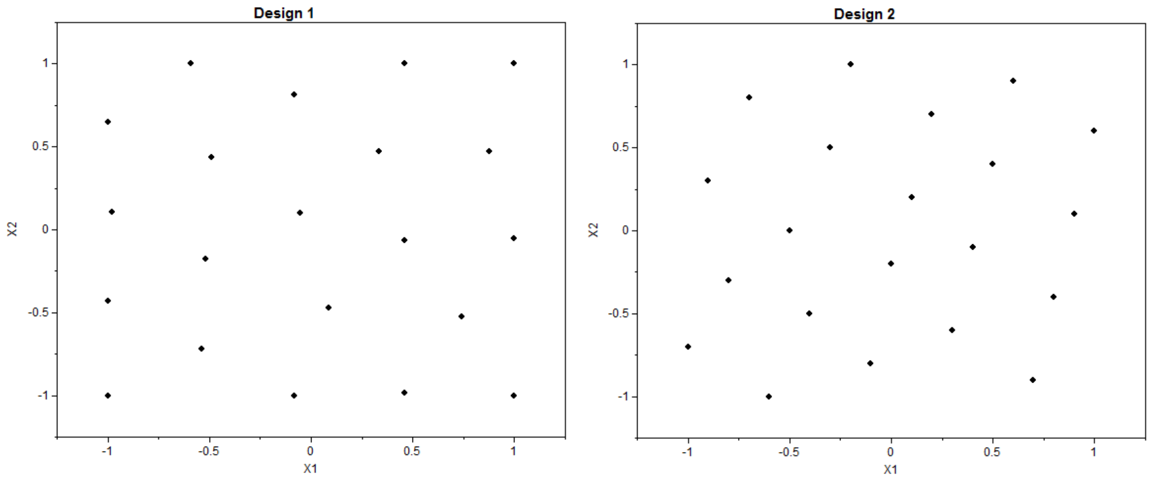

Figure 12. Point placement for two two-factor, 21-run Latin hypercube designs.

Theoretical Prediction Variance

The prediction variance for the five two-factor, 21-run Latin hypercube designs was calculated in terms of a fifth-order polynomial, and plotted in an FDS plot shown in Figure 13. The prediction variance for Design 2 visibly separates from the other designs around the 90th percentile. Looking at the summary statistics, for Design 1 and Design 2 side by side as presented in Table 3, it can be seen that the separation occurs even earlier (by about the median), with sharp increases by the 75th percentile and beyond. This results in a maximum prediction variance increase of almost 2000-fold from the maximum prediction variance of Design 1, as detailed in Table 3.

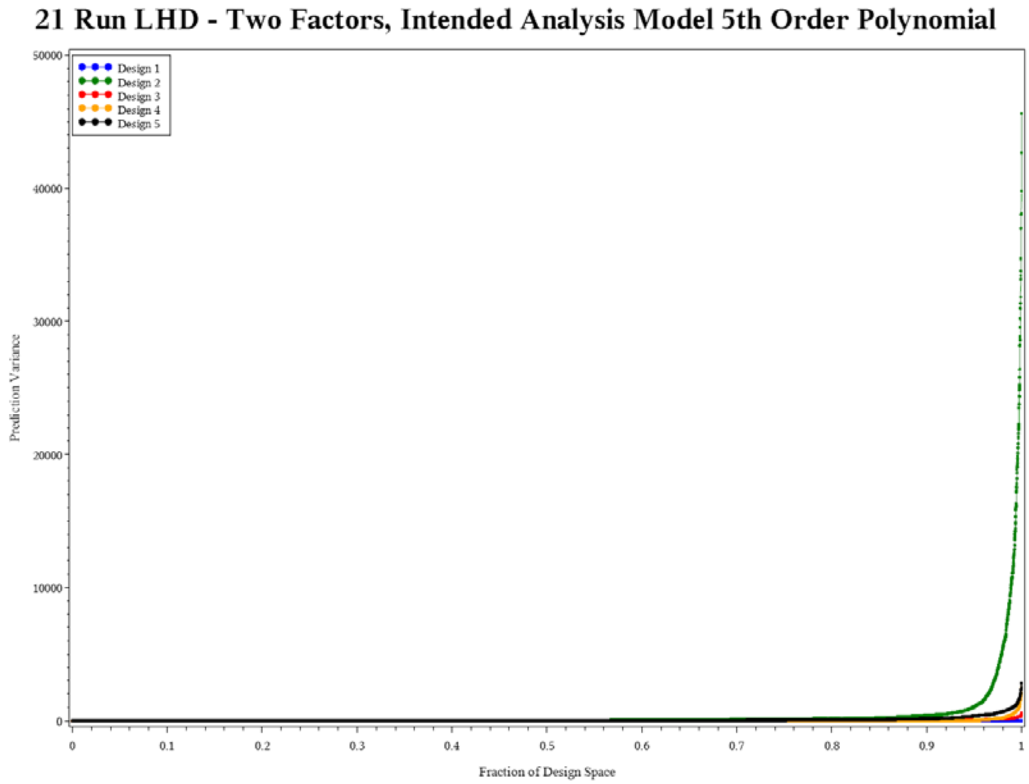

Figure 13. FDS plot for replicated two-factor, 21-run Latin hypercube designs, in terms of a fifth-order polynomial model.

Table 3. Summary statistics for two two-factor, 21-run Latin hypercube designs, in terms of a fifth-order polynomial model.

	Design 1	Design 2
Mean	1.59	427.83
Minimum	0.51	0.57
5th Percentile	0.69	1.31
10th Percentile	0.77	2.10
1st Quartile	0.93	6.34
Median	1.29	23.22
3rd Quartile	1.97	112.14
90th Percentile	2.71	390.01
95th Percentile	3.35	1039.71
Maximum	23.58	45631.39

The prediction variance over the design space is plotted in Figure 14, showing that the largest prediction variance is found in the (-1, 1) corner for Design 1. This happens to be the only corner in the design without an observation.

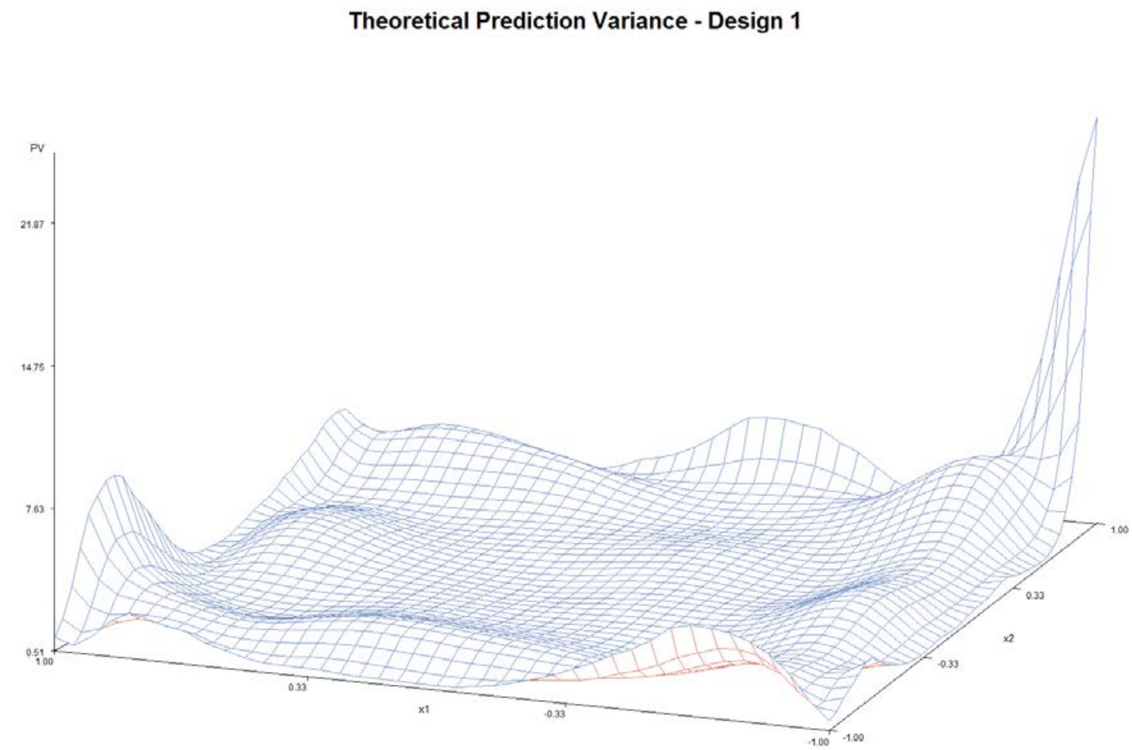

Figure 14. Theoretical prediction variance for Design 1, in terms of a fifth-order polynomial.

Given the scales on the Z-axis, it can be seen that Design 2 has very large prediction variance in the (-1, 1) and (1, -1) corners of the design space (Figure 15), corresponding to the largest gaps seen in the coverage. The variance in each corner is so high that the ambient variance of the rest of the space is muted. Design 1 has its largest prediction variance in the (-1, 1) corner of the design space as well, but it is more on the scale of the prediction variance seen elsewhere in the design space.

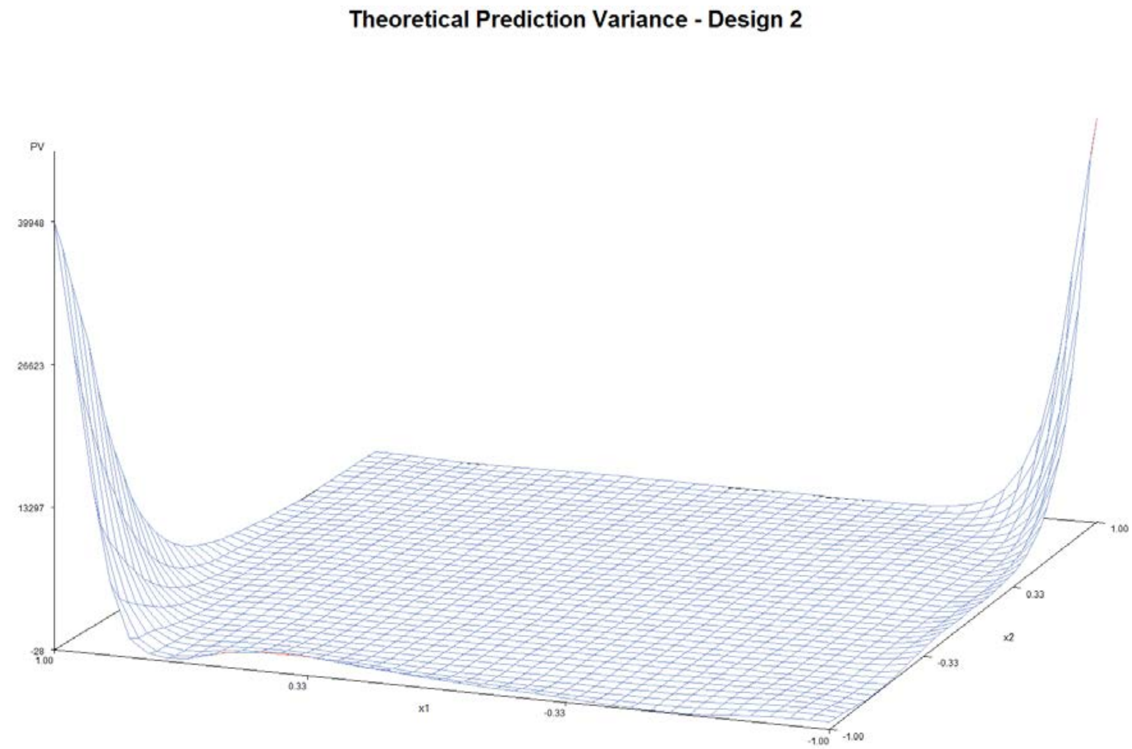

Figure 15. Theoretical prediction variance for Design 2, in terms of a fifth-order polynomial.

In general, the impact of the design itself on the prediction variance is much reduced as the I-optimal points are added. This is an intuitive result, as the intent of the I-optimality criteria is to minimize the average scaled prediction variance over the design space. As an example, Table 4 summarizes the range of the mean and maximum prediction variances for designs with two factors and a fifth-order polynomial as the intended analysis model. It can be seen that the variability reduces dramatically as I-optimal points are added to the Latin hypercube designs.

Table 4. Maximum prediction variance values observed across five two-factor, 21-run Latin hypercube designs, in terms of a fifth-order polynomial model.

	L21	L19I2	L17I4	L15I6	L13I8	L11I10	I21
Maximum PV Range	45608	432	71	107	24	4.8	1.8
Mean PV Range	426	1.7	0.5	0.7	0.3	0.1	0.01

The evaluation of the prediction variance of the same two designs with respect to the GASP model is shown in Table 5. The summary statistics show that while Design 2 still has a higher prediction variance than Design 1, the variance between designs is on a much smaller scale.

Table 5. Summary statistics for two 21-run Latin hypercube designs, in terms of a GASP model.

	Design 1	Design 2
Mean	0.0526	0.0402
Minimum	0.0000	0.0000
5th Percentile	0.0080	0.0042
10th Percentile	0.0152	0.0078
1st Quartile	0.0337	0.0156
Median	0.0522	0.0283
3rd Quartile	0.0681	0.0435
90th Percentile	0.0890	0.0804
95th Percentile	0.1014	0.1283
Maximum	0.3046	0.4359

The prediction variance for each design is plotted in Figures 16 (Design 1) and 17 (Design 2). The relative prediction patterns are similar to those seen for the polynomial models, with the maximum variance seen in the (-1, 1) corner for Design 1, and larger variance in the the (-1, 1) and (1, -1) corners of Design 2.

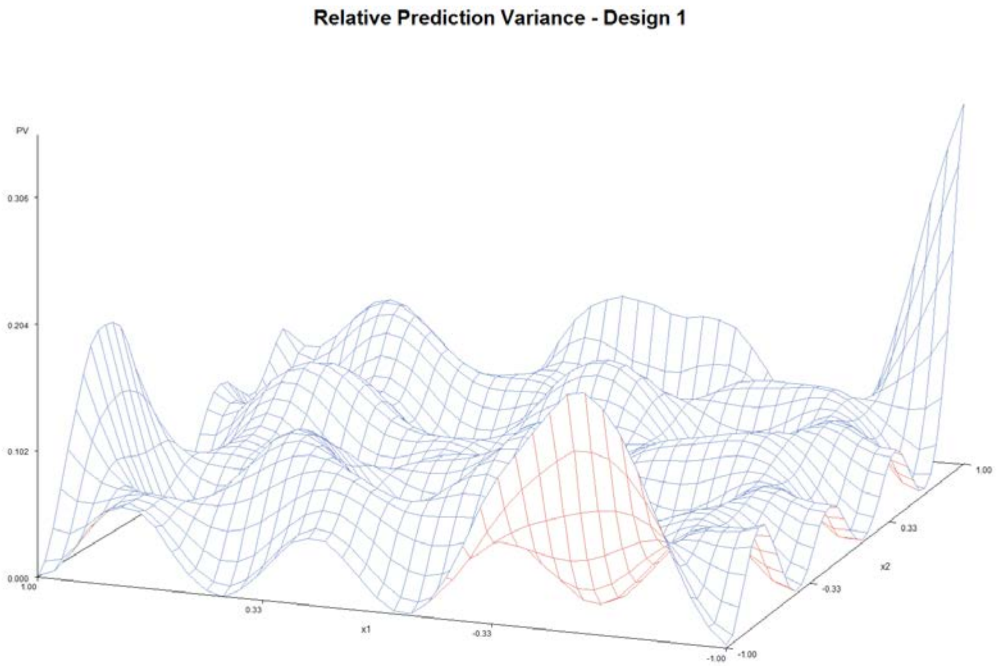

Figure 16. Theoretical prediction variance for Design 1, in terms of a GASP model (both $\theta_i = 3$).

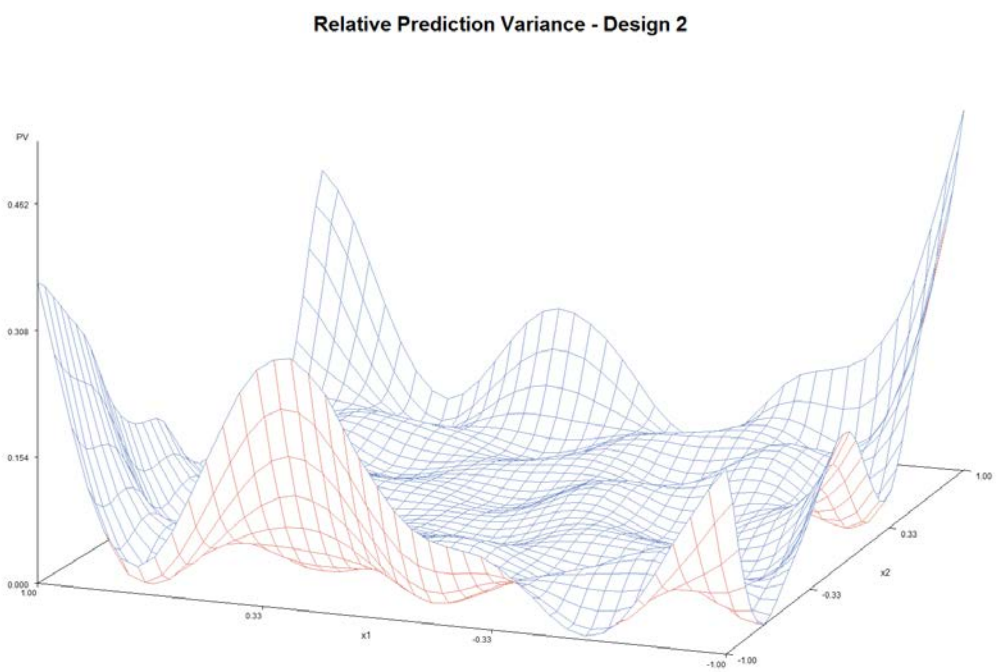

Figure 17. Theoretical prediction variance for Design 2, in terms of a GASP model (both $\theta_i = 3$).

Prediction Performance

Using the same 21-run Latin hypercube designs as previously evaluated for prediction variance, it can be seen in Figure 18 that the form of the predicted surface is affected by the variance properties of the design. Logically following from the theoretical prediction properties of the designs, it can be seen that the prediction capability in the (-1, 1) and (1, -1) corners is reduced for Design 2, although the departure is markedly smaller in the GASP model than the polynomial.

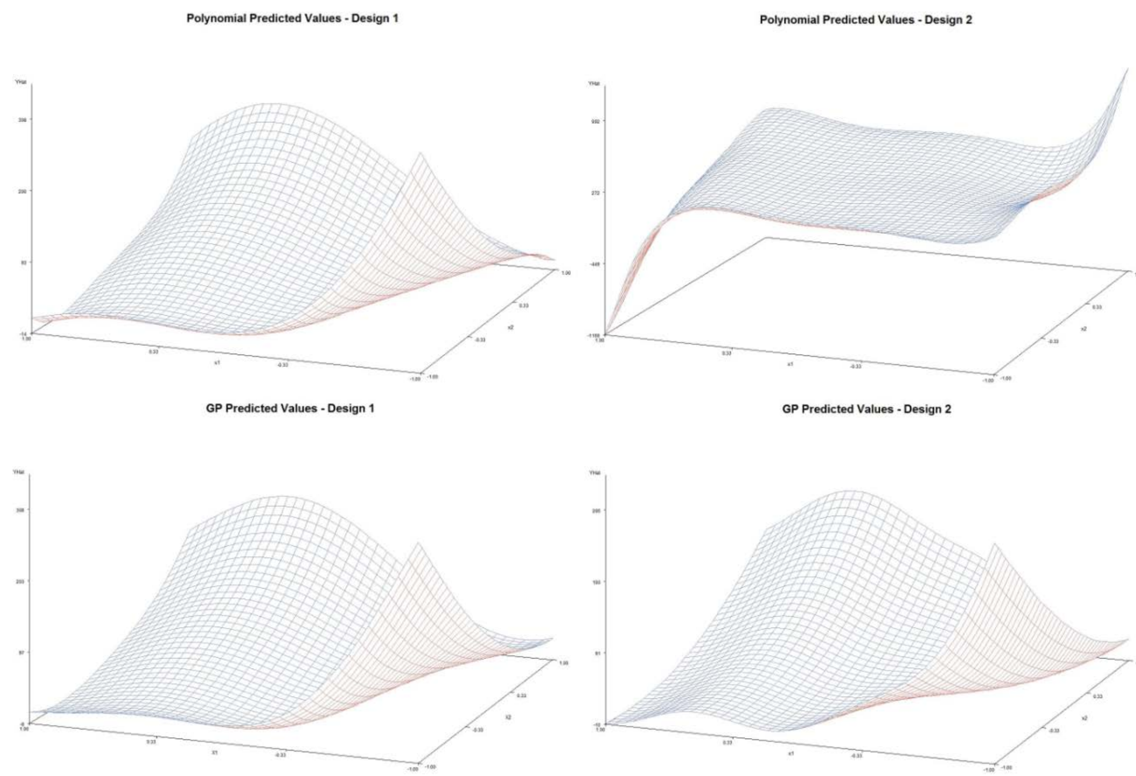

Figure 18. Predicted values for Test Function 1 for Designs 1 and 2, analyzed using fifth-order polynomials or GASP models.

Empirical Root Mean Squared Error

Finally, all of the repeated designs were analyzed under two scenarios – one in which Test Function 1 was used in a deterministic fashion, and another in which normally distributed random error was added to simulate a stochastic process. In both cases, deterministic and stochastic, the designs perform as expected for the polynomials. The full I-optimal designs consistently have the lowest RMSE, while the full Latin hypercube designs consistently have the highest RMSE. As the number of I-optimal points in the design increases, the RMSE decreases. There was no apparent relationship between the mixture of design points and the RMSE for the GASP model, differences seemed to be solely related to sample size (as sample size increased, RMSE decreased) and whether the response was deterministic or stochastic (higher RMSE were evidenced in the stochastic case).

In general, the fitted error for the GASP models was higher than that of the polynomials for small designs (n = 6 and 10). As the number of design points increases, the GASP models begin to perform comparably to the polynomials in terms of fitted error, which corresponds to work by Loeppky, Sacks, and Welch (2008) that indicates that the GASP model works well given 10 times the number of factors' worth of runs. The GASP models also begin to perform comparably or better than the polynomials as error is introduced into the system.

As an example, the results for the two-factor, fourth-order designs are presented in the form of box plots. Figure 19 displays the RMSE values for each of the augmented space-filling designs fit to the responses with no random error, while Figure 20 includes random error in the test function. Results from the GASP models and polynomials are presented side by side for comparison.

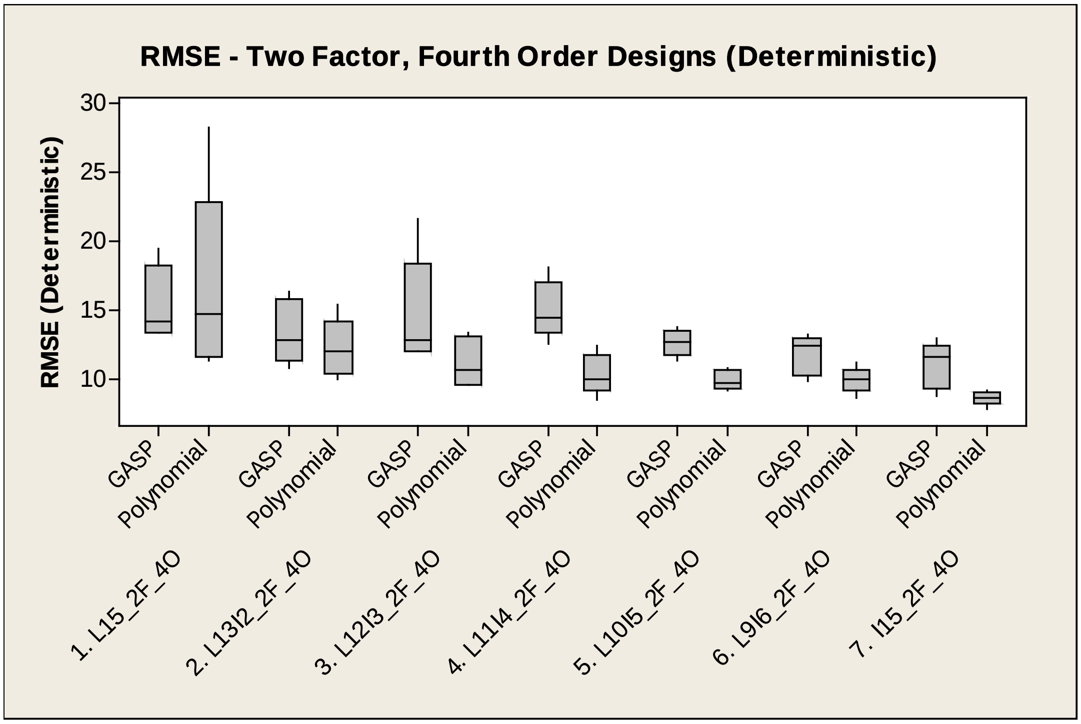

Figure 19. Deterministic RMSE for two-factor, fourth-order designs.

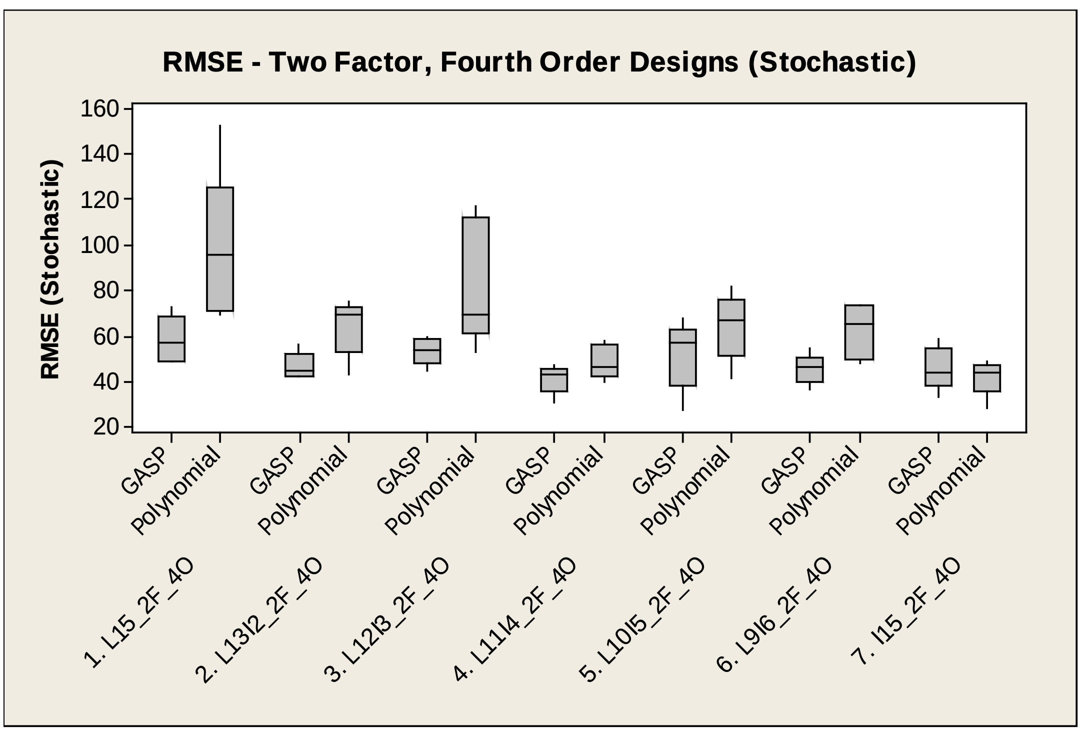

Figure 20. Stochastic RMSE for two-factor, fourth-order designs.

