Constructing Nearly Orthogonal Latin Hypercubes for Any Nonsaturated Run-Variable Combination

ALEJANDRO S. HERNANDEZ, THOMAS W. LUCAS, and MATTHEW CARLYLE, United States Naval Postgraduate School

We present a new method for constructing nearly orthogonal Latin hypercubes that greatly expands their availability to experimenters. Latin hypercube designs have proven useful for exploring complex, high-dimensional computational models, but can be plagued with unacceptable correlations among input variables. To improve upon their effectiveness, many researchers have developed algorithms that generate orthogonal and nearly orthogonal Latin hypercubes. Unfortunately, these methodologies can have strict limitations on the feasible number of experimental runs and variables. To overcome these restrictions, we develop a mixed integer programming algorithm that generates Latin hypercubes with little or no correlation among their columns for most any determinate run-variable combination—including fully saturated designs. Moreover, many designs can be constructed for a specified number of runs and factors—thereby providing experimenters with a choice of several designs. In addition, our algorithm can be used to quickly adapt to changing experimental conditions by augmenting existing designs by adding new variables or generating new designs to accommodate a change in runs.

1. INTRODUCTION

Experiments using computer models are increasingly essential in scientific research, national defense, and public policy debates. Continual improvements in computational power against a backdrop of rising costs and other challenges often associated with physical experimentation make computer experimentation an increasingly attractive option. In situations with a dearth of real-world experimental data, computer simulations are often used to help understand complex issues. Indeed, in many important areas it is not practical to conduct many, or even any, physical experiments;

for instance, the long-term effects of various policies on global climate, emergency response to large-scale nuclear accidents, potential major military conflicts, etc.

Computer models used in these areas may contain many thousands of input variables and can take many days to run [Kleijnen et al. 2005]. Researchers have several techniques to effectively extract information from these simulations. Among them are experimental designs that are specifically developed for efficiently exploring high-dimensional computer models. For example, Sanchez and Sanchez [2005] use Walsh functions to quickly generate very large (hundreds of factors), two-level, resolution V fractional factorial designs and central composite designs. Of course, the information obtainable by analyzing the data from an experiment depends critically on the design; for instance, a quadratic response is not identifiable for a quantitative input variable from a two-level design.

Often, especially in exploratory analysis, we desire designs that can accommodate fitting a wide variety of metamodels [Santner et al. 2003]. For such situations, Latin hypercube (LH) sampling [McKay et al. 1979] has proven to be an invaluable technique. In fact, LHs are the predominant design for experiments involving computer simulations [Buyske and Trout 2001]. A key reason for this is that they are easily obtainable (e.g., LHs are available in many simulation software packages) and come with minimal restrictions on the number of factors and sampling budget. In addition, the resultant output data allow analysts to fit many different diverse models to multiple outputs from a single experimental set. In practice, LHs are often used to simultaneously screen many factors for significance and to fit complex metamodels to a handful of dominant variables. This flexibility also extends to visual investigations of the data [Sanchez et al. 2012], as we get many viewpoints from which to observe the relationships between inputs and outputs.

LH designs may come with unwanted properties; chiefly, unacceptable correlations among the columns of the design matrix that may hinder many statistical procedures, such as regression and regression trees [Montgomery 2005; Kim and Loh 2003]. To mitigate this problem, many researchers [Iman and Conover 1982; Florian 1992; Owen 1994; Ye 1998; Steinberg and Lin 2006; Cioppa and Lucas 2007; Joseph and Hung 2008; Pang et al. 2009; Moon et al. 2011] have developed algorithms to reduce or eliminate correlations among input variables in LHs. An LH with zero correlation between any of its input variables is an orthogonal Latin hypercube (OLH). In this article, a nearly orthogonal Latin hypercube (NOLH) is defined as an LH with a maximum absolute pairwise correlation no greater than 0.05 between any two input variables [Hernandez 2008]. Although this criterion is somewhat arbitrary, designs meeting it suffer minimal adverse multicollinearity effects.

As desirable as OLH and NOLH designs are for computer experimentation, their application is limited because of the scarcity of available design dimensions (i.e., combinations of the number of runs n and factors k) due to restrictions that many algorithms impose when creating them. A common restriction among existing algorithms is a strong reliance on design dimensions that are functions of powers of 2. Unfortunately, in practice, experimental conditions seldom fall precisely into the currently available dimensions. Thus, analysts must modify experimental conditions to fit the framework of catalogued OLHs or NOLHs, or use designs that possess greater correlation among input variables and deal with the consequences. The limited number of readily available NOLHs motivates our study.

In this article, we present a new method for constructing NOLH designs that breaks through the design dimensional barriers that afflict previous efforts. Our discussion in Section 2 defines our terminology, details the progression of OLH and NOLH efforts, and designates the measures by which we differentiate LHs of the same class (i.e., identical n and k). Section 3 contains our mixed integer program optimization

formulation to address the design problem to create NOLHs for most any situation in which $n > k$. Section 4 demonstrates the power and flexibility of our new method, while the last section summarizes our findings.

2. BACKGROUND

This section describes the experimental setting, defines key terms, introduces the measures by which we compare designs, and summarizes previous algorithms that reduce or eliminate correlations among input variables. We also outline the construction of lattice LHs to provide the basis for simplifying the mathematical formulation of the design problem and for discussing our approach.

2.1. Experimental Setting

We consider experimental settings involving a computer simulation in which users can precisely choose the input values and outputs can be exactly determined. The *design matrix* is the complete specification of input settings for each variable over a set of runs. For this discussion, the model contains k continuous variables that we wish to vary in n computational experiments over a k dimensional hyperrectangle. We denote the $n \times k$ design matrix as $\boldsymbol{X}$, where row i of $\boldsymbol{X}$ corresponds to the ith experiment, and column j corresponds to the jth variable. Thus, X_i^j is the value the experimenter sets for factor j in run i. We further denote the jth column of X as X^j and the ith row as X_i. Finally, let z_i be an outcome generated by the ith experiment.

Analysts often construct metamodels to quantify the relationship between the input variables $(X^1, X^2, \ldots, X^k)$ and the resultant outputs $(z_1, z_2, \ldots, z_n)$ from the computational experiments. A metamodel is a "model of a model" that approximates the simulation. A good metamodel is one that makes parsimonious use of the variables, is simple to understand, and whose outputs closely match those of the simulation. A variety of statistical methods are used to build metamodels. Barton [1998] lists parametric polynomial response surface approximation, splines, neural networks, spatial correlation models (like Kriging), and frequency domain methods among the approaches used. He notes that using linear regression to build an approximate response surface is the most common technique used. A design that facilitates many possible metamodels is preferred.

Metamodels provide analysts with insight into their complicated simulations by identifying the most important factors (usually a small proportion of the total possible) and interesting features, such as nonlinear responses, interactions, and change-points. While predictive accuracy is an important consideration in many design contexts, it is less critical in our situation, as the simulation model can usually be run at any desired setting.

