Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

WORK SMARTER, NOT HARDER: A TUTORIAL ON DESIGNING AND CONDUCTING SIMULATION EXPERIMENTS

Susan M. Sanchez
Paul J. Sánchez

Operations Research Department
Naval Postgraduate School
Monterey, CA 93943, USA

Hong Wan

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907, USA

ABSTRACT

Simulation models are integral to modern scientific research, national defense, industry and manufacturing, and in public policy debates. These models tend to be extremely complex, often with thousands of factors and many sources of uncertainty. To understand the impact of these factors and their interactions on model outcomes requires efficient, high-dimensional design of experiments. Unfortunately, all too often, many large-scale simulation models continue to be explored in ad hoc ways. This suggests that more simulation researchers and practitioners need to be aware of the power of designed experiments in order to get the most from their simulation studies. In this tutorial, we demonstrate the basic concepts important for designing and conducting simulation experiments, and provide references to other resources for those wishing to learn more. This tutorial (an update of previous WSC tutorials) will prepare you to make your next simulation study a simulation experiment.

1 INTRODUCTION

In June 2008, a new supercomputer called the "Roadrunner" was unveiled. This bank of machines was assembled from components originally designed for the video game industry; it cost $133 milion, and is capable of doing a petaflop (a quadrillion operations per second). The New York Times coverage stated that "*petaflop machines like Roadrunner have the potential to fundamentally alter science and engineering*" by allowing researchers to "*ask questions and receive answers virtually interactively*" and "*perform experiments that would previously have been impractical*" (Markoff 2008). Yet let us take a closer look at the practicality of a brute-force approach. Suppose a simulation has 100 factors, each factor has two levels (low and high) of interest, and we decide to look at all combinations of these 100 factors. A single replication of this experiment would take over 40 million years on the Roadrunner, even if each of the $2^{100} \approx 10^{30}$ simulation runs consisted of a single machine instruction! A decade later, the world's most powerful supercomputer is the Summit, with an astounding 200 petaflop capacity (Simonite 2018). Nonetheless, even the Summit would require over 178 millenia to perform 2^{100} machine instructions – let alone 2^{100} simulation runs!

Efficient design of experiments can break this curse of dimensionality at a tiny fraction of the cost. For example, suppose we want to study 100 factors and all their two-way interactions. One screening design we could use (a resolution 5 fractional factorial, described in Section 3.3) specifies 32768 specific combinations of the factor levels to evaluate. How quickly can we finish such an experiment? On a desktop computer with a simulation that takes a full second to run, each replication of this experiment takes under 9.5 hours; even if the simulation takes a more reasonable one minute to run, we can finish this experiment on an 8-core desktop (under $3000) in 2.85 days. Other designs are even more efficient, and provide more detailed insights into the simulation model's behavior.

The field called Design of Experiments (DOE) has been around for a long time. Many of the classic experiment designs can be used in simulation studies. We discuss a few in this paper to explain the concepts

and motivate the use of experiment design. However, the settings in which real-world experiments are performed can be quite different from the simulation environment – therefore, a framework specifically geared toward simulation experiments is beneficial.

Before undertaking a simulation experiment, it is useful to think about *why* this experiment is needed. Simulation analysts and their clients might seek to (i) *develop a basic understanding* of a particular simulation model or system, (ii) *find robust* decisions or policies, or (iii) *compare the merits* of various decisions or policies. The goal will influence the way the study should be conducted (Kleijnen et al. 2005).

We focus on setting up single-stage experiments to address the first goal, and touch briefly on the second. Although the examples in this paper are very simple simulation models, the same types of designs have been extremely useful for investigating more complex simulation models in a variety of application areas. For a detailed discussion of the philosophy and tactics of simulation experiments, a more extensive catalog of potential designs, and a comprehensive list of references, see Kleijnen et al. (2005) or Sanchez et al. (2012); other useful references are Kleijnen (2017), or Chapter 12 of Law (2014).

The benefits of designed experiments are tremendous. Once you realize how much insight and information can be obtained in a relatively short amount of time from a well-designed experiment, DOE should become a regular part of the way you approach your simulation projects.

2 BASIC CONCEPTS

2.1 Definitions and Notation

One of the first things an experimenter or tester must do to design a good experiment is identify the experiment's factors. In DOE parlance, *factors* are the input (or independent) variables that are thought might have some impact on *responses* (i.e., experimental outputs). In general, an experiment might have many factors, each of which might assume a variety of values, called *levels* of the factor in DOE. A primary goal of many DOEs is to identify which of the factors are really important for which responses, and which are not and can thus be dropped from further consideration, greatly reducing the experimental effort and simplifying the task of interpreting the results. Also, of the important factors, we would like to identify the nature of the impact on the responses (e.g., increasing, linear, quadratic), and whether the levels of some factors influence the effects that other factors have (called *factor interactions*).

To identify appropriate designs, it is often useful to classify the factors along several dimensions:

- *Quantitative* or *qualitative*. Quantitative factors naturally take on numerical values, while qualitative factors do not (though they might be assigned numerically coded values).
- *Discrete* or *continuous* (quantitative factors only). Discrete factors can have levels only at certain separated values; an example would be the number of x-ray machines in a hospital, which would have to be a non-negative integer, presumably with some upper bound. Continuous factors can assume any real value, perhaps within some range, such as the speed at which a vehicle is operated.
- *Binary* or not. Binary factors are naturally constrained to just two levels, like the classification of a part as either defective or non-defective. Non-binary factors could take on more than two values, but might still be tested at only two levels, typically "low" and "high," or might be allowed to assume (many) more than two levels in the experiment.
- *Controllable* or *uncontrollable*. In a simulation experiment, all factors are manipulated and controlled, but in reality factors might be controllable or not. For example, the degree or nature of enemy jamming of a communications system would be controlled in a simulation, but not in an actual fight. This can affect how the experimenter interprets the estimates of the effects of factors.