As can be seen by comparing Figure 19 and Figure 20, although the error in the stochastic case is higher for both GASP models and the polynomials, the GASP models are performing as well or better than the polynomials in the stochastic case. In the deterministic case, the RMSE for the polynomial models was much smaller than that of the GASP models.

Conclusions

The results presented give insight into how hybrid space-filling designs perform with respect to prediction variance properties for the linear regression model and the GASP model. The designs are compared to both solely space-filling and solely optimal designs.

One of the benefits of a computer simulation models is the ability to build up a design sequentially, without concern for blocking or randomization. Note that in deterministic models replication and randomization are not needed and in stochastic models randomization can be controlled through the random number generator. Either way, in computer simulation experiments the space-filling-hybrid design is an excellent choice. Due to the potentially large impact of the design itself, the theoretical prediction capabilities should be evaluated prior to running the experiment. Either type of model can be credibly fit after running the hybrid design, and after the experiments are completed the experimenter has a better idea of what modeling strategy to use. At this point the design can be augmented with a criterion that is optimal for that strategy, be it a polynomial model or a GASP model.

While some might question the use of the space-filling design for polynomials at all, it is important to remember that in advance of any experimentation it is impossible to know whether a polynomial model of any order will prove to be adequate. Using a space-filling design for initial exploration makes considerable practical sense.

CHAPTER 4 – BRIDGE DESIGN PROPERTIES

Bridge designs were introduced by Jones, Johnson, Montgomery, and Steinberg (2012) as a compromise between Latin hypercube designs and D-optimal designs. They are intended for use when a polynomial is judged to be a promising candidate for modeling the response. The algorithm for generating a bridge design ensures that the resulting design will be D-optimal for a specified polynomial, subject to the constraint that any one-dimensional projection will maintain a minimum distance between points. This results in a design that takes advantage of the efficiency of D-optimal designs for fitting a polynomial model to the response, while avoiding the potential replication inherent in traditional optimal designs.

This chapter seeks to evaluate the performance of bridge designs in comparison with its parent components, as well as the commonly used I-optimal design. This is done to better understand the prediction properties of the bridge design, and to better understand the situations in which it may best be applied.

Methodology

The JMP script that was used in the original work was extended to allow D-optimal creation given full third through fifth-order models. Similar to Chapter 3, the designs chosen for comparison were a maximin Latin hypercube design, a D-optimal and an I-optimal design. All comparators were chosen to as frequently used, easily generated with commonly available software, and ultimately flexible in terms of sample size.

As in Chapter 3, the designs were to be evaluated using similar methods as Johnson, Montgomery, Jones, and Parker (2010). The prediction capabilities of the designs are assessed theoretically using the theoretical prediction variance for a polynomial model, based on a random sample of 10,000 points in the design space. The designs are also compared using the design efficiencies. JMP evaluates several design diagnostics, assessing the efficiency of the design as assessed by several alphabetic optimality criteria. D-optimality maximizes the determinant of the information matrix. G-optimality minimizes the maximum scaled prediction variance across the design space. A-optimality minimizes the average variance of the coefficient estimates. The average prediction variance is evaluated as an analog for I-optimality, which minimizes the average prediction variance across the design space. Finally, test functions are set to act as response variables to compare the empirical results of the fitted models for each design.

Designs with two to five factors were generated, with underlying models specified as second to fifth-order, for a total of 16 different factor-order combinations. Four sample sizes per factor-order combination were evaluated, starting with the minimum number of design points necessary to fit the intended model. The minimum was then increased by two, and four, and doubled, with sample sizes for all generated models presented in Table 6.

Table 6. Number of runs necessary for each design combination (number of factors, underlying model order, and number of runs).

	Second Order				Third Order				Fourth Order				Fifth Order			
Factors	p	p + 2	p + 4	2p	p	p + 2	p + 4	2p	p	p + 2	p + 4	2p	p	p + 2	p + 4	2p
2	6	8	10	12	10	12	14	20	15	17	19	30	21	23	25	42
3	10	12	14	20	20	22	24	40	35	37	39	70	56	58	60	112
4	15	17	19	30	35	37	39	70	70	72	74	140	126	128	130	252
5	21	23	25	42	56	58	60	112	126	128	130	252	252	254	256	504

As noted in Chapter 2, in addition to the number of factors, the number of runs, and the intended regression model, a minimum spacing distance between points in any one-dimensional projection must be set. The minimum distance set for the bridge design points was 0.04, unless the minimum recommended distance ($\delta \leq 1 / (n - 1)$) for the maximum number of runs in each factor-order combination was less than 0.04. In those cases, the minimum distance was set to meet the minimum for all designs within that combination.

Results

The theoretical comparison of the design types will be presented by the intended model order of the response variable, second through fifth-order, followed by empirical results using the Gaussian process model to fit test function responses with two and three factors.

Second-Order Designs

Figure 21 shows examples of designs generated assuming an underlying second-order model, with two factors. The circled points in the optimal designs show the locations of replicates. The D-optimal design places replicates at (1, 1), (1, -1), and (-1, 1), while the I-optimal design places four runs at the center point. Looking at the designs, it can be seen that the bridge design places points in similar locations to both of the optimal designs. Although there are a few points that are placed close together, the bridge design is free of replicated points.

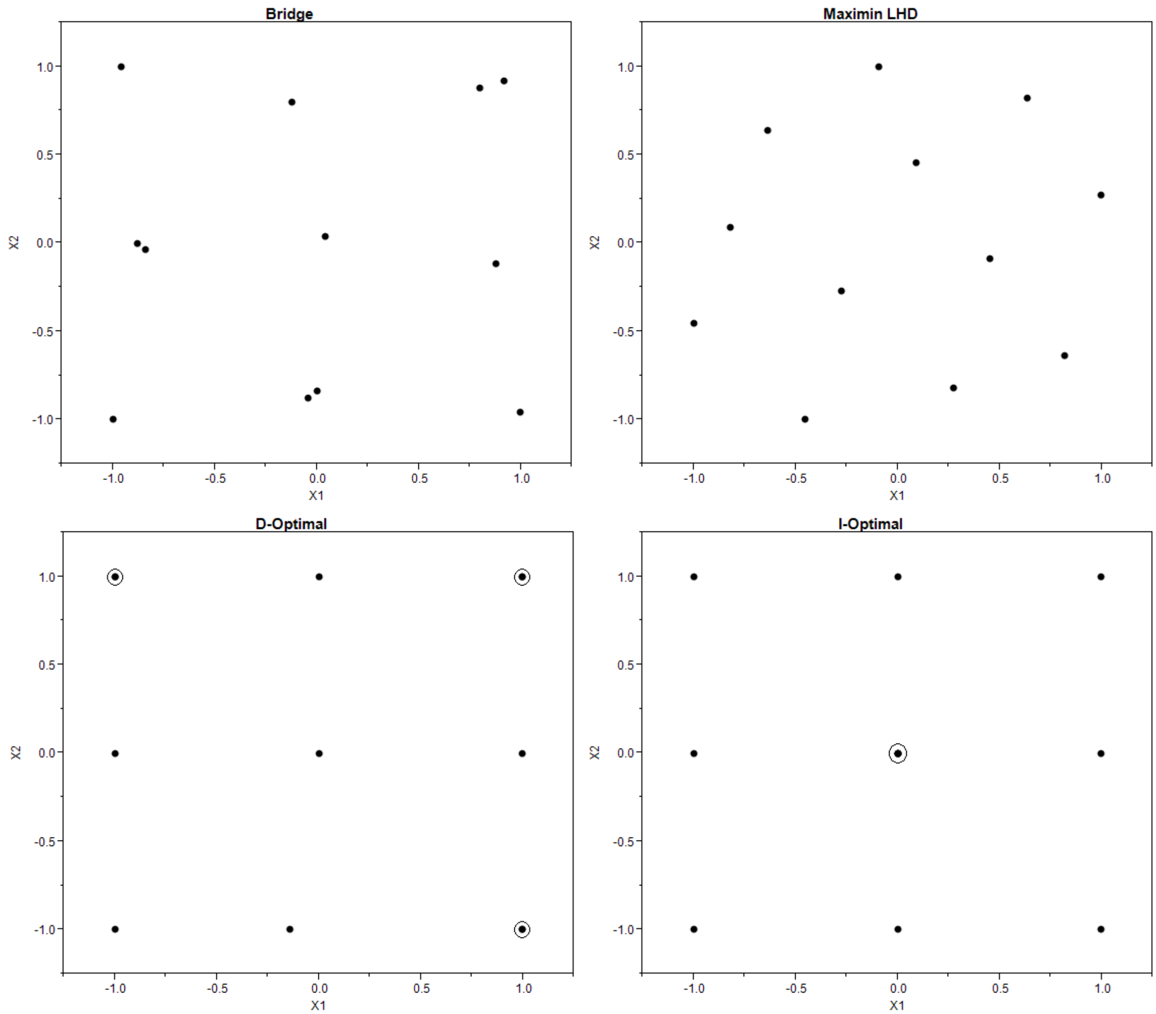

Figure 21. Two-factor, 12-run designs generated assuming a full quadratic model.

The prediction variance results for second-order designs with two to five factors are shown in Table 7. The bridge designs consistently display smaller prediction variance than the Latin hypercube designs, particularly in terms of maximum variance (in fact there is only one case in which the maximum prediction for the bridge design exceeds that of the Latin hypercube design, at five factors and 42 runs). In general, the bridge designs perform comparably to the optimal designs. The majority of the bridge designs had median prediction variance within 10% of the comparable D-optimal design, and within 36% of the comparable I-optimal design. The maximum prediction variance was more likely to be much larger than the optimal designs, but was no more than twice as large for any design of two or three factors.

Table 7. Prediction variance estimates for designs generated assuming a full quadratic model.

	n = p = 6 runs				n = p + 2 = 8 runs				n = p + 4 = 10 runs				n = 2p = 12 runs			
2 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.46	0.48	0.43	0.45	0.42	0.37	0.36	0.30	0.26	0.34	0.27	0.29	0.23	0.27	0.26	0.20
5%	0.63	0.57	0.54	0.50	0.45	0.39	0.39	0.33	0.32	0.35	0.33	0.29	0.26	0.29	0.29	0.20
25%	0.91	0.70	0.74	0.69	0.53	0.47	0.45	0.38	0.37	0.38	0.39	0.31	0.29	0.32	0.35	0.21
50%	1.15	0.86	0.93	0.84	0.61	0.59	0.52	0.45	0.41	0.45	0.43	0.34	0.32	0.38	0.40	0.26
75%	1.46	1.07	1.11	1.00	0.70	0.98	0.62	0.60	0.47	0.60	0.48	0.40	0.39	0.51	0.45	0.37
90%	1.65	1.57	1.51	1.20	0.79	1.61	0.73	0.82	0.57	0.98	0.52	0.48	0.48	0.80	0.49	0.46
95%	1.75	2.05	1.75	1.41	0.84	2.14	0.85	0.95	0.63	1.37	0.54	0.51	0.54	1.11	0.52	0.50
max	1.85	6.94	2.61	2.15	1.09	5.07	1.21	1.32	0.93	3.95	0.78	0.78	0.84	3.05	0.74	0.77

	n = p = 10 runs				n = p + 2 = 12 runs				n = p + 4 = 14 runs				n = 2p = 20 runs			
3 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.44	0.53	0.42	0.40	0.37	0.41	0.37	0.31	0.28	0.34	0.31	0.31	0.18	0.24	0.26	0.19
5%	0.55	0.67	0.53	0.50	0.45	0.48	0.47	0.36	0.32	0.40	0.33	0.33	0.22	0.26	0.31	0.20
25%	0.64	0.85	0.63	0.60	0.53	0.57	0.58	0.40	0.38	0.48	0.36	0.36	0.26	0.31	0.35	0.21
50%	0.74	1.04	0.73	0.68	0.61	0.72	0.67	0.47	0.49	0.61	0.39	0.39	0.35	0.38	0.40	0.25
75%	0.87	1.46	0.86	0.77	0.71	1.03	0.78	0.59	0.66	0.93	0.47	0.47	0.49	0.59	0.46	0.31
90%	1.05	2.28	1.08	1.00	0.84	1.67	0.90	0.79	0.81	1.45	0.52	0.52	0.59	0.91	0.51	0.37
95%	1.17	3.15	1.26	1.20	0.93	2.25	1.00	0.99	0.89	1.93	0.55	0.55	0.65	1.18	0.53	0.42
max	2.29	16.32	2.60	2.76	1.59	8.13	1.41	2.62	1.34	5.92	0.77	0.77	1.20	4.50	0.60	0.72

	n = p = 15 runs				n = p + 2 = 17 runs				n = p + 4 = 19 runs				n = 2p = 30 runs			
4 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.44	0.49	0.40	0.38	0.31	0.39	0.37	0.26	0.29	0.46	0.39	0.25	0.11	0.23	0.24	0.15
5%	0.57	0.64	0.71	0.54	0.42	0.49	0.51	0.32	0.37	0.62	0.47	0.29	0.16	0.30	0.33	0.16
25%	0.71	0.86	0.94	0.67	0.51	0.64	0.63	0.38	0.48	0.78	0.55	0.33	0.31	0.35	0.38	0.18
50%	0.84	1.16	1.10	0.74	0.67	0.85	0.74	0.49	0.66	1.07	0.62	0.41	0.52	0.45	0.42	0.23
75%	1.01	1.91	1.27	0.84	0.88	1.31	0.86	0.63	0.88	1.71	0.69	0.51	0.74	0.69	0.46	0.29
90%	1.22	3.18	1.41	0.98	1.11	1.96	0.96	0.77	1.10	2.81	0.76	0.62	0.94	1.09	0.51	0.35
95%	1.38	4.47	1.52	1.11	1.28	2.43	1.02	0.88	1.25	4.29	0.80	0.70	1.08	1.41	0.53	0.40
max	2.96	23.66	2.08	2.97	2.59	6.68	1.36	1.64	2.48	27.57	1.19	1.62	2.03	4.49	0.63	0.69

	n = p = 21 runs				n = p + 2 = 23 runs				n = p + 4 = 25 runs				n = 2p = 42 runs			
5 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.35	0.53	0.43	0.35	0.32	0.56	0.30	0.25	0.24	0.48	0.29	0.22	0.09	0.22	0.22	0.11
5%	0.54	0.91	0.63	0.47	0.52	0.83	0.41	0.33	0.35	0.66	0.38	0.27	0.22	0.28	0.28	0.13
25%	0.74	1.48	0.79	0.56	0.69	1.18	0.48	0.41	0.64	0.88	0.44	0.35	0.50	0.34	0.31	0.17
50%	1.02	2.32	0.91	0.65	0.92	1.66	0.57	0.50	0.98	1.12	0.50	0.44	0.80	0.47	0.34	0.22
75%	1.39	3.86	1.05	0.81	1.25	2.51	0.67	0.62	1.39	1.63	0.59	0.54	1.25	0.75	0.36	0.27
90%	1.79	6.57	1.20	1.03	1.61	3.83	0.77	0.77	1.87	2.67	0.67	0.64	1.79	1.15	0.39	0.33
95%	2.08	9.42	1.28	1.21	1.88	5.33	0.82	0.87	2.20	3.82	0.72	0.71	2.20	1.45	0.40	0.36
max	4.14	56.17	1.77	3.38	4.62	19.64	1.14	1.89	6.64	16.97	1.04	1.54	6.14	4.48	0.51	0.63

The minimum, maximum, and median prediction variance were plotted in Figure 22 to visually compare the prediction variance across the designs. The median was chosen for presentation rather than the mean, since in some cases the maximum prediction variance was large enough to skew the mean. The minimum and maximum are represented by the bottom and top of the vertical lines, respectively, and the median by the arrows. The Latin hypercube was omitted, since the prediction variance results were so much larger than the other design types that its inclusion increased the y-axis scale to a point in which it was difficult to distinguish differences between the bridge and optimal designs. There are few designs with two and three factors in which the maximum prediction variance of the bridge design is smaller than that of the optimal designs. As the number of factors increases however, the maximum variance of the bridge design begins to greatly exceed that of the optimal designs, in some cases as much as 11 times larger. This is likely due to the fact that in smaller design spaces, the optimal designs tend to place replicates, which does not occur in the larger design spaces.

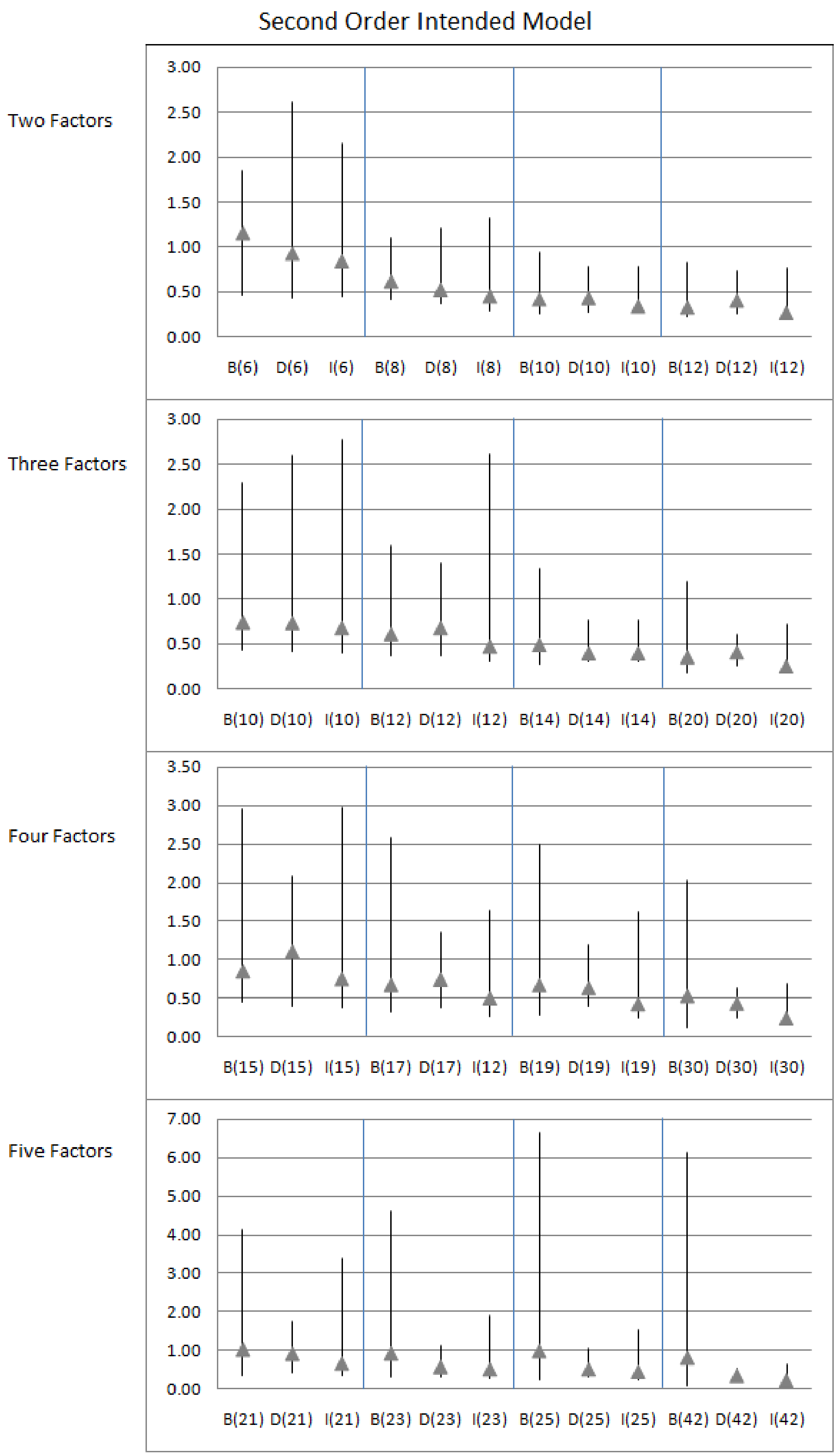

Figure 22. Prediction Variance (minimum, median, and maximum) for designs generated intending to be fit with a second-order polynomial model.

Table 8 presents the design efficiencies for the designs generated assuming an underlying second-order model.

Table 8. Design efficiencies for designs generated intended to be fit with a second-order polynomial model.

	n = p = 6 runs				n = p + 2 = 8 runs				n = p + 4 = 10 runs				n = 2p = 12 runs			
2 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	63.61	51.55	66.67	65.57	64.64	47.58	72.25	68.36	62.49	49.35	72.87	71.09	59.24	47.02	73.93	64.81
G-efficiency	61.79	50.08	59.18	59.90	71.38	48.46	72.87	67.41	73.23	55.23	80.07	86.11	76.95	54.49	78.76	79.47
A-efficiency	53.02	40.20	51.69	53.68	60.19	34.13	65.30	59.70	58.93	42.11	66.86	66.32	55.68	40.92	66.80	58.54
Avg. Pred Var	0.88	0.95	0.85	0.78	0.53	0.83	0.50	0.51	0.42	0.56	0.39	0.37	0.36	0.49	0.34	0.34

	n = p = 10 runs				n = p + 2 = 12 runs				n = p + 4 = 14 runs				n = 2p = 20 runs			
3 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	53.40	38.68	62.21	61.56	52.56	40.69	68.17	61.87	49.65	40.05	70.18	70.18	43.20	38.31	70.33	63.89
G-efficiency	66.70	44.73	62.48	68.31	68.85	48.84	73.32	62.94	69.15	50.64	93.21	92.55	69.84	54.37	75.30	81.65
A-efficiency	47.30	27.96	49.59	49.18	47.90	32.28	56.99	50.32	44.54	32.26	66.06	66.06	39.40	33.27	62.24	59.26
Avg. Pred Var	0.79	1.33	0.80	0.78	0.65	0.96	0.62	0.63	0.59	0.82	0.43	0.43	0.46	0.57	0.35	0.32

	n = p = 15 runs				n = p + 2 = 17 runs				n = p + 4 = 19 runs				n = 2p = 30 runs			
4 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	45.17	29.43	61.52	57.49	40.81	33.75	64.32	56.58	37.45	28.09	66.22	58.79	28.77	30.45	69.66	58.73
G-efficiency	57.65	35.24	61.34	61.78	62.16	45.78	72.80	72.16	58.94	23.52	75.41	73.41	47.74	46.23	81.53	78.60
A-efficiency	38.34	18.34	44.40	44.82	34.18	26.61	53.03	46.52	30.87	14.57	55.39	50.75	22.97	24.30	59.89	53.94
Avg. Pred Var	0.94	1.94	0.96	0.84	0.91	1.21	0.69	0.68	0.89	2.19	0.59	0.57	0.75	0.77	0.36	0.33

	n = p = 21 runs				n = p + 2 = 23 runs				n = p + 4 = 25 runs				n = 2p = 42 runs			
5 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	31.93	22.19	64.01	53.20	31.32	25.23	66.16	57.63	27.14	25.77	66.69	62.09	17.80	27.27	69.02	59.43
G-efficiency	47.20	17.08	69.35	57.19	51.26	23.52	85.64	72.03	41.73	31.40	85.56	75.09	31.46	45.11	89.78	76.03
A-efficiency	23.55	7.52	46.49	38.41	23.48	12.09	55.60	46.06	18.59	15.63	57.67	53.50	10.59	21.75	60.90	55.63
Avg. Pred Var	1.46	4.76	0.87	0.94	1.33	2.95	0.62	0.71	1.54	2.05	0.56	0.57	1.60	0.83	0.33	0.32

The D-efficiency results are plotted in Figure 23, and the G and A-efficiencies follow similar patterns. The bridge designs perform very comparably to the optimal designs for cases with two and three factors. While their relative performance does decline a bit as the number of factors and runs increases, their performance is still superior to the Latin hypercube designs until the sample size is increased to 2*p* in the four and five-factor designs.

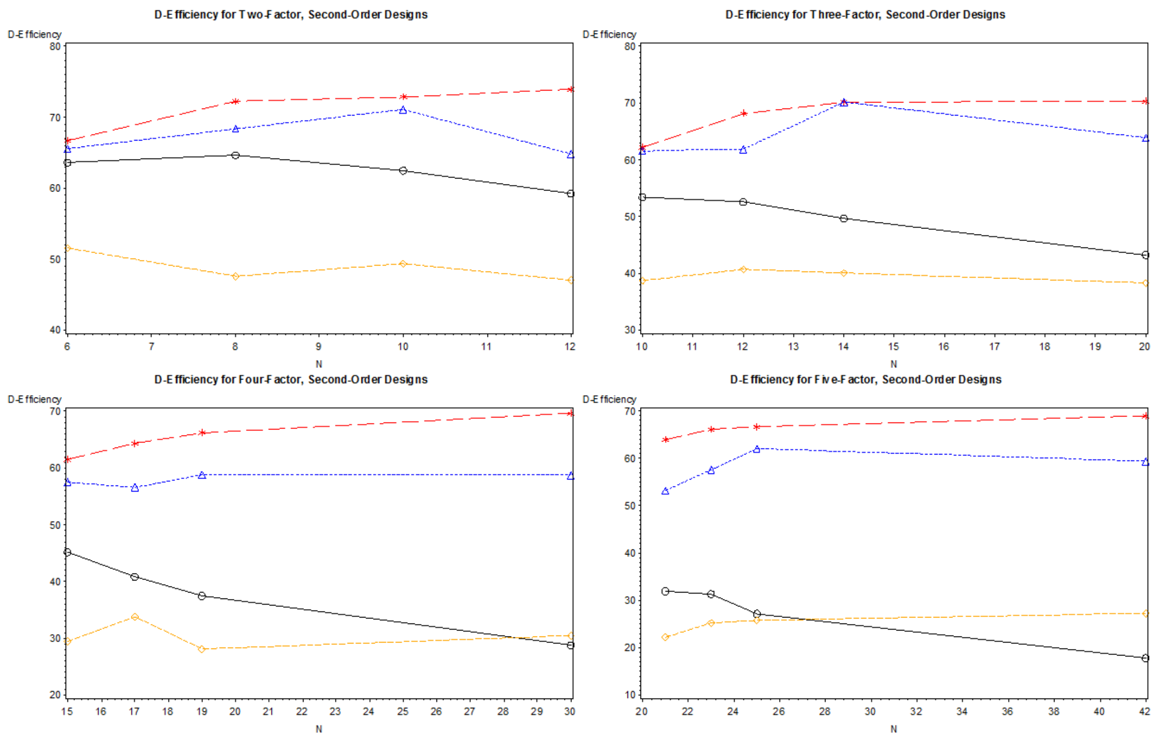

Figure 23. D-efficiencies for second-order designs.