2.2. Latin Hypercube Designs

McKay et al. [1979] first proposed LH sampling and described it as follows: for each input variable X^j, "all portions of its distribution [are] represented by input values" (p. 56) by dividing its range into "n strata of equal marginal probability $1/n$, and [sampling] once from each stratum" (p. 56). Following Owen's [1994] notation, the ith element in the jth column, X_i^j, is determined by $X_i^j = F_j^{-1}(\frac{(\pi_j(i) - U_{ij})}{n})$, for $i = 1, \ldots, n$ and $j = 1, \ldots k$, where $\pi_j(1), \ldots, \pi_j(n)$ is one of the $n!$ possible random permutations of $1, \ldots, n$ in which all $n!$ permutations are equally likely. F_j, for $j = 1, \ldots k$, are continuous and invertible distribution functions. U_{ij}, for $i = 1, \ldots, n$ and $j = 1, \ldots k$ are independent and identically distributed uniform [0, 1] random variables. Many analysts choose F_j to be a uniform distribution and take a fixed value in each stratum (e.g., the median); see Koehler and Owen [1996]. In this situation, the design points all fall on a lattice in

k-space [Patterson 1954]. In such a case, creating an LH corresponds to independently generating k permutations of the first n natural numbers and appropriately scaling the columns to cover the variables' ranges. It is this lattice variant that we use in this article. The projection of a lattice LH into any one-dimensional factor's subspace results in n equally spaced points—which maximizes the minimum distance between any two points in that dimension.

2.3. Measures for Assessing Designs

For any given n and k, there exist $(n!)^k$ lattice LHs, of which $(n!)^{k-1}$ are distinct. If experimenters rely solely on chance, they may get an LH that has high correlations among the input variables and/or poor space-filling properties. In a linear regression model an orthogonal design is desirable since it gives uncorrelated estimates of the coefficients, avoids partial confounding, and enhances the performance of many other procedures, such as classification and regression tree models [Kim and Loh 2003]. A good *space-filling* design is one in which the design points are scattered throughout the experimental region. Such a design is less likely to miss an interesting localized effect.

Correlation is the most commonly used criterion for assessing LHs [Owen 1994; Tang 1998], and it is our primary measure. Computation of the correlation coefficient between any two vector columns, X^i and X^j, in the design matrix, X, is straightforward:

$$\rho_{ij} = \frac{\sum_{b=1}^{n}[(X_b^i - \bar{X}^i)(X_b^j - \bar{X}^j)]}{\sqrt{\sum_{b=1}^{n}(X_b^i - \bar{X}^i)^2 \sum_{b=1}^{n}(X_b^j - \bar{X}^j)^2}}, \tag{1}$$

where $\bar{X}^i$ and $\bar{X}^j$ are the mean values of the ith and j^{th} columns in X.

In an $n \times k$ design matrix, there are $\binom{k}{2}$ pairwise correlations, of which the largest in magnitude defines the degree of nonorthogonality in the design. We express the maximum absolute pairwise (*map*) correlation as

$$\rho_{map} = \max\{|\rho_{ij}|, \forall (i \neq j)\} \tag{2}$$

and concentrate on minimizing it to find OLH and NOLH designs. By minimizing ρ_{map}, we bound the worst-case pairwise correlation. For a design involving many factors, a low average absolute correlation does not guarantee that all pairwise correlations are small.

While we focus on minimizing correlation, this does not guarantee a good LH. For example, points that are uniformly distributed over an X (the main diagonal and the reverse diagonal) in the unit square are orthogonal; however, such a design has poor space-filling properties and does not allow the fitting of interactions in addition to the linear terms. Therefore, in addition to correlation, we assess LHs based on space-filling properties. Our general approach, as illustrated in Section 4, is to generate many NOLHs and choose for experimentation the one with the best space-filling properties (or some other criteria, such as its ability to fit higher-order terms).

The design points will ideally be distributed as uniformly as possible throughout the k dimensional design space. Our choice of space-filling measure is the computationally efficient modified L_2 discrepancy. Fang et al. [2000a, p. 238] state that *the L_∞ discrepancy*, equivalent to the Kolmogorov-Smirnov statistic, "is probably the most commonly used measurement for discrepancy…and has been universally accepted in quasi-Monte-Carlo methods and number theoretic methods." Unfortunately, as they note, "one disadvantage of [this] measure is that it is expensive to compute." When the L_∞ discrepancy is too computationally onerous, as for many of the designs in this article, the *modified L_2 discrepancy* (with the designs normalized to [0,1] in each dimension), is often used instead (Hickernell 1998; Fang et al. 2000b). Thus, this article

uses ML_2 discrepancy (see Equation (3)) to measure the space-filling of a design, with smaller values preferred over large ones.

$$ML_2 = \left(\frac{4}{3}\right)^k - \frac{2^{1-k}}{n} \sum_{d=1}^{n} \prod_{i=1}^{k}(3 - x_{di}^2) + \frac{1}{n^2} \sum_{d=1}^{n} \sum_{j=1}^{n} \prod_{i=1}^{k}[2 - \max(x_{di}, x_{ji})]. \qquad (3)$$

If a design has a low ML_2, then all of the projections of the design onto subsets of the k variables will also have good uniformity. While LHs have proven valuable in obtaining insight from high-dimensional models, when higher dimensional uniformity is required; for instance, in minimizing integration error or variance by the point set, Quasi-Monte Carlo methods [L'Ecuyer 2009] and uniformity techniques [Fang et al. 2000a] may be appropriate.

2.4. Previous Approaches to Creating OLHs and NOLHs

Efforts to reduce correlations among the columns of an existing LH have been numerous. They include Iman and Conover [1982], Florian [1992], and Owen [1994]. While these methods often suffice, especially when n is large relative to k, they do not guarantee an OLH or even an NOLH. For smaller values of n, their performance degrades when k is close to n [Hernandez 2008].

A recent series of algorithms generate LHs with little or no correlation among the columns in the design matrix. Ye [1998] developed a procedure to create OLHs when the number of runs, n, is a power of 2 plus 1. Specifically, Ye's use of Kroenecker products enables construction of an OLH with $2m - 2$ columns for an experiment with $2^m + 1$ runs, where m is a positive integer.

Cioppa and Lucas [2007] extend Ye's method for constructing OLH designs to $m + \binom{m-1}{2}$ factors in $2^m + 1$ computational experiments. However, they note that these designs often have poor space-filling properties, subsequently developing an approach that improves the space-filling properties by allowing an acceptable level of nonorthogonality. The result is a family of NOLH designs for various combinations of k input variables in n experiments, which are labeled N_k^n. We denote an OLH for k factors in n runs as O_k^n. Cioppa [2002] selectively catalogued NOLHs with good space-filling properties for up to 29 factors within 257 runs. Ang [2006] extended Cioppa's methodologies to create OLHs and NOLHs for $n = 2^m + 1$ and $k = 1 + \sum_{j=1}^{w} \binom{m-1}{j}$, where $w \leq m - 1$ and m is integer > 0.