Throughout this paper, *simulation model* denotes any model that is evaluated using a computer. Simulation models come in many flavors. There are deterministic simulations (e.g., numerical solutions of differential equations, where the same set of inputs always produces the same output) and stochastic simulations (where the same set of simulation inputs may produce different output unless the random-

number streams are carefully controlled). Simulations that model a process that occurs over time can also be characterized as terminating or non-terminating, depending on the stopping conditions. For ease of presentation we assume that terminating simulations are used; the simulation stops after either a pre-specified amount of simulation time has elapsed, or when a specific event or condition occurs.

Mathematically, let $X_1, \ldots, X_k$ denote the k factors in our experiment, and let Y denote a response of interest. Sometimes, graphical methods are the best way to gain insight about the Y's, but often we are interested in constructing *response surface metamodels* that approximate the relationships between the factors and the responses with statistical models. Regression metamodels are one class that is typically used (see, e.g., Barton 2015; Kleijnen 2017; Law 2014; or Sanchez et al. 2012).

First, suppose that the X_i's are all quantitative, although they can be discrete or continuous. A *main-effects metamodel* means we assume

$$Y = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \varepsilon, \tag{1}$$

where the ε's are independent random errors with mean zero. Ordinary least-squares regression assumes that the ε's in (1) are also identically distributed, but the regression coefficients are still unbiased estimators of the β_i even if the underlying variance is not constant.

To explore any quadratic effects, we will include terms like X_1^2 as potential explanatory variables for Y. Similarly, two-way interactions are terms like $X_1 X_2$. A *second-order metamodel* includes quadratic effects and two-way interactions, although it is best for numerical stability to fit this after centering the quadratic and interaction terms, as in (2):

$$Y = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{k} \beta_{i,i} (X_i - \overline{X}_i)^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{i,j} (X_i - \overline{X}_i)(X_j - \overline{X}_j) + \varepsilon. \tag{2}$$

Some statistical packages do this centering automatically. It is worth noting that regression can also be used when some of the X's are qualitative – in fact, the ANOVA (analysis of variance) technique commonly used for experiment designs with qualitative X's is a special case of regression.

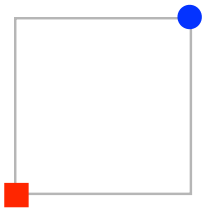
A *design* is a matrix where every column corresponds to a factor, and the entries within the column are settings for this factor. Each row represents a particular combination of factor levels, and is called a *design point*. If the row entries correspond to the actual settings that will be used, these are called *natural levels*. Coding the levels in a standardized way is a convenient way to characterize a design. Different codes are possible, but for quantitative data the low and high levels are often coded as -1 and $+1$, respectively, for arithmetic convenience (an example for a particular design is shown later). In this paper, each repetition of the whole design matrix is called a *replication* and we generally assume that the replications are independent. Let n_d be the number of design points, and n_r be the number of replications. The total number of experimental units is $n_{tot} = n_d n_r$.

We remark that the terminology can be slightly different, depending on what source you read or what software you use. For example, many sources dealing with design experiments in non-simulation settings refer to a combination of factor settings as a run. We prefer design point, because the term "run" has a specific meaning to simulators – and a single run can be associated with many (nearly) independent observations if, e.g., we use batch means to investigate steady-state simulations. Similarly, some designs allow the number of replications to differ by design point – but they may not count the original observations as replications! In these situations, a classical statistician might describe a Monte Carlo simulation with 10 runs at each of two parameter settings as a 2-run experiment with 9 replicates, or a 2-run experiment with 10 replicates. Despite differences in wording, the basic concepts remain the same – the design is a matrix, the design points tell us what factor level combinations to run, and replication provides information about the response variability at the design points.

2.2 Pitfalls to Avoid

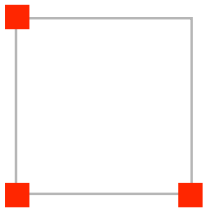
Two common types of simulation studies are ill-designed experiments. The first can occur if several people each suggest an "interesting" combination of factor settings, so a handful of design points end up being explored where many levels change simultaneously. Consider an agent-based simulation model of the child's game, where two teams (blue and red) each try to "capture the flag" of the opposition. Suppose that only two design points are used, corresponding to different settings for the speed (X_1) and stealth (X_2) of the blue team, with the results in Figure 1a. (Instead of providing numerical response values, a blue circle is used to represent a "good" average outcome for the blue team, while a red square represents a "bad" average outcome.) One person might claim these results show that high stealth is of primary importance, another that speed is the key to success, and a third that they are equally important. There is *no way* to resolve these differences of opinion without collecting more data. In statistical terms, the effects of stealth and speed are said to be *confounded*. In practice, simulation models easily have dozens or hundreds of potential factors. A handful of haphazardly chosen scenarios, or a trial-and-error approach, can use up a great deal of time without addressing the fundamental questions.

Speed	Stealth	Success?
Low	Low	No
High	High	Yes



(a)

Speed	Stealth	Success?
Low	Low	No
High	Low	No
Low	High	No



(b)

Figure 1: Two poor designs for capture-the-flag: (a) confounded effects, and (b) one-at-a-time sampling.

The second type of study that can be problematic occurs when people start with a "baseline" scenario and vary one factor at a time. Revisiting the capture-the-flag example, suppose the baseline corresponds to low stealth and low speed. Varying each factor, in turn, to its high level yields the results of Figure 1b. It appears that *neither* factor is important, so someone using the simulation results to decide how to choose a team would not know how (or if) to proceed. But combining the results of Figure 1a and b, it is clear that success requires both high speed and high stealth. This means that the factors interact – and if there are interactions, one-at-a-time sampling will never uncover them!