Third-Order Designs

Figure 24 shows examples of designs generated assuming an underlying third-order model, with two factors. It can be seen that the bridge design places the points in the corners of the design rather than the center similar to both of the optimal designs, but again without replicated points. The D-optimal design has 6 replicated points located at (1, 1), (1, 0), (1, -1), (0, -1), (-1, 1) and (-1, -1), while the I-optimal design has 5 replicated points located at (0.5, 0.5), (0.5, -0.5), (0, -1), (-0.5, 0.5), and (-0.5, -0.5).

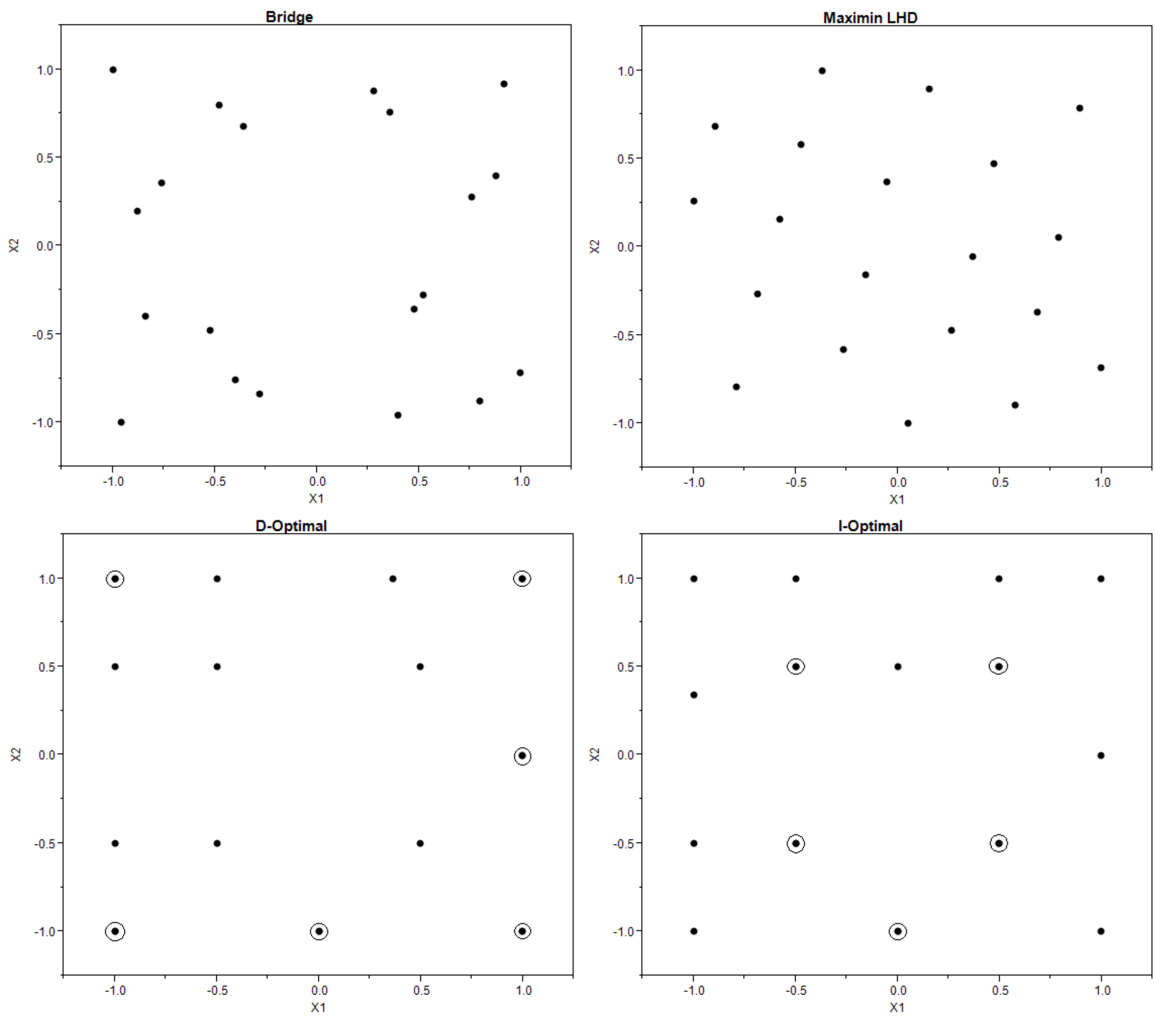

Figure 24. Two-factor, 20-run designs generated assuming a full cubic model.

The prediction variance results shown in Table 9 illustrate that there are no cases in which the maximum prediction variance for the bridge design exceeds that of the Latin hypercube design. The median prediction variance is comparable between the bridge designs and the optimal designs, in most cases within 25%. The maximum prediction variance for the bridge designs was generally no more than 70% higher than the comparable optimal design, although for the three-factor designs the multiple was as high as 13 times higher. The prediction variance results are illustrated in Figure 25, presenting the minimum, median, and maximum for the bridge designs and optimal designs similarly to the second-order designs.

Table 9. Prediction variance estimates for designs generated assuming a third-order polynomial model.

	n = p = 10 runs				n = p + 2 = 12 runs				n = p + 4 = 14 runs				n = 2p = 20 runs			
2 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.57	0.50	0.51	0.52	0.47	0.40	0.38	0.38	0.38	0.31	0.39	0.27	0.26	0.19	0.26	0.20
5%	0.63	0.97	0.61	0.61	0.52	0.58	0.45	0.45	0.42	0.42	0.45	0.32	0.28	0.24	0.34	0.22
25%	0.82	4.12	0.85	0.75	0.59	0.72	0.55	0.55	0.48	0.51	0.53	0.40	0.31	0.31	0.40	0.27
50%	0.95	11.74	1.02	0.86	0.66	0.92	0.62	0.62	0.52	0.66	0.59	0.51	0.34	0.35	0.45	0.31
75%	1.12	28.87	1.31	0.98	0.73	1.23	0.67	0.67	0.63	0.91	0.64	0.59	0.43	0.54	0.49	0.37
90%	1.27	111.43	1.56	1.17	0.91	1.92	0.73	0.73	0.81	2.02	0.69	0.68	0.65	0.91	0.52	0.49
95%	1.34	351.69	1.64	1.37	1.04	3.76	0.79	0.79	0.94	3.35	0.71	0.74	0.74	1.25	0.53	0.56
max	2.10	2455.60	1.92	3.59	1.55	19.61	0.94	0.94	1.48	20.89	0.87	0.91	1.33	4.34	0.69	0.84

	n = p = 20 runs				n = p + 2 = 22 runs				n = p + 4 = 24 runs				n = 2p = 40 runs			
3 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.49	0.45	0.44	0.44	0.31	0.46	0.43	0.33	0.33	0.31	0.36	0.25	0.10	0.15	0.25	0.13
5%	0.64	0.89	0.71	0.63	0.46	0.68	0.56	0.47	0.42	0.53	0.56	0.37	0.18	0.25	0.34	0.18
25%	0.84	1.49	0.89	0.78	0.64	0.94	0.68	0.58	0.55	0.65	0.68	0.47	0.36	0.30	0.41	0.23
50%	1.03	3.05	1.04	0.88	0.89	1.34	0.76	0.67	0.78	0.80	0.76	0.55	0.68	0.38	0.45	0.27
75%	1.45	6.56	1.30	1.00	1.56	2.30	0.83	0.79	1.32	1.24	0.85	0.66	1.36	0.64	0.50	0.35
90%	2.11	14.97	1.69	1.13	2.38	4.51	0.95	0.92	2.10	2.61	0.96	0.82	2.24	1.23	0.53	0.42
95%	2.67	40.21	1.98	1.18	2.91	8.53	1.06	1.02	2.75	4.03	1.04	0.93	2.95	1.74	0.55	0.48
max	6.79	561.73	2.86	2.06	6.77	113.52	2.06	2.68	7.82	33.31	1.30	2.86	8.67	9.10	0.63	1.18

	n = p = 35 runs				n = p + 2 = 37 runs				n = p + 4 = 39 runs				n = 2p = 70 runs			
4 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.41	0.56	0.48	0.39	0.32	0.45	0.37	0.33	0.33	0.36	0.36	0.24	0.07	0.16	0.23	0.10
5%	0.67	2.49	0.80	0.62	0.52	0.72	0.66	0.51	0.50	0.69	0.63	0.40	0.15	0.27	0.32	0.16
25%	0.84	9.63	1.04	0.77	0.74	0.98	0.84	0.64	0.64	0.92	0.77	0.51	0.30	0.32	0.38	0.21
50%	1.05	30.86	1.26	0.90	0.96	1.49	1.00	0.75	0.84	1.29	0.91	0.61	0.60	0.43	0.41	0.26
75%	1.40	109.20	1.53	1.07	1.32	3.00	1.23	0.88	1.17	2.54	1.08	0.74	1.00	0.77	0.45	0.32
90%	1.76	324.09	1.78	1.31	1.74	6.03	1.51	1.08	1.54	5.18	1.26	0.92	1.34	1.42	0.49	0.39
95%	1.97	624.50	1.96	1.56	2.01	9.71	1.69	1.24	1.80	7.97	1.36	1.04	1.54	2.06	0.53	0.43
max	4.92	6352.14	2.86	4.10	5.10	85.05	3.04	3.36	3.77	43.47	2.54	2.88	3.03	14.92	0.72	1.05

	n = p = 56 runs				n = p + 2 = 58 runs				n = p + 4 = 60 runs				n = 2p = 112 runs			
5 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.45	0.85	0.47	0.38	0.37	0.45	0.51	0.30	0.42	0.52	0.52	0.25	0.09	0.17	0.24	0.08
5%	0.80	1.87	0.83	0.65	0.64	1.12	0.84	0.51	0.64	1.30	0.74	0.44	0.16	0.29	0.37	0.14
25%	1.03	3.51	1.08	0.82	0.82	1.88	1.08	0.64	0.81	2.32	0.90	0.57	0.27	0.34	0.42	0.19
50%	1.21	6.06	1.30	0.98	0.98	3.25	1.30	0.77	0.96	3.90	1.05	0.68	0.46	0.45	0.45	0.24
75%	1.46	11.53	1.57	1.19	1.23	6.55	1.59	0.96	1.18	7.57	1.21	0.83	0.68	0.82	0.48	0.30
90%	1.75	24.35	1.85	1.46	1.52	14.22	1.88	1.23	1.42	17.62	1.37	1.02	0.87	1.54	0.50	0.37
95%	1.96	41.88	2.01	1.69	1.72	21.70	2.06	1.46	1.58	30.88	1.47	1.17	0.99	2.16	0.52	0.41
max	4.58	458.03	3.36	4.77	3.13	125.66	2.72	6.07	3.36	444.07	2.42	2.85	1.74	12.05	0.65	0.82

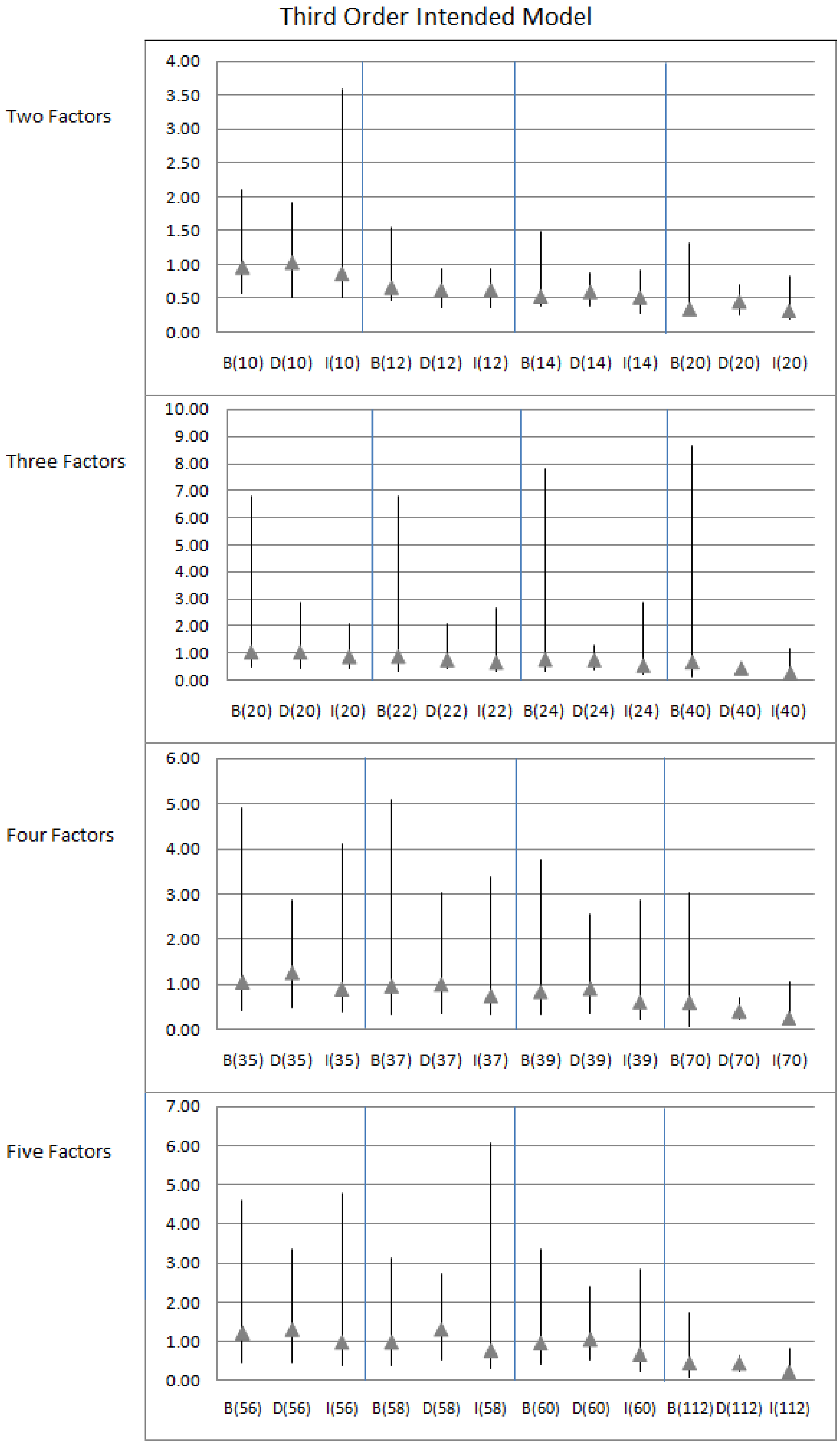

Figure 25. Prediction Variance (minimum, median, and maximum) for designs generated intending to be fit with a third-order polynomial model.

Design efficiencies for the designs assuming an underlying third-order polynomial model are presented in Table 10. The average prediction variance of the Latin hypercube designs is very large at small sample size, and decreases quickly with the inclusion of additional runs.

Table 10. Design efficiencies for designs generated intending to be fit with a third-order polynomial model.

	n = p = 10 runs				n = p + 2 = 12 runs				n = p + 4 = 14 runs				n = 2p = 20 runs			
2 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	22.12	10.56	24.83	22.79	22.26	19.74	25.68	25.68	21.48	16.00	25.96	24.43	19.05	17.56	26.10	23.19
G-efficiency	18.87	1.47	18.11	20.24	20.77	11.32	24.13	24.32	21.70	12.16	24.36	26.25	21.91	20.27	22.56	28.45
A-efficiency	5.44	0.08	5.72	6.65	6.84	2.97	8.11	8.11	6.86	3.31	8.16	8.78	6.88	5.78	7.45	9.58
Avg. Pred Var	6.22	407.55	5.93	5.14	4.14	9.43	3.49	3.49	3.53	7.26	2.98	2.76	2.47	2.94	2.28	1.77

	n = p = 20 runs				n = p + 2 = 22 runs				n = p + 4 = 24 runs				n = 2p = 40 runs			
3 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	12.91	10.69	20.85	21.46	11.64	12.16	21.85	19.99	11.74	12.86	22.68	19.86	7.73	12.56	23.34	19.33
G-efficiency	16.58	3.78	19.72	22.85	15.45	7.55	25.40	24.55	15.55	14.36	19.19	27.52	12.06	19.29	24.09	30.60
A-efficiency	3.53	0.45	5.59	6.28	3.10	1.48	7.17	6.90	3.28	3.33	5.70	7.59	2.08	4.47	6.90	9.17
Avg. Pred Var	9.55	73.56	6.04	5.35	9.84	20.59	4.27	4.43	8.58	8.47	4.92	3.69	8.05	3.78	2.45	1.84

	n = p = 35 runs				n = p + 2 = 37 runs				n = p + 4 = 39 runs				n = 2p = 70 runs			
4 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	13.24	7.10	19.43	16.58	12.01	9.09	20.02	18.00	12.28	9.46	20.77	18.24	8.71	10.53	23.13	19.05
G-efficiency	17.34	0.99	16.08	22.02	17.40	9.02	17.91	24.88	19.96	9.74	21.53	27.86	16.32	20.47	26.14	34.90
A-efficiency	3.65	0.05	4.11	5.23	3.40	1.71	4.58	6.06	3.82	1.69	5.06	6.88	2.90	4.00	7.21	9.45
Avg. Pred Var	9.16	616.55	8.16	6.41	9.31	18.75	6.93	5.24	7.86	17.87	5.97	4.37	5.76	4.21	2.34	1.77

	n = p = 56 runs				n = p + 2 = 58 runs				n = p + 4 = 60 runs				n = 2p = 112 runs			
5 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	13.16	6.51	19.20	15.23	13.21	7.46	20.22	15.14	13.08	7.36	20.38	16.76	10.29	9.17	23.56	18.64
G-efficiency	17.04	3.53	17.22	17.96	18.43	5.56	19.12	26.09	20.58	3.34	22.93	28.53	22.09	19.19	27.13	39.17
A-efficiency	3.17	0.31	3.75	4.42	3.75	0.71	3.94	5.38	3.75	0.44	5.00	6.21	3.96	3.49	6.75	9.81
Avg. Pred Var	10.54	109.49	8.93	7.56	8.60	45.65	8.23	6.00	8.31	71.78	6.28	5.03	4.22	4.85	2.49	1.71

The D-efficiencies of the designs are plotted in Figure 26. The performance of the bridge designs is closer to that of the optimal designs at smaller sample size, rather than larger sample size. As with the prediction variance results, the three-factor bridge designs do not perform as well the designs with other factor levels. The four and five-factor bridge designs still fall short of the Latin hypercube designs when the sample size is increased to 2*p*.

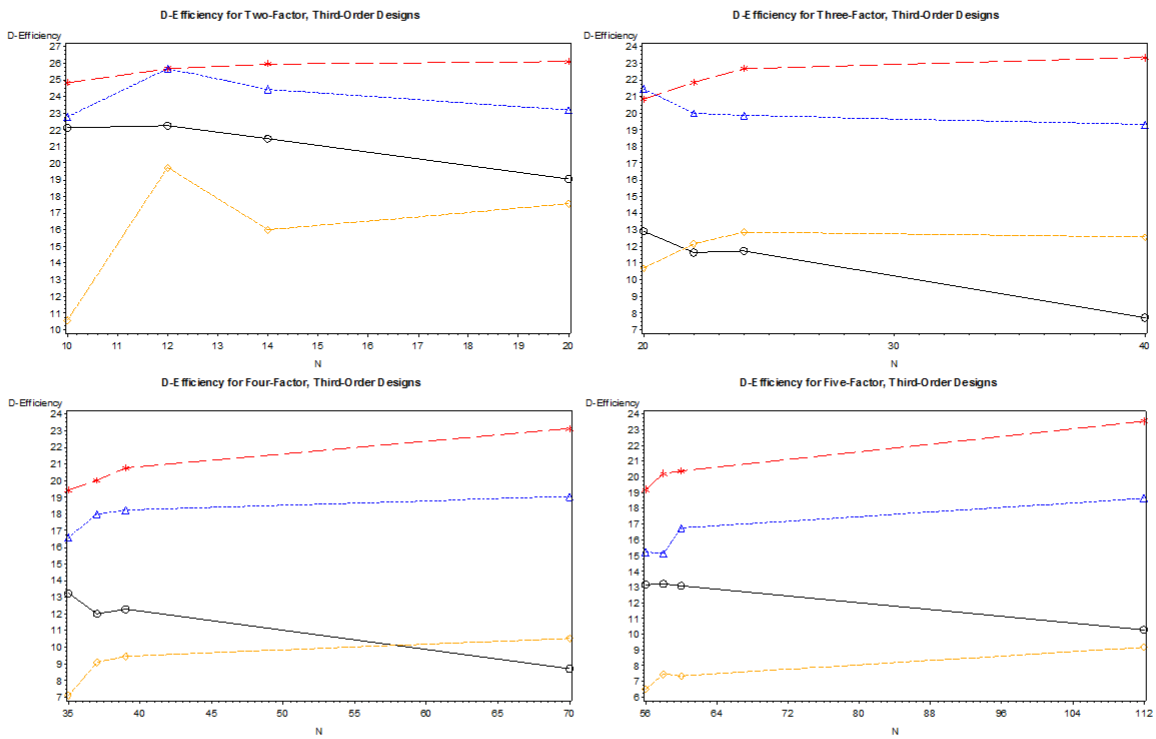

Figure 26. D-efficiencies for third-order designs.

Fourth-Order Designs

Figure 27 shows examples of designs generated assuming an underlying fourth-order model, with two factors. There are four replicated points in each of the optimal designs, all the corner points in the D-optimal design, while the I-optimal design has three replicates at the center point as well as replicates at (0.7, 0) and (-0.7, -0.6).

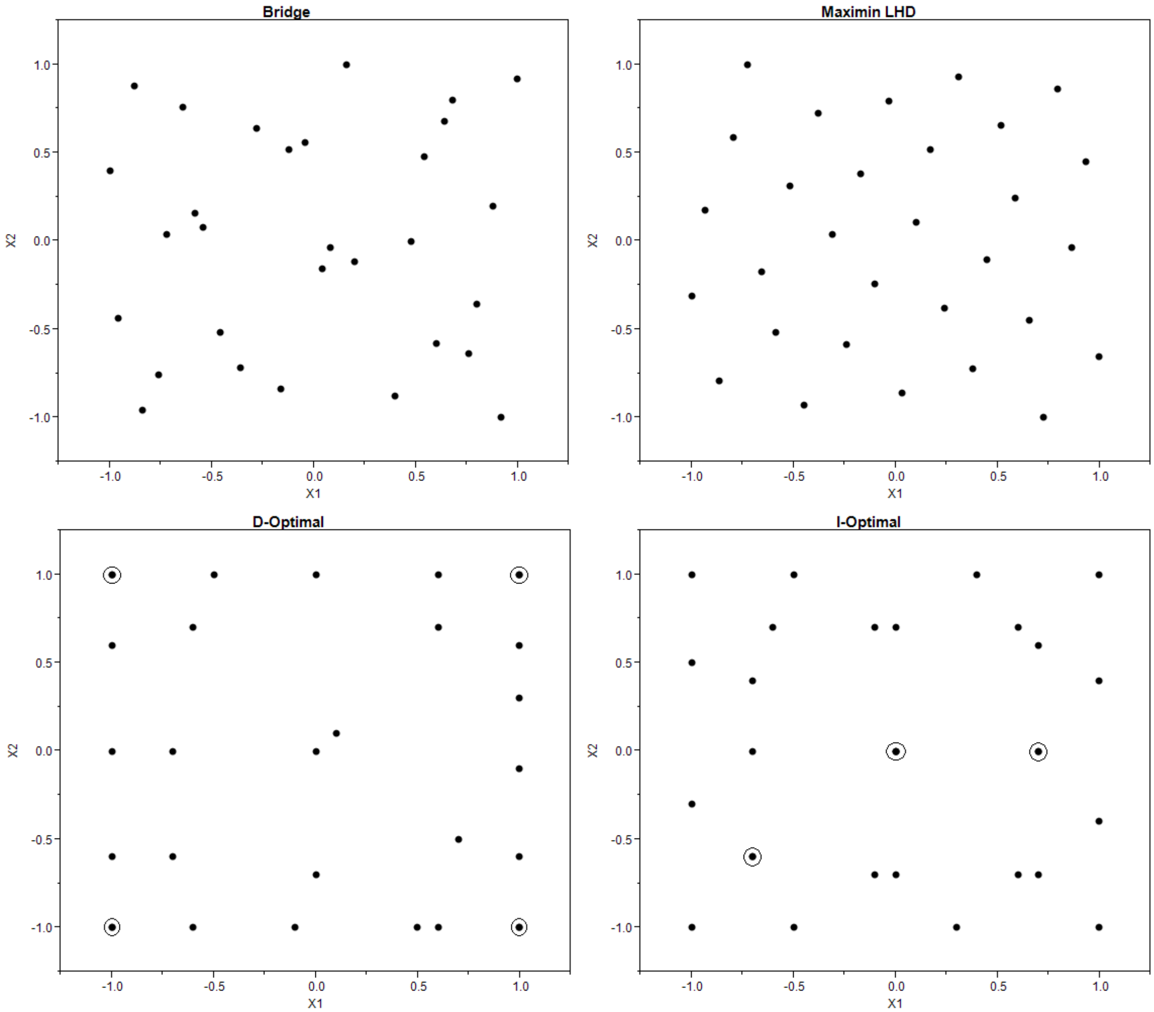

Figure 27. Two-factor, 30-run designs generated assuming a full fourth-order model.

Table 11 presents the prediction variance across the fourth-order designs. There is an interesting disparity in how the median prediction variance compares between design types as opposed to the maximum prediction variance, for designs with two to four factors. The bridge designs actually perform better (have smaller prediction variance) than the D-optimal designs for the majority of two to four-factor designs, and are within 60% of the prediction variance of the I-optimal designs. While the maximum prediction variance of the bridge design is still smaller than that of the Latin hypercube design for all but the 252-run case, it is orders of magnitude larger than that of either of the optimal designs in many cases. The maximum prediction variance for the bridge designs is generally within five times the prediction variance of the optimal designs at two to four factors, but both median and maximum prediction variance are much greater for the bridge designs than the optimal designs in the five-factor case (up to 215 times greater). It could be that the bridge design generation algorithm begins to break down as the number of potential exchanges has reached such high dimensionality.