Steinberg and Lin [2006] rotate two-level factorial designs to construct OLHs for $n = 2^h$, with h a power of 2, and the maximum number of factors being $h \times B_h$, where $B_h = \lfloor \frac{n-1}{h} \rfloor$, with $\lfloor x \rfloor$ the largest integer $\leq x$. For instance, for $n = 16$ runs, $h = 4$ and $B_h = 3$, an O_{12}^{16} design is possible. Pang et al. [2009] propose a general construction method for OLH in which Steinberg and Lin's [2006] approach is a special case ($p = 2$). Pang et al. [2009] show that an OLH may be constructed for $n = p^d$, where p is a prime number and d is a power of 2. Additionally, the number of factors that may be addressed is also a function of d.

The success that these recent efforts have had in creating OLH and NOLH designs is substantial, but still subject to stringent constraints in their dimensionality. Steinberg and Lin [2006] recognize that "the primary limitation to our method is the severe sample size constraint" (p. 287). Likewise, Pang et al. [2009] acknowledge that "the primary limitation to the [our] method is the sample size constraint" (p. 1726). Our work overcomes these constraints.

To illustrate the constraints of these algorithms, we select values for n up to 65 such that OLHs or NOLHs are, or are not, available. Table I reveals numerous gaps in n

N	Maximum k for Each Method				
	Ye [1998]	Cioppa [2002]	Ang [2006]	Steinberg and Lin [2006]	Pang et al. [2009]
9	4	**UNAVAILABLE**			4
10–15	**UNAVAILABLE**				
16	**UNAVAILABLE**			12	12
17	6	7	8	**UNAVAILABLE**	
18–24	**UNAVAILABLE**				
25	**UNAVAILABLE**				6
26–32	**UNAVAILABLE**				
33	8	11	16	**UNAVAILABLE**	
34–48	**UNAVAILABLE**				
49	**UNAVAILABLE**				8
50–64	**UNAVAILABLE**				
65	10	16	32	**UNAVAILABLE**	

and k combinations from these techniques, underscoring the sparseness of the family of existing OLH and NOLH designs.

Sample sizes for the NOLHs developed in this article need not be related to a power of 2, and are feasible for many k (typically all $k < n$). In many cases, our algorithm creates saturated NOLHs, which to date, no other method achieves. A significant advantage of our technique is that it can adapt to major changes in the experiment. Since our designs can be created for most any situation in which $k < n$, our method facilitates the addition or subtraction of design points. In addition, its application to existing OLHs or NOLHs to extend the number of factors often preserves much of the nearly orthogonal properties of the original design, while maintaining the same sample size.

3. A MIXED INTEGER PROGRAMMING APPROACH TO MINIMIZE ρ_{MAP}

Our design approach is based on a mixed integer program (MIP) that optimally selects input levels for a given column in the design to minimize the maximum correlation between that column and all other columns. An optimization routine to create an LH presents a number of challenges; the objective function is nonlinear in the entries of the design matrix and not differentiable everywhere. In addition, all decision variables are natural numbers. Moreover, each column of X must be a permutation of the first n natural numbers $1, \ldots, n$. Thus, the lattice construct for a given n and k equates to a total of $(n!)^k$ possible LHs, making an exhaustive search to find an optimal LH daunting, and as n and k increase—impossible. Fang et al. [2000b] identify a similar issue in their own work and note that such problems are most likely nondeterministic, polynomial-time hard [Bazaraa et al. 2004], additionally confirming their intractability for large n and k.

To transform the overall problem into a solvable linear program, we focus on one column of the design matrix in formulating the mathematical model. This is broadly similar to the nonlinear optimization approach of Stinstra et al. [2003], who find maximin designs by using a constrained sequential heuristic to iterate toward a solution.

The single column problem is guaranteed an optimal solution. In most cases, applying the program iteratively to update one selected design column at a time, until no further improvement in the overall design matrix is possible, produces NOLHs. This threshold-accepting process is a local search [Aarts and Lenstra 1997] within the neighborhood of the most updated design matrix. The ideal optimal solution for the overall problem is $\rho_{map} = 0$, signaling a completely orthogonal design. However, we

typically stop iterations when ρ_{map} meets the threshold of 0.05, since the resultant designs are minimally affected by multicollinearity.

$$v_{map} \equiv \max_{i \neq j} \left| \sum_{b=1}^{n} (X_b^i - \overline{X}^i (X_b^j - \overline{X}^j) \right|. \tag{4}$$

The following paragraphs detail the steps to formulate the mathematical model for minimizing ρ_{map}. Much as Owen [1998] points at the utility of the LH lattice for integration, we exploit the lattice construct of design matrix X to simplify the problem of minimizing ρ_{map} into an equivalent problem for minimizing:

This follows since $\sum_{b=1}^{n} (X_b^i - \bar{X}^i)^2$ from Equation (1) is a constant, as is the average of any column that has the first n natural numbers as its values (i.e., $\bar{X}^i = n(n+1)/2$ for all i), regardless of order. We use $\bar{X} = n(n+1)/2$ as the mean value for each column in the formulations below. We establish Equation (4) as the objective function for the optimization problem. In the following formulations we equate the objective function (i.e., v_{map} in Equation (4)) to the variable V. We note that the nonlinearity of the objective function only results from product terms involving X_b^i and X_b^j, and the absolute value expression. Using standard math programming notation [Bertsimas and Tsitsiklis 1997; Bazaraa et al. 2004], we express our initial nonlinear formulation of this optimization problem in Model A:

Model A (n, k):
INDICES
i runs (alias j) $i = 1, \ldots, n$
l factors (alias h) $l = 1, \ldots, k$
DECISION VARIABLES
X_i^l level of lth factor on the ith run
V variable representing the value v_{map}
OBJECTIVE and CONSTRAINTS

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

$$\text{s.t.} \quad V \geq \sum_{i=1}^{n} (X_i^l - \bar{X})(X_i^h - \bar{X}) \quad \forall l \neq h \tag{A1}$$

$$V \geq - \sum_{i=1}^{n} (X_i^l - \bar{X})(X_i^h - \bar{X}) \quad \forall l \neq h \tag{A2}$$

$$X_i^l \neq X_j^l \quad \forall i \neq j, l \tag{A3}$$

$$1 \leq X_i^l \leq n, \text{integer} \quad \forall i, l \tag{A4}$$

The objective function A0 is simply V; the value v_{map}. Because the absolute value is a nonlinear expression, constraints A1 and A2 separate it into expressions that we can transform into linear terms. These actions are consistent when dealing with optimization problems that have nonlinear objective functions [Rardin 1998; Hillier and Lieberman 2005]. Constraints A3 and A4 require that each column of X be a permutation of the integers from 1 to n.

