Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

IMPROVED EFFICIENT, NEARLY ORTHOGONAL, NEARLY BALANCED MIXED DESIGNS

Hélcio Vieira, Jr.

Technological Institute of Aeronautics
Praça Marechal Eduardo Gomes, 50
12228-900, São José dos Campos, BRAZIL

Susan M. Sanchez

Naval Postgraduate School
1 University Circle
Monterey, CA 93943, USA

Karl Heinz Kienitz
Mischel Carmen Neyra Belderrain

Technological Institute of Aeronautics
Praça Marechal Eduardo Gomes, 50
12228-900, São José dos Campos, BRAZIL

ABSTRACT

Designed experiments are powerful ways to gain insights into the behavior of complex simulation models. In recent years, many new designs have been created to address the large number of factors and complex response surfaces that often arise in simulation studies, but handling discrete-valued or qualitative factors remains problematic. We proposed a framework for generating, with a (given) limited number of design points n, a design which is nearly orthogonal and also nearly balanced for any mix of factor types (categorical, numerical discrete, and numerical continuous) and/or mix of factor levels.

Our approach can be used to create designs with low maximum absolute pairwise correlation, low imbalance level, and high $\boldsymbol{D}$-optimality for simulation problems with mixed factor types. Our mixed designs are much more efficient than existing alternatives.

1 INTRODUCTION

The field of statistical design of experiments (DoE) was born in the 1920's through the pioneering work of Fisher (2000) in the agriculture arena. The basic principles of DoE are the use of *randomization*, *replication*, and *control* to allow the analyst to make statistically valid inferences about the behavior of a system. As noted by Montgomery (2005), "[T]here is not a single area of science and engineering that has not successfully employed statistically designed experiments."

Simulation is one of those fields, and we refer the reader to Sanchez and Wan (2009), Kleijnen (2007), Law (2007), or Santner, Williams, and Notz (2003) to find out more about conducting experiments in simulation settings. Large-scale simulation experiments often have more complex goals than physical experiments. These goals include: developing a broad understanding of a complex system; identifying robust aspects of the system; and comparing alternative system configurations (Kleijnen et al. 2005, Sanchez et al. 2011). Classical designs typically cannot be used in the simulation environment without making restrictive or unwarranted assumptions. Fortunately, recent advances in DoE is expanding the design portfolio available to analysts, improving their ability to conduct large-scale simulation experiments.

In this paper, we focus on single-stage experiments. The experimental design is an $n \times p$ matrix of factor settings, with a row corresponding to each of n design points and a column corresponding to each of p factors. The title of this work has several terms that we now formally clarify.

978-1-4577-2109-0/11/$26.00 ©2011 IEEE

- We define *mixed designs* as designs with different factor types (categorical, discrete and continuous) and/or different factor levels (e.g., factor 1 with 10 levels, factor 2 with 5 levels, factor 3 with 2 levels, etc.). Throughout this paper, we use the terms "qualitative" and "categorical" interchangeably, and may refer to discrete and continuous factors as "quantitative" or "numerical."
- A design is said to be *balanced* if the number of objects in each of the levels of each column is equal. We call a design *nearly balanced* if the number of objects in each level of each factor differs from the ideal by no more than α. Put mathematically: $(1-\alpha)\lambda_c \leq \omega_{cl} \leq (1+\alpha)\lambda_c$, $\forall l, c$, where $0 \leq \alpha < 1$ is the percentage of allowed imbalance, $\lambda_c = {}^{n}\!/\!_{\beta_c}$ is the ideal number of objects in each level in column c, n is the number of design points, β_c is the number of levels in column c, and ω_{cl} is the number of objects in level l in column c. The specification of the imbalance value is subjective and problem dependent. From our point of view, an acceptable imbalance value is less than 20%.
- Let ρ_{map} denote the maximum absolute pairwise correlation between any two factors (columns). An *orthogonal design* has $\rho_{map} = 0$. If a design has $0 < \rho_{map} \leq 0.05$, it is called a *nearly orthogonal design*.
- Finally, we characterize a design as *efficient* if the number of design points is acceptable. Again, this concept is subjective and is problem driven.

The above concepts are important, especially for simulation studies, for several reasons. Simulation models usually have different factor types and factor levels, and designs that accommodate this variety are needed. The balance property allows correct analysis of non-normal heteroscedastic experiments (see Bathke (2007)). Orthogonality makes it possible to model the effect of one factor independently of other factors (see, e.g., Montgomery (2005) and Ryan (2008)). Finally, despite the ready availability of high-speed computing processors, brute-force computation cannot be used to explore large-scale simulation experiments. Real-world simulation studies face restrictions due to time, cost, number of computers available for experimentation, etc. They need efficient designs, although the number of design points is not the overriding consideration.