Table 11. Prediction variance estimates for designs generated assuming a full fourth-order polynomial model.

	n = p = 15 runs				n = p + 2 = 17 runs				n = p + 4 = 19 runs				n = 2p = 30 runs			
2 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.49	0.59	0.53	0.55	0.46	0.45	0.53	0.48	0.41	0.36	0.46	0.46	0.19	0.22	0.27	0.23
5%	0.71	0.77	0.74	0.69	0.59	0.57	0.61	0.54	0.49	0.45	0.56	0.49	0.23	0.24	0.37	0.24
25%	0.81	1.24	0.90	0.78	0.69	0.75	0.68	0.60	0.54	0.58	0.62	0.55	0.27	0.30	0.41	0.26
50%	0.91	2.56	1.02	0.86	0.79	1.09	0.73	0.67	0.62	0.74	0.68	0.59	0.35	0.40	0.44	0.32
75%	1.00	7.89	1.18	0.94	0.91	1.73	0.81	0.80	0.77	1.16	0.73	0.64	0.52	0.50	0.49	0.38
90%	1.22	21.96	1.40	1.00	1.29	3.55	0.89	0.91	1.24	2.42	0.78	0.73	0.93	0.91	0.53	0.51
95%	1.52	32.60	1.56	1.04	1.72	6.98	0.95	0.98	1.61	4.33	0.82	0.83	1.22	1.31	0.54	0.59
max	4.04	383.55	1.94	2.95	3.34	65.51	1.44	2.71	3.43	89.20	1.04	1.87	2.61	6.96	0.65	0.84

	n = p = 35 runs				n = p + 2 = 37 runs				n = p + 4 = 39 runs				n = 2p = 70 runs			
3 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.51	0.52	0.53	0.54	0.40	0.49	0.47	0.46	0.40	0.46	0.47	0.38	0.13	0.20	0.26	0.18
5%	0.70	1.41	0.81	0.67	0.58	0.70	0.69	0.55	0.53	0.56	0.61	0.47	0.20	0.24	0.38	0.19
25%	0.88	7.59	1.01	0.75	0.72	1.12	0.87	0.65	0.66	0.78	0.73	0.57	0.38	0.32	0.42	0.24
50%	1.01	35.76	1.25	0.82	0.89	2.09	1.00	0.74	0.83	1.09	0.82	0.65	0.87	0.40	0.45	0.29
75%	1.24	165.04	1.63	0.92	1.43	5.28	1.22	0.83	1.41	2.03	0.96	0.74	2.19	0.66	0.48	0.36
90%	1.75	768.53	2.05	1.09	2.35	13.57	1.49	0.96	2.36	5.65	1.19	0.86	4.29	1.60	0.50	0.44
95%	2.15	1583.46	2.30	1.25	3.03	27.13	1.70	1.08	3.03	11.31	1.32	0.96	6.02	2.76	0.52	0.50
max	5.16	14999.43	2.96	3.99	12.53	235.70	2.66	2.71	9.20	314.31	1.65	2.34	22.20	25.85	0.58	1.04

	n = p = 70 runs				n = p + 2 = 72 runs				n = p + 4 = 74 runs				n = 2p = 140 runs			
4 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.57	0.70	0.61	0.48	0.53	0.65	0.63	0.44	0.46	0.49	0.58	0.41	0.12	0.18	0.30	0.14
5%	0.82	2.11	0.91	0.64	0.76	2.23	0.84	0.54	0.65	0.94	0.87	0.49	0.16	0.22	0.39	0.16
25%	1.02	7.91	1.13	0.73	0.95	7.47	1.00	0.64	0.78	1.59	1.08	0.59	0.26	0.32	0.42	0.21
50%	1.27	24.69	1.38	0.81	1.15	20.99	1.17	0.73	0.99	2.74	1.28	0.68	0.54	0.42	0.44	0.26
75%	1.75	77.25	1.72	0.95	1.51	63.22	1.41	0.87	1.45	7.18	1.53	0.81	1.06	0.78	0.47	0.33
90%	2.37	289.06	2.15	1.16	2.04	162.21	1.77	1.04	2.09	21.66	1.81	0.97	1.54	1.67	0.49	0.40
95%	2.83	612.57	2.43	1.34	2.45	282.93	2.05	1.19	2.55	41.62	2.02	1.11	1.82	2.77	0.50	0.45
max	7.02	16516.75	4.58	3.59	6.53	5523.04	3.99	3.55	7.73	969.18	3.33	3.04	3.98	25.10	0.59	0.86

	n = p = 126 runs				n = p + 2 = 128 runs				n = p + 4 = 130 runs				n = 2p = 252 runs			
5 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.34	2.36	0.70	0.50	0.33	0.75	0.64	0.44	0.34	0.68	0.68	0.41	0.10	0.19	0.32	0.11
5%	1.74	136.36	1.06	0.63	2.14	2.70	1.06	0.56	1.86	1.52	0.97	0.49	0.22	0.26	0.41	0.14
25%	8.67	3228.33	1.28	0.73	15.17	6.13	1.28	0.65	14.00	3.01	1.17	0.63	0.49	0.34	0.44	0.19
50%	24.55	15080.85	1.52	0.85	44.32	12.90	1.48	0.76	40.74	5.80	1.38	0.76	1.07	0.46	0.46	0.24
75%	51.19	50296.05	1.83	1.06	95.01	32.32	1.71	0.94	86.64	14.79	1.64	0.94	2.37	0.92	0.48	0.31
90%	87.91	156595.02	2.13	1.36	155.92	90.22	1.96	1.20	144.77	37.45	1.91	1.20	4.68	2.10	0.50	0.38
95%	116.95	370891.48	2.33	1.61	200.15	162.46	2.15	1.43	186.75	64.86	2.11	1.42	6.82	3.37	0.52	0.43
max	442.78	5584383.58	4.39	4.81	595.50	3190.28	4.04	5.28	748.01	841.47	3.51	3.45	55.15	28.10	0.60	0.81

The widening gap in prediction variance between the bridge designs and the optimal designs can be seen in Figure 28. In particular, it is worth noting the expansion of the y-axis for the five factor case accommodating the maximum prediction variance of the bridge designs.

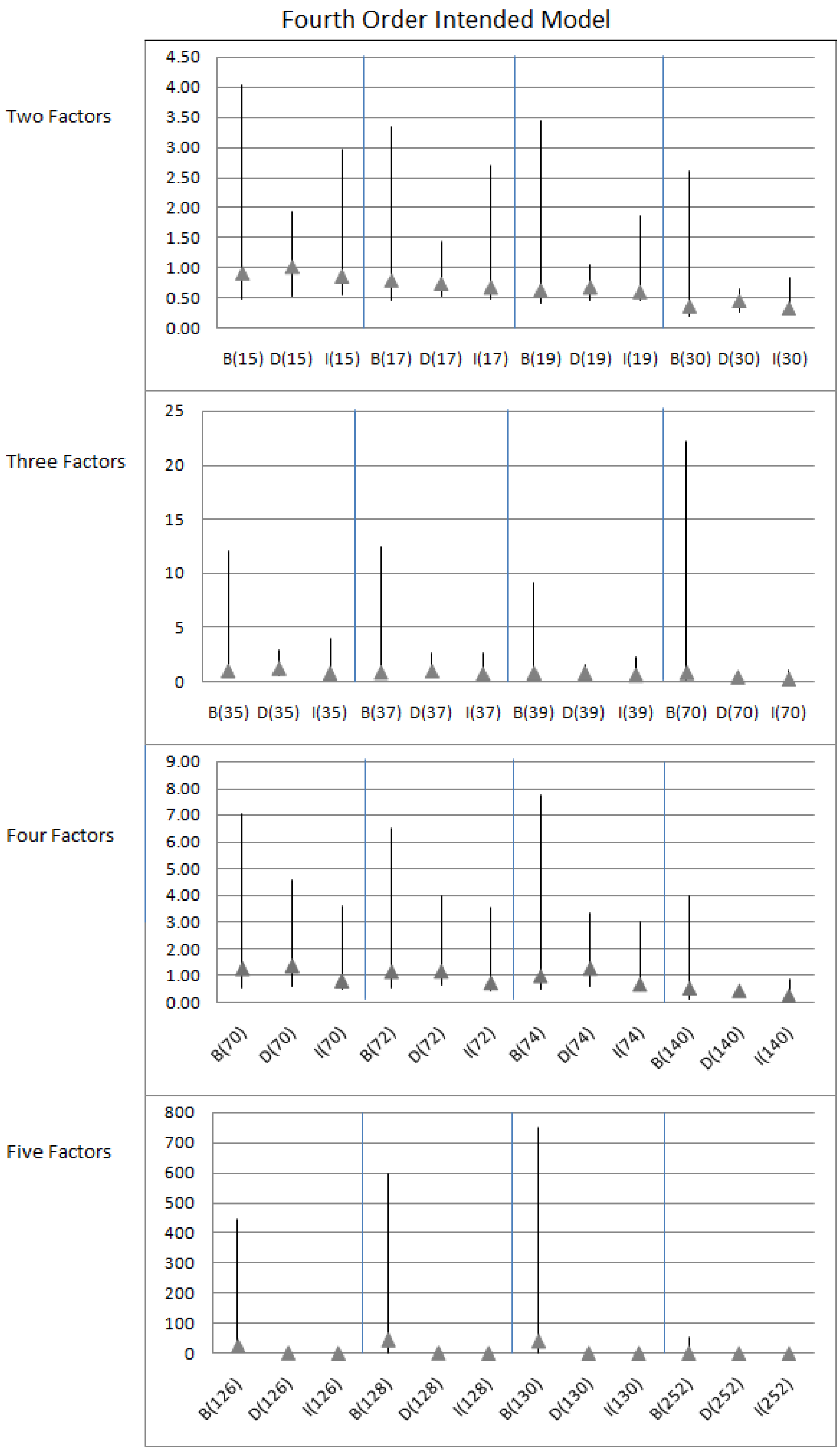

Figure 28. Prediction Variance (minimum, median, and maximum) for designs generated intending to be fit with a fourth-order polynomial model.

The design efficiencies for the fourth-order designs are presented in Table 12. The bridge designs are comparable to the optimal designs when there are only two factors included, and for three factors at small sample size.

Table 12. Design efficiencies for designs generated intending to be fit with a fourth-order polynomial model.

	n = p = 15 runs				n = p + 2 = 17 runs				n = p + 4 = 19 runs				n = 2p = 30 runs			
2 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	9.97	7.73	12.36	11.73	9.40	8.72	12.82	11.39	9.10	7.49	13.00	11.82	8.32	9.38	13.23	11.42
G-efficiency	14.48	2.84	13.93	15.84	15.19	7.80	18.05	15.70	14.84	7.99	18.09	16.60	14.90	12.98	17.34	19.25
A-efficiency	2.61	0.30	2.95	3.05	2.58	1.22	3.57	3.14	2.59	1.41	3.72	3.50	2.55	2.65	3.56	3.91
Avg. Pred Var	13.06	123.00	11.52	11.30	11.73	25.00	8.43	9.59	10.44	19.49	7.26	7.72	6.72	6.52	4.77	4.36

	n = p = 35 runs				n = p + 2 = 37 runs				n = p + 4 = 39 runs				n = 2p = 70 runs			
3 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	6.66	3.57	9.46	8.20	5.36	4.63	9.89	8.66	5.53	5.15	10.38	9.04	2.72	5.80	10.81	8.83
G-efficiency	11.80	0.54	12.36	15.84	11.82	3.66	14.36	17.19	13.05	7.04	18.44	18.36	6.03	12.14	19.56	21.78
A-efficiency	1.74	0.01	2.13	2.36	1.43	0.44	2.43	2.61	1.55	0.91	2.99	2.91	0.51	1.70	3.25	3.49
Avg. Pred Var	19.52	2577.68	16.04	14.50	22.38	78.37	13.25	12.34	19.60	36.55	10.30	10.46	32.78	10.25	5.23	4.86

	n = p = 70 runs				n = p + 2 = 72 runs				n = p + 4 = 74 runs				n = 2p = 140 runs			
4 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	4.91	3.32	8.45	7.04	4.93	3.56	8.68	7.14	4.88	3.63	8.83	7.21	3.74	4.76	10.19	8.05
G-efficiency	11.44	0.77	13.31	16.44	11.00	0.64	14.29	18.17	11.75	2.87	13.79	18.06	13.82	13.20	20.89	23.66
A-efficiency	1.21	0.02	1.78	2.15	1.25	0.03	1.91	2.34	1.26	0.24	1.94	2.43	1.30	1.52	3.15	3.50
Avg. Pred Var	27.98	1472.82	19.17	15.85	26.46	1394.66	17.21	14.08	25.31	139.03	16.58	13.19	12.92	11.56	5.40	4.82

	n = p = 126 runs				n = p + 2 = 128 runs				n = p + 4 = 130 runs				n = 2p = 252 runs			
5 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
D-efficiency	1.03	2.62	8.12	6.11	0.84	2.87	8.26	6.25	0.81	3.00	8.27	6.54	1.39	4.25	10.23	7.94
G-efficiency	1.96	0.02	12.55	17.32	1.36	1.29	13.28	17.37	1.69	2.29	13.62	17.33	7.35	12.41	21.88	26.13
A-efficiency	0.05	0.00	1.57	1.83	0.03	0.05	1.59	2.03	0.03	0.15	1.61	2.06	0.37	1.28	3.04	3.60
Avg. Pred Var	694.58	1442189.00	21.48	18.48	1220.26	644.46	20.89	16.42	1042.47	219.68	20.29	15.86	45.90	13.99	5.59	4.67

The D-efficiencies for the fourth-order designs are plotted in Figure 29. The bridge designs maintain better efficiency characteristics than the Latin hypercube designs for two to four factors with sample sizes of $p + 4$ or less, but are less efficient than the Latin hypercubes for all sample sizes at five factors.

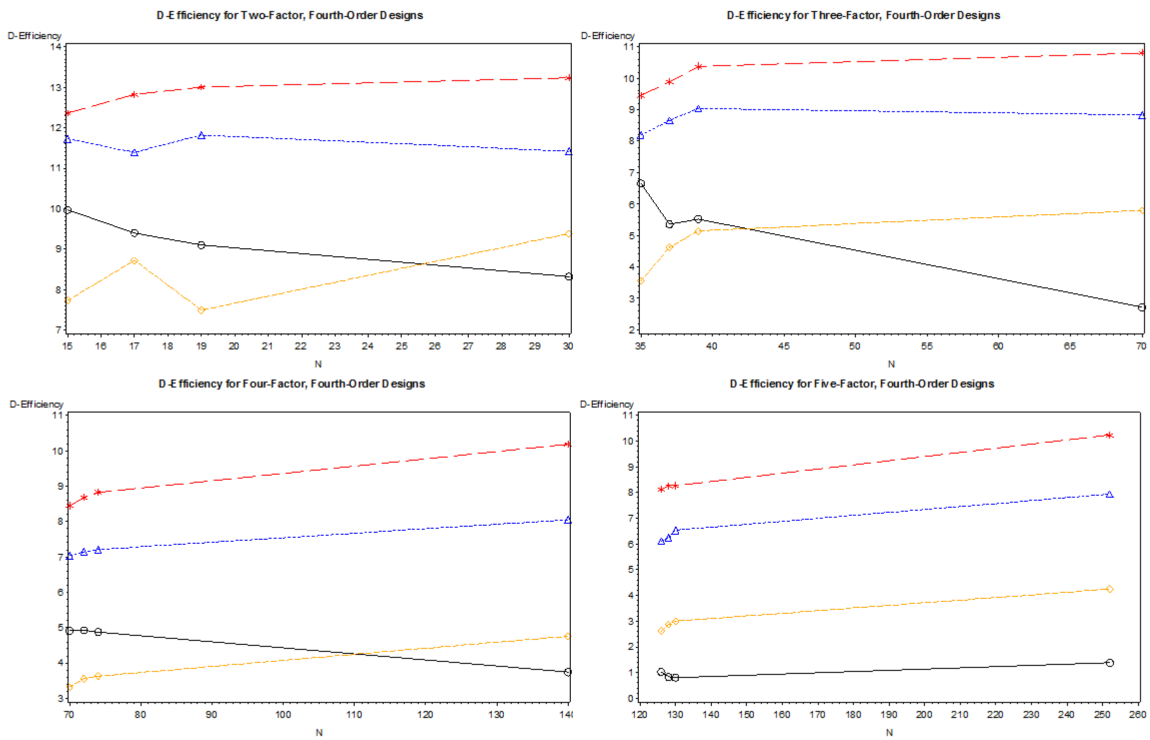

Figure 29. D-efficiencies for fourth-order designs.

Fifth-Order Designs

The algorithm was extended to generate designs intended for full fifth-order polynomials. Fifth-order designs would converge for up to four factors, but not for five or more factors. The prediction variance results are presented in Table 13. Fifth-order bridge designs with two and three factors indicate that the resulting models are not nearly as efficient as the optimal designs, with results that are orders of magnitude greater than the optimal designs although still lower than the Latin hypercubes. With four factors, however, the prediction variance results for four-factor designs are even orders of magnitude greater (5,000 to 82,000 times greater) than the Latin hypercube designs under 252 runs. Given this trending, it follows that fifth-order bridge designs with five factors would likely have unacceptable performance even if the algorithm could be streamlined to allow convergence.

Table 13. Prediction variance estimates for designs generated assuming a full fifth-order polynomial model.

	n = p = 21 runs				n = p + 2 = 23 runs				n = p + 4 = 25 runs				n = 2p = 42 runs			
2 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.61	0.60	0.55	0.60	0.53	0.50	0.54	0.48	0.45	0.46	0.47	0.46	0.17	0.20	0.29	0.24
5%	0.77	0.86	0.81	0.72	0.61	0.61	0.70	0.57	0.53	0.51	0.65	0.53	0.22	0.24	0.40	0.27
25%	0.89	1.30	0.95	0.82	0.69	0.74	0.78	0.64	0.61	0.67	0.71	0.59	0.32	0.28	0.44	0.29
50%	0.99	2.08	1.07	0.89	0.81	1.09	0.88	0.73	0.72	1.09	0.77	0.66	0.45	0.40	0.47	0.33
75%	1.21	5.06	1.25	0.98	0.97	2.31	1.02	0.87	0.89	2.20	0.84	0.74	1.33	0.56	0.51	0.40
90%	2.95	13.28	1.46	1.08	1.77	6.70	1.17	1.00	1.87	7.70	0.95	0.85	4.98	0.94	0.54	0.50
95%	5.57	45.88	1.60	1.18	2.68	18.29	1.22	1.10	3.09	18.00	1.16	0.94	10.45	1.45	0.55	0.59
max	24.08	6866.20	2.27	4.18	18.37	149.35	1.42	2.15	16.39	508.44	1.59	2.32	84.99	13.45	0.59	0.90

	n = p = 56 runs				n = p + 2 = 58 runs				n = p + 4 = 60 runs				n = 2p = 112 runs			
3 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	0.53	0.66	0.57	0.50	0.40	0.53	0.52	0.48	0.32	0.42	0.52	0.43	0.13	0.15	0.28	0.15
5%	0.71	1.12	0.83	0.71	0.61	0.97	0.79	0.65	0.58	0.67	0.69	0.59	0.20	0.22	0.41	0.21
25%	1.02	2.33	1.04	0.80	0.97	2.50	0.95	0.75	0.92	1.06	0.82	0.68	0.40	0.29	0.44	0.24
50%	2.51	5.80	1.21	0.87	2.60	6.12	1.10	0.83	2.17	1.83	0.92	0.76	1.36	0.42	0.46	0.30
75%	9.63	23.86	1.50	0.98	9.05	21.51	1.30	0.94	6.74	4.63	1.03	0.85	5.81	0.70	0.48	0.38
90%	25.89	76.76	1.99	1.15	21.29	83.26	1.52	1.10	15.12	18.04	1.16	0.98	13.15	1.59	0.49	0.47
95%	41.87	178.58	2.32	1.33	30.99	175.59	1.64	1.29	21.72	36.26	1.26	1.08	18.38	2.98	0.50	0.53
max	264.04	3780.01	3.89	5.49	125.77	2605.84	2.34	5.78	154.86	1039.29	1.70	2.93	115.68	40.77	0.58	1.29

	n = p = 126 runs				n = p + 2 = 128 runs				n = p + 4 = 130 runs				n = 2p = 252 runs			
4 Factors	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt	Bridge	Mm LHD	D-opt	I-opt
min	1.72	0.89	0.64	0.59	2.13	0.70	0.66	0.46	1.32	0.64	0.59	0.44	0.12	0.14	0.27	0.11
5%	2585.41	2.25	1.00	0.76	427.59	1.91	0.99	0.66	264.89	1.61	0.95	0.65	0.21	0.21	0.42	0.17
25%	253499.15	6.53	1.28	0.86	60213.48	5.41	1.22	0.79	28666.57	4.04	1.19	0.75	0.44	0.31	0.45	0.21
50%	2021798.81	17.00	1.55	0.96	426892.34	14.78	1.43	0.91	204136.08	11.19	1.45	0.84	1.04	0.45	0.48	0.27
75%	10697490.93	51.51	1.92	1.11	1897125.57	47.01	1.76	1.10	957099.74	37.08	1.73	0.98	2.91	0.81	0.50	0.34
90%	39690057.12	191.91	2.38	1.37	6569503.96	162.14	2.20	1.41	3265246.91	140.13	2.02	1.19	7.10	2.04	0.53	0.42
95%	82132436.85	497.86	2.72	1.62	12588436.58	363.74	2.53	1.66	6479017.75	313.32	2.24	1.38	11.57	3.65	0.55	0.48
max	1821915341.92	22120.34	4.77	7.14	192995324.13	21057.88	4.74	5.97	95433533.21	18984.84	3.82	5.99	64.87	65.50	0.65	0.99

Theoretical Properties Summary

The theoretical properties of bridge designs have been evaluated in terms of prediction variance and design efficiencies. The bridge designs maintain good qualities in terms of each for smaller designs. The difference between the bridge designs and the optimal designs increases as the design complexity increases, either number of factors or underlying model. Although they may still be appropriate for use when a Gaussian process model is intended for modeling the response, bridge designs of five factors or more with an underlying fourth order model would not be recommended when a polynomial is intended for use, nor would bridge designs with underlying fifth-order models.

Empirical Model Fitting Results

The bridge designs have been shown to be comparable to the optimal design types and superior to the Latin hypercube designs in terms of prediction variance properties assuming an underlying polynomial model. The Gaussian process model was fit to the two-factor and three-factor designs to evaluate how the different designs handle departures from the underlying assumptions.

A two-dimensional test function used previously by Jones, Johnson, Montgomery, and Steinberg (2012) in the introduction of bridge designs was used to compare design performance in the case that a Gaussian process model was to be fit. The equation is

$$\eta(x) = \exp[x_1/(x_2+3)] - x_1 \sin(10x_2)$$

and the surface is illustrated in Figure 30.

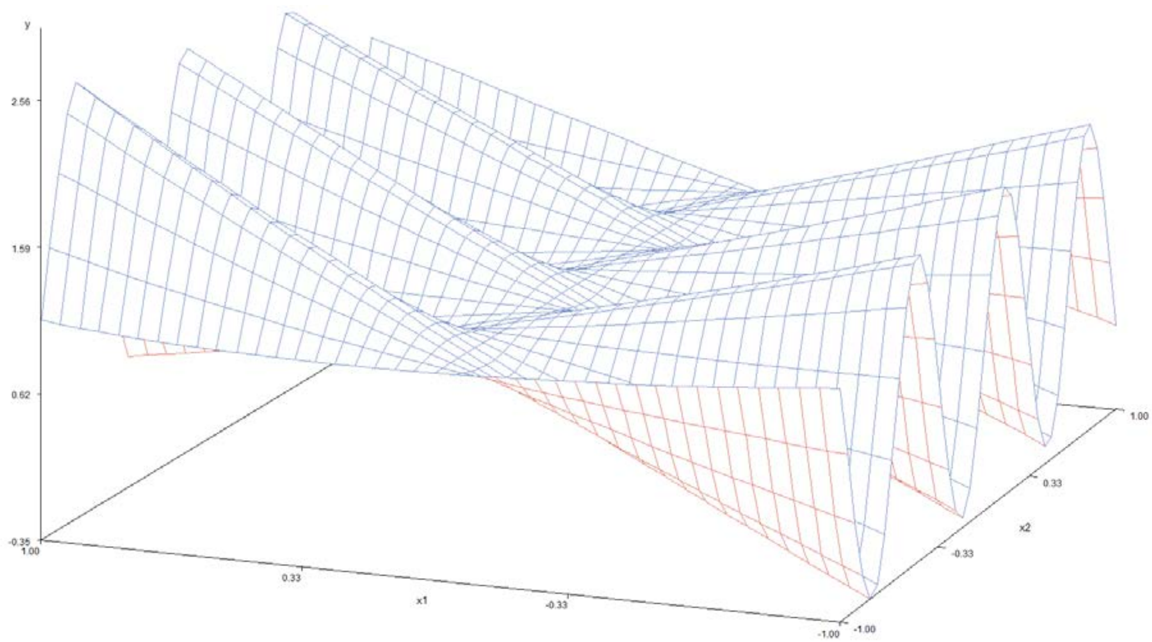

Figure 30. Test Function 1 surface plot.

The Gaussian process model was fit to each of the designs with two-factors, including bridge and optimal designs generated assuming second, third, or fourth order polynomials would be used for analysis (while no analysis model was necessary for the maximin Latin hypercubes). The mean, median, and maximum squared prediction error (SPE) demonstrated across a test set of 10,000 randomly sampled points throughout the design space were recorded. Of the 12 cases tested, the bridge design performed better (had smaller squared error) than the other designs most if not all of the time, as seen in Table 14. The bridge designs likely perform superior to the optimal designs due to the fact that there are no replicates included in the design.

Table 14. Percent of cases in which the bridge design SPE is smaller than the comparator design for Test Function 1.

	Mean SPE	Median SPE	Maximum SPE
Mm LHD	92%	83%	83%
D-Opt	100%	100%	67%
I-Opt	83%	92%	67%

The mean, median, and maximum squared prediction error are illustrated in Figure 31, and the superiority of the bridge design can be seen clearly, particularly for the mean and median.