Unfortunately, our desired formulation cannot be directly solved due to the nonlinear (A1 and A2) and nonconvex (A3) constraints. To address the nonconvex nature of the feasible region [Bazaraa et al. 2004], we reformulate A3 by introducing binary variables [Wolsey 1998]. We represent one run (design point) i of factor m, with a set of n binary

variables, one for each possible level j of the factor on that run: $Y_{i,j}^m$, $j = 1, 2, \ldots, n$. For instance, a solution, $Y_{3,7}^4 = 1$, means that for the 4th factor, the value level is set at 7 in the 3rd run of the experiment.

Finally, focusing on one factor (column $= m$) of the design matrix to optimize, we treat the elements in all other columns (l) of the design as "fixed" values, denoted $\hat{X}_i^l$, $\forall l \neq m$ and $i = 1, \ldots, n$. These constant values eliminate the variable product terms in A1 and A2. Thus, for a given factor, m, we find level settings that minimize v_{map} between column m and every other column in the design. In this instance, the resulting levels for factor m guarantees the lowest value for V and is optimal. Our model is now a single-column MIP for a specified column, m, in a design with n runs and k factors. After collecting terms, our final model equates to the following formulation, which we refer to as *VMIN* for the remainder of the article.

VMIN (m, k, n)

INDICES

i runs (alias j) $i = 1, \ldots, n$

l factors $l = 1, \ldots, k$

DATA (DESIGN MATRIX)

$\hat{X}_i^l$ level value of factor l for run i in the given design $\boldsymbol{X}$

DECISION VARIABLES

$Y_{i,j}^m$ equals 1 (0 otherwise) if factor m is set to level j in run i

V maximum absolute value of v_{map} for any two columns in $\boldsymbol{X}$

OBJECTIVE and CONSTRAINTS

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

$$\text{s.t.} \quad V \geq \sum_{i=1}^n \left(X_i^l - \bar{X}\right) \sum_{j=1}^n j Y_{i,j}^m \qquad \forall l \neq m \tag{V1}$$

$$V \geq -\sum_{i=1}^n \left(X_i^l - \bar{X}\right) \sum_{j=1}^n j Y_{i,j}^m \quad \forall l \neq m \tag{V2}$$

$$\sum_{j=1}^n Y_{i,j}^m = 1 \qquad \forall i \tag{V3}$$

$$\sum_{j=1}^n Y_{i,j}^m = 1 \qquad \forall j \tag{V4}$$

$$Y_{i,j}^m \in \{0, 1\} \qquad \forall i, j \tag{V5}$$

We code this formulation in the General Algebraic Modeling System (GAMS), using the Cplex solver on a standard laptop computer with 2.0 gigabytes of RAM. This optimization code is available at http:/harvest.nps.edu, under section "NOLH design GAMS code." [Sanchez 2012]. GAMS is a specialized programming language for optimization problems. We instruct GAMS to use the Cplex solver [IBM 2012], a very powerful integer programming routine that solves our problems quickly and, in almost every case, to a provably optimal solution using a combination of a proprietary IBM heuristic that quickly finds feasible solutions and automatic generation of cutting planes at the root node, followed by standard branch and bound applied to the resulting problem and any incumbent solution. The initial solution to this single column optimization problem is the maximum absolute pairwise correlation of the original design matrix.

Even as a one-column MIP, VMIN can rapidly become large as n and k increase. The model contains $2(n^2 + n + k - 1)$ constraint equations and n^2 binary variables. A relatively

$m=6$	j=1	j=2	j=3	j=4	j=5	j=6	j=7	j=8	j=9	j=10	j=11	j=12	j=13	j=14	j=15	j=16	j=17		Column 6
i=1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0		9
i=2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		4
i=3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		16
i=4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		13
i=5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0		15
i=6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		6
i=7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0		14
i=8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0		7
i=9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0		8
i=10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		11
i=11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1
i=12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		2
i=13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		12
i=14	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		10
i=15	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		3
i=16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		5
i=17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		17

Fig. 1. A design with $n = 17$, $k = 7$ has a total of 289 binary variables for any factor in an iteration of VMIN. For factor (or column) $m = 6$, VMIN optimizes the order of the elements 1 through 17 to minimize $\mathbf{V}$. Cells with an entry of 1 specify the value level (j) for the specified run (i). We translate the actual value level for each run from the matrix and present it on the right.

small design, $n = 33$, involves 2,262 constraint equations and 1,089 binary variables for just one factor. However, for a problem of this size, our formulation provides a solution for any column, typically within minutes.

The two-dimensional solution for this mathematical problem is best seen in matrix form. Figure 1 shows the solution for 289 binary variables when $m = 6$ (i.e., we are solving for X^6) and $n = 17$. Each column equates to a value level (j) that may be set for the ith run (row). VMIN populates each cell of this 17×17 matrix with a 0 or 1 to minimize V. Thus, the matrix contains the solutions for the variables $Y_{i,j}^{m=6}$, $i = 1, 2, \ldots, 17$ and $j = 1, 2, \ldots, 17$. Entries of 1 equate to the complete specification for the order of elements in the 6th factor (column) of the design matrix, shown on the right-hand side of the figure.

Any LH can be used to initialize the procedure. However, a good initial solution (design) helps the optimization routine solve the problem more quickly, and begin in a neighborhood that contains an acceptable v_{map}. Fang et al. [2000b] start with an arbitrary choice of U-type design to initiate their optimization routine. Our starting solution for VMIN results from quickly generating many (up to 1,000) random Latin hypercubes (RLHs), and selecting the one with minimum ρ_{map} [Hernandez 2008].

An iterative application of VMIN constructs the entire NOLH. After determining the best value levels for one column, a new column is selected, continuing until no further improvement is possible or the correlation threshold is met. This local search algorithm seeks a maximum improvement for one column at each step [Aarts and Lenstra 1997]. The problem that emerges is choosing the direction of the next step (i.e., selection of the next column to optimize).

Fang et al. [2000b] use a local search that focuses on a complete submatrix of the design, which they eliminate, reconstruct, and test for "all possible permutations of the deleted elements," (p. 286) finally selecting the set with least discrepancy relative to the rest of the design matrix. Consequently, a method for specifying the next submatrix is not a consideration. However, a large neighborhood quickly makes this approach computationally unaffordable.

In our method, we preclude the need to search all possible permutations in the neighborhood of the current design by specifying the direction of the search. We use the correlation mean square (ms) to select the next column to optimize. In other studies,

Table II. Applying VMIN to the Best of 1,000 RLHs Created This N_{12}^{16}.