There are two common approaches to dealing with mixed factors. The first approach involves using an orthogonal array, which is a balanced design suitable for any type of factor (qualitative and/or quantitative). The second approach involves constructing separate designs for quantitative and categorical factors, and then *crossing* the designs. Typically, a discrete factor is treated as categorical if it has only a handful of levels, or continuous (perhaps with rounding) otherwise; note that too much rounding can destroy the orthogonality of the design. Unfortunately, both these approaches can be extremely inefficient and lead to enormous designs ($n \gg p$) if there are many discrete or categorical factors with several levels. Also, the catalogue of orthogonal arrays for large p is extremely limited, particularly if the factors take on different numbers of levels.

Recently, we have successfully used mixed integer programming (MIP) to construct designs that are suitable for discrete-valued factors without treating them as continuous or requiring them all to have the same numbers of levels. In Vieira Junior et al. (2011b), we create ***orthogonal, balanced*** designs for quantitative (discrete and/or continuous) factors. In Vieira Junior et al. (2011a), we relax the balance requirement, and provide a MIP formulation suitable for constructing ***nearly orthogonal, nearly balanced*** designs for quantitative (discrete and/or continuous) factors.

The purpose of this paper is to propose a framework for generating, with a (given) limited number of design points n, a design which is nearly orthogonal and also nearly balanced for any mix of factor types (categorical, numerical discrete, and numerical continuous) and/or number of factor levels. The organization of the rest of this paper is as follows. In Section 2, we present technical background, and discuss the drawbacks of the crossed design and orthogonal array approaches in more detail. Our MIP formulation appears in Section 3. In Section 4 we provide some examples, and our concluding remarks appear in Section 5.

2 TECHNICAL BACKGROUND

We are interested in designs able to provide the analyst with a broad understanding of the simulation over the region of interest in exploratory simulation studies. When factors are continuous, space-filling designs are useful for exploratory studies because they provide insight about the simulation behavior throughout the region of interest. An analogy for discrete-valued factors is that they take on many (perhaps all) of the potential levels of interest. For example, a design where x assumes levels $\beta_x \in \{0,1\}$ (in weeks) is less space-filling than a design where x assumes levels $\beta_x \in \{1,2,\ldots,7\}$ (in days). For categorical factors, we assume that β_x may need to be large in order to adequately reflect the complexity of the real-world situation being modeled and/or that the number of categorical factors is big.

2.1 Designs for Categorical Factors

Orthogonal arrays (OAs) have played an important role in experimental design (see Hedayat, Sloane, and Stufken (1999) for more information). These arrays possess some properties that allow them to be used for analysis of any type of data (numerical and/or categorical). For example, consider an $n \times p$ matrix, where the elements in column x are from the set of integers $\{1,2,\ldots,\beta_x\}$ for some integer $\beta_x \leq n$. If the array has the property that any subarray of size $n \times g$ contains all possible combinations of values equally often as rows, the OA is said to have "*strength g*." Orthogonality is important because it allows one to estimate the effect of one factor independently of the others.

In order to achieve this desirable characteristic, orthogonal arrays must "save" several degrees of freedom to allow a subsequent analysis of the collected data (the reason will be shown in Subsection 2.1.1). In order to "save" degrees of freedom, classical DoE requires the number of design points to be greater than the number of factors. Design points are often called "runs" in statistical literature, but in this paper we use "design points" because the terms "run" and "replication" are often used interchangeably in simulation studies. When the number of levels each factor possesses is big, the required number of design points is **much** greater than the number of factors.

2.1.1 Indicator Variable Representation

If any of the factors are categorical, it is necessary to work with indicator (also known as "dummy") variables. "The design column for a factor level is constructed as the zero-one indicator of that factor level minus the indicator of the last level ... [In this fashion, the design matrix] achieves full rank unless there are missing cells or other incidental collinearity" (SAS Institute 2005). Other indicator variable codings are possible, such as a two-level 0/1 coding with the omitted factor representing the baseline, but this three-level coding assures that when regression models are fit to the resulting data, the intercept represents the overall mean response. An example of the construction of indicator variables for a four-level categorical factor is given in Table 1.

Table 1: Example of indicator-variable construction.

Categorical factor	Level 1 indicator	Level 2 indicator	Level 3 indicator
1	1	0	0
2	0	1	0
3	0	0	1
4	-1	-1	-1

2.1.2 Drawbacks of Using OAs for Mixed Factor Experiments

From Table 1, it is easy to understand why orthogonal arrays need to "save" so many degrees of freedom: each categorical factor is transformed into $\beta_x - 1$ new factors. Doing so means that at least $1+\sum_{x=1}^{j}(\beta_x - 1)$ design points are needed for an experiment involving j categorical factors, where β_x is the number of categories for factor x.

Now suppose that the experiment includes quantitative factors as well as categorical factors. If OAs are to be used, a (numerical) discrete factor x with β_x levels will use $\beta_x - 1$ degrees of freedom as above. In contrast, if x is treated as a quantitative factor, then a single degree of freedom is sufficient for estimating the main effect of x (two degrees of freedom can be used to estimate a quadratic relationship, and so forth). Clearly, treating the factor as quantitative is more efficient if a parsimonious representation of the response's dependence on x can be obtained.