The pitfalls of using a poor design seem obvious on this toy problem, but the same mistakes are made far too often in larger studies of more complex models. When only a few variations from a baseline are conducted, there may be many factors that change but a few that decision makers think are "key." If they are mistaken, changes in performance from the baseline scenario may be attributed to the wrong factors. Similarly, many analysts change one factor at a time from their baseline scenario, but fail to understand that this approach implicitly assumes that there are no interaction effects. This assumption may be unreasonable unless the region of exploration is very small.

Another pitfall to avoid is more subtle. The statistical DOE literature focuses, in large part, on comparing designs in terms of the number of design points or the precision of specific factor effect estimates (e.g., main effects) based on assumed response behavior. This means there is a tendency to limit the investigation to a very small number of factors and limit the number of levels for each factor. This mindset is counterproductive for simulation experiments, particularly given the availability of computing clusters and the relative time required to create (vs. run) the model. It is better to gather enough data, via larger designs and more than one replication, to be able to explore the simulation's performance without resorting to lots of simplifying assumptions or relying on series of small experiments that may need to be back-tracked.

2.3 Choosing Factors

Potential factors in simulation experiments include the *input parameters* or *distributional parameters* of a simulation model. For example, a simulation model of a repair facility might have both quantitative factors (such as the number of mechanics of different types, or the mean time for a particular task) and qualitative factors (such as priority rules).

Generating a list of the potential inputs to a simulation model is one way of coming up with an initial factor list. However, factors need not correspond directly to simulation inputs. For example, suppose two inputs are the mean times μ_1 and μ_2 required for a specific agent to process messages from class 1 and class 2, respectively, where message class 2 is considered more complex than message class 1. Varying μ_1 and μ_2 independently may either result in unrealistic situations where $\mu_1 > \mu_2$, or require the analyst to select narrow factor ranges. Instead, we could use μ_1 as one factor to represent the capabilities of the agent, and vary the ratio μ_2/μ_1 over a range of interesting values (say, 1.1 to 2.0) to represent the relative difference in message complexity. Of course, the factor values have to be changed to come up with the inputs to our simulation, but this is a straightforward task. This idea may, in fact, represent factors more intuitively. For the standard M/M/1 queueing example, characterizing queue behavior as a function of arrival rate and traffic intensity may be more informative than as a function of arrival rate and service rate – because we know that the queue is non-stationary when the traffic intensity $\rho \geq 1$.

2.4 Sample-Size Issues

In live experiments, where data are extremely expensive, the total sample size is often very small. This affects the choice of an experiment design as well as the number of replications.

In simulation experiments, where a major portion of the effort often occurs in model development, the total sampling budget may not be so constrained. This increases the set of potential designs that can be used, and it may be possible to generate a great deal of information (even hundreds of thousands of runs) in a relatively short time. We discuss this further in Section 3.

2.5 Non-terminating Simulations

Different types of simulation studies involve different types of *experimental units*. For a static Monte Carlo simulation, where no aspect of time is involved, the experimental unit is a single observation. For time-stepped or discrete-event stochastic simulation studies, it more often is a run or a batch, yielding an averaged or aggregated output value. When runs form the experimental units for non-terminating simulations, and steady-state performance measures are of interest, care must be taken to delete data during the simulation's warm-up period before performing the averaging or aggregation. Details may be found in any simulation textbook, such as Law (2014) or Kelton et al. (2011).

3 POTENTIAL EXPERIMENT DESIGNS

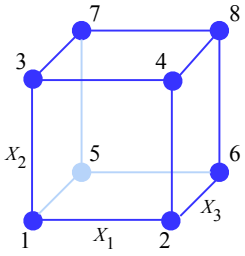
Many designs are available in the literature. We focus on a few basic types that we have found particularly useful for simulation experiments. Factorial or gridded designs are straightforward to construct and readily explainable – even to those without statistical backgrounds. Coarse grids (2^k factorials) are most efficient if we can assume that the simulation response is well-fit by a model with only linear main effects and interactions, while fine grids (more than two levels for factors) provide greater detail about the response and greater flexibility for constructing metamodels of the responses. When the number of factors is large, then more efficient designs are required. We have found Latin hypercubes to be good general-purpose designs for exploring complex simulation models when little is known about the response surfaces. Two-level designs called *resolution 5 fractional factorials* (R5-FFs) allow us to investigate the linear main effects and interactions of many factors simultaneously; they are potential choices either when factors have only two qualitative settings, or when practical considerations dictate that only a few levels be used for quantitative

input factors. Expanding these R5-FFs to central composite designs provides some information about nonlinear behavior in simulation response surfaces.

Factorials (or gridded designs) are perhaps the easiest to discuss: they examine all possible combinations of the factor levels for each of the X_i's. A shorthand notation for the design is m^k, which means k factors are investigated, at m levels for each factor, in a total of m^k design points. *Crossed designs*, where different sets of factors are investigated at different numbers of levels are written as, e.g., $m_1^{k_1} \times m_2^{k_2}$, where k_1 factors are evaluated at m_1 levels each, and another k_2 factors are evaluated at m_2 levels each.

3.1 2^k Factorial Designs (Coarse Grids)

The simplest factorial design is a 2^k because it requires only two levels for each factor. These can be low and high, often denoted -1 and $+1$ (or $-$ and $+$). 2^k designs are very easy to construct. Start by calculating the number of design points $N = 2^k$. The first column alternates -1 and $+1$, the second column alternates -1 and $+1$ in groups of 2, the third column alternates in groups of 4, and so forth by powers of 2. Conceptually, 2^k factorial designs sample at the corners of a hypercube defined by the factors' low and high settings. The left of Figure 2 shows an example for a 2^3 design. Envisioning a 2^4 or larger design is left to the hyperimaginative reader.