Final NOLH After VMIN Application: $\rho_{map} = 0.029$												
	k1	**k2**	**k3**	**k4**	**k5**	**k6**	**k7**	**k8**	**k9**	**k10**	**k11**	**k12**
n1	13	11	5	5	9	16	13	4	16	14	15	5
n2	10	1	7	14	5	13	15	12	8	6	9	14
n3	15	3	9	11	1	4	1	3	14	12	2	9
n4	11	4	13	10	12	12	16	9	5	8	1	4
n5	1	6	8	4	3	6	11	11	10	5	8	11
n6	14	14	4	7	7	14	6	16	11	3	6	12
n7	12	9	11	6	15	1	9	6	13	2	13	16
n8	8	7	10	3	10	8	10	1	6	1	7	1
n9	7	2	16	2	8	10	2	15	7	13	16	7
n10	2	8	3	9	14	2	12	13	15	15	5	6
n11	3	5	1	15	11	11	4	2	4	7	14	10
n12	16	10	6	13	13	3	7	14	2	10	11	3
n13	9	15	12	8	4	5	14	5	1	16	10	15
n14	6	12	2	1	6	9	5	8	3	9	4	8
n15	4	16	14	16	2	7	8	10	12	4	12	2
n16	5	13	15	12	16	15	3	7	9	11	3	13

Owen [1994] uses the correlation root mean square (*rms*) to compare LHS designs, and Joseph and Hung [2008] use *rms* as one of two objectives in their multiobjective algorithm to develop LHs. The *ms* for a column m is the average squared correlation coefficient (ρ_{lm}) between m and all other columns in a design with k columns:

$$\rho_{ms}^2(m) = \frac{\sum_{l \neq m}^{k} \rho_{lm}^2}{k - 1} \tag{5}$$

The largest $\rho_{ms}^2(m)$ identifies the column that is likely the most problematic, thereby having the greatest potential for improvement, and thus we use this criterion to select the next column to optimize. A possible alternative to this approach is to simply select the column containing ρ_{map}; however, this method results in ties among the columns. At a minimum, there would be two columns from which to select, necessitating another measure to break the tie. Thus far, $\rho_{ms}^2(m)$ provides a unique direction in the next step of the local search and has proven sufficiently effective for our purposes.

4. THE PROCESS FOR BREAKING DIMENSIONAL BARRIERS IN THE CONSTRUCTION OF NEW NOLH DESIGNS

In this section, we apply VMIN in different scenarios to demonstrate its effectiveness and versatility. Our approach creates many NOLHs, which are often beyond the dimensional limits of other procedures. Furthermore, we provide an explicit illustration of how VMIN extends the dimensions of existing OLH and NOLH designs, as well as how it adjusts to new experimental conditions, thus demonstrating the adaptive nature of this algorithm.

The combinations of n and k that are the most difficult for correlation reduction techniques are those where n is relatively small (less than 50) and $k \to n$ [Hernandez, 2008]. Therefore, we focus on this class of designs for our demonstration. Steinberg and Lin [2006] provide a test case, where $n = 16$ and $k = 12$, O_{12}^{16}, with which we make comparisons throughout this section.

We begin construction of an N_{12}^{16} by generating 1,000 RLHs and selecting the one with the smallest value of ρ_{map}. The result is an LH with a ρ_{map} value of 0.476. In 20 minutes, VMIN transforms this initial LH into an N_{12}^{16} with $\rho_{map} = 0.029$; see Table II.

Table III. Applying VMIN to a New RLH Generates Another N_{12}^{16}

Unique NOLH After VMIN Application: $\rho_{map} = 0.032$												
	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10	k11	k12
n1	7	16	4	13	11	10	4	16	9	4	2	10
n2	4	6	10	14	12	9	3	6	15	2	7	3
n3	5	13	16	7	3	14	8	11	13	10	16	5
n4	16	12	12	5	13	15	15	4	16	8	3	13
n5	11	7	7	3	4	16	6	7	2	3	11	4
n6	12	10	1	4	10	6	1	10	14	16	13	7
n7	1	11	3	10	5	12	13	5	3	9	6	12
n8	3	2	15	12	15	11	10	13	6	15	9	9
n9	8	4	2	9	16	8	12	2	8	5	15	6
n10	13	9	8	15	2	1	16	9	11	7	8	2
n11	6	3	11	6	1	5	2	1	10	11	1	11
n12	9	14	13	2	14	3	9	8	1	12	4	1
n13	15	5	14	8	9	4	5	12	4	1	12	16
n14	2	8	6	1	7	2	14	14	12	6	10	14
n15	14	1	5	11	6	13	11	15	7	14	5	8
n16	10	15	9	16	8	7	7	3	5	13	14	15

An important benefit of our approach is that it can produce many NOLHs of the same design dimension, while many other techniques to construct an OLH or NOLH in the literature are deterministic algorithms that provide a single design. Using new random number streams to yield different initial RLHs and applying VMIN, we develop new NOLHs with the same design dimensions. Table III shows a new VMIN-generated N_{12}^{16} with $\rho_{map} = 0.032$ that was initiated using the best of 1000 new RLHs—which had a $\rho_{map} = 0.506$.

An examination of the NOLH designs in Tables II and Table III shows that the properties of the two design matrices are different, although their dimensions are exactly the same. To distinguish between the two NOLH designs we classify them with their ρ_{map} value, designated $N_k^n(\rho_{map})$, which we now apply to the two designs: $N_{12}^{16}(0.029)$ and $N_{12}^{16}(0.032)$. If desired, we could generate an almost inexhaustible supply of different N_{12}^{16} designs.

A selection of NOLHs allows the analyst to use other properties for distinguishing among them. Although our approach does not attempt to directly balance correlation reduction and space-filling measures as do Joseph and Hung [2008], eliminating concern for high correlation enables the analysts to concentrate on other needs—for example, space-filling and/or higher order correlations. We extend our example and measure ML_2 for each, $N_{12}^{16}(0.029)$ with $ML_2 = 2.74$ and $N_{12}^{16}(0.032)$ with $ML_2 = 2.82$, and decide that $N_{12}^{16}(0.029)$ is more advantageous.

We compare the new VMIN designs with currently catalogued $n = 16$ and $k = 12$ designs from Fang's Web site[1] and Steinberg and Lin's [2006] OLH. We match them with our $N_{12}^{16}(0.029)$ design in terms of ρ_{map} and ML_2. Table IV summarizes the comparison of all three designs. While an analyst who requires minimal correlation among input variables would choose the OLH from Steinberg and Lin [2006] or the NOLH from VMIN, a need for a space-filling scheme points to VMIN as the best choice. In a case when both properties are required, VMIN offers the most advantage among these designs. It is interesting to note that $N_{12}^{16}(0.029)$ has a lower ML_2 than Fang's presumed optimal design (thereby dominating it). This underscores that using

---

[1](http://www.math.hkbu.edu.hk/UniformDesign/).

Table IV.
A comparison of three $n = 16$, $k = 12$ designs; one from Fang, another from Steinberg and Lin, and one from VMIN, in terms of correlation and space-filling measures, shows that VMIN offers advantages in both criteria.

	Fang[2011]	Steinberg and Lin[2006]	VMIN
ρ_{map}	0.20	0	0.029
ML_2	2.78	2.92	2.740

Table V.
A comparison of ρ_{map} summary statistics after Florian (1992) and VMIN application on an RLH for $n = 16$ and $k = 12$.