OAs are most efficient if all the β_x are small, so there is a temptation to set $\beta_x = 2$ for any quantitative factor x. However, the resulting designs will have poor space-filling behavior, and so are far less useful for exploratory studies than other designs. But if the β_x are large, then the size of the OA can be immense.

In summary, using an OA for a mixed factor experiment will likely require an excessively large number of design points—particularly if there are several discrete or continuous factors.

2.2 Space-filling Designs for Continuous Factors

Randomly generated Latin hypercubes (LHs) have been widely used for computational experiments (Sacks et al. 1989). They tend to have good space-filling and orthogonality behavior if $n \gg p$, but when $n \approx p$ they can perform quite poorly. Cioppa and Lucas (2007) constructed efficient, space-filling, nearly orthogonal Latin hypercubes (NOLHs) that have proven useful for investigating continuous factors in a number of studies. To overcome the limited combinations of p and n for which NOLHs were available, Hernandez et al. (2011) developed a mixed integer programming approach that allows for the construction of nearly orthogonal Latin hypercubes for non-saturated cases ($2 < p < n$).

2.2.1 Drawbacks of Using Rounded NOLHs for Mixed Factor Experiments

One issue relating to all of the designs of both Cioppa and Lucas (2007) and Hernandez et al. (2011) is that they are constructed for continuous-valued factors. Applying them to discrete-valued factors requires rounding. A limited amount of rounding is acceptable, but if there are several factors with small numbers of levels this can destroy the near-orthogonality of the designs.

If rounding a particular design M causes problems, there are a few steps the analyst can take to mitigate these problems. First, the analyst could construct a new design based on $n' > n$ design points to see if the additional granularity in the base design reduces the correlations induced by rounding. For the designs of Cioppa and Lucas (2007), the available n's are $2^p + 1$ for $p = 4(1)8$, so the number of design points is essentially doubled each time n increases. Hernandez et al. (2011) greatly expand the available combinations of p and n for which NOLHs are available for continuous factors so that n need not grow so rapidly, but even so, achieving good orthogonality in the presence of rounding is not guaranteed. Alternatively, the analyst could construct several designs and stack them until suitable near-orthogonality is achieved. However, this is an ad hoc method. If the original NOLH (for continuous factors) has n design points, then each stack has $\approx n$ design points as well.

Even if the rounding problem is solved, the NOLH can deal only with numerical (discrete and/or continuous) factors.

2.3 Designs for Mixed Numerical Factors

In the previous Sections, we discuss how neither OAs or NOLHs may be suitable for handling designs involving a mixture of continuous, discrete, and categorical factors. If suitable designs can be created for each type of factor separately, then these smaller designs can be crossed to obtain one that, overall, is close to orthogonal. For example, OAs can be used for factors that are categorical, or discrete with a limited number of levels. NOLHs or other space-filling designs could be used for continuous factors, and for discrete factors with many levels of interest. However, if designs D_1 and D_2 have n_1 and n_2 design points, respectively, then the crossed design $D_1 \times D_2$ will have $n_1 \times n_2$ design points.

Our recent work takes a more direct approach for constructing designs for mixed factors. In Vieira Junior et al. (2011b), we extend and enhance the mixed integer programming (MIP) formulation of Hernandez et al. (2011) in order to construct orthogonal designs, or improve existing orthogonal arrays, for experiments involving quantitative factors with limited numbers of levels of interest. Subsequently, we relax the requirement for balance and orthogonality, and present a MIP formulation for constructing nearly orthogonal, nearly balanced designs for mixed factors (Vieira Junior et al. 2011a). We now provide a brief description of this formulation, in order to facilitate the presentation of our new extension which incorporates qualitative factors.

Let $M = [a_{rc}]_{n \times j}$ denote a design matrix with n rows and j columns, and for notational convenience let $\overline{c}$ and s_c denote the mean and standard deviation of column c, respectively. The sample pairwise correlation between two columns x and y of this matrix is given by (1).

$$\rho_{xy} = \frac{\sum_{r=1}^{n} (a_{rx} - \overline{x})(a_{rc} - \overline{y})}{(n-1)s_x s_y}. \tag{1}$$

Now, fix the values of all columns in M except column x; this means that the a_{ry}, $\overline{y}$, and s_y are all constants for $y \neq x$. Define ρ^*_{xy} as given by (2).

$$\rho^*_{xy} = \rho_{xy}(n-1)s_x = \frac{\sum_{r=1}^{n} (a_{rx} - \overline{x})(a_{ry} - \overline{y})}{s_y}. \tag{2}$$

If we constrain the factor x to be balanced, then $\rho^*_{xy} \propto \rho_{xy}$. If we allow only a small imbalance on x, then $\rho^*_{xy} \overset{\sim}{\propto} \rho_{xy}$, where $\overset{\sim}{\propto}$ means approximately proportional.

Now, if a design is nearly orthogonal, that means that $|\rho_{xy}| < 0.05$, but mathematical programming approaches cannot deal directly with this form of an objective function. Fortunately, we can define a quantity v and constrain it to satisfy $v \geq \max_{y \neq x} \rho^*_{xy}$ and $v \geq -\max_{y \neq x} \rho^*_{xy}$. If we can identify values for the x_r so that $v = 0$, then column x is orthogonal to all other columns in M.