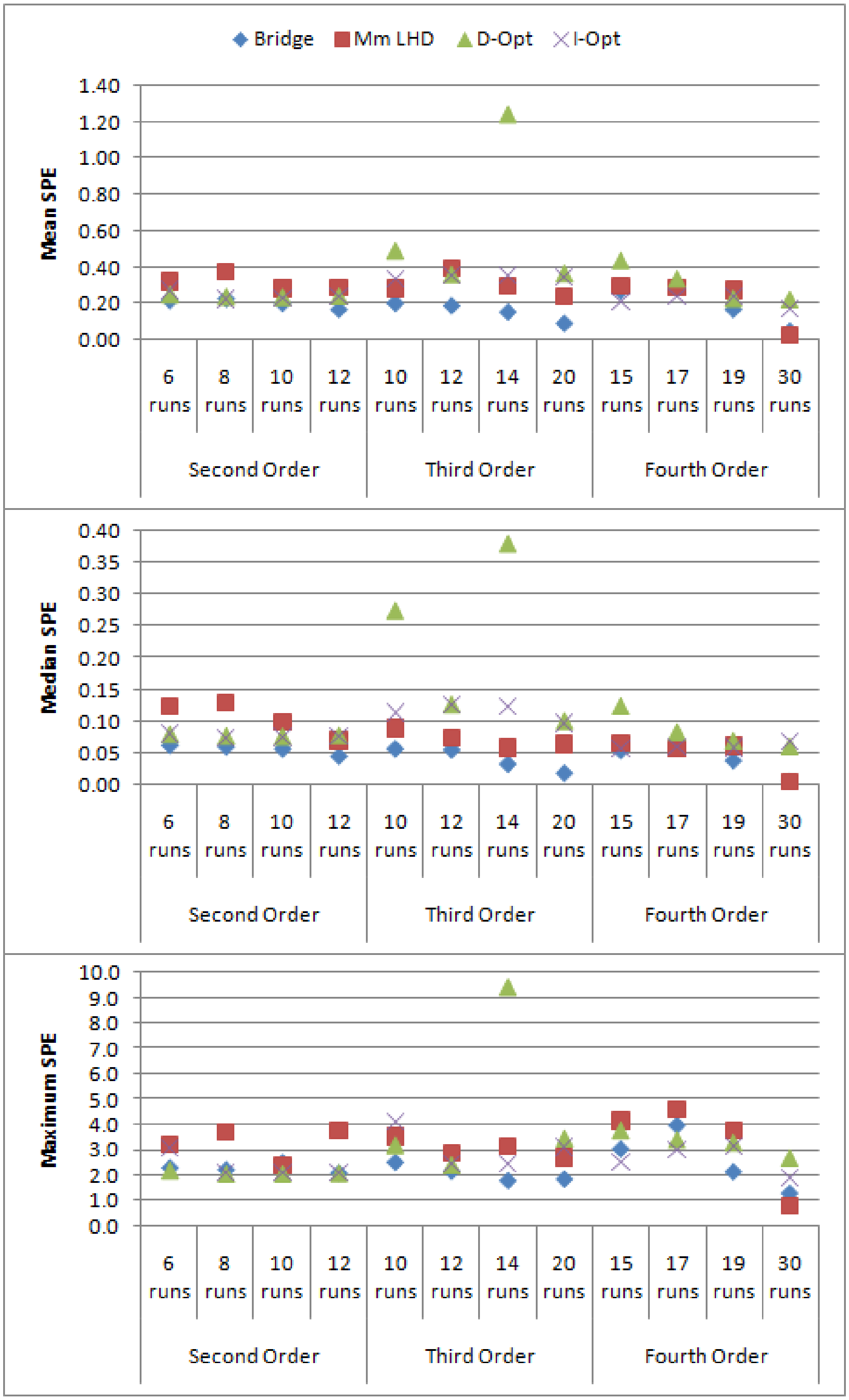

Figure 31. Squared prediction error results (mean, median, and maximum) for Test Function 1.

The same methods were used to evaluate the fitting of a second two-factor test equation, to see if there would be a difference in performance results based on surface complexity. The second test function was used in Santner, Williams, and Notz (2003), and first appeared in Brainin (1972). The function is

$$y = \left(x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$$

$$x_1 \in (-5,10), x_2 \in (0,15)$$

The resulting surface (with x_1 and x_2 scaled from -1 to 1) is presented in Figure 32.

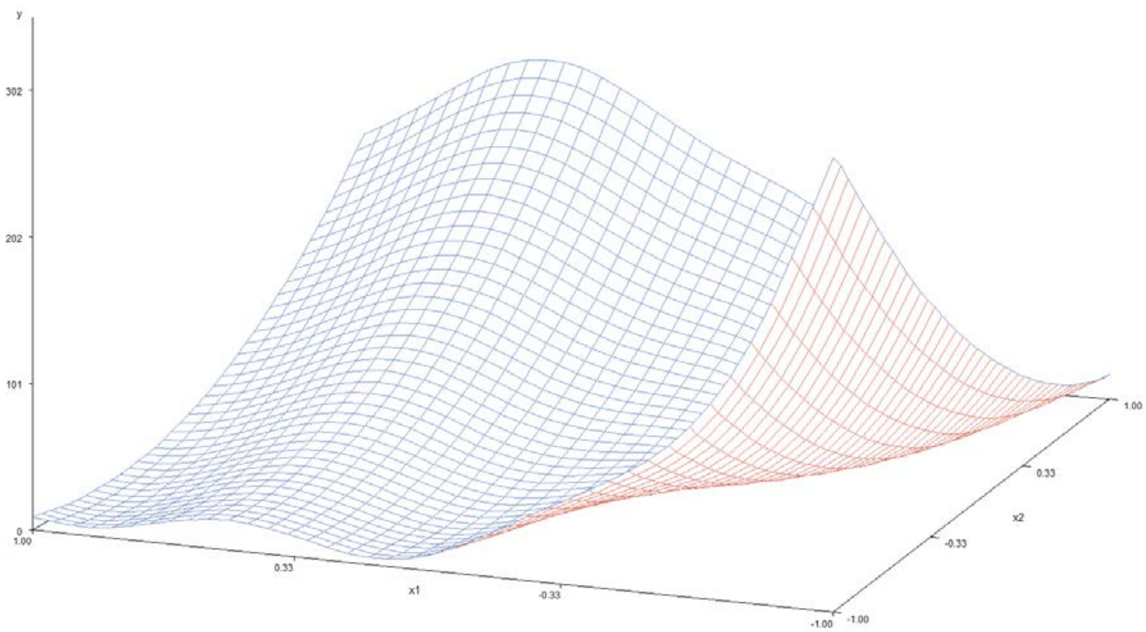

Figure 32. Test Function 2 surface plot.

The range of the response surface of the two equations is quite different. The response surface for Test Function 1 ranges from -0.36 to 2.57, while the response surface for Test Function 2 ranges from 0.4 to 308.1. However, the surface of Test Function 2 appears to be less complex.

The comparative performance of the bridge designs is not as superior as for Test Function 1, and is presented in Table 15. The difference is particularly notable for the

Latin hypercube, which performs much more comparably in terms of mean and median SPE. The bridge design has smaller median SPE than the Latin hypercube of like size in only one out of the 12 cases tested. The bridge designs still perform better than both of the optimal designs a majority of the time.

Table 15. Percent of cases in which the bridge design SPE is smaller than the comparator design for Test Function 2.

	Mean SPE	Median SPE	Maximum SPE
Mm LHD	33%	8%	92%
D-Opt	67%	67%	58%
I-Opt	83%	67%	58%

The mean, median, and maximum SPE are graphed in Figure 33. The interplay between the bridge designs and Latin hypercube designs can be seen easily. The designs track closely together for the mean SPE, while the Latin hypercube design performs better for the median SPE and the bridge design performs better for the maximum SPE.

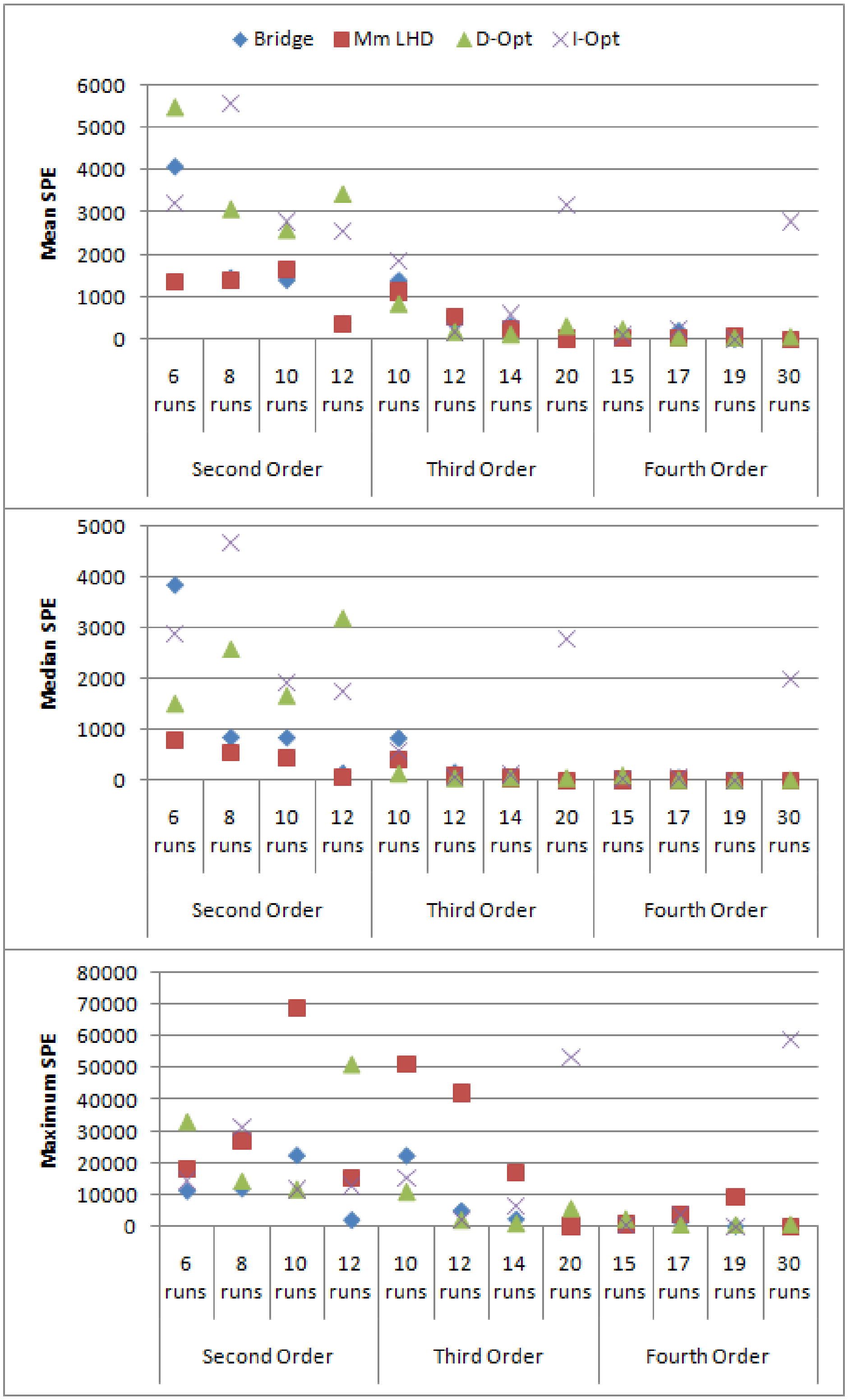

Figure 33. Squared prediction error results (mean, median, and maximum) for Test Function 2.

The last test equation was found in Dette and Pepelyshev (2010), including three factors. The region of interest is the $[0, 1]^3$ cube rather than the $[-1, 1]^3$ cube, so each of the designs was scaled accordingly.

$$\eta(x) = 4\left(x_1 - 2 + 8x_2 - 8x_2^2\right)^2 + \left(3 - 4x_2\right)^2 + 16\sqrt{x_3 + 1}\left(2x_3 - 1\right)^2$$

The bridge designs with three factors perform better than the other design types in terms of SPE much of the time, as presented in Table 16.

Table 16. Percent of cases in which the bridge design SPE is smaller than the comparator design for Test Function 3.

	Mean SPE	Median SPE	Maximum SPE
Mm LHD	50%	42%	67%
D-Opt	67%	58%	42%
I-Opt	67%	75%	58%

The mean, median, and maximum SPE are plotted in Figure 34. In looking at the results this way, it can be seen that the places where the bridge design falls short of the optimal designs is when the underlying model intended for analysis during the design generation phase was assumed to be a fourth-order polynomial.

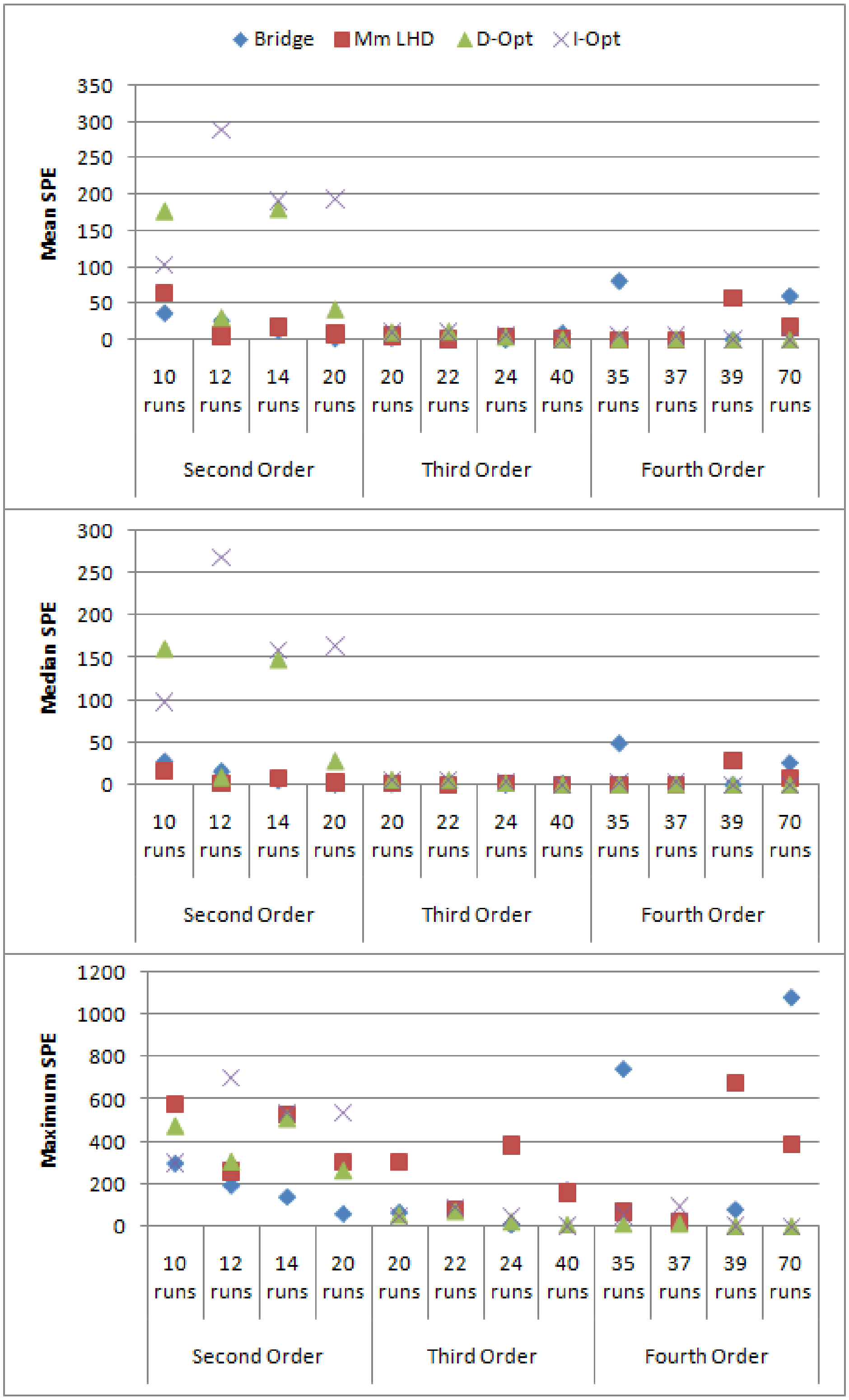

Figure 34. Squared prediction error results (mean, median, and maximum) for Test Function 3.

The results for the three test functions show that the bridge designs are excellent choices for modeling. For the two-factor test functions, the results showed that the bridge design performed superior to the optimal designs a majority of the time in terms of squared prediction error, while being comparable or superior to the Latin hypercube design. For the three-factor test function, the bridge designs performed better than the other designs tested primarily for second and third-order underlying models.

Conclusions

Bridge designs were evaluated in comparison to maximin Latin hypercube designs as well as D and I-optimal designs. The theoretical properties associated with prediction variance and design efficiencies were evaluated in terms of the underlying polynomial models specified during design generation, and the prediction properties in terms of a Gaussian process model were evaluated empirically.

In conclusion, bridge designs are judged to be good choices for computer experiments when the underlying model is hypothesized to be a second or third-order polynomial, or a fourth-order polynomial of up to four factors. They maintain much of the favorable properties of optimal designs, while avoiding pure replicates as well as incidental replicates that would provide little additional information to the design in the case of deterministic models or those in which factors may be insignificant. This makes them attractive for alternative modeling strategies as well, including the commonly used Gaussian process model.

CHAPTER 5 – AUGMENTED BRIDGE DESIGNS

In the previous chapter, it was determined that bridge designs perform well as compared to the Latin hypercubes and traditional optimal designs, balancing an increase in prediction variance (PV) over an optimal design with more desirable space-filling properties. This chapter focuses more on the scenario in which a polynomial model does turn out to be the most appropriate model for analyzing the response. It was hypothesized that augmenting the bridge designs with even a few optimal design points might reduce the prediction variance to help bring the performance more in line with the optimal designs in terms of the prediction variance associated with the polynomial model. A second research question involved whether augmentation with higher order optimal points could be an effective method to hedge against model misspecification in the case that a higher order model was required.

Since the hybrid space-filling designs detailed in Chapter 3 are already a combination of space-filling and I-optimal points, they were not considered for augmentation testing. If additional points were available for inclusion in the preliminary experimentation stage, the total sample size could be included in the initial generation of the design.

Methodology

The catalog of bridge designs created for the work in Chapter 4 is used as a basis to evaluate how augmentation affects the theoretical prediction variance in advance of any model-fitting attempts. The bridge designs range from two to five-factors, and are generated with second, third, or fourth-order polynomial models specified for analysis. Sample sizes range from the minimum number of points necessary to fit the full pre-specified model to twice the minimum number of necessary points for each factor-order combination are created with different sample sizes, with sample sizes presented in Table 17.

Table 17. Design size for base bridge designs to be augmented with I-optimal points.

	Second Order				Third Order				Fourth Order			
Factors	p	p + 2	p + 4	2p	P	p + 2	p + 4	2p	p	p + 2	p + 4	2p
2	6	8	10	12	10	12	14	20	15	17	19	30
3	10	12	14	20	20	22	24	40	35	37	39	70
4	15	17	19	30	35	37	39	70	70	72	74	140
5	21	23	25	42	56	58	60	112	126	128	130	252

The optimization objective of an I-optimal design is to minimize the average prediction variance over the design space, hence it was chosen as the criteria to provide candidate augmentation points. I-optimal designs were generated for each of the factor-order combinations, in order to provide candidate sets for augmentation. The default number of runs suggested by JMP was specified for the original design size. Prior to any augmentation attempt, the I-optimal designs are reduced to ensure that they include only unique points that do not replicate any in the base bridge design. Table 18 presents the original sample sizes for each I-optimal design, as well as the resulting sample size of candidate points in parentheses. In augmenting the designs with two or more I-optimal points, the number of candidate augmentation options increases combinatorically.

Table 18. Sample sizes for I-optimal designs generated as candidate sets for bridge design augmentation.

	Order		
Factors	2	3	4
2	12 (8)	16 (11)	20 (19 or 20)
3	16 (14)	24 (24)	40 (40)
4	20 (19)	40 (39 or 40)	76 (76)
5	28 (27 or 28)	60 (60)	132 (131 or 132)

A test set of 10,000 randomly sampled points within the design space was used to test the effect of adding each candidate point to the base bridge design. The prediction variance for the original design is calculated across the test set, and then again with the addition of the each candidate point in turn. For each addition, the percentage reduction in the mean prediction variance and maximum variance was calculated. In many cases, the point that results in the greatest mean prediction variance reduction differs from the point that reduces the maximum prediction variance, and hence the average of the mean and maximum prediction variance reductions was taken as a measure that may balance the two objectives. SAS macros were written to automate the augmentation and prediction variance evaluation.

The first case tested involves augmenting bridge designs with I-optimal points of the same model order, to evaluate whether the prediction variance can be reduced to levels similar to their counterpart optimal designs. The second case augments bridge designs with I-optimal points of a higher order, in an attempt to mitigate the increased prediction variance that would be associated with model misspecification in the design generation phase. Second-order bridge designs with enough runs to support fitting a third-order model were augmented with third-order I-optimal points, and the prediction variance calculated across the test space assuming a third-order analysis model. Similarly, third-order bridge designs with sufficient runs to fit a fourth-order model were augmented with fourth-order I-optimal points.

For the sake of clarity, from here on, a bridge design that was generated assuming that a second-order polynomial would be used for analysis will be referred to as a second-order bridge design, as well as for third and fourth-order. This convention will also be used for I-optimal designs, with an I-optimal design generated assuming a second-order polynomial would be used for analysis being referred to as a second-order I-optimal design, and so on.

Results

To illustrate the method, a two-factor, second-order bridge design with six runs was selected to be augmented with points from an I-optimal design with 12 points. After omitting replicates and overlap with the base design, there were eight I-optimal points remaining as candidates for augmentation. It was found that candidate point 4, located at (0, 0), would result in the greatest reduction in both mean and maximum prediction variance, reducing the mean from 1.18 to 0.65, and the maximum from 1.85 to 1.40. Figure 35 shows the prediction variance for the original design, and Figure 36 the prediction variance for the design augmented with a single point at (0, 0). It can be seen that the addition of the candidate point reduces the variance across the design space, and flattens the hump in prediction variance in the center of the original design space in particular (Figure 36).

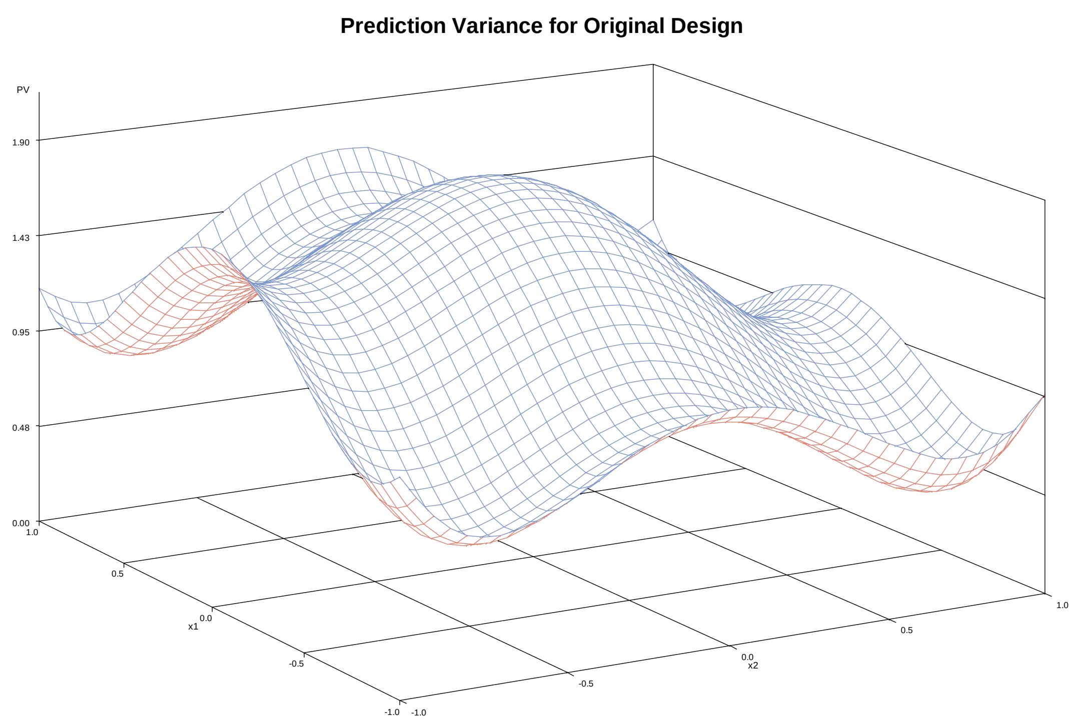

Figure 35. Prediction variance for the original two-factor, second-order bridge design with six runs.

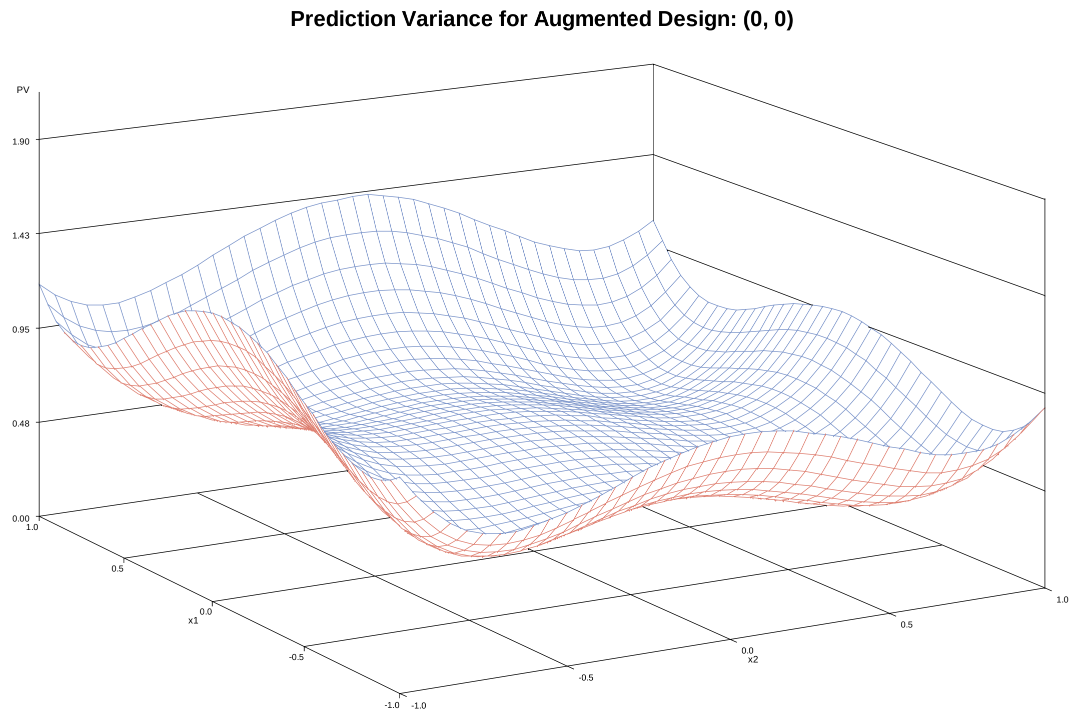

Figure 36. Prediction variance for the two-factor, second order bridge design with six runs augmented with candidate point 4, (0, 0).

Same-Order Augmentation

The results for augmenting bridge designs with one or two I-optimal points are presented by intended analysis model order. For each design, the point (and pair of points) that result in the greatest reduction in mean prediction variance and the greatest reduction in maximum prediction variance are presented. Since those two points (or pairs) are different in many cases (i.e., different points impact the reduction in mean vs. the maximum prediction variance), the point (and pair of points) that results in the greatest average reduction across mean and maximum prediction variance is captured as well.

For cases in which the single augmentation points that resulted in the greatest reduction in the mean prediction variance and the maximum prediction variance were different, special attention was paid to the addition of that pair. While in most cases adding that pair of points performed well, only in very few cases did it result in the optimal reduction in prediction variance across all potential pairs.