Statistic for ρ_{map}	Technique		
	Initial RLH	Post-Florian	Post-VMIN
Mean	0.458	0.121	0.033
Median	0.456	0.118	0.032
Range	0.385—0.509	0.082—0.191	0.024—0.044

heuristics to facilitate optimization does not necessarily guarantee an optimum is found.

Other methods, such as Florian's [1992] updated sampling technique, can generate LHs of this dimension with reduced correlation. However, an NOLH directly from an RLH using Florian is not easily achieved [Cioppa 2002]. We compare Florian's method with VMIN, applying both to 30 unique RLHs with $n = 16$ and $k = 12$. In each trial the initial RLH is the best of 1,000 RLHs. After applying Florian and VMIN separately to the initial RLH, we discover that VMIN outperforms Florian in every case. On average, VMIN reduces correlation over 71% better than Florian. In a number of cases, there is more than an 80% greater reduction of ρ_{map} using VMIN than when applying Florian. While VMIN results in 30 different NOLHs of the same design dimension, Florian's method produces none. A comparison of before and after ρ_{map} statistics from these 30 test cases is presented in Table V.

Our approach's adaptability to many different scenarios, as well as changing situations, is clear. As discussed, there are techniques to build a design that addresses 12 factors with just 16 runs. However, a new requirement that adds two factors for study without increasing the number of runs, while simultaneously keeping NOLH properties, is a difficult challenge. To the authors' knowledge, no other construction method to date can accommodate this change under the same conditions.

The given situation provides an opportunity to build an N_{14}^{16}, using either Fang's [2011] uniform design or Steinberg and Lin's [2006] O_{12}^{16} as a starting point. For this illustration, we select the Steinberg and Lin design, which we scale to the first n natural numbers, and append two columns. The entries for the two new columns are each simply a random permutation of the first 16 natural numbers. The initial design has $\rho_{map} = 0.61$.

VMIN quickly transforms the extended Steinberg and Lin (2006) design into an N_{14}^{16} with $\rho_{map} = 0.047$; see Table VI. This minimal sacrifice in orthogonality results in enabling the analyst to address two additional factors without increasing the required number of runs. Significantly, VMIN recognizes columns 13 and 14 as the most problematic and only modifies them in the process, thereby keeping the base O_{12}^{16} intact. The analyst now has the flexibility to conduct trade-offs between these new columns and the original columns of O_{12}^{16}, and to develop a new N_{12}^{16} with perhaps better properties in space-filling or other design needs.

As previously explained, we may generate a completely new RLH with $n = 16$ and $k = 14$, and apply VMIN, and we do so. An initial RLH of the same dimension has $\rho_{map} = 0.509$. Application of Florian's method results in a reduced correlation of

Table VI.
The final NOLH design after applying VMIN shows that Steinberg and Lin's [2006] OLH remains intact. VMIN modifies only columns 13 and 14 to reduce the design's overall correlation.

	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10	k11	k12	k13	k14
n1	16	6	4	10	5	3	14	5	13	7	16	6	**7**	**4**
n2	15	8	8	2	12	14	3	12	9	15	15	8	**14**	**6**
n3	14	5	12	14	2	9	9	15	4	10	1	11	**5**	**2**
n4	13	7	16	6	15	8	8	2	8	2	2	9	**15**	**7**
n5	12	14	3	12	3	12	5	3	6	1	10	13	**10**	**13**
n6	11	16	7	4	14	5	12	14	2	9	9	15	**4**	**11**
n7	10	13	11	16	8	2	2	9	11	16	7	4	**12**	**15**
n8	9	15	15	8	9	15	15	8	15	8	8	2	**1**	**10**
n9	8	2	2	9	13	7	16	6	3	12	5	3	**9**	**14**
n10	7	4	6	1	4	10	1	11	7	4	6	1	**3**	**12**
n11	6	1	10	13	10	13	11	16	14	5	12	14	**11**	**16**
n12	5	3	14	5	7	4	6	1	10	13	11	16	**2**	**9**
n13	4	10	1	11	11	16	7	4	12	14	3	12	**8**	**3**
n14	3	12	5	3	6	1	10	13	16	6	4	10	**13**	**5**
n15	2	9	9	15	16	6	4	10	5	3	14	5	**6**	**1**
n16	1	11	13	7	1	11	13	7	1	11	13	7	**16**	**8**

Final NOLH from Extended Steinberg and Lin Design, $n = 16$, $k = 14$: $\rho_{map} = 0.047$

$\rho_{map} = 0.124$. VMIN creates an N_{14}^{16} with $\rho_{map} = 0.041$. As an extension of this exercise (not shown here), we add another column to generate a saturated NOLH design, N_{15}^{16}, which of course includes 14 other N_{14}^{16} designs with the elimination of any column [Hernandez 2008].

The speed in which VMIN generates these new NOLHs is a departure from previous methods. In ten separate RLHs we recorded the time VMIN needed to solve one column. In developing N_{12}^{16} designs, VMIN solved any single column to a guaranteed optimal solution in less than three minutes. We also recorded the amount of time VMIN required to solve the entire design problem, resulting in an NOLH. The median time was less than 24 minutes, with the longest time less than 29 minutes. Similarly, we computed times when constructing complete N_{14}^{16} designs from raw RLHs. The median time was less than 36 minutes, with the longest time less than one hour. This relatively small cost in computational time is an added advantage for experimenters and opens opportunities for customized NOLH designs instead of a static library of designs.

Joseph and Hung [2008] developed LHs using simulated annealing to minimize a weighted linear combination of a correlation measure and a maximin space-filling metric. Specifically, they minimize $w_1 \rho^2 + w_2 \phi_p$, where ρ^2 is defined by Equation (6) and ϕ_p by Equation (7), and scaled to [0,1]. Using $w_1 = 0.5$ and $w_2 = 0.5$, Joseph and Hung generate an LH for $n = 9$ and $k = 4$. They show that their new design performs well against designs of similar dimensions by Morris and Mitchell [1995], Ye [1998], and Fang et al. [2002].

$$\rho^2 = \frac{\sum_{i=2}^{k} \sum_{j=1}^{i-1} \rho_{ij}^2}{k(k-1)2^{-1}}, \tag{6}$$

where ρ_{ij} is the linear correlation between columns i and j.

$$\phi_p = \left(\sum_{i=1}^{\binom{n}{2}} \frac{1}{d_i^p} \right)^{1/p}, \tag{7}$$

Table VII.
A comparison of the minimum ML_2 of 100 VMIN designs and an LH from Joseph and
Hung [2008] for $n = 9$ and $k = 4$.