Design Point	Factor settings X_1	X_2	X_3	Additional analysis terms X_1X_2	X_1X_3	X_2X_3	$X_1X_2X_3$
1	-1	-1	-1	+1	+1	+1	-1
2	+1	-1	-1	-1	-1	+1	+1
3	-1	+1	-1	-1	+1	-1	+1
4	+1	+1	-1	+1	-1	-1	-1
5	-1	-1	+1	+1	-1	-1	+1
6	+1	-1	+1	-1	+1	-1	-1
7	-1	+1	+1	-1	-1	+1	-1
8	+1	+1	+1	+1	+1	+1	+1

Figure 2: 2^3 factorial design, graphically and in matrix form, with numbered design points.

Factorial designs have several nice properties. They let us examine more than one factor at a time, so they can be used to identify important interaction effects. They are also *orthogonal* designs: the pairwise correlation between any two columns (factors) is equal to zero. This simplifies the analysis of the output (Y's) we get from running our experiment, because estimates of the factors' effects ($\hat{\beta}_i$'s) and their contribution to the explanatory power (R^2) of the regression metamodel will not depend on what other explanatory terms are present in the regression metamodel. From Figure 2, the design itself is specified by the first three columns, but there are seven different terms (three main effects, three two-way interactions, and one three-way interaction) that we could consider estimating from a 2^3 factorial experiment. The columns for the interactions are calculated by simply multiplying the appropriate factor columns. However, since we also want to estimate the intercept (overall mean), that means there are eight things we could try to estimate from eight data points. That will not work – we will always need at least one degree of freedom (d.f.) for estimating error (and preferably, a few more) to estimate the metamodel for a model of the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 + \beta_{123} X_1 X_2 X_3 + \varepsilon. \quad (3)$$

A similar relationship holds as we increase the number of factors k.

So, what do people do with a factorial design? One possibility is to *replicate* the design to get more d.f. for error. Estimating eight effects from eight observations (experimental units) is not possible, but estimating eight effects from 16 observations is simple. Replication also makes it easier to detect smaller effects by reducing the underlying standard errors associated with the estimates of the β's. In simulation experiments replication is quite important for another reason: the response variability can differ dramatically across design points, and understanding the behavior of the response variability may be as important (or more important) than understanding the behavior of the response mean.

Another option is to *make simplifying assumptions.* The most common approach is to assume that some higher-order interactions do not exist. In the 2^3 factorial of Figure 2, one d.f. would be available for estimating error if the three-way interaction could safely be ignored. We could then fit a second-order regression model to the results. Similarly, if we have data for a single replication of a 2^4 factorial design but can assume there is no four-way interaction we have one d.f. for error; if we can assume there are no three-way or four-way interactions, we have five d.f. for error estimation. Making simplifying assumptions sounds dangerous, but it can be a good approach. Over the years, statisticians conducting field experiments have found that often, if there are interactions present, the main effects also show up unless you "just happen" to set the low and high levels so the effects cancel. There is also a rule of thumb stating that the magnitudes of two-way interactions are at most about 1/3 the size of main effects, and the magnitudes of three-way interactions are at most about 1/3 the size of the two-way interactions, etc. Whether or not this holds for experiments on simulations of complex systems is not yet certain – we may expect to find stronger interactions in a simulation of a supply chain or humanitarian assistance operations than when growing potatoes. Consequently, we advocate checking (at a minimum) for second-order interactions.

3.2 m^k Factorial Designs (Finer Grids)

Examining each of the factors at only two levels (the low and high values of interest) means we have no idea how the simulation behaves for factor combinations in the interior of the experimental region. Finer grids can reveal complexities in the landscape. When each factor has three levels, the convention is to use -1, 0 and 1 (or $-$, 0, and $+$) for the coded levels. Consider the capture-the-flag example once more. Figure 3 shows the (notional) results of two experiments: a 2^2 factorial (on the left) and an 11^2 factorial (on the right). For the 2^2 factorial, all that can be said is that when speed and stealth are both high, the agent is successful. Much more information is conveyed by the 11^2 factorial: here we see that if the agent can achieve a minimal level of stealth, then speed is more important. In both subgraphs the blue circles – including the upper right-hand corner – represent good results, the tan triangles in the middle represent mixed results, and the red squares on the left-hand side and bottom represent poor results.

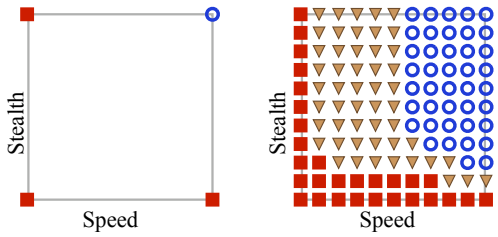


Figure 3: 2^2 and 11^2 factorial designs for capture-the-flag.

When we study more than two factors, a scatterplot matrix of the design points is a useful graph for visualizing the design – it shows the projections of the full design onto each pair of factors. Consider the left-most graph in Figure 4 for a 2^4 factorial. This graph contains cells of subplots of the design points for pairs of factors at a time. For instance, the third cell over in the top row plots the (X_3, X_1) factor combinations; the third cell down in the left column is just its transpose, plotting the pairs (X_1, X_3), so carries the same information. The second graph in Figure 4 contains the scatterplot matrix for a 4^4 factorial. Note that all subgraphs in Figure 4 just show dots corresponding to the factor settings; they do not show any color-coding for the responses.

The larger the value of m for an m^k factorial design, the better its space-filling properties. Yet despite the greater detail provided, and the ease of interpreting the results, fine grids are not suitable for more than a handful of factors because of their massive data requirements. A 2^{20} requires n_d over one million, a 5^{10} requires $n_d > 9.7$ million, and a 10^{10} factorial requires 10 billion design points.