The reduced prediction variance statistics are then compared between the original bridge design and comparable D and I-optimal designs as in Chapter 4. Since the prediction variance associated with the bridge design is nearly always less than that of a comparable maximin Latin hypercube design, the Latin hypercube design was omitted from the comparison. The bridge design was augmented with the point(s) that resulted in the greatest reduction in the averaged mean and maximum prediction variance.

Second-Order Designs

Table 19 presents the results for augmenting second-order bridge designs with one and two second-order I-optimal points. With the addition of a single point, the reduction in the mean prediction variance ranged from 7.5% to 44.5%. Intuitively, the larger reductions in prediction variances were seen with the smaller designs, since the new point represents a larger proportion of the total information for the design. The reduction in maximum prediction variance ranged from 3.9% to 37.4%, and was less associated with design size.

The addition of a second point reduces the mean prediction variance of the second-order designs by an additional 6.1%-8.8% (for a reduction of 13.6%-52.9% over the base design). The maximum prediction variance of the second-order designs reduces by an additional 3.2%-24.3% (8.5%-44.9% overall).

Table 19. Augmentation results for bridge designs generated with underlying second-order polynomial models.

	2 Factors	n = p = 6 runs			n = p + 2 = 8 runs			n = p + 4 = 10 runs			n = 2p = 12 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	4	4	4	4	6	4	4	7	4	4	2	4
	Point	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(1, -1)	(0, 0)	(0, 0)	(1, 0)	(0, 0)	(0, 0)	(-1, 1)	(0, 0)
	Percentage	44.5%	24.2%	34.3%	27.0%	14.4%	16.0%	14.0%	3.9%	7.7%	10.0%	4.4%	5.6%
2 points	Index	4 & 7	3 & 7	4 & 7	4 & 5	6 & 8	4 & 6	4 & 5	6 & 8	6 & 8	4 & 7	5 & 6	5 & 7
	Points	(0, 0)	(0, -1)	(0, 0)	(0, 0)	(1, -1)	(0, 0)	(0, 0)	(1, -1)	(1, -1)	(0, 0)	(0, 1)	(0, 1)
		(1, 0)	(1, 0)	(1, 0)	(0, 1)	(1, 1)	(1, -1)	(0, 1)	(1, 1)	(1, 1)	(1, 0)	(1, -1)	(1, 0)
	Percentage	52.9%	39.0%	43.9%	33.2%	19.0%	24.1%	21.8%	16.8%	13.5%	17.1%	8.5%	11.0%

	3 Factors	n = p = 10 runs			n = p + 2 = 12 runs			n = p + 4 = 14 runs			n = 2p = 20 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	8	1	6	8	8	8	2	14	12	6	14	14
	Point	(0, 0, 0)	(-1, -1, -1)	(0, -1, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(-1, 0, 0)	(1, 1, 1)	(1, 0, 0)	(0, -1, 0)	(1, 1, 1)	(1, 1, 1)
	Percentage	16.7%	9.4%	10.1%	14.7%	5.8%	10.3%	11.7%	7.2%	6.7%	7.5%	9.1%	6.7%
2 points	Index	8 & 13	4 & 14	6 & 13	8 & 13	12 & 14	8 & 9	2 & 9	1 & 9	1 & 2	6 & 7	5 & 14	5 & 14
	Point	(0, 0, 0)	(-1, 1, -1)	(0, -1, 0)	(0, 0, 0)	(1, -1, 1)	(0, 0, 0)	(-1, 0, 0)	(-1, -1, -1)	(-1, -1, -1)	(0, -1, 0)	(-1, 1, 1)	(-1, 1, 1)
		(1, 0, 0)	(1, 1, 1)	(1, 0, 0)	(1, 0, 0)	(1, 1, 1)	(0, 0, 1)	(0, 1, 0)	(0, 1, 0)	(-1, 0, 0)	(0, 0, -1)	(1, 1, 1)	(1, 1, 1)
	Percentage	25.3%	18.2%	18.9%	22.0%	9.0%	14.9%	20.0%	12.5%	14.2%	13.6%	13.6%	11.1%

	4 Factors	n = p = 15 runs			n = p + 2 = 17 runs			n = p + 4 = 19 runs			n = 2p = 30 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	11	3	3	18	16	8	8	19	12	9	9	9
	Point	(0, 0, 0, 0)	(-1, -1, 1, 1)	(-1, -1, 1, 1)	(1, 1, 0, -1)	(1, 0, 1, 1)	(0, -1, 0, 0)	(0, -1, 0, 0)	(1, 1, 1, 0)	(0, 1, 1, 1)	(0, 0, -1, 0)	(0, 0, -1, 0)	(0, 0, -1, 0)
	Percentage	15.5%	22.1%	16.0%	8.0%	8.8%	8.1%	9.1%	15.3%	11.7%	9.1%	10.0%	9.5%
2 points	Index	11 & 18	3 & 16	3 & 16	8 & 18	2 & 8	8 & 9	8 & 12	12 & 19	8 & 12	9 & 12	10 & 15	6 & 10
	Point	(0, 0, 0, 0)	(-1, -1, 1, 1)	(-1, -1, 1, 1)	(0, -1, 0, 0)	(-1, -1, 0, -1)	(0, -1, 0, 0)	(0, -1, 0, 0)	(0, 1, 1, 1)	(0, -1, 0, 0)	(0, 0, -1, 0)	(0, 0, 0, -1)	(-1, 1, 0, 1)
		(1, 1, 0, -1)	(1, 0, 1, 1)	(1, 0, 1, 1)	(1, 1, 0, -1)	(0, -1, 0, 0)	(0, 0, -1, 0)	(0, 1, 1, 1)	(1, 1, 1, 0)	(0, 1, 1, 1)	(0, 1, 1, 1)	(1, -1, 1, -1)	(0, 0, 0, -1)
	Percentage	23.3%	36.8%	26.6%	15.8%	17.5%	16.3%	16.7%	21.4%	18.5%	17.7%	19.2%	17.3%

	5 Factors	n = p = 21 runs			n = p + 2 = 23 runs			n = p + 4 = 25 runs			n = 2p = 42 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	17	16	17	10	21	10	5	19	19	23	23	23
	Point	(0, 1, 1, 1, 0)	(0, 1, -1, -1, 1)	(0, 1, 1, 1, 0)	(0, -1, 0, 0, 0)	(1, -1, 1, 1, 1)	(0, -1, 0, 0, 0)	(-1, 0, -1, -1, 0)	(1, -1, -1, 1, -1)	(1, -1, -1, 1, -1)	(1, 1, -1, -1, -1)	(1, 1, -1, -1, -1)	(1, 1, -1, -1, -1)
	Percentage	16.0%	4.4%	9.0%	11.9%	7.9%	8.1%	13.1%	37.4%	23.5%	8.6%	15.0%	11.8%
2 points	Index	17 & 26	4 & 17	4 & 17	10 & 15	9 & 21	6 & 9	5 & 17	2 & 19	15 & 19	3 & 15	17 & 23	17 & 23
	Point	(0, 1, 1, 1, 0)	(-1, -1, 1, 1, -1)	(-1, -1, 1, 1, -1)	(0, -1, 0, 0, 0)	(-1, 1, 1, -1, -1)	(-1, 0, 1, 1, 1)	(-1, 0, -1, -1, 0)	(-1, -1, -1, 1, 1)	(0, 0, 1, 0, 1)	(-1, -1, 1, 1, -1)	(1, -1, -1, -1, 1)	(1, -1, -1, -1, 1)
		(1, 1, 0, 0, 0)	(0, 1, 1, 1, 0)	(0, 1, 1, 1, 0)	(0, 0, 1, 0, 1)	(1, -1, 1, 1, 1)	(-1, 1, 1, -1, -1)	(0, 1, 1, 1, 0)	(1, -1, -1, 1, -1)	(1, -1, -1, 1, -1)	(0, 1, -1, -1, 1)	(1, 1, -1, -1, -1)	(1, 1, -1, -1, -1)
	Percentage	23.0%	28.8%	25.1%	20.7%	24.3%	19.9%	20.2%	44.9%	29.7%	16.5%	25.4%	20.7%

The mean and maximum prediction variance for the second-order bridge and optimal designs is plotted in Figure 37. For underlying second order polynomial models, the bridge designs already perform comparably to the optimal designs in many cases, particularly for smaller designs (factors and runs). In cases in which the bridge design has demonstrably higher prediction variance than the optimal designs, such as the second-order, five-factor designs, the reduction in prediction variance due to the augmentation is still not enough to bring the prediction variance to comparable levels.

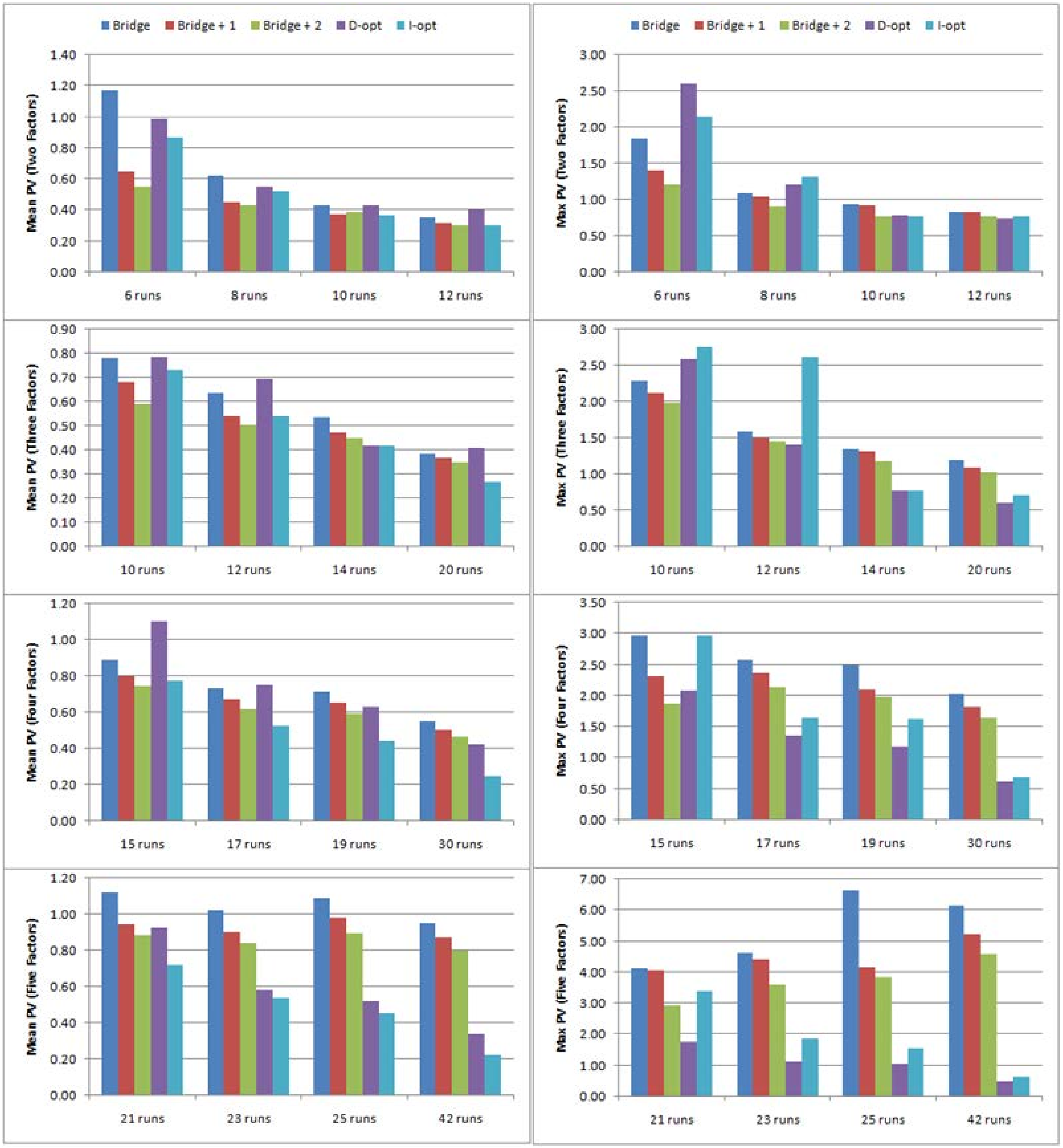

Figure 37. Mean and maximum prediction variance for original and augmented second-order designs.

Third-Order Designs

The results for augmenting third-order bridge designs with one and two third-order I-optimal points are presented in Table 20. The reduction in mean prediction variance ranged from 3.8% to 18.3% with the addition of a single I-optimal point, while the reduction in maximum prediction variance ranged from 4.1% to 26.3%.

The addition of a second I-optimal point reduces the mean prediction variance by an additional 3.2%-13.0% (for a reduction of 7.0%-29.7% over the base design). The maximum prediction variance reduces by an additional 3.2%-21.4% (11.2%-37.2% overall).

Table 20. Augmentation results for bridge designs generated with underlying third-order polynomial models.

	2 Factors	n = p = 10 runs			n = p + 2 = 12 runs			n = p + 4 = 14 runs			n = 2p = 20 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	4	7	7	4	6	6	9	1	4	6	11	11
	Point	(-0.5, -0.5)	(0, 1)	(0, 1)	(-0.5, 0.5)	(0, 1)	(0, 1)	(0.5, 0.5)	(-1, -1)	(-0.5, -0.5)	(0, 1)	(1, 1)	(1, 1)
	Percentage	15.9%	21.9%	14.8%	9.0%	4.1%	5.5%	8.1%	5.8%	5.5%	5.4%	11.8%	8.1%
2 points	Index	4 & 8	1 & 7	4 & 7	4 & 7	6 & 9	6 & 9	4 & 9	1 & 7	1 & 7	2 & 6	9 & 11	9 & 11
	Point	(-0.5, -0.5)	(-1, -1)	(-0.5, -0.5)	(-0.5, 0.5)	(0, 1)	(0, 1)	(-0.5, -0.5)	(-1, -1)	(-1, -1)	(-1, 0)	(1, -1)	(1, -1)
		(0.5, -0.5)	(0, 1)	(0, 1)	(0.5, -0.5)	(1, -1)	(1, -1)	(0.5, 0.5)	(0, 1)	(0, 1)	(0, 1)	(1, 1)	(1, 1)
	Percentage	28.9%	28.7%	24.2%	17.3%	15.1%	12.8%	15.4%	13.0%	12.1%	10.6%	24.1%	16.2%

	3 Factors	n = p = 20 runs			n = p + 2 = 22 runs			n = p + 4 = 24 runs			n = 2p = 40 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	21	21	21	14	3	3	3	23	23	5	5	5
	Point	(1, 0, -1)	(1, 0, -1)	(1, 0, -1)	(0, 1, 0.5)	(-1, -0.5, 0.5)	(-1, -0.5, 0.5)	(-1, -0.5, 0.5)	(1, 1, -0.5)	(1, 1, -0.5)	(-1, 1, -0.5)	(-1, 1, -0.5)	(-1, 1, -0.5)
	Percentage	10.2%	17.5%	13.9%	13.2%	15.8%	12.0%	13.3%	23.1%	18.2%	18.3%	19.2%	18.8%
2 points	Index	21 & 22	20 & 21	21 & 22	12 & 14	4 & 12	4 & 12	3 & 22	11 & 23	22 & 23	5 & 12	6 & 13	6 & 13
	Point	(1, 0, -1)	(1, -1, 1)	(1, 0, -1)	(0, -0.5, 1)	(-1, 0.5, -1)	(-1, 0.5, -1)	(-1, -0.5, 0.5)	(0, -1, 0.5)	(1, 0, 0.5)	(-1, 1, -0.5)	(-1, 1, 1)	(-1, 1, 1)
		(1, 0, 0.5)	(1, 0, -1)	(1, 0, 0.5)	(0, 1, 0.5)	(0, -0.5, 1)	(0, -0.5, 1)	(1, 0, 0.5)	(1, 1, -0.5)	(1, 1, -0.5)	(0, -0.5, 1)	(0, 1, -1)	(0, 1, -1)
	Percentage	19.7%	26.2%	22.7%	22.1%	37.2%	27.2%	23.9%	29.7%	25.7%	29.7%	32.4%	27.7%

	4 Factors	n = p = 35 runs			n = p + 2 = 37 runs			n = p + 4 = 39 runs			n = 2p = 70 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	14	38	38	7	28	28	18	31	31	24	13	13
	Point	(-0.5, -0.5, 0, -0.5)	(1, 1, -1, -1)	(1, 1, -1, -1)	(-1, 0, 0.5, -0.5)	(0.5, 1, 0.5, -1)	(0.5, 1, 0.5, -1)	(-0.5, 1, 0, 0.5)	(1, -1, 1, 1)	(1, -1, 1, 1)	(0.5, -1, 0, 0.5)	(-0.5, -0.5, -1, 0.5)	(-0.5, -0.5, -1, 0.5)
	Percentage	7.5%	23.5%	14.9%	7.1%	26.3%	15.7%	4.9%	11.4%	7.1%	6.6%	11.2%	8.5%
2 points	Index	14 & 38	31 & 38	31 & 38	7 & 12	7 & 38	7 & 38	18 & 33	10 & 25	10 & 33	21 & 24	3 & 21	21 & 24
	Point	(-0.5, -0.5, 0, -0.5)	(1, -1, 1, 1)	(1, -1, 1, 1)	(-1, 0, 0.5, -0.5)	(-1, 0, 0.5, -0.5)	(-1, 0, 0.5, -0.5)	(-0.5, 1, 0, 0.5)	(-1, 1, -0.5, -1)	(-1, 1, -0.5, -1)	(0, 0.5, -0.5, 1)	(-1, -1, -0.5, 0)	(0, 0.5, -0.5, 1)
		(1, 1, -1, -1)	(1, 1, -1, -1)	(1, 1, -1, -1)	(-0.5, -1, 0.5, 1)	(1, 1, -1, -1)	(1, 1, -1, -1)	(1, -0.5, 0.5, 0)	(0.5, -0.5, 1, 1)	(1, -0.5, 0.5, 0)	(0.5, -1, 0, 0.5)	(0, 0.5, -0.5, 1)	(0.5, -1, 0, 0.5)
	Percentage	12.6%	31.6%	20.9%	12.3%	33.3%	22.2%	9.6%	14.6%	11.2%	12.2%	17.8%	14.5%

	5 Factors	n = p = 56 runs			n = p + 2 = 58 runs			n = p + 4 = 60 runs			n = 2p = 112 runs		
		Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	15	48	15	17	46	46	39	28	28	32	32	32
	Point	(-1, 1, -1, 1, -1)	(1, -1, 1, 1, 1)	(-1, 1, -1, 1, -1)	(-0.5, -1, 0, -0.5, -0.5)	(1, -1, 0, -0.5, 1)	(1, -1, 0, -0.5, 1)	(0.5, 0.5, -1, -0.5, -0.5)	(0, -0.5, 1, 0, -1)	(0, -0.5, 1, 0, -1)	(0, 1, -0.5, -0.5, 1)	(0, 1, -0.5, -0.5, 1)	(0, 1, -0.5, -0.5, 1)
	Percentage	6.9%	17.5%	11.7%	4.3%	7.9%	5.7%	4.1%	19.7%	10.6%	3.8%	8.5%	6.1%
2 points	Index	15 & 23	28 & 48	28 & 48	17 & 30	7 & 17	7 & 17	31 & 39	6 & 38	31 & 38	24 & 32	32 & 47	32 & 46
	Point	(-1, 1, -1, 1, -1)	(0, -0.5, 1, 0, -1)	(0, -0.5, 1, 0, -1)	(-0.5, -1, 0, -0.5, -0.5)	(-1, -0.5, 0, 0.5, -0.5)	(-1, -0.5, 0, 0.5, -0.5)	(0, 0.5, 1, 0.5, -0.5)	(-1, -1, 1, 1, -0.5)	(-1, -1, 1, 1, -0.5)	(-0.5, 1, -1, 0, 0)	(0, 1, -0.5, -0.5, 1)	(0, 1, -0.5, -0.5, 1)
		(-0.5, 0, 0.5, -1, 1)	(1, -1, 1, 1, 1)	(1, -1, 1, 1, 1)	(0, 0.5, 1, -1, 0.5)	(-0.5, -1, 0, -0.5, -0.5)	(-0.5, -1, 0, -0.5, -0.5)	(0.5, 0.5, -1, -0.5, -0.5)	(0.5, 0, 0.5, -1, -1)	(0.5, 0, 0.5, -1, -1)	(0, 1, -0.5, -0.5, 1)	(1, -1, 1, -0.5, -1)	(1, -1, 0, -0.5, 1)
	Percentage	12.0%	27.1%	17.5%	8.0%	11.2%	9.4%	8.0%	26.2%	16.5%	7.0%	11.7%	8.5%

The prediction variance for the comparative designs in third-order is illustrated in Figure 38. As with the second-order designs, for smaller designs the prediction variance was already comparable to the optimal designs. In cases where the difference between designs begins to widen, such as the cases where the original sample size was set to twice the minimum number of parameters needed to fit the full polynomial model, the reduction in prediction variance for the augmented bridge designs is not large enough to make them approximate the optimal designs.

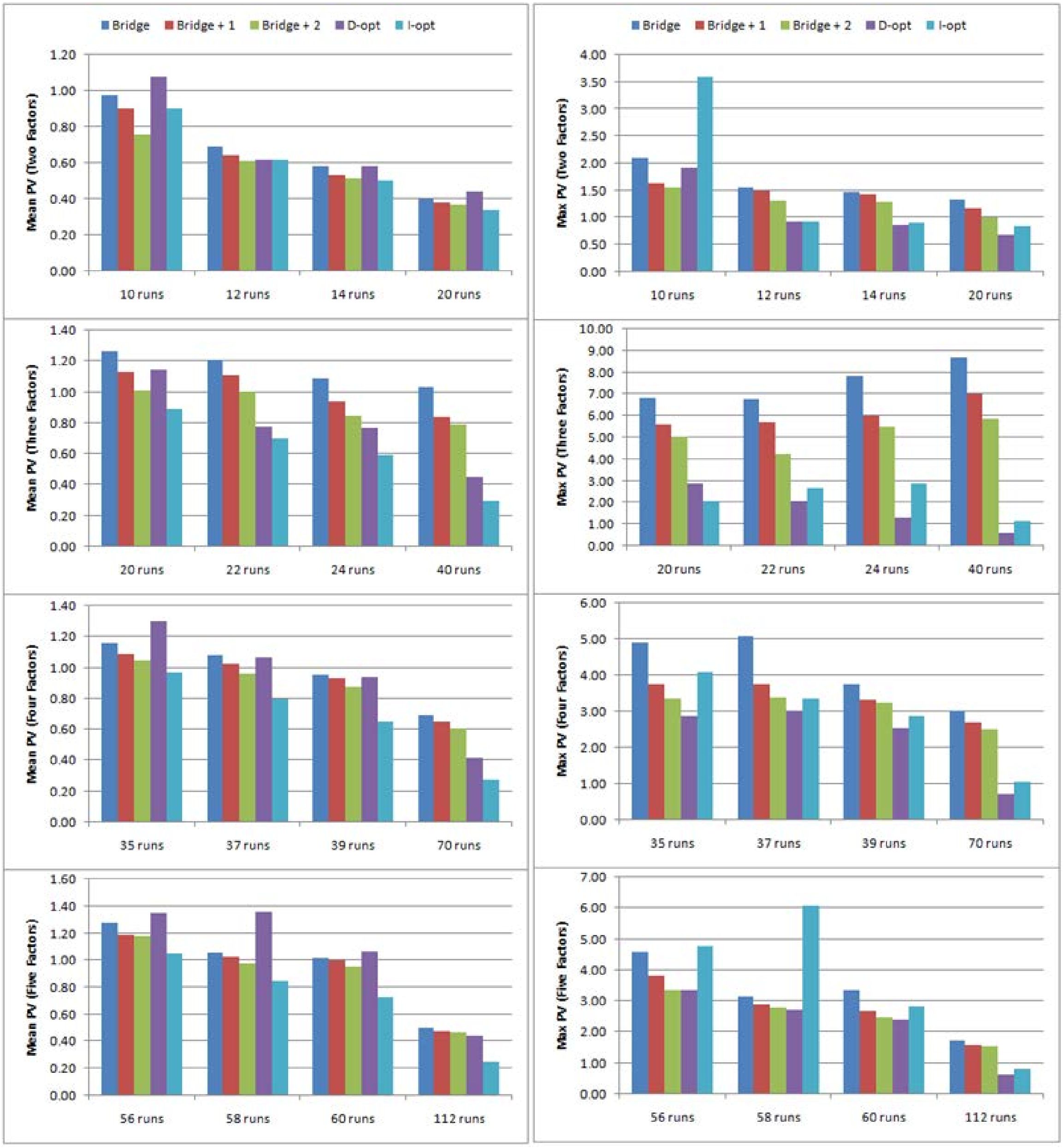

Figure 38. Mean and maximum prediction variance for original and augmented third-order designs.

Fourth-Order Designs

Results for augmenting fourth-order bridge designs with a one and two fourth-order I-optimal points are presented in Table 21. The reduction in mean prediction variance associated with the addition of a single I-optimal point ranged from 3.4% to 16.9%, and the reduction in maximum prediction variance ranged from 4.4% to 37.4%.

The addition of a second point reduces the mean prediction variance of the fourth-order designs by an additional 3.9%-13.5% (for a reduction of 8.7%-30.1% over the base design). The maximum prediction variance of the fourth-order designs reduces by an additional 2.0%-25.0% (11.1%-44.5% overall).