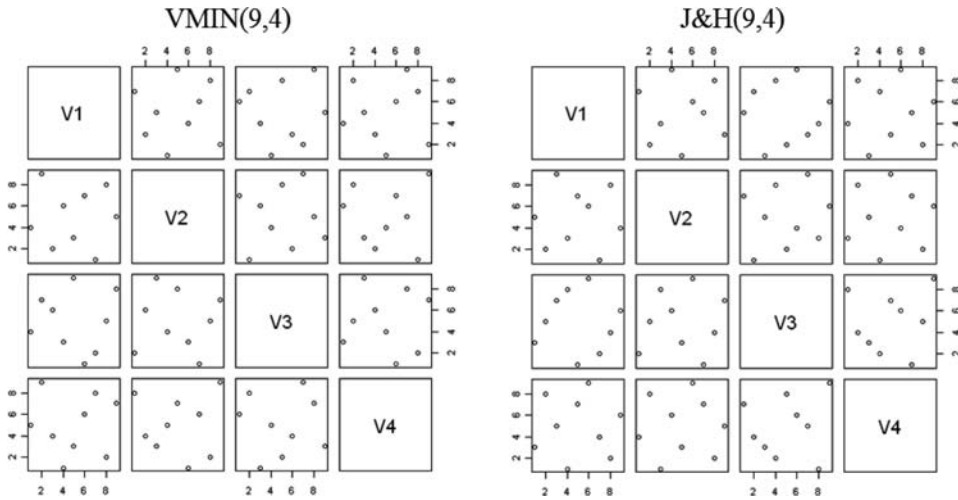| Design Matrix	VMIN				Joseph and Hung [2008]			
	9	5	8	7	1	5	3	3
	6	7	1	6	2	2	5	8
	4	6	3	1	3	9	7	5
	1	4	4	5	4	3	8	1
	7	1	2	8	5	7	1	7
	5	3	9	3	6	6	9	9
	2	9	7	9	7	1	2	4
	8	8	5	2	8	8	4	2
	3	2	6	4	9	4	6	6
ρ_{map}	0				0.117			
ρ	0				0.063			
ϕ_p	.1498				.1049			
ML_2	0.0485				0.0519			

Fig. 2. The pairwise projections of the VMIN generated LH and the one from Joseph and Hung [2008] for $n = 9$ and $k = 4$.

with d_i the rectangular distances between the $\binom{n}{2}$ pairs of points and p a positive integer (set to 15).

As an alternative to Joseph and Hung's design, VMIN was applied on 100 RLHs, and the orthogonal design with the best *ML-2* discrepancy was selected for comparison. Table VII displays the designs and their correlation and space-filling measures. We see that VMIN produces a design that is preferred by all of the correlation metrics. In addition, each design does better in the respective space-filling measure that was the focus in its construction. Figure 2 displays the projections of the nine design points into the six two-dimensional subspaces defined by all pairs of factors.

5. CONCLUSIONS

Latin hypercubes have proven their utility as easy-to-generate designs that allow experimenters to explore the relationships between a large number of input variables and multiple output variables. However, the degree of correlation among its input variables

Table VIII.
VMIN fills in gaps in the available catalogue of NOLH and OLH designs. Many of the VMIN-based NOLHs are saturated.

n	Maximum k for Each Method					
	Ye	Cioppa	Ang	Steinberg and Lin	Pang et al,	VMIN
9	4	**UNAVAILABLE**			4	**6**
14	**UNAVAILABLE**					**12**
16	**UNAVAILABLE**			12	12	**15**
17	6	7	8	**UNAVAILABLE**		**16**
19	**UNAVAILABLE**					**18**
24	**UNAVAILABLE**					**23**
25	**UNAVAILABLE**				6	**24**
32	**UNAVAILABLE**					**31**
33	8	11	16	**UNAVAILABLE**		**32**
47	**UNAVAILABLE**					**46**
49	**UNAVAILABLE**				8	**47**
64	**UNAVAILABLE**					**63**
65	10	16	32	**UNAVAILABLE**		

can affect its utility. Efforts to develop LHs with little or no correlation continue, but are stifled by strict dimensional constraints.

Our research makes OLHs and NOLHs available to experimenters in most any experimental situation. The new algorithm presented in this article enables construction of an NOLH for almost any n and $k < n$. Saturated NOLHs (S-NOLH) such as N_{15}^{16}, N_{16}^{17}, N_{18}^{19}, N_{24}^{25}, N_{32}^{33}, N_{46}^{47}, and N_{63}^{64} are only some of the new designs from our work [Hernandez 2008] that are available at http:/harvest.nps.edu [Sanchez 2012].

The results of our study greatly expand the inventory of NOLHs available to experimenters. In the "VMIN" column of Table VIII, which is an update of Table I, we introduce some of our new NOLHs that fill dimensional gaps discussed in Sections 1 and 2. For instance, Pang et al. [2009] develop an O_6^{25} design for a 25-run experiment, while in a later study Sun et al. [2009] produce an O_4^{25}. Yet, there are no OLH or NOLHs that cover more than six variables for 25 or less runs. Our algorithm addresses this concern. We quickly develop NOLHs, $N_6^n, n = 9, 10, \ldots, 14$ and OLHs, $O_6^n, n = 15, 16, \ldots, 23$. VMIN fills much of the remaining space with N_{12}^{14}, N_{15}^{16}, N_{16}^{17}, N_{18}^{19}, and N_{24}^{25} designs [Hernandez 2008]. Table VIII presents some specific values of n for which we have developed an S-NOLH, demonstrating how we fill gaps in k for a specified n.

The utility of our designs often emerge in experimental situations where no OLH or NOLH designs exist, even in theoretical dimensions. For example, the Peace Support Operations Model (PSOM)—a high-dimensional, interactive simulation—is an important training tool in humanitarian assistance and peace-keeping operations [Body and Marston 2007]. Such a model contains thousands of variables, many of which are uncertain, including the leadership and reputation of coalitions, as well as numerous population characteristics.

In Marlin's [2009] study to quantitatively analyze PSOM, he sought efficient experimental designs (OLH and NOLH) to test and explore a scenario used by the multinational forces in Iraq. He was concerned that classic fractional factorial designs may not deal with the range of possible response surfaces associated with PSOM. Although Marlin concentrated on just a few vignettes, the series of experiments still had hundreds of variables. One phase of these cumulative experiments required him to address 66 factors. Previously, no OLHs or NOLHs existed in this dimension. However,

our work produced designs for this situation. Marlin's study results have contributed to an understanding of PSOM and have helped the United Kingdom's Defence Science and Technology Laboratory, in partnership with U.S. Joint Staff J-8, to improve the model and cooperation for training in peacekeeping and stability operations. The United States Partnership for Peace Training and Education Center now uses PSOM as a primary tool for its global program.

We have presented significant advantages in using our technique for constructing experimental designs. Our algorithm readily and quickly creates new NOLHs for most any design dimension required, to include S-NOLHs. Additionally, we can repeat the process and create unique designs for the same design dimension, allowing the experimenter to use criteria other than the degree of correlation to select one that best fits experimental goals. Our process can extend existing OLHs and NOLHs to address new situations, while maintaining the orthogonal properties of the original design. We offer our new method to develop OLH and NOLH designs of any determinate dimension to the scientific community to increase the effective utilization of computer simulations in the sciences.