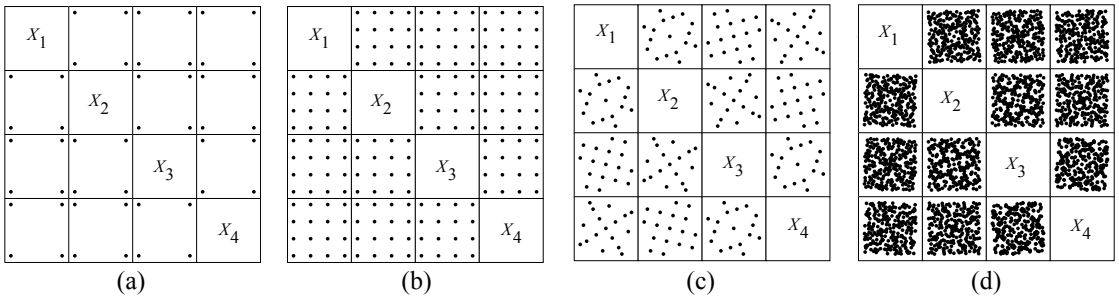


Figure 4: Scatterplot matrices for selected factorial and nearly orthogonal Latin hypercube (NOLH) designs: (a) 2^4 factorial with 16 design points, (b) 4^4 factorial with 256 design points, (c) NOLH with 17 design points, and (d) NOLH with 257 design points.

Considering the number of high-order interactions we *could* fit but may not believe are important (relative to main effects and two-way or possibly three-way interactions), this seems like a lot of wasted effort. It means we need *smarter, more efficient* types of experimental designs if we are interested in exploring many factors.

3.3 2^{k-p} Resolution 5 Fractional Factorial and Central Composite Designs

Sometimes, many factors take on only a few levels. In these cases, we can consider variations of gridded designs. If we are willing to assume that some high-order interactions are not important, we can cut down (perhaps dramatically) the number of runs required to examine a fixed number of factors using a *fractional factorial* design.

Graphically, these sample at a carefully-chosen fraction of the corner points on the hypercube. The left-most cube in Figure 5 shows the sampling for a 2^{3-1} factorial design, i.e., investigating three factors, each at two levels, in only $2^{3-1} = 4$ runs. There are two points on each of the left and right faces of the cube, and each of these faces has one instance of X_2 at each level and one instance of X_3 at each level, so we can isolate the effect for factor X_1. Similarly, averaging the results for the top and bottom faces allows us to estimate the effect for factor X_2, and averaging the results for the front and back faces allows us to estimate the effect for factor X_3.

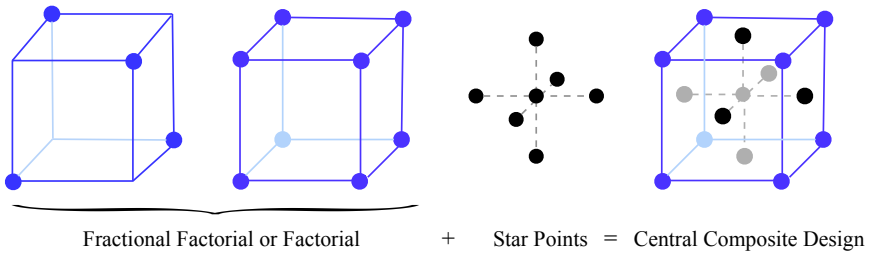


Figure 5: Examples of fractional factorial and central composite designs.

Saturated or nearly-saturated fractional factorials are often called *screening designs* because they can be useful for eliminating factors that are unimportant (though they will not do a good job of revealing the underlying structure of the response surface if there truly are strong interactions but we ignore them when setting up the experiment.) They are very efficient (relative to full factorial designs) when there are many factors. For example, 64 runs could be used for a single replication of a design involving 63 factors, or two replications of a design involving 32 factors. Screening designs that enable estimates of only main

effects are called resolution 3 fractional factorials (R3-FFs); designs that provide valid estimates of main effects in the presence of two-way interactions (without allowing the analyst to estimate the interaction effects) are called resolution 4 fractional factorials (R4-FFs). More recently, Xing et al. (2013) propose analysis-method-directed supersaturated designs for high-dimensional screening experiments.

Resolution 5 fractional factorials (R5-FFs) allow all main effects and two-way interactions to be fit, and may be more useful for simulation analysts than saturated or nearly-saturated designs. Sanchez and Sanchez (2005) developed a method, based on discrete-valued Walsh functions, for rapidly constructing very large R5-FFs – a short program generates designs up to a $2^{120-105}$ in under a minute. The gains in efficiency (as compared to full factorials) are dramatic enough to be worth mentioning again: running a 2^{100} full factorial would require over 40 million years on the world's fastest supercomputer in 2009, while a R5-FF requires only $2^{100-85} = 32768$ design points.

For quantitative factors, an R5-FF can be extended to a *central composite design* (CCD) that lets the analyst estimate all full second-order models (i.e., main effects, two-way interactions, and quadratic effects). Start with a 2^k factorial or R5 2^{k-p} fractional factorial design. Add a center point and two "star points" for each of the factors. In the coded designs, if -1 and $+1$ are the low and high levels, respectively, then the center point occurs at $(0,0,...,0)$, the first pair of star points are $(-c,0,...,0)$ and $(c,0,...,0)$; the second pair of star points are $(0,-c,0,...,0)$ and $(0,+c,0,...,0)$, and so on. If $c = 1$ the star points fall on the face of the cube, but other values of c can be used. A graphical depiction of a CCD for $k = 3$ appears in Figure 5. Using the efficient R5-FFs of Sanchez and Sanchez (2005) as the base designs, a CCD for 10 factors requires 152 design points, while a 3^{10} factorial requires over 59000 design points. Once again, it is clear that a brute force (full factorial) approach is impossible when k is even moderately large, but efficient designs make experimentation both practical and informative.

3.4 Space-filling Designs

Latin hypercube (LH) designs are quite flexible and efficient for quantitative factors. They have some of the space-filling properties of factorial designs with fine grids, but require orders of magnitude less sampling. Once again, let k denote the number of factors, and let $n_d \geq k$ denote the number of design points. Every column of the LH design is a permutation of the n_d equally-spaced factor levels. Figure 6 lists a random LH with $k = 2$ and $m = 11$, and provides a picture of results that might arise by using this experiment design for our capture-the-flag simulation. Compare this design to those of Figure 3. Unlike the 2^2 factorial design, the LH design provides some information about what happens in the center of the experimental region, but requires far less effort than the 11^2 factorial design.