Vieira Junior et al. (2011a) show that, with suitable constraints, one can use a mathematical programming approach to optimize v as a linear function of the entries in a particular column x_r. A MIP formulation is required because integer-valued variables are used in the design construction process. Applying this MIP sequentially allows new designs to be constructed. Specifically, start by randomly creating a one-column matrix $M = [a_{rc}]_{n \times 1}$ with the desired levels and, sequentially, add a new column corresponding to a new factor, and solve the MIP.

2.3.1 MIP Formulation for Numerical Factors

If all the factors are numerical (continuous and/or discrete), the MIP formulation of Vieira Junior et al. (2011a) can be used to construct designs. This MIP is provided in (3), and has the following characteristics:

INPUTS

$M = [a_{rc}]_{n \times j}$ A design matrix with n rows and j columns;
x The column of M to optimize;
β_x The number of levels ($\leq n$) associated with the factor in column x ($x = 1, \ldots, j$);
α The maximum allowable imbalance for any factor ($0 \leq \alpha < 1$).

VARIABLES

x_r Entry in the r^{th} row of column x

FORMULATION

$$
\begin{array}{lll}
Min\, v & & \\
s.t. & & \\
(i) & v \geq \frac{1}{s_c} \sum\limits_{r=1}^{n} \left(x_r - \frac{1}{n} \sum\limits_{k=1}^{n} x_k \right) (a_{rc} - \bar{c}) & c = 1,\, 2,\, \ldots, x-1,\, x+1, \ldots,\, j \\
(ii) & v \geq -\frac{1}{s_c} \sum\limits_{r=1}^{n} \left(x_r - \frac{1}{n} \sum\limits_{k=1}^{n} x_k \right) (a_{rc} - \bar{c}) & c = 1,\, 2,\, \ldots, x-1,\, x+1, \ldots,\, j \\
(iii) & \sum\limits_{l=1}^{\beta_x} \theta_{rl} = 1 & r = 1,\, 2,\, \ldots,\, n \\
(iv) & x_r = \sum\limits_{l=1}^{\beta_x} l\theta_{rl} & r = 1,\, 2,\, \ldots,\, n \\
(v) & \sum\limits_{r=1}^{n} \theta_{rl} \leq (1+\alpha) \left\lceil \frac{n}{\beta_x} \right\rceil & l = 1,\, 2,\, \ldots,\, \beta_x \\
(vi) & \sum\limits_{r=1}^{n} \theta_{rl} \geq (1-\alpha) \left\lfloor \frac{n}{\beta_x} \right\rfloor & l = 1,\, 2,\, \ldots,\, \beta_x \\
(vii) & \theta_{rl} \in \{0,1\} & r = 1,\, 2,\, \ldots,\, n;\; l = 1,\, 2,\, \ldots,\, \beta_x
\end{array}
\tag{3}
$$

Here, $\lceil b \rceil$ is the smallest integer greater than b, and $\lfloor b \rfloor$ is the greatest integer smaller than b.

As discussed above, constraints (i) and (ii) ensure that $v \geq \rho^*_{xy}$ and $v \geq -\rho^*_{xy}$, i.e., $v \geq |\rho^*_{xy}|$ regardless of the sign of ρ^*_{xy}, for all $y \neq x$. Constraint (iii) assures that only one of the β_x levels will be assigned to x_r. The translation from these binary indicators to their integer equivalents (i.e., from θ_{rl} to x_r) is accomplished by (iv). The imbalance limits are guaranteed by the constraints (v) and (vi). Finally, constraint (vii) ensures that θ_{rl} can assume only the values 0 or 1.

2.3.2 Implementation

In real-world-simulation problems, the numbers of levels and numbers of design points are usually not small. This makes the size of the branch and bound tree large (with $(\beta_x)^n$ alternatives), restricting its full inspection in a reasonable amount of time. Consequently, we allow the MIP algorithm to perform its search for a limited, pre-specified amount of time t and consider at the current best solution v^*. At that time, if the optimized $v^* = \min \max\limits_{y} |\rho^*_{xy}| \neq 0$, we calculate the $\rho_{map} = \max\limits_{x \neq y} |\rho_{xy}|$. If it is less than or equal to 5%, we accept the optimized column and move forward to create new ones. If $\rho_{map} > 0.05$, then we run the MIP algorithm again, giving it more time to perform its search. This last procedure should be repeated until $\rho_{map} \leq 0.05$. If $v^* = 0$, then an orthogonal column has been found and it is not necessary to calculate the new value of ρ_{map}.

2.3.3 Pitfalls to Avoid

A mistake that might be made by someone unfamiliar with experimental design is to use a column of a design matrix intended for a numerical factor to represent the (coded) levels of a categorical factor. We now give a small example to show why this is such a bad idea.