Table 21. Augmentation results for bridge designs generated with underlying fourth-order polynomial models.

		n = p = 15 runs			n = p + 2 = 17 runs			n = p + 4 = 19 runs			n = 2p = 30 runs		
	2 Factors	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	12	17	17	8	2	2	8	17	17	19	4	4
	Point	(0.1, -0.1)	(0.7, 1)	(0.7, 1)	(-0.2, 0.1)	(-1, 0.5)	(-1, 0.5)	(-0.3, -1)	(0.7, 1)	(0.7, 1)	(1, -0.1)	(-0.9, 1)	(-0.9, 1)
	Percentage	8.3%	18.0%	10.4%	8.9%	13.9%	11.0%	7.4%	22.9%	13.2%	8.5%	9.2%	6.5%
2 points	Index	5 & 12	4 & 17	4 & 17	2 & 11	2 & 16	2 & 16	8 & 19	14 & 20	8 & 17	14 & 19	4 & 14	4 & 19
	Point	(-0.9, 1)	(-0.7, 0)	(-0.7, 0)	(-1, 0.5)	(-1, 0.5)	(-1, 0.5)	(-0.3, -1)	(0.5, -1)	(-0.3, -1)	(0.5, -1)	(-0.9, 1)	(-0.9, 1)
		(0.1, -0.1)	(0.7, 1)	(0.7, 1)	(0.1, -0.1)	(0.7, 1)	(0.7, 1)	(1, -0.1)	(1, 0.8)	(0.7, 1)	(1, -0.1)	(0.5, -1)	(1, -0.1)
	Percentage	15.3%	31.3%	18.4%	16.8%	23.3%	18.6%	13.4%	26.3%	17.8%	15.2%	11.1%	11.7%

		n = p = 35 runs			n = p + 2 = 37 runs			n = p + 4 = 39 runs			n = 2p = 70 runs		
	3 Factors	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	24	36	22	4	27	27	37	22	22	2	12	4
	Point	(0.3, -0.7, 1)	(1, -0.3, 0.7)	(0.1, 1, -1)	(-1, 0.3, -1)	(0.6, -1, 0.5)	(0.6, -1, 0.5)	(1, -0.1, -0.8)	(0.1, 1, -1)	(0.1, 1, -1)	(-1, -0.6, 0.6)	(-0.7, 1, 0.5)	(-1, 0.7, 0.9)
	Percentage	6.1%	11.9%	7.3%	6.1%	16.2%	11.0%	7.5%	18.0%	11.4%	13.6%	29.7%	18.1%
2 points	Index	20 & 24	22 & 27	22 & 27	4 & 19	4 & 19	4 & 15	16 & 37	2 & 22	2 & 22	2 & 24	7 & 12	2 & 12
	Point	(0, 0, -0.2)	(0.1, 1, -1)	(0.1, 1, -1)	(-1, 0.3, -1)	(-1, 0.3, -1)	(-1, 0.3, -1)	(-0.3, 0, -0.9)	(-1, -1, 1)	(-1, -1, 1)	(-1, -0.6, 0.6)	(-0.9, 0.2, 0.1)	(-1, -0.6, 0.6)
		(0.3, -0.7, 1)	(0.6, -1, 0.5)	(0.6, -1, 0.5)	(-0.1, 1, 1)	(-0.1, 1, 1)	(-0.1, 1, 1)	(1, -0.1, -0.8)	(0.1, 1, -1)	(0.1, 1, -1)	(0.5, -0.6, -1)	(-0.7, 1, 0.5)	(-0.7, 1, 0.5)
	Percentage	10.5%	18.1%	12.1%	12.4%	41.2%	26.8%	13.3%	35.3%	21.9%	21.6%	34.2%	25.9%

		n = p = 70 runs			n = p + 2 = 72 runs			n = p + 4 = 74 runs			n = 2p = 140 runs		
	4 Factors	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	39	57	39	39	26	26	68	30	30	53	66	31
	Point	(0, -0.1, 0, 0)	(0.8, -0.9, 0.2, -0.5)	(0, -0.1, 0, 0)	(0, -0.1, 0, 0)	(-0.5, -0.7, -1, -0.7)	(-0.5, -0.7, -1, -0.7)	(1, -0.2, -0.1, 0.3)	(-0.3, -1, -1, 1)	(-0.3, -1, -1, 1)	(0.6, 0.3, 0.6, 1)	(1, -0.7, -0.5, -1)	(-0.3, -0.6, 0.6, 1)
	Percentage	7.0%	6.7%	5.6%	7.7%	15.5%	9.4%	5.1%	21.2%	12.2%	5.5%	6.8%	5.5%
2 points	Index	36 & 39	20 & 70	20 & 28	16 & 39	12 & 23	26 & 44	32 & 68	30 & 76	30 & 70	43 & 53	21 & 66	31 & 42
	Point	(-0.1, 0.2, 1, 0.4)	(-0.7, -1, -0.1, -0.1)	(-0.7, -1, -0.1, -0.1)	(-0.5, 0.3, 0.8, -0.5)	(-1, 1, -0.1, 1)	(-0.5, -0.7, -1, -0.7)	(-0.3, 0.6, -0.5, 1)	(-0.3, -1, -1, 1)	(-0.3, -1, -1, 1)	(0.3, -1, 0.5, 0.5)	(-0.7, -1, 1, 0.6)	(-0.3, -0.6, 0.6, 1)
		(0, -0.1, 0, 0)	(1, 0, 0.5, -1)	(-0.5, 1, -0.9, 0.5)	(0, -0.1, 0, 0)	(-0.6, 0.5, -0.6, -1)	(0.3, 0.4, -0.4, 0)	(1, -0.2, -0.1, 0.3)	(1, 1, 0.9, -1)	(1, 0, 0.5, -1)	(0.6, 0.3, 0.6, 1)	(1, -0.7, -0.5, -1)	(0.3, -1, -0.7, -0.5)
	Percentage	11.6%	23.6%	15.3%	11.9%	23.1%	14.8%	9.2%	26.1%	16.2%	10.4%	13.0%	10.5%

		n = p = 126 runs			n = p + 2 = 128 runs			n = p + 4 = 130 runs			n = 2p = 252 runs		
	5 Factors	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)	Mean	Max	Avg(Mean, Max)
1 point	Index	82	110	82	94	56	94	63	48	35	21	52	52
	Point	(0.4, -0.8, -0.6, -1, -0.6)	(1, -1, -1, -1, -1)	(0.4, -0.8, -0.6, -1, -0.6)	(0.6, 1, 0.7, 0.8, 0.5)	(-0.4, 0.8, -0.6, -0.5, -0.7)	(0.6, 1, 0.7, 0.8, 0.5)	(-0.1, -1, 0, 0.3, 0)	(-0.5, -1, -0.2, -0.8, -1)	(-0.8, 0.4, 1, -0.5, 0.4)	(-1, 1, -1, -0.8, -0.4)	(-0.5, 1, -1, -1, 1)	(-0.5, 1, -1, -1, 1)
	Percentage	14.2%	7.8%	9.0%	16.9%	19.7%	17.0%	14.4%	29.1%	19.1%	4.8%	37.4%	20.8%
2 points	Index	70 & 82	55 & 82	55 & 82	63 & 116	56 & 63	68 & 70	29 & 63	43 & 114	43 & 114	21 & 59	16 & 21	16 & 21
	Point	(0.1, 0, 1, 0, -0.4)	(-0.4, 0.6, 0.8, 0.1, -1)	(-0.4, 0.6, 0.8, 0.1, -1)	(-0.1, -1, 0, 0.3, 0)	(-0.4, 0.8, -0.6, -0.5, -0.7)	(0, 0.1, 0.3, -1, 0.1)	(-0.9, 1, 0, 0.2, -0.9)	(-0.6, -0.3, -1, -0.9, -0.9)	(-0.6, -0.3, -1, -0.9, -0.9)	(-1, 1, -1, -0.8, -0.4)	(-1, 0.5, -0.4, -1, 0.8)	(-1, 0.5, -0.4, -1, 0.8)
		(0.4, -0.8, -0.6, -1, -0.6)	(0.4, -0.8, -0.6, -1, -0.6)	(0.4, -0.8, -0.6, -1, -0.6)	(1, -0.7, 0.6, 0, 0)	(-0.1, -1, 0, 0.3, 0)	(0.1, 0, 1, 0, -0.4)	(-0.1, -1, 0, 0.3, 0)	(1, -1, 1, 0.7, -0.9)	(1, -1, 1, 0.7, -0.9)	(-0.2, 0.2, 1, -1, 1)	(-1, 1, -1, -0.8, -0.4)	(-1, 1, -1, -0.8, -0.4)
	Percentage	23.9%	19.8%	21.3%	30.1%	34.2%	30.7%	27.9%	41.1%	30.0%	8.7%	44.5%	26.1%

The prediction variance for the comparative designs in fourth-order is presented in Figure 39. While the bridge designs have comparable mean prediction variance to the optimal designs in many cases (two, three, or four factor designs with sample size less than twice the minimum number of points necessary for fitting the full polynomial model), the maximum prediction variance for each design is much larger in every case. The reduction associated with the addition of the I-optimal points is not large enough to bring the maximum variance of the bridge designs down to comparable levels with the optimal designs.

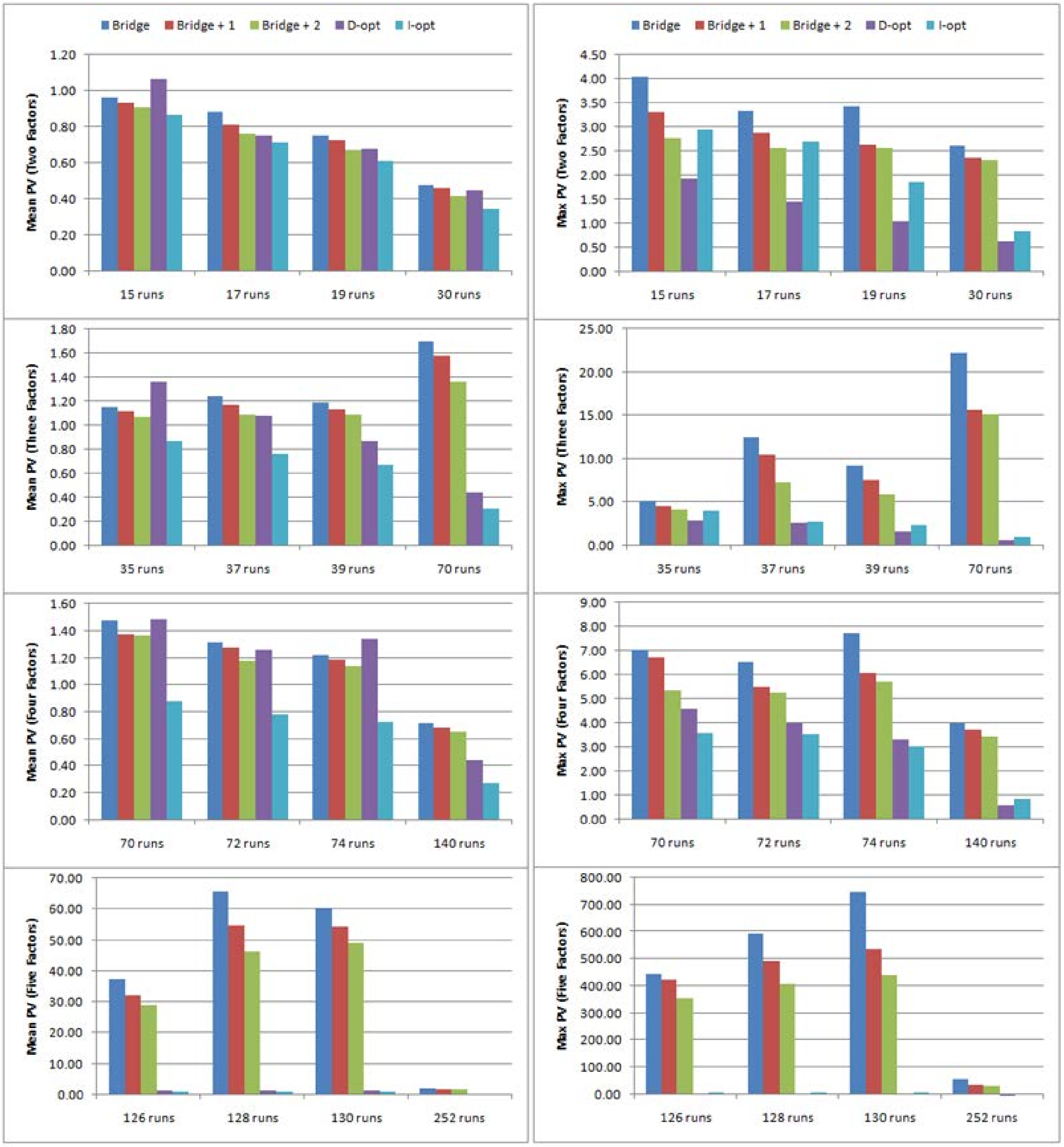

Figure 39. Mean and maximum prediction variance for original and augmented fourth-order designs.

Augmentation With Higher Order Optimal Points

The previous augmentation results have been for cases in which a bridge design generated with a specified order of underlying model is augmented with I-optimal points generated assuming the same underlying model. Given that the intended analysis model must be specified during the design generation phase, at which point the true underlying model is unknown, an additional question arises as to whether augmentation could be useful in mitigating the increased variance that would be associated with model misspecification.

<u>Second-Order Designs Augmented With Third-Order Design Points</u>

A third-order model can be fit to a two-factor design if there are 10 or more design points, so there are two existing two-factor bridge designs that could be tested to assess the effect of augmentation with third-order I-optimal points. Results are presented in Table 22. For the bridge design with 10 original points, the addition of a single third-order I-optimal point reduces the mean prediction variance by 46.1%, or the maximum prediction variance by 37.2%. The results for the 12-point bridge design are even better, with a single I-optimal point reducing the mean prediction variance by 84.4% and the maximum by 90.2%. Adding a second point reduces the mean or the maximum prediction variance by 86% of the baseline for the 10-point design, and 93.3% and 95.2% respectively for the 12-point design. While the gain in terms of percentage points is small for the addition of a third I-optimal point for both designs, the prediction variance is still substantially reduced.

Visualizing the change in prediction variance for the two-factor, second-order bridge design with 10 runs, the prediction variance under a third-order model for the original design is plotted in Figure 40. The resulting prediction variance after the addition of a single point at at (-0.5, 0.5) is presented in Figure 41.

Table 22. Two-factor, second-order bridge designs augmented with one, two, or three third-order I-optimal points.

2 Factors		n = p + 4 = 10 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	7.839	20.818	---
	Index	4	6	4
	Point	(-0.5, 0.5)	(0, 1)	(-0.5, 0.5)
	Augmented PV	4.22	13.19	---
	Percentage Reduction	46.1%	36.7%	37.2%
2 points	Index	3 & 4	4 & 8	4 & 8
	Points	(-0.5, -0.5)	(-0.5, 0.5)	(-0.5, 0.5)
		(-0.5, 0.5)	(0.5, 0.5)	(0.5, 0.5)
	Augmented PV	1.04	2.84	---
	Percentage Reduction	86.7%	86.3%	86.4%
3 points	Index	3 & 4 & 8	4 & 7 & 8	4 & 7 & 8
	Point	(-0.5, -0.5)	(-0.5, 0.5)	(-0.5, 0.5)
		(-0.5, 0.5)	(0.5, -0.5)	(0.5, -0.5)
		(0.5, 0.5)	(0.5, 0.5)	(0.5, 0.5)
	Augmented PV	0.82	2.31	---
	Percentage Reduction	89.6%	88.9%	89.2%

		n = 2p = 12 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	15.184	71.606	---
	Index	6	6	6
	Point	(0, 1)	(0, 1)	(0, 1)
	Augmented PV	2.37	6.99	---
	Percentage Reduction	84.4%	90.2%	87.3%
2 points	Index	3 & 7	6 & 10	7 & 8
	Points	(-0.5, -0.5)	(0, 1)	(0.5, -0.5)
		(0.5, -0.5)	(1, 0)	(0.5, 0.5)
	Augmented PV	1.01	3.45	---
	Percentage Reduction	93.3%	95.2%	94.0%
3 points	Index	3 & 4 & 7	3 & 4 & 6	4 & 6 & 8
	Point	(-0.5, -0.5)	(-0.5, -0.5)	(-0.5, 0.5)
		(-0.5, 0.5)	(-0.5, 0.5)	(0, 1)
		(0.5, -0.5)	(0, 1)	(0.5, 0.5)
	Augmented PV	0.79	2.17	---
	Percentage Reduction	94.8%	97.0%	95.8%

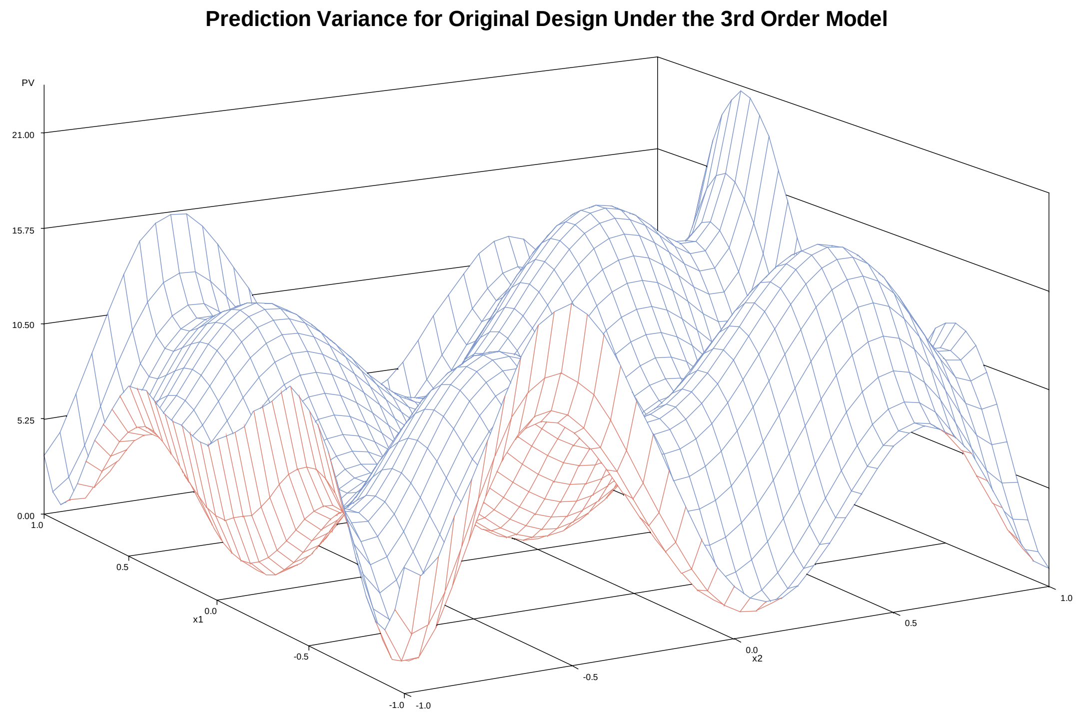

Figure 40. Prediction variance for a two-factor, second-order bridge design under a third-order model.

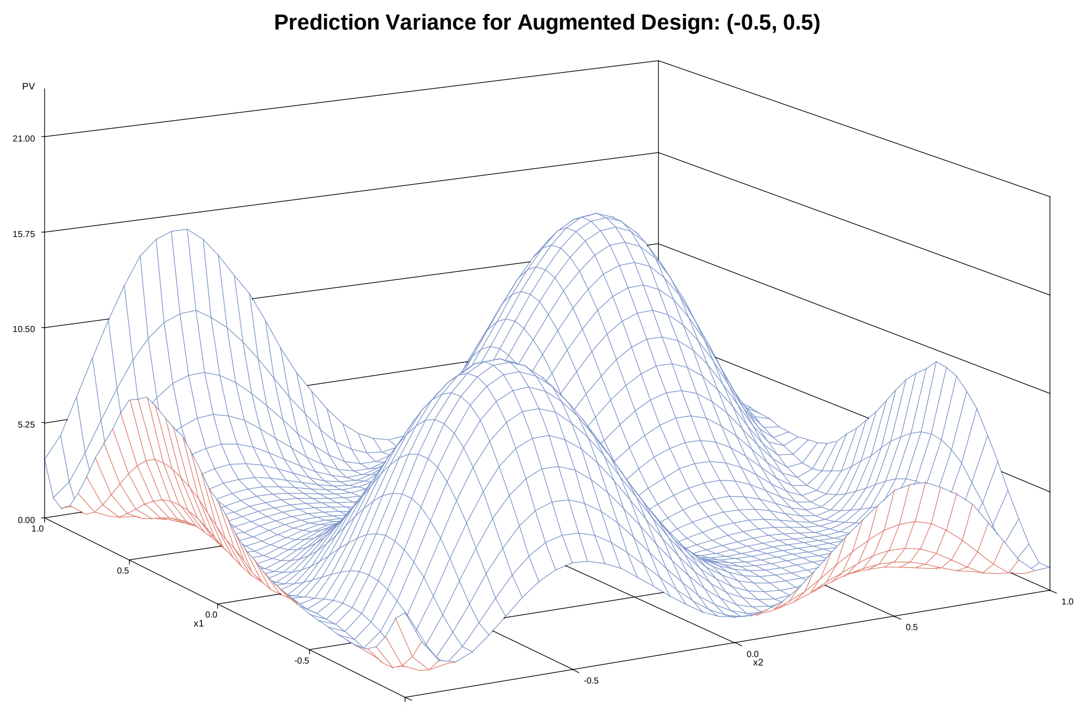

Figure 41. Prediction variance for a two-factor, second-order bridge design under a third-order model, augmented with a single point at (-0.5, 0.5).

A third-order model can be fit to a three-factor design if there are 20 or more design points, so there is one existing three-factor bridge design that can be tested to assess the effect of augmentation with third-order I-optimal points. The baseline mean and maximum prediction variance are quite high, but the addition of a single point reduces the mean prediction variance by 82.4%, or the maximum prediction variance by 88.7% (85.0% averaged for the same point). With the addition of a second point, the prediction variance further reduces, taking the mean prediction variance down by 94.9% and the maximum by 96.7% (95.5% averaged for the same point). As with the two-factor designs, the gains for the addition of the third I-optimal point are much reduced in terms of the percentage from baseline, but the prediction variance itself is 40-50% smaller than that of the design augmented with two points. Results are presented in Table 23.

Table 23. Three-factor, second-order bridge design augmented with one, two, or three third-order I-optimal points.

3 Factors		n = 2p = 20 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	56.565	888.002	---
	Index	11	20	11
	Point	(0, -1, 0.5)	(1, 0, -1)	(0, -1, 0.5)
	Augmented PV	9.93	100.79	---
	Percentage Reduction	82.4%	88.7%	85.0%
2 points	Index	11 & 13	4 & 21	11 & 13
	Point	(0, -1, 0.5)	(-1, 0.5, -1)	(0, -1, 0.5)
		(0, 1, -1)	(1, 0, 0.5)	(0, 1, -1)
	Augmented PV	2.90	29.55	---
	Percentage Reduction	94.9%	96.7%	95.5%
3 points	Index	11 & 13 & 22	9 & 21 & 22	11 & 13 & 22
	Point	(0, -1, 0.5)	(-0.5, 0.5, 0)	(0, -1, 0.5)
		(0, 1, -1)	(1, 0, 0.5)	(0, 1, -1)
		(1, 1, -0.5)	(1, 1, -0.5)	(1, 1, -0.5)
	Augmented PV	1.79	15.14	---
	Percentage Reduction	96.8%	98.3%	97.5%

Fitting a third-order model in four factors would require 35 points, so there are no existing four-factor second-order bridge designs in the catalog that would be sufficient. A new bridge design was generated with 35 runs, assuming an underlying second-order model, and results for its augmentation are presented in Table 24. While the prediction variance reduction is not quite as dramatic as designs with two- and three-factors, the reduction is still quite high.

Table 24. Four-factor, second-order bridge design augmented with one, two, or three third-order I-optimal points.

4 Factors		n = 35 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	123.14	2086.41	---
	Index	29	29	29
	Point	(0.5, 1, 0.5, 1)	(0.5, 1, 0.5, 1)	(0.5, 1, 0.5, 1)
	Augmented PV	47.62	521.46	---
	Percentage Reduction	61.3%	75.0%	68.2%
2 points	Index	8 & 29	29 & 39	8 & 29
	Point	(-1, 0.5, -1, 0)	(0.5, 1, 0.5, 1)	(-1, 0.5, -1, 0)
		(0.5, 1, 0.5, 1)	(1, 1, 1, 0)	(0.5, 1, 0.5, 1)
	Augmented PV	26.74	259.78	---
	Percentage Reduction	78.3%	87.5%	82.9%
3 points	Index	8 & 21 & 29	8 & 21 & 29	8 & 21 & 29
	Point	(-1, 0.5, -1, 0)	(-1, 0.5, -1, 0)	(-1, 0.5, -1, 0)
		(0, 0.5, -0.5, 1)	(0, 0.5, -0.5, 1)	(0, 0.5, -0.5, 1)
		(0.5, 1, 0.5, 1)	(0.5, 1, 0.5, 1)	(0.5, 1, 0.5, 1)
	Augmented PV	17.22	167.15	---
	Percentage Reduction	86.0%	92.0%	89.0%

As with the four-factor second-order case, there are no existing five-factor second-order bridge designs with sufficient runs to fit a third-order model. A new bridge design was generated with the minimum required 56 runs, assuming an underlying second-order model, with augmentation results presented in Table 25. The baseline mean and maximum prediction variance associated with fitting a third-order model are quite high, and the reduction in prediction variance brought about with the augmentation of only a few third-order I-optimal points is excellent.

Table 25. Five-factor, second-order bridge design augmented with one, two, or three third-order I-optimal points.

5 Factors		n = 56 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	1149.52	29800.50	---
	Index	17	2	17
	Point	(-0.5, -1, 0, -0.5, -0.5)	(-1, -1, -1, 1, 1)	(-0.5, -1, 0, -0.5, -0.5)
	Augmented PV	33.29	972.25	---
	Percentage Reduction	97.1%	97.6%	97.3%
2 points	Index	17 & 52	13 & 17	17 & 52
	Point	(-0.5, -1, 0, -0.5, -0.5)	(-1, 0.5, 0.5, -0.5, 0.5)	(-0.5, -1, 0, -0.5, -0.5)
		(1, -0.5, 0.5, 1, 0)	(-0.5, -1, 0, -0.5, -0.5)	(1, -0.5, 0.5, 1, 0)
	Augmented PV	16.09	267.34	---
	Percentage Reduction	98.6%	99.3%	99.0%
3 points	Index	2 & 9 & 22	31	2 & 9 & 22
	Point	(-1, -1, -1, 1, 1)	(-1, -1, -1, 1, 1)	(-1, -1, -1, 1, 1)
		(-1, 0, 1, 1, 1)	(-0.5, 1, 0, -1, 0)	(-1, 0, 1, 1, 1)
		(-0.5, 0, 0, 1, -1)	(0, 0.5, 1, -0.5, -0.5)	(-0.5, 0, 0, 1, -1)
	Augmented PV	10.74	197.70	---
	Percentage Reduction	99.1%	99.5%	99.2%

Third-Order Designs Augmented With Fourth-Order Design Points

Moving from second-order to third-order base designs, the baseline prediction variance of all models tested increased greatly, with the additional terms required for fitting a fourth-order model, particularly for designs with more than two factors. As a result, while the additional percentage reduction seen in adding additional points may seem modest, the reduction in the actual prediction variance can be quite large.

Prediction variance results for augmenting a two-factor, third-order bridge designs with fourth-order I-optimal points are presented in Table 26. The addition of each additional point substantially reduces the prediction variance under the higher order model.

Table 26. Two-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points.

2 Factors		n = 2p = 20 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	3.303	21.819	---
	Index	9	15	9
	Point	(-0.2, 0.1)	(0.7, -0.6)	(-0.2, 0.1)
	Augmented PV	1.31	9.23	---
	Percentage Reduction	60.2%	57.7%	58.3%
2 points	Index	9 & 17	3 & 18	3 & 9
	Points	(-0.2, 0.1)	(-1, 0.5)	(-1, 0.5)
		(0.7, 1)	(1, -0.9)	(-0.2, 0.1)
	Augmented PV	0.94	5.09	---
	Percentage Reduction	71.6%	76.7%	73.6%
3 points	Index	9 & 17 & 19	10 & 17 & 18	9 & 11 & 19
	Point	(-0.2, 0.1)	(-0.1, 0.9)	(-0.2, 0.1)
		(0.7, 1)	(0.7, 1)	(0.1, -0.6)
		(1, -0.1)	(1, -0.9)	(1, -0.1)
	Augmented PV	0.79	3.99	---
	Percentage Reduction	76.1%	81.7%	78.2%

The prediction variance for the original design is plotted in Figure 42, and Figure 43 shows the prediction variance after the addition of a single point at (-0.2, 0.1). In particular, the increased prediction variance at the center of the design is flattened.