DP	Speed	Stealth	DP	Speed	Stealth
1	1	11	7	10	1
2	3	5	8	4	2
3	7	7	9	11	8
4	2	3	10	8	9
5	5	10	11	9	6
6	6	4			

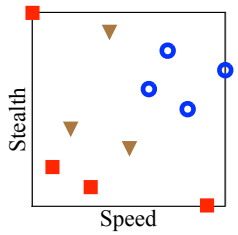


Figure 6: Random Latin hypercube for capture-the-flag.

Random LH's may not work well unless $n_d >> k$, but other LH designs are available. Minimax or maximin LH's have good space-filling properties, and algorithms or packages for constructing them can be found in most statistical software packages. Cioppa and Lucas (2007) construct *nearly orthogonal Latin hypercube* (NOLH) designs that have good space-filling and orthogonality properties for small or moderate k ($k \leq 29$). Portions of two of their designs are shown in Figure 4c and d: an NOLH design with 17

design points, and an NOLH design with 257 design points. The two-dimensional space-filling behavior of the NOLH compares favorably with that of the 4^4 factorial for roughly 1/15 the computational effort, so experimenters concerned about the level of computational effort might prefer the latter. Alternatively, experimenters considering the use of the 4^4 factorial (and thus willing to run 256 design points) might prefer the NOLH with 257 design points (just one more) – and gain the ability to examine a much denser set of factor-level combinations, as well as explore up to 25 additional factors using the same design! The benefits of LH sampling are greatest for large k. Assuming that a single design point takes one second to run, each replication of a 29-factor experiment would take under five minutes using an NOLH design, but over 17 years using a 2^{29} factorial design. More recently, Hernandez et al. (2012) use a mixed integer programming approach to generate sets of Latin hypercubes that are saturated or nearly-saturated. These extend Latin hypercube designs for simulation studies with larger numbers of factors. MacCalman et al. (2018) develop NOLH designs that permit the estimation of full second-order response models when the number of factors is relatively small.

Replicating the design allows us to determine whether or not a constant error variance is a reasonable characterization of the simulation's performance, and is highly recommended. If we have the time and budget for even more sampling, then two or more different Latin hypercubes or NOLHs can be *stacked* to obtain a larger design with better space-filling properties. Stacking two designs means running both sets of design points; one way to obtain two different designs from the same NOLH matrix is to reassign the factors to different columns of the experiment design matrix.

When discrete-valued factors with limited numbers of levels are present, then rounded NOLH designs may no longer retain their near-orthogonal properties. The nearly orthogonal-and-balanced (abbreviated NOB or NOAB) mixed designs of Vieira et al. (2013) are suitable in these situations. One general-purpose design with $n_d = 512$ allows for simultaneously investigating up to 300 factors: 20 each with discrete numbers of levels m ($m = 2, 3, \ldots, 11$) and 100 continuous-valued factors.

3.5 Robust Design Methods

A distinction can be made between decision factors that can be controlled in the real world, and noise factors that cannot be controlled during actual operations. For example, in a simulation of search-and-rescue operations after a natural disaster, the decision factors might include the communication systems, available equipment, or number of people on the rescue team. Noise factors might include weather conditions, the number and location of those in need of rescue, and the skill levels of the emergency medical technicians. An alternative to an exploratory analysis that seeks to understand how these noise factors affect the responses is a *robust design* approach, where the goal of the experiment(s) is to identify design points that yield good performance across the range of noise factor settings – in other words, to identify *robust* systems, rather than systems that are effective only against specific threat and environmental conditions – particularly if these correspond to the most favorable settings for threat and environmental factors. The robust design philosophy was pioneered by Taguchi (1987) for manufactured-product design, where it has been successfully used to achieve high-quality products while keeping costs in line; it also facilitates the evaluation of trade-offs between quality and cost. An important consideration for the simulation community is that the robust design philosophy explicitly requires analysts to consider variances, as well as means, in assessing system performance.

The classification of factors as either decision or noise factors may affect the choice of the design. Generally, we are interested in fitting metamodels that explain the relationship between the decision factors (and their interactions, etc.) and the response. Interactions among noise factors may affect the variability of the response but are not of direct interest, while (noise factor)$\times$(decision factor) interactions show up as unequal response variances across different decision-factor combinations.

Applying robust design principles to simulation experiments is discussed in Sanchez (2000). A more detailed discussion and examples appear in Kleijnen et al. (2005), where *identifying robust systems and processes* is considered one of three primary goals of simulation experiments.

4 DESIGN RECOMMENDATIONS

Selecting a design can be an art, as well as a science. Clearly, the number of factors and the mix of different factor types (binary, qualitative or discrete with a limited number of levels, discrete with many levels, or continuous) play important roles. But these are rarely cast in stone – particularly during exploratory analysis. The experimenter has control over how factors are grouped, how levels are determined, etc. Even if these are specified, different experimenters may prefer different designs.

Having said that, we have strong preferences for large-scale designs that exhibit good space-filling behavior. This is a "big data" view of simulation experiments (Sanchez 2015, Sanchez and Sanchez 2017) – and so our advice may be fundamentally different than what you would find from a statistician used to working with physical experiments. Fortunately, while generating some of the designs suitable for large-scale (rather than small-scale) simulation experiments may have required a lot of effort, that does not mean they are any more difficult to use. For example, a spreadsheet for the NOAB design of Vieira et al. (2013) discussed in Section 3.4 is available online (https:\harvest.nps.edu), and can readily be used to create custom designs involving up to 300 factors by simply filling in low and high values, along with the decimals required, for each factor.