Table 2 shows two categorical factors and their respective indicator variables, where x_i is the i^{th} categorical factor and x_i^j is the indicator variable for the j^{th} level of the i^{th} categorical factor. The correlation between x_1 and x_2 is $\rho_{x_1 x_2} = 0$; i.e., they are orthogonal to each other (at least with the orthogonality definition we use). Despite being orthogonal in the original levels, when we analyze the corresponding indicator variables, the correlation between x_1^3 and x_2^2 is $\rho_{x_1^3 x_2^2} = -1$; i.e., they are perfectly correlated with each other. If the statistical analysis states that the level three of factor one is the main responsible for the measured outcome variability, it cannot be assessed if this variability was due to level three of factor one, to level two of factor two or to a combination of both. This is called "confounding" in DoE terminology.

Table 2: Example of correlation problems with categorical variables.

Categorical Factors		Indicator Representations					
x_1	x_2	x_1^1	x_1^2	x_1^3	x_2^1	x_2^2	x_2^3
1	3	1	0	0	0	0	1
2	1	0	1	0	1	0	0
3	4	0	0	1	-1	-1	-1
4	2	-1	-1	-1	0	1	0

This situation exists even if one of the factors is numerical. If x_1 were numerical instead of categorical, we still would have problems with correlation: $\rho_{x_1 x_2^3} = -0.632$. This means that the columns constructed using the MIP of (3) cannot be used to define the levels of categorical factors. We also remark that the MIP formulation of (3) cannot be used to directly construct indicator variables, except in one special case. If all categorical factors have only two potential levels, then each categorical factor requires a single indicator column. A design could be constructed using the coded values $\{1,2\}$ for each of these indicators, and the results could be converted back to the original units for the associated categorical factors.

3 CONSTRUCTING MIXED DESIGNS THAT INCLUDE QUALITATIVE FACTORS

Our new formulation uses the same basic ideas the previous one used (the new correlation calculus and the sequential creation of columns instead of generating the whole matrix in one step). However, in order to be able to deal with categorical factors, we move to an indicator variable view of the factors, as described in Section 2.1.1. This leads us to modify some constraints and add others. We briefly describe the motivation for the modifications, then present the new formulation and discuss some of the new constraints in more detail.

First, we need new notation to allow for the construction of $\beta_x - 1$ indicator variables (i.e., $\beta_x - 1$ columns) for each categorical variable, rather than a single column for each numerical factor. We let x_r^i represent the value in the r^{th} row of the i^{th} indicator variable associated with categorical factor x; we modify variable θ_{rl} to θ_{rl}^i for the same reason. Second, several constraints are needed to ensure that the indicator variable columns are constructed correctly. These columns should contain entries $x_r^i \in \{-1,0,1\}$ but not necessarily in equal proportions: zeroes will be more prevalent if β_x is large. Related to this, constraints enforcing near-balance of the design are not concerned with the numbers of zeros in indicator variable columns.

3.1 MIP Formulation for Categorical Factors

Equation (4) gives our new MIP, which works for qualitative (categorical) factors.

INPUTS

$M = [a_{rc}]_{n \times j}$ A design matrix with n rows and j columns;
x The categorical factor to be added to M;
β_x The number of levels ($\leq n$) associated with the categorical factor x;
α The maximum allowable imbalance for a factor ($0 \leq \alpha < 1$).

VARIABLES

x_r^i Entry in the r^{th} row of the i^{th} indicator variable column for x

FORMULATION

$$
\begin{array}{lll}
\multicolumn{3}{l}{Min\ v} \\
\multicolumn{3}{l}{s.t.} \\
(i) & v \geq \frac{1}{s_c} \sum_{r=1}^{n} \left(x_r^i - \frac{1}{n} \sum_{k=1}^{n} x_k^i \right) (a_{rc} - \bar{c}) & c = 1,2,\ldots,j;\ i = 1,2,\ldots,\beta_x - 1 \\
(ii) & v \geq -\frac{1}{s_c} \sum_{r=1}^{n} \left(x_r^i - \frac{1}{n} \sum_{k=1}^{n} x_k^i \right) (a_{rc} - \bar{c}) & c = 1,2,\ldots,j;\ i = 1,2,\ldots,\beta_x - 1 \\
(iii) & \sum_{l=1}^{3} \theta_{rl}^i = 1 & r = 1,2,\ldots,n;\ i = 1,2,\ldots,\beta_x - 1 \\
(iv) & x_r^i = \sum_{l=1}^{3} (l-2)\theta_{rl}^i, & r = 1,2,\ldots,n;\ i = 1,2,\ldots,\beta_x - 1 \\
(v) & \sum_{r=1}^{n} \theta_{rl}^i \leq (1+\alpha) \left\lceil \frac{n}{\beta_x} \right\rceil & l = 1,3;\ i = 1,2,\ldots,\beta_x - 1 \\
(vi) & \sum_{r=1}^{n} \theta_{rl}^i \geq (1-\alpha) \left\lfloor \frac{n}{\beta_x} \right\rfloor & l = 1,3;\ i = 1,2,\ldots,\beta_x - 1 \\
(vii) & \sum_{i=1}^{\beta_x - 1} \theta_{r3}^i \leq 1 & r = 1,2,\ldots,n \\
(viii) & \sum_{i=1}^{\beta_x - 1} \theta_{r2}^i \leq \beta_x - 2 & r = 1,2,\ldots,n \\
(ix) & \theta_{r1}^i - \theta_{r1}^1 = 0 & r = 1,2,\ldots,n;\ i = 2,3,\ldots,\beta_x - 1 \\
(x) & \theta_{rl}^i \in \{0,1\} & r = 1,2,\ldots,n; l = 1,2,3;\ i = 1,2,\ldots,\beta_x - 1
\end{array}
\tag{4}
$$