ACKNOWLEDGMENTS

REFERENCES

AARTS, E. AND LENSTRA, J.K. 1997. *Local Search in Combinatorial Optimization*. Wiley.

ANG, J. K. 2006. Extending orthogonal and nearly orthogonal Latin hypercube designs for computer simulation and experimentation. Master's thesis. Naval Postgraduate School, Monterey, CA.

BARTON, R. R. 1998. Simulation metamodels In *Proceedings of the Winter Simulation Conference*. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, Eds., vol 1, IEEE, 167–174.

BAZARAA, M. S., JARVIS, J. J., AND SHERALI, H. D. 2004. *Linear Programming and Network Flows* 4th Ed. Wiley.

BERTSIMAS, D. AND TSITSIKLIS, J. N. 1997. *Introduction to Linear Optimization*. Athena Scientific, Nashua, NH.

BODY, H. AND MARSTON, C. 2007. The peace support operations model: origins, development, philosophy and use. *J. Defense Model. Simul. Appl. Methodol. Technol. 8*, 2, 69–77.

BUYSKE, S. AND TROUT, R. 2001. *Advanced Design of Experiments*. Statistics 591 Lecture Series, Rutgers University.

CIOPPA, T. M. 2002. Efficient nearly orthogonal and space-filling experimental designs for high-dimensional complex models. Doctoral dissertation. Monterey, CA: Naval Postgraduate School.

CIOPPA, T. M. AND LUCAS, T. W. 2007, March. Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics 49*, 1, 45–55.

FANG, K. T. 2011. The uniform design. http://www.ath.hkbu.edu.hk/UniformDesign/.

FANG, K. T., LIN, D. K. J., WINKER, P., AND ZHANG, Y. 2000a. Uniform design: Theory and application. *Technometrics 42*, 3, 237–248.

FANG, K. T., MA, C. X., AND WINKER, P. 2000b. Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs. *Math. Comput. 71*, 237, 275–296.

FANG, K. T., MA, C. X., AND WINKER, P. 2002. Centered L2-discrepency of random sampling and Latin hypercube design. *Math. Comput. 71*, 275–296.

FLORIAN, A. 1992. An efficient sampling scheme: Updated Latin hypercube sampling. *Probabil. Engin. Mech. 7*, 123–130.

HERNANDEZ, A. S. 2008. Breaking barriers to design dimensions in nearly orthogonal Latin hypercubes. Doctoral dissertation. Naval Postgraduate School, Monterey, CA.

HICKERNELL, F. J. 1998. A generalized discrepancy and quadrature error bound. *Math. Comput. 67*, 299–322.

HILLIER, F. S. AND LIEBERMAN, G. J. 2005. *Introduction to Operations Research*. McGraw-Hill.

IMAN, R. L. AND CONOVER, W. J. 1982. A distribution-free approach to inducing rank correlation among input variables. *Commun. Statist. Simul. Comput. 11*, 3, 311–334.

IBM. 2012. IBM ILOG Cplex Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

JOSEPH, V. R. AND HUNG, Y. 2008. Orthogonal-maximin Latin hypercube designs. *Statistica Sinica 18*, 171–186.

KIM, L. AND LOH, H. 2003. Classification trees and bivariate linear discriminant node models. *J. Graphical Statist. 12*, 512–530.

KLEIJNEN, J. P. C., SANCHEZ, S. M., LUCAS, T. W., AND CIOPPA, T. M. 2005. A user's guide to the brave new world of designing simulation experiments. *INFORMS J. Comput. 17*, 3, 263–289.

KOEHLER, J. R. AND OWEN, A. B. 1996. Computer experiments. *Handbook Statist. 13*, 261–308.

L'E CUYER, P. 2009. Quasi-Monte Carlo methods with applications in finance. *Finance Stochastics 13*, 3, 307–349.

MARLIN, B. 2009. Ascertaining validity in the abstract realm of PMESII simulation models: An analysis of the peace support operations model PSOM. Masters thesis. Naval Postgraduate School, Monterey, CA.

MCKAY, M. D., BECKMAN, R. J., AND 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, 239–245.

MOON, H., DEAN, A., AND SANTNER, T. J. 2011. Algorithms for generating maximin Latin hypercube and orthogonal designs. *J. Statist. Theory Pract. 5*, 1, 81–98.

MORRIS, M. D. AND MITCHELL, T. J. 1995. Exploratory designs for computer experiments. *J. Statist. Planning Inference 43*, 381–402.

MONTGOMERY, D. C. 2005. *Design and Analysis of Experiments*. Wiley.

OWEN, A. B. 1994. Controlling correlations in Latin hypercube samples. *J. Amer. Statist. Assoc. Theory Methods 89*, 428, 1517–1522.

OWEN, A. B. 1998. Latin supercube sampling for very high-dimensional simulations. *ACM Trans. Model. Comput. Simul. 8*, 1, 71–102.

PANG, F., LIU, M. Q., AND LIN, D. K. J. 2009. A construction method for orthogonal Latin hypercube designs with prime power levels. *Statistica Sinica 19*, 1721–1728.

PATTERSON, H. D. 1954. The errors of lattice sampling. *J. Royal Statist. Soc. Series B Methodol. 16*, 1, 140–149.

RARDIN, R. L. 1998. *Optimization in Operations Research*. Prentice Hall, Inc.

SANCHEZ, S. M. 2012. NOLH designs spreadsheet and code. http://harvest.nps.edu.

SANCHEZ, S. M., LUCAS, T. W., SANCHEZ, P. J., NANNINI, C. J., AND WAN, H. 2012. Designs for large-scale simulation experiments, with application to defense and homeland security. *The Design and Analysis of Computer Experiments. Vol. 3: Special Designs and Applications*, K. Hinkelmann, Ed., Wiley, 413–441.

SANCHEZ, S. M. AND SANCHEZ, P. J. 2005. Very large fractional factorial and central composite designs. *ACM Trans. Model. Comput. Simul. 15*, 4, 362–377.

SANTNER, T. J., WILLIAMS, B. J., AND NOTZ, W. 2003. *The Design and Analysis of Computer Experiments*. Springer.

STINSTRA, E., STEHOUWER, P., DEN HERTOG, D., AND VESTJENS, A. 2003. Constrained maximin designs for computer experiments. *Technometrics 45*, 4, 340–346.

STEINBERG, D. M. AND LIN, D. K. J. 2006. A construction method for orthogonal Latin hypercube designs. *Biometrika 93*, 2, 279–288.

SUN, F., LIU, M. Q., AND LIN, D. K. J. 2009. Construction of orthogonal Latin hypercube designs. *Biometrika 96*, 971–974.

TANG, B. 1998. Selecting Latin hypercubes using correlation criteria. *Statisca Sinica 8*, 965–977.

WOLSEY, L. A. 1998. *Integer Programming*. Wiley.

YE, K. Q. 1998. Orthogonal column Latin hypercubes and their application in computer experiments. *J. Amer. Statist. Assoc. Theory Methods 93*, 444, 1430–1439.