5 GAINING INSIGHT

Design of experiment approaches, coupled with analytic and graphical methods such as response-surface methodology and data-mining techniques, can be useful for all the goals mentioned in section 1. They help the experimenter develop a better understanding of the simulation, and in turn help develop a better understanding of the system. Insights gained from simulation experiments can be used in many ways. Results can be used to evaluate or improve the simulation model. By identifying important factors, interactions, and nonlinear effects, the experimenter can improve their understanding of the simulation's behavior, find robust solutions, or raise questions to be explored in subsequent experiments. Thresholds, plateaus, or other interesting features of the response surfaces might provide guidance about situations that are particularly good (or particularly bad).

For classical design of experiments, some statisticians recommend deciding on the metamodel type before selecting an experiment design. For example, Barton (2015) recommends that analysts interested in using regression metamodels should decide on the desired polynomial order (typically, first-order or second-order) before selecting the design. However, we now provide some examples to highlight the vast difference in insights that can be obtained if you do not unnecessarily restrict yourself to small designs, but instead take a large-scale, data farming view where, from the outset, you opt for space-filling designs.

If a crystal ball revealed the metamodel form (e.g., first-order or second-order polynomial) before we began our experiment, then a space-filling design and a so-called 'optimal' design for that metamodel will both provide insights about the important factors, interactions, and quadratic effects – as long as we have enough data to determine statistically significant effects. One difference between using a space-filling design and a minimally-sufficient factorial-based design is that we automatically get indications of lack-of-fit if the responses turn out to be more complicated. While some might see this at first as a drawback, because the residuals may not fit the regression model assumptions, we see this as a benefit. Let us go back to the capture-the-flag example – where we have added a rather complicated underlying response surface that has flat areas, areas of abrupt transition, and with smooth changes – and suppose we use a 65-dp NOLH to explore the space. Figure 7(a) shows a contour plot of the actual response based on an 11×11 grid. The regression metamodel in (b) looks quite similar to that which would be obtained using the 2^{11} factorial data – both have a hard time capturing threshold effects. The partition tree metamodel in (c) has a hard time capturing diagonal transitions (more on partition trees is coming soon). The Gaussian process (or kriging) and hybrid regression/partition metamodels in (d) and (e) appear closer to the actual surface, although the Gaussian process does less well around some edges of the plot. All of the metamodels in (b)–(e) give much more information than the 2^2 factorial experiment in Figure 3.

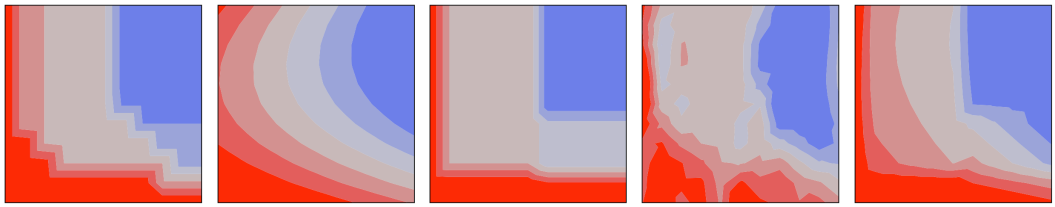


Figure 7: Capture-the-flag contour plots. (a) displays the actual response based on an 11×11 grid. The remaining contours are metamodels following a 65-dp NOLH: (b) 2nd-order regression metamodel, (c) partition tree with five splits, (d) Gaussian process metamodel, and (e) regression/partition metamodel.

Sometimes graphs are enough to provide insight. Figure 8 shows some results for the well-known deterministic combat model of Dewar et al. (1996). Here, only two factors are varied: the reinforcement block size for both Red and Blue forces take on 201 distinct levels. The graph in Figure 8(a) shows the results of a 4^2 full factorial design, while that in Figure 8(b) shows the results of a 201^2 full factorial design (Vinyard and Lucas 2002, Sanchez and Lucas 2002), where the shaded blue results represent a win for Blue. Neither regression, logistic regression, or kriging models on the small design in Figure 8(a) reveal the pervasive regions of chaotic non-monotonicity in the response surface. At the time the runs for Figure 8(b) were made in 2001, they required eight hours of CPU time on a single Pentium III PC – a computational effort well worth the additional insights. With current computers, these runs would require mere minutes of computation. Of course, using a space-filling design would be a much more efficient way to reveal regions of complex or chaotic behavior, and allow us to investigate a larger number factors.

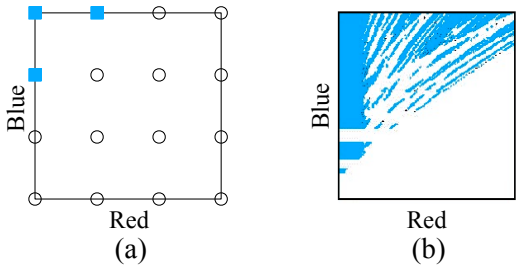


Figure 8: Results of (a) 4^2 and (b) 201^2 full factorial designs for a deterministic combat model.

The previous two examples have focused on a small number of factors. For a large number of factors, it is less likely that a single, simple metamodel form is appropriate for all responses of interest – consequently, it is even more beneficial to use flexible, space-filling designs. A partition tree (also called a classification and regression tree), mentioned earlier, is a nonparametric metamodeling technique that we find useful for both screening and model-fitting purposes. It starts with all the data in a single group, then searches among all factors to find the binary partitioning of a factor that splits the data into two groups resulting in the largest improvement in R^2. This process can be repeated until the analyst deems the marginal improvement is unimportant. We illustrate this with some results from a fleet management simulation experiment involving 50 replications of an experiment with 19 factors and 1040 design points (Marlow et al. 2015). Of interest are two responses related to the ashore flight hours per "tail" of a fleet of Naval helicopters: Y_{avg} is a measure of the average, and Y_{spread} is a measure of the spread. In Figure 9(a), the first two splits of the partition tree for Y_{spread} reveal that the tail rotation heuristic used is most important (the ideal Y_{spread} is zero). A more detailed metamodel for Y_{spread} includes two other decision factors. The metamodel for Y_{avg}, where high values are desired, is dominated by noise factors. The scatter plot of these two responses in Figure 9(b) shows that while many dps are capable of achieving high Y_{avg}, there are vast differences in their corresponding Y_{spread}. Alternatives in the lower right are robust.