As in Formulation (3), constraints (*i*) and (*ii*) ensure that $v \geq |\rho_{x^i y}^*|$ regardless of the sign of $\rho_{x^i y}^*$. Constraint (*iii*) assures that only one of the three possible levels will be assigned to x_r^i, and constraint (*iv*) performs that assignment. The imbalance limits are guaranteed by the (*v*) and (*vi*); note that these are enforced only for non-zero values of the indicator variables.

Constraints (*vii*)–(*ix*) are needed to construct the indicator variables properly. Specifically:

- No two indicator variables can have 1's assigned to the same row if they correspond to the same categorical factor. For example, if we have $x_1^1 = \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 1 \end{pmatrix}^T$, then the column vector $\begin{pmatrix} 1 & 0 & 1 & -1 & -1 & 0 \end{pmatrix}^T$ is not an allowable solution for x_1^2. Despite being orthogonal to each other, the value 1 in the first row of both vectors would be interpreted as "set factor x_1 to levels 1 and 2 for design point 1," which is not possible. Constraint (*vii*) assures that multiple level assignments do not occur within a particular design point (row) of M;

- There must **not** be a row in the set of indicator variables filled only with 0's. The reason is that every level of the categorical factor must be represented by exactly one of the indicator vectors. If we have 0 in a row at all indicator vectors, it means that none of them are "indicating" the level that that row has in the categorical factor. This is assured by the constraint (*viii*); and
- All indicator variables for the same categorical factor must have -1 assigned to same rows. This is assured by the constraint (*ix*).

Finally, constraint (*x*) specifies that the θ^i_{rl} are binary-valued variables.

When constructing a design for categorical factors only, begin by specifying a reasonable number of design points. Given j qualitative factors, we need $n > \sum_{x=1}^{j}(\beta_x - 1)$ design points. Then, randomly generate the first categorical factor and, based on the levels of this first factor, construct the set of indicator columns in the same fashion Table 1 did. This set of indicator variables will be the initial design matrix M. In each subsequent iteration, construct the appropriate indicator columns for another categorical factor by using Formulation (4). After all of the qualitative factors have been added, then the indicator columns for each categorical factor in the design matrix M should be replaced by a single column that lists the categorical levels (in original units) for that factor to facilitate experimentation.

3.2 MIP Approach for Qualitative and Quantitative Factors

When generating a design for a mix of categorical and numerical factors, begin by specifying a reasonable number of design points n. If N_{cat}, N_{disc}, and N_{cts} represent the numbers of categorical, discrete, and continuous factors, respectively, then we need $n > N_{disc} + N_{cts} + \sum_{x=1}^{N_{cat}}(\beta_x - 1)$. A design for the categorical factors should be created as described in at 3.1. Once a suitable design has been constructed for the categorical factors, then iteratively add numerical columns, one at a time by using Formulation (3). After all of the numerical factors have been added, then the indicator columns for each categorical factor in the design matrix M should be replaced by a single column that lists the categorical levels (in original units) for that factor to facilitate experimentation.

3.3 Other Implementation Issues

As described in at 2.3.2, it may be necessary to run the MIP for a specified amount of time t and then stop to see if a suitable design has been obtained, rather than attempting to let the MIP run to completion. If no design that meets the desired balance and correlation properties can be found in a timely manner, consider increasing n and/or t, and starting over.

4 RESULTS

Our motivation for creating these nearly balanced, nearly orthogonal mixed designs arose from numerous simulation studies in a variety of application areas related to defense and national security. Rather than provide details about the factors, settings, results, and interpretation for any single study, we provide brief descriptions of the design characteristics for two recent simulation experiments.

Before proceeding, a more detailed discussion of design efficiency is in order. We have already shown that large-scale models cannot be explored using brute-force methods. However, it is not the case that designs should be compared solely in terms of the number of design points n. Heterogeneous variances are pervasive in simulation, meaning that multiple replications $b > 1$ are needed. The time required for a single run at a single design point is typically not constant, so that the total computational effort is not necessarily proportional to the number of design points n, or even the total number of runs nb. Most of our experiments are performed on computing clusters, where multiple runs are conducted in parallel. This means that the time required to complete all the runs is more important than either n or nb. Finally, there is substantial benefit to the analyst if they can analyze and interpret the results of a single experiment, rather than having to go through an iterative sequence of experiments that build on information from earlier ones (e.g., beginning with a screening experiment, then moving to a higher-resolution design for a limited number of factors, then cross-checking to ensure that they have not missed important terms, and repeating this process). Unless the time for an individual run is quite large, we have found that designs with $3k \leq n \leq 10k$ provide a good mix of efficiency, statistical power, and analysis flexibility.