Prediction Variance for Original Design Under the Fourth Order Model

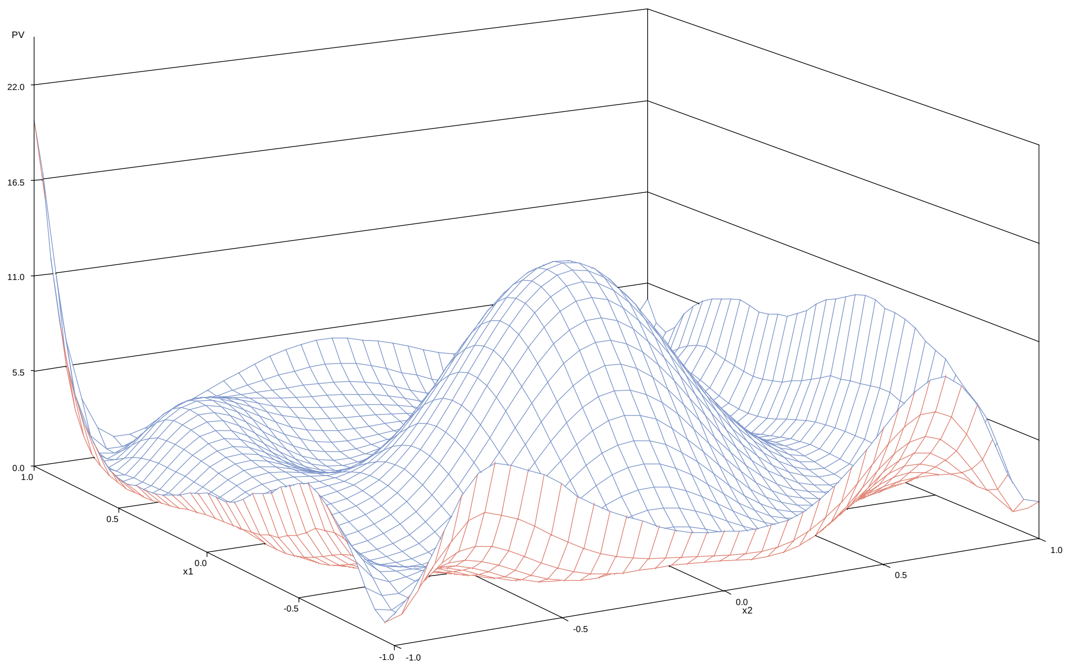

Figure 42. Prediction variance for a two-factor, third-order bridge design under a fourth-order model.

Prediction Variance for Augmented Design: (-0.2, 0.1)

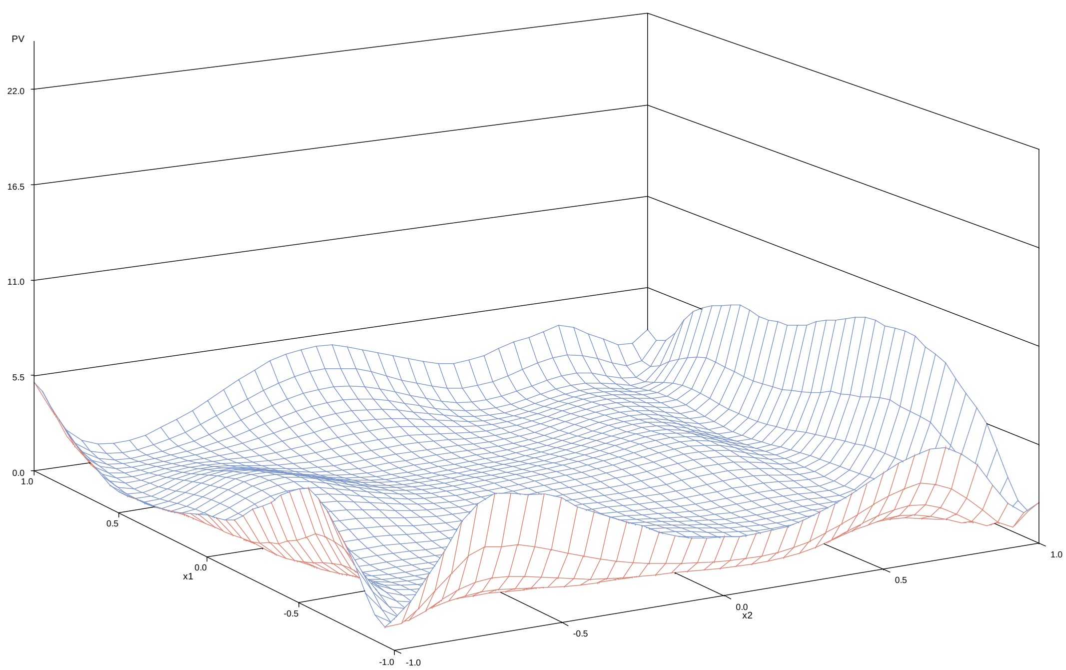

Figure 43. Prediction variance for a two-factor, third-order bridge design under a fourth-order model, augmented with a single point at (-0.2, 0.1).

Prediction variance results for augmenting a three-factor, third-order bridge designs with fourth-order I-optimal points are presented in Table 27. The addition of each additional point substantially reduces the prediction variance under the higher order model, although the maximum prediction variance is still quite high even after the addition of three points.

Table 27. Three-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points.

3 Factors		n = 2p = 40 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	90.428	1815.701	---
	Index	32	36	36
	Point	(0.8, 1, -0.6)	(1, -0.3, 0.7)	(1, -0.3, 0.7)
	Augmented PV	32.01	956.18	---
	Percentage Reduction	64.6%	47.3%	51.1%
2 points	Index	9 & 32	9 & 32	9 & 32
	Point	(-0.7, -1, -1)	(-0.7, -1, -1)	(-0.7, -1, -1)
		(0.8, 1, -0.6)	(0.8, 1, -0.6)	(0.8, 1, -0.6)
	Augmented PV	16.63	231.48	---
	Percentage Reduction	81.6%	87.3%	84.4%
3 points	Index	1 & 8 & 28	17 & 22 & 38	2 & 19 & 40
	Point	(-1, -1, -0.1)	(-0.2, 0.6, 0.7)	(-1, -1, 1)
		(-0.9, 0.2, 0.1)	(0.1, 1, -1)	(-0.1, 1, 1)
		(0.6, 0.6, -0.9)	(1, 0.6, 0.1)	(1, 1, 1)
	Augmented PV	10.66	143.10	---
	Percentage Reduction	88.2%	92.1%	89.5%

For the four and five-factor, third-order designs, the number of candidate augmentation options became unrealistically large for augmenting with three points. In order to streamline the augmentation algorithm, only candidate points that resulted in reductions in the averaged mean and maximum prediction variance above the 70th percentile when added individually were considered. The 70th percentile was chosen because all points that resulted in maximum reductions in the mean, maximum, or averaged mean and maximum in either one or two-point augmentations all fell at or above the third quartile. Sensitivity analysis was done varying the threshold down to the median, but the results were not markedly different.

As with the three-factor designs, the baseline prediction variance of the four and five-factor designs was also quite large. Similarly, while the additional percentage reduction seen in adding additional points may seem modest, the reduction in the actual prediction variance was be quite large. Results for four and five-factor, third-order bridge designs augmented with fourth-order I-optimal points are presented in Tables 28 and 29, respectively.

Table 28. Four-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points.

4 Factors		n = 2p = 70 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	1014.15	22133.96	---
	Index	22	65	22
	Point	(-0.7, -0.4, 1, -1)	(1, -1, 1, 0)	(-0.7, -0.4, 1, -1)
	Augmented PV	250.84	5079.78	---
	Percentage Reduction	75.3%	77.1%	74.9%
2 points	Index	22 & 71	22 & 71	22 & 71
	Point	(-0.7, -0.4, 1, -1)	(-0.7, -0.4, 1, -1)	(-0.7, -0.4, 1, -1)
		(1, 0.4, 1, -0.5)	(1, 0.4, 1, -0.5)	(1, 0.4, 1, -0.5)
	Augmented PV	117.13	1317.71	---
	Percentage Reduction	88.5%	94.0%	91.2%
3 points	Index	2 & 18 & 19	19 & 20 & 23	2 & 19 & 20
	Point	(-1, 0, 1, 1)	(1, -1, -1, -1)	(-1, 0, 1, 1)
		(0.8, -1, 1, -1)	(1, -1, 0.7, 1)	(1, -1, -1, -1)
		(1, -1, -1, -1)	(1, 0.7, -1, 0.4)	(1, -1, 0.7, 1)
	Augmented PV	92.69	1088.33	---
	Percentage Reduction	90.9%	95.1%	92.5%

Table 29. Five-factor, third-order bridge design augmented with one, two, or three fourth-order I-optimal points.

5 Factors		n = 126 runs		
		Mean	Max	Avg(Mean, Max)
1 point	Baseline PV	765.84	16492.72	---
	Index	55	108	61
	Point	(-0.4, 0.6, 0.8, 0.1, -1)	(0.9, 0.4, 1, 1, -1)	(-0.2, 0.7, 0.2, -0.1, 0.8)
	Augmented PV	139.75	2042.96	---
	Percentage Reduction	81.8%	87.6%	84.5%
2 points	Index	32 & 55	32 & 90	32 & 55
	Point	(-0.9, 1, 1, -1, 0.7)	(-0.9, 1, 1, -1, 0.7)	(-0.9, 1, 1, -1, 0.7)
		(-0.4, 0.6, 0.8, 0.1, -1)	(0.6, -0.7, 0.5, 0.6, -1)	(-0.4, 0.6, 0.8, 0.1, -1)
	Augmented PV	36.15	269.65	---
	Percentage Reduction	95.3%	98.4%	96.8%
3 points	Index	8 & 10 & 11	6 & 19 & 27	8 & 10 & 11
	Point	(-0.9, 1, 1, 0.3, 0)	(-0.9, -0.2, -0.3, 1, 1)	(-0.9, 1, 1, 0.3, 0)
		(-0.6, 0.9, 0.6, 0.7, 1)	(0.3, -1, 0.7, -1, 1)	(-0.6, 0.9, 0.6, 0.7, 1)
		(-0.5, -1, -0.7, 0.2, 1)	(0.6, 1, 0.7, 0.8, 0.5)	(-0.5, -1, -0.7, 0.2, 1)
	Augmented PV	31.02	244.56	---
	Percentage Reduction	96.0%	98.5%	97.1%

Discussion

If the total sample size of a design needs to be large enough to fit a higher order model, one might question why a design with a lower order model would be generated and then augmented. In point of fact, it would be more efficient to simply optimize the design for the higher order model, which would de facto include the terms from the lower order model.

The JMP script that generates the bridge design includes options for full polynomial models, but not intermediate models where only certain terms are included. If only certain higher order terms are of interest, it would be useful to use the methods described to generate designs suitable for fitting those specific models. For instance, if a second-order design plus the pure cubic terms in two factors was of interest, the $x_1^2x_2$ and $x_1x_2^2$ interactions would be unnecessary. The second-order bridge design could be augmented with the I-optimal points for the specific model of interest, and the complete design could be implemented in a minimum of eight runs total. The methodology proposed here would require a design of more than 10 runs, but the same principles apply. The augmentation of a bridge design with I-optimal points generated under the specific model of interest can quickly bring down the additional prediction variance associated with additional terms, however, while keeping the sample size and potential replication to a minimum.

One additional parameter that may help to evaluate the impact of adding a point to the base design is the variance of the prediction variance. It is likely that a point that reduces the variance of the prediction variance has an impact in the reduction of the prediction variance across the design space. In cases in which there are different points that result in the greater reduction of the mean prediction variance and the maximum prediction variance, there is likely a disparity in terms of the effect on the variance of the prediction variance as well.

Conclusions

There is little utility found in augmenting a bridge design with I-optimal points with the same underlying model. While the reduction in mean prediction variance could be as much as 44.5% with the addition of a single point, in general the augmented bridge design still displays larger prediction variance than a comparable optimal design. Since bridge designs are particularly useful for early stage experimentation, when it is unclear what type of model will best fit the response surface, it would make more sense to save the additional runs considered for augmentation for a secondary phase of experimentation after the initial models are created.

In the case of protecting against model misspecification, however, the augmentation of bridge designs has great potential. In augmenting bridge designs of lower order with even a few I-optimal points of higher order, the reductions in prediction variance associated with the higher order model are substantial. In fact, the results presented are for the worst case scenario, since the assumed models used for the estimation of the prediction variance are for the full model. In practice, this methodology would be most effective in the case that only a subset of terms was of interest. In that case, the I-optimal model for the specific model would be generated as a candidate set, and the sample size of the overall design could be minimized at the number of terms that would be necessary to fit the model. The hybrid space-filling designs could also be tailored in this fashion, augmenting the space-filling portion with I-optimal points that are generated based on the specific model of interest, however the potential for replication in the case of insignificant terms would be higher than that of the augmented bridge designs.

These results have all been for cases in which the design is being improved prior to any experimental simulations being run. The methodology should also work for cases in which the experiments have been run and initial models generated. If the initial results under polynomial models are promising, additional points could be added to help bring the variance down. If the modeler is confident in the initial results and is only looking for refinement, the I-optimal designs used for candidate sets could be further streamlined by including only terms that appear to be significant. In particular, if a higher order model than originally specified is indicated, the original design could be augmented with optimal points of a higher order.

CHAPTER 6 – CONCLUSIONS AND FUTURE WORK

This work has evaluated different designs that meld traditional optimal designs with the commonly used space-filling designs used in computer experiments. The results show that these composite designs can be quite useful, taking advantages of the positive aspects of each type while helping to mitigate the weaknesses of the other.

Hybrid space-filling designs that are generated as Latin hypercubes augmented with I-optimal points were compared to designs of each contributing component. The results presented give insight into how hybrid space-filling designs perform with respect to prediction variance properties for analysis with either a linear regression model or a Gaussian process model.

The bridge designs further the integration of the disparate design types. Unlike the hybrid designs, they ensure that there is zero replication of factors in any one-dimensional projection, strengthening their relevance for computer experiments with deterministic outcomes. They out-perform pure space-filling designs in terms of prediction variance and alphabetic efficiency, and maintain comparability with pure optimal designs especially for smaller factors and lower order polynomial models.

Coming full circle, the bridge designs were augmented with small numbers of I-optimal design points in order to reduce the prediction variance while introducing a minimum of replication potential. The augmentation of bridge designs with I-optimal points of the same model order was found to be relatively ineffective. In the case of smaller designs (in terms of number of factors and sample size), the prediction variance of the bridge designs was already comparable to that of corresponding optimal designs. In the case of larger designs where prediction variance reduction would be desirable, however, even though the addition of one or two I-optimal points could reduce the mean prediction variance by as much as 44.5% it was not a large enough reduction to approach the performance of the optimal designs.

The concept of augmentation shows great promise for mitigating the issue of increased variance associated with model misspecification, however. Since the generation of the bridge design is dependent on the predicted model intended for analysis of the response, which is unknown prior to any experimentation, there is the potential that the experimenter choses poorly. This work illustrates that adding a few I-optimal points of a higher order than that of the base bridge design can result in a much reduced prediction variance with respect to the higher order model. There is greater flexibility in specifying the intended analysis model for the I-optimal design to be used as a candidate set than there is in the original bridge design. This means that the resulting augmented design could be engineered to give better information on a broader range of polynomial models for the response at a minimized sample size with small potential for replication.

One of the benefits of a computer simulation models is the ability to build up a design sequentially, particularly without concern for blocking or randomization. These composite designs are excellent starting points for experimentation, given that they allow for the credible fitting of either polynomials or other models. Due to the potentially large impact of the design itself, the theoretical prediction capabilities should be evaluated prior to running the experiment. They also provide an immediately intuitive functionality for augmentation after running the initial design, particularly in the case that a polynomial is judged to be appropriate, if additional information is desired to refine the model.

Future Work

In order to preserve comparability between the hybrid space-filling designs, each factor-order combination was only studied for a single sample size. It could be illustrative to evaluate the performance of larger hybrid designs, in cases in which polynomials are suitable or other modeling methods are anticipated.

Bridge designs are generated as D-optimal Latin hypercube designs, but it could be interesting to employ other optimal design criteria. In particular, given that it is the maximum prediction variance which differs most from optimal designs of comparable size and model order, a design that merges a space-filling design with G-optimality to minimize the maximum prediction variance could have interesting properties.

The bridge and comparator designs were evaluated in terms of design efficiency criteria (D, A, G, and average prediction variance as a surrogate for I-efficiency). In most cases the different efficiency values followed similar patterns, but in cases where they vary it could be of use to create a desirability function that may help to evaluate which design could be best given the experimenter's priorities.

A comparison between the properties of the bridge design and the generalized maximin Latin hypercube designs introduced by Dette and Pepelyshev (2010) would be of great interest, since the two designs have similar goals of seeking a compromise between between optimal and space-filling designs.

In evaluating the prediction variance of both hybrid space-filling designs and bridge designs, there were locations noted in the design space in which the prediction variance was especially high. It would be of interest to evaluate whether an augmentation strategy that places design points at the locations with maximum prediction variance would perform well to reduce the prediction variance quickly and hence improve prediction performance.

For the augmentation of bridge designs, it is possible that commercially available software may be able to achieve the same goals, albeit with less flexibility. JMP provides design augmentation functionality, given an anticipated analysis model, allowing for the addition of either D or I-optimal points. This was not previously tested since the software requires a minimum number of additional points to be added, and the original goal of this work had been to keep the potential for replication small.

The algorithms developed for Chapter 5 could also be used to augment bridge designs with points from another bridge design of higher order, rather than an optimal design. In this way, depending on the minimum distance specified between points in the design generation, zero replication would be maintained in the case of an insignificant factor. This is unlikely to have a large impact in the context of the current work, since the number of optimal points added represented a small percentage of the total design, but if true model-order hybrid designs were desired it could be effective.

Finally, all the results presented in this work have been for cases in which the design is being improved prior to any experimental simulations being run. The methodology should also work for cases in which the experiments have been run and initial models generated, but could be tested through sequential experimentation applications.

REFERENCES

Allen, T.T., Bernshteyn, M.A., and Kabiri-Bamoradian, K. (2003). "Constructing Meta-Models for Computer Experiments," *Journal of Quality Technology* **35**(3), pp. 264 – 274.

Ankenman, B., Nelson, B.L., Staum, J. (2010). "Stochastic Kriging for Simulation Metamodeling," *Operations Research*, **58**(2), pp. 371-382.

Ba, S., Joseph, V.R. (2011). "Multi-layer designs for computer experiments," *Journal of the American Statistical Association*, **495**(106), pp. 1139-1149.

Barton, R.R., Meckesheimer, M. (2006). "Chapter 18: Metamodel-Based Simulation Optimization," *Handbook in OR & MS*, **13**, pp. 535-574.

Bingham, D., Sitter, R.R., Tang, B. (2009). "Orthogonal and nearly orthogonal designs for computer experiments," *Biometrika*, **96**(1), pp. 51-65.

Branin, F.H., Jr. (1972). "Widely convergent method for finding multiple solutions of simultaneous nonlinear equations," *IBM Journal of Research and Development*, **16**(5), pp. 504–522.

Bursztyn, D. and Steinberg, D.M. (2006). "Comparison of Designs for Computer Experiments," *Journal of Statistical Planning and Inference* **136**, pp. 1103 – 1119.

Calise, F., Palombo, A., Vanoli, L. (2010). "Maximization of primary energy savings of solar heating and cooling systems by transient simulations and computer design of experiments," *Applied Energy*, **87**, pp. 524-540.

Chen V., Tsui, K-L., Barton, R., and Meckesheimer, M. (2006). "A Review on Design, Modeling and Applications of Computer Experiments," *IEE Transactions* **38**, pp. 273 – 291.

Currin, C., Mitchell, T., Morris, M., Ylvisaker, D. (1991). "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, **86**(416), pp. 953-963.

Dette, H., Pepelyshev, A. (2010). "Generalized Latin hypercube design for computer experiments," *Technometrics*, **52**(4), pp. 421-429.

Fang, K.T., Li, R., Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Boca Raton: Taylor & Francis Group.

Fries, A., Hunter, W.G. (1980). "Minimum aberration 2^{k-p} designs," *Technometrics*, **22**(4), pp. 601-608.

Hussain, M.F., Barton, R.R., and Joshi, S.B. (2002). "Metamodeling: Radial Basis Functions, Versus Polynomials," *European Journal of Operational Research* **138**, pp. 142 – 154.

Iman, R.L., Conover, W.J. (1982). "A distribution-free approach to inducing rank correlation among input variables," *Communications in Statistics, Part B – Simulation and Computation*, **11**, pp. 311-334.

Jin, R., Chen, W., Sudjianto, A. (2005). "An efficient algorithm for constructing optimal design of computer experiments," *Journal of Statistical Planning and Inference*, **134**, pp. 268-287.

Johnson, M.E., Moore, L.M., Ylvisaker, D. (1990). "Minimax and maximin distance designs," *Journal of Statistical Planning and Inference*, **26**, pp. 131-148.

Johnson, R.T., Montgomery, D.C., Jones, B., and Parker, P.A. (2010). "Comparing Computer Experiments Using High Order Polynomial Metamodels," *Journal of Quality Technology*, 42(1), pp. 86-102.

Johnson, R.T., Montgomery, D.C., and Jones, B. (2011). "An Empirical Study of the Prediction Performance of Space-Filling Designs," *International Journal of Experimental Design and Process Optimisation*, **2**(1), pp. 1-18.

Jones, B. and Johnson, R.T. (2009). "The Design and Analysis of the Gaussian Process Model," *Quality and Reliability Engineering International*, **25**, pp. 515-524.

Jones, B., Johnson, R.T., Montgomery, D.C., Steinberg, D. M. (2012). "Bridge Designs for Modeling Systems with Small Error Variance," submitted to *Technometrics*.

Joseph, V.R., Hung, Y. (2008). "Orthogonal-maximin Latin hypercube designs," *Statistica Sinica*, **18**, pp. 171-186.

Joseph, V.R., Hung, Y., Sudjianto, A. (2008). "Blind kriging: a new method for developing metamodels," *Journal of Mechanical Design*, **130**, pp. 031102-1-031102-8.

Kleijnen, J.P.C., van Beers, W.C.M. (2004). "Application-driven sequential designs for simulation experiments: Kriging metamodelling," *Journal of the Operational Research Society*, **55**, pp. 876-883.

Li, W., Lin, D.K.J. (2003). "Optimal foldover plans for two-level fractional factorial designs," *Technometrics*, **45**(2), pp. 142-149.

Loeppky, J.L., Moore, L.M., Williams, B.J. (2010). "Batch sequential designs for computer experiments," *Journal of Statistical Planning and Inference*, **140**, pp. 1452-1464.

Loeppky, J.L., Sacks, J., and Welch, W. (2008). "Choosing the Sample Size of a Computer Experiment: A Practical Guide," Technical Report Number 170, *National Institute of Statistical Sciences*.

McKay, M.D., Beckman, R.J., Conover, W.J. (1979). "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, **21**(2), pp. 239-245.

Montgomery, D.C. (2009). *The Design and Analysis of Experiments*, 7th ed. John Wiley and Sons, New York, NY.

Montgomery, D.C., Peck, E.A., Vining, G. (2012). *Introduction to Linear Regression Analysis*, 5th ed. John Wiley and Sons, New York, NY.

Morris, M.D., Mitchell, T.J., and Ylvisaker, D. (1993). "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," Technometrics **35**, pp. 243 – 255.

Morris, M.D., Mitchell, T.J. (1995). "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, **43**, pp. 381-402.

Myers, R.H., Montgomery, D.C., and Anderson-Cook, C.M. (2009). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 3rd ed*. New York, NY: John Wiley and Sons.

Owen, A.B. (1992). "Orthogonal arrays for computer experiments, integration and visualization," *Statistica Sinica*, **2**, pp. 439-452.

Owen, A.B. (1994). "Controlling correlations in Latin hypercube samples," *Journal of the American Statistical Association*, **89**(428), pp. 1517-1522.

Park, J-S. (1994). "Optimal Latin-hypercube designs for computer experiments," *Journal of Statistical Planning and Inference*, **39**, pp. 95-111.

Ranjan, P., Bingham, D., Michailidis, G. (2008). "Sequential experiment design for contour estimation from complex computer codes," *Technometrics*, **50**(4), pp. 527-541.

Regniere, J., Sharov, A. (1999). "Simulating temperature-dependent ecological processes at the sub-continental scale: male gypsy moth flight phrenology as an example," *International Journal of Biometeorolgy*, **42**, pp. 146-152.

Sacks, J., Schiller, S.B., Welch, W.J. (1989). "Designs for Computer Experiments," *Technometrics*, **31**(1), pp. 41-47.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*, Springer Series in Statistics, Springer – Verlag, New York.

Shewry, M.C., Wynn, H.P. (1987). "Maximum Entropy Sampling," *Journal of Applied Statistics*, **14**, pp. 165-170.

Stein, M. (1987). "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, **29**(2), pp. 143-151.

Steinberg, D.M., Lin, D.K.J. (2006). "A construction method for orthogonal Latin hypercube designs," *Biometrika*, **93**(2), pp. 279-288.

Storlie, C.B. and Helton, J.C. (2008). "Multiple predictor smoothing methods for sensitivity analysis: Example results," *Reliability Engineering & System Safety* **93**, pp. 55 – 77.

Tang, B. (1993). "Orthogonal array-based Latin hypercubes," *Journal of the American Statistical Association*, **88**(424), pp. 1392-1397.

van Beers, W.C.M, Kleijnen, J.P.C. (2003). "Kriging for interpolation in random simulation," *Journal of the Operational Research Society*, **54**, pp. 255-262.

Ventriglia, Francesco (2011). "Effect of filaments within the synaptic cleft on the response of excitatory synapses simulated by computer experiments," *BioSystems*, **104**, pp. 14-22.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992). "Screening, Predicting, and Computer Experiments," *Technometrics* **34**(1), pp. 15 – 25.

Williams, B.J., Santner, T.J., Notz, W.I. (2000). "Sequential design of computer experiments to minimize integrated response functions," *Statistica Sinica*, **10**, pp. 1133-1152.

Ye, K.Q. (1998). "Orthogonal column Latin hypercubes and their application in computer experiments," *Journal of the American Statistical Association*, **93**(444), pp. 1430-1439.

Ye, K.Q., Li, W., Sudjianto, A. (2000). "Algorithmic construction of optimal symmetric Latin hypercube designs," *Journal of Statistical Planning and Inference*, **90**, pp. 145-159.

Zahran, A., Anderson-Cook, C.M., and Myers, R.H. (2003). "Fraction of Design Space to Assess the Prediction Capability of Response Surface Designs," *Journal of Quality Technology* **35**, pp. 377 – 386.