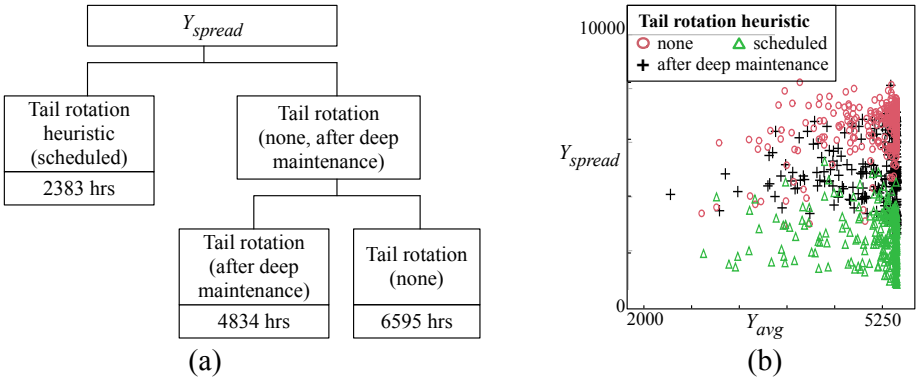


Figure 9: Robust design results for ashore hours per tail from a fleet management simulation: (a) first two splits of a partition tree for Y_{spread}, and (b) scatterplot of two responses.

6 FINDING OUT MORE

For more on the philosophy and tactics of designing simulation experiments, examples of graphical methods that facilitate gaining insight into the simulation model's performance, and extensive literature surveys, we refer the reader to Sanchez et al. (2012) or Kleijnen et al. (2005). Books that discuss experiment designs for simulation include Santner et al. (2003), Law (2014), and Kleijnen (2017). For experiments where it is very time-consuming to run a single replication, there are other single-stage designs (often used for physical experiments) that require fewer runs than fractional factorial designs. Some of these designs appear in the above references; others can be found in experiment design texts such as Box et al. (2005) or Ryan (2007).

Finally, the benefits of efficient experiment designs are often more tangible if you see how they are used in practice. Designs like the ones described in this paper have assisted the U.S. military and several allied countries in a series of international data farming workshops. Interdisciplinary teams of officers and analysts develop and explore agent-based simulation models to address questions of current interest to the U.S. military and allies, such as network-centric operations, effective use of unmanned vehicles, peace support operations, and more. Sanchez and Lucas (2002) provide an overview of issues in modeling and analysis aspects of agent-based simulation. A humanitarian assistance scenario is discussed in Kleijnen et al. (2005). Lucas et al. (2007) describe several defense and homeland security applications: critical infrastructure protection, non-lethal capabilities in a maritime environment, and emergency first response to a crisis event. The website of the SEED Center for Data Farming (https:\harvest.nps.edu) also has links to many papers, design spreadsheets and software, and over 200 student theses covering a wide range of applications (SEED Center for Data Farming 2018). These provide more details about statistical and graphical analysis of the simulation results, along with implementation issues regarding leveraging high-performance computing assets.

7 CONCLUSIONS

The process of building, verifying, and validating a simulation model can be arduous – but once complete, then it is time to have the model work for you. One extremely effective way of accomplishing this is to use designed experiments to help explore your simulation model. This tutorial has touched on a few designs that we have found particularly useful, but other design and analysis techniques exist. Our intent was to open your eyes to the benefits of DOE, and convince you to make your next simulation study a simulation *experiment*. As we have shown, if you are interested in exploring the behavior of a simulation model with more than a handful of input factors, efficient experiment designs are readily available – and help you to work smarter, rather than harder, in order to gain insights from your simulation.

ACKNOWLEDGMENTS

This is an update of previous tutorials, most recently Sanchez and Wan (2015). DoD Distribution Statement: Approved for public release; distribution is unlimited. The views expressed in this document are those of the authors and do not necessarily reflect the official policy or position of the DoD or the U.S. Government.

REFERENCES

Barton, R. R. 2015. "Tutorial: Simulation Metamodeling". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz et al., 1765–1779. Piscataway, New Jersey: IEEE.

Box, G. E. P., W. G. Hunter, and J. S. Hunter. 2005. *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. 2nd ed. New York: Wiley.

Cioppa, T. M., and T. W. Lucas. 2007. "Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes". *Technometrics* 49(1):45–55.

Dewar, J., J. Gillogly, and M. Juncosa. 1996. "Non-Monotonicity, Chaos, and Combat Models". *Military Operations Research* 2(2):37–49.

Hernandez, A. S., T. W. Lucas, and M. Carlyle. 2012. "Enabling Nearly Orthogonal Latin Hypercube Construction for any Non-Saturated Run-Variable Combination". *ACM Transactions on Modeling and Computer Simulation* 22(4):20:1–20:17.

Kelton, W. D., J. S. Smith, and D. Sturrock. 2011. *Simio and Simulation: Modeling, Analysis, Applications*. 2nd ed. New York: McGraw-Hill.

Kleijnen, J. P. C. 2017. "Regression and Kriging Metamodels With Their Experimental Designs in Simulation: A Review". *European Journal of Operational Research* 256:1–16.

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.

Law, A. M. 2014. *Simulation Modeling and Analysis*. 5th ed. New York: McGraw-Hill.

Lucas, T. W., S. M. Sanchez, F. Martinez, L. R. Sickinger, and J. W. Roginski. 2007. "Defense and Homeland