4.1 First Design

The Naval Postgraduate School's *SEED Center for Data Farming* (http://harvest.nps.edu/) is conducting a study of the United States Marine Corps' Total Life Cycle Management Assessment Tool (TLCM-AT). The objectives of the study include assessing the model's sensitivities, identifying critical input data, determining robust strategies, and generating distributions on future possibilities. The study is intended to complement other ongoing Validation & Verification (V&V) activities.

TLCM-AT has a large number of quantitative and qualitative inputs. Additionally, there are sources of significant uncertainty associated with many of these inputs, e.g., failure rates, operational tempo, etc. This project leverages the benefits of using state-of-the-art experimental design techniques, coupled with high-performance computing, to investigate the model's behavior over a range of inputs.

We developed a design for this study that involved 15 continuous factors, 5 qualitative factors with 2 levels, 2 qualitative factors with 3 levels, and 5 qualitative factors with 5 levels. The $\boldsymbol{D}$-optimality, maximum imbalance value, and ρ_{map} of the design are, respectively, 99.97%, 10%, and 1.98%. Our design has 100 design points.

For comparison purposes, consider the crossing approach. For the categorical factors, a full factorial would require $2^5 \times 3^2 \times 5^5 = 900,000$ design points. No OAs capable of handling all 12 categorical factors are available in the online libraries of orthogonal arrays (Sloane 2007, Kuhfeld 2010), although one can be constructed by crossing two smaller designs—one that handles up to 20 2-level factors and two 3-level factors in 36 runs, the other that handles up to six 5-level factors in 25 runs. For the continuous factor design, our rule of thumb suggests that between 45 and 150 design points is reasonable, although it is possible to use as few as 16 design points. Crossing these designs yields overall design matrices with the number of design points ranging from $36 \times 25 \times 16 = 14,400$ for the smallest design (that someone familiar with OAs could construct) up to $900,000 \times 150 = 135,000,000$ design points (if a full factorial is used). With only 100 design points, our design is much more efficient.

4.2 Second Design

Cizek (2010) studies the launching of UAVs (Unmanned Aerial Vehicles) from submarines. UAVs provide the submarine with a more detailed tactical picture of the battlefield. The study aims to analyze how UAV capabilities affect a submarine's ability to accomplish a maritime interdiction mission.

We created a design that mixed all type of factors: categorical, discrete and continuous. The UAV/submarine simulation model has four categorical factors with 2, 3, 3 and 3 levels; four discrete factors with 8, 11, 21 and 41 levels; and 37 continuous factors. The $\boldsymbol{D}$-optimality, maximum imbalance value, and ρ_{map} of the design are, respectively, 99.97%, 10%, and 0.94%, respectively. Our design has 468 design points.

There are no suitable OAs readily available for the categorical factors, even though this is a much smaller problem than the first example. There is a design capable of handling up to four 3-level factors in 9 design points; by doubling the number of design points, we can accommodate the single 2-level factor as well. If we decide to treat the 8-level discrete factor as categorical, we need $9 \times 2 \times 8 = 144$ design points for the "qualitative" factor design. The size of a full factorial for these five factors is 432. OAs are not available for 11-, 21-, or 41- level factors, so without the MIP formulation the analyst would probably treat these as continuous factors, appropriately rounded (although the 11-level factor could be treated as categorical). With 39–40 "continuous" factors, sizes of overall designs obtained by crossing would range from $144 \times 40 = 5,760$ (treating the 11-, 21- and 41- level discrete factors as continuous) to

$432 \times 11 \times 39 = 185,328$ design points (treating the 21- and 41- level discrete factors as continuous), or over 1.5 billion design points (using factorials for all categorical and discrete factors). Once again, our design is more efficient by several orders of magnitude.

5 CONCLUSIONS

We proposed a mixed integer programming formulation that, for a (given) limited number of design points n, generates a design which is nearly orthogonal and also nearly balanced for any mix of factor types (categorical, numerical discrete and numerical continuous) and/or number of factor levels. Our proposal can be used to create designs with low maximum absolute pairwise correlation, low imbalance level, and high $\boldsymbol{D}$-optimality for simulation problems with any type of factors. The designs we construct require orders of magnitude fewer design points than other approaches.

These new designs greatly expand the portfolio of designs available for analysts conducting large-scale simulation experiments. Consequently, there are much greater opportunities for gaining insights about the behavior of complex simulation models (and the real-world situations they represent) in a timely manner.

Interesting problems for future research involve the study of high-order aliasing (e.g., aliasing of main effects and interactions) and how our MIP formulation might be expanded to diminish adverse alias effects. A related topic is that of explicitly incorporating space-filling requirements into our MIP formulation.

REFERENCES

Bathke, A. 2007. The ANOVA F test can still be used in some balanced designs with unequal variances and nonnormal data. *Journal of Statistical Planning and Inference* 49 (2): 413–422.

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

Cizek, R. 2010. *Determining the important features for a submarine-launched uav*. M.S. thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California.

Fisher, R. A. 2000. *Statistical methods for research workers*. Edinburgh, Scotland: Oliver and Boyd.

Hedayat, A. S., N. J. Sloane, and J. Stufken. 1999. *Orthogonal arrays: Theory and applications*. New York: Springer-Verlag.

Hernandez, A. S., T. W. Lucas, and M. Carlyle. 2011. Enabling nearly orthogonal latin hypercube construction for any non-saturated run-variable combination. Working paper, Department of Operations Research, Naval Postgraduate School.

Kleijnen, J. P. C. 2007. *Design and analysis of simulation experiments*. New York: Springer.

Kleijnen, J. P. C., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa. 2005. A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing* 17 (3): 263–289.

Kuhfeld, W. F. 2010. Orthogonal arrays. http://support.sas.com/techsup/technote/ts723.html.

Law, A. M. 2007. *Simulation Modeling and Analysis*. 4^{th} ed. New York: McGraw-Hill.

Montgomery, D. 2005. *Design and Analysis of Experiments*. 6^{th} ed. Hoboken, New Jersey: Wiley.

Ryan, T. P. 2008. *Modern regression methods*. 2^{nd} ed. Hoboken, New Jersey: Wiley.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. Design and analysis of computer experiments (includes comments and rejoinder). *Statistical Science* 4:409–435.

Sanchez, S. M., T. W. Lucas, P. J. Sanchez, C. J. Nannini, and H. Wan. 2011. Designing large-scale simulation experiments, with applications to defense and homeland security. In *Design and Analysis of Experiments, Vol. 3: Special Designs and Applications*, ed. K. Hinkelmann, Chapter 12. Hoboken, New Jersey: Wiley.

Sanchez, S. M., and H. Wan. 2009. Better than a petaflop: The power of efficient experimental design. In *Proceedings of the 2009 Winter Simulation Conference*, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 60–74. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Santner, T. J., B. J. Williams, and W. I. Notz. 2003. *The design and analysis of computer experiments*. New York: Springer-Verlag.

SAS Institute, I. 2005. JMP©8.0 help - nominal factors.

Sloane, N. J. 2007. Orthogonal arrays. http://www2.research.att.com/ njas/oadir/.

Vieira Junior, H., S. M. Sanchez, K. H. Kienitz, and M. C. N. Belderrain. 2011a. Efficient nearly orthogonal nearly balanced mixed designs. Working paper, Department of Operations Research, Naval Postgraduate School, Monterey, CA.

Vieira Junior, H., S. M. Sanchez, K. H. Kienitz, and M. C. N. Belderrain. 2011b. Generating and improving orthogonal designs by using mixed integer programming. *European Journal of Operational Research* 215 (3): 629–638.

AUTHOR BIOGRAPHIES

HÉLCIO VIEIRA, JUNIOR, a Lieutenant Colonel in the Brazilian Air Force, is a PhD candidate at the Technological Institute of Aeronautics in São José dos Campos, Brazil. He earned a Bachelors in Aeronautics Science at the Brazilian Air Force Academy and a Master's degree in Operations Research from the Federal University of Rio de Janeiro, RJ, Brazil. His research interest is in the area of Design of Experiments, Simulation Optimization, and Discrete-Event Simulation applied to military problems. His email address is junior_hv@yahoo.com.br.

SUSAN M. SANCHEZ is a Professor of Operations Research at the Naval Postgraduate School, where she holds a joint appointment in the Graduate School of Business & Public Policy and serves as a Co-director of the SEED Center for Data Farming. Her research interests include experimental design for simulation studies, data-intensive statistics, and robust selection. She has a Ph.D. in Operations Research from Cornell University. She is currently the ASA representative to the WSC Board of Directors, and Chair of the Design of Experiments Subgroup of NATO's *MSG-088: Data Farming in Support of NATO*. Her e-mail is ssanchez@nps.edu and her web address is http://faculty.nps.edu/smsanche.

KARL HEINZ KIENITZ was born in São Paulo, Brazil. He currently is an Associate Professor of Systems and Control at Technological Institute of Aeronautics (ITA), São José dos Campos, Brazil. He holds B.S. and M.E.E. degrees from ITA and a PhD degree in Electrical Engineering from the Swiss Federal Institute of Technology Zürich. His main research interests are in optimization, decision and robust control for aerospace applications. Karl is a Senior Member of the IEEE. His email address is kienitz@ita.br.

MISCHEL CARMEN NEYRA BELDERRAIN is a Professor at Technological Institute of Aeronautics, São José dos Campos, Brazil. She earned a bachelor in Operations Research, from Universidad Nacional Mayor de San Marcos; a Master in Computer Science – Optimization, from Federal University of Rio de Janeiro, RJ, Brazil; and Ph.D. in Operations Research from Technological Institute of Aeronautics. Her research interests include the following subjects: Multicriteria Decision Making Methods, Problem Structuring Methods (PSM) and OR Education. Her email address is carmen@ita.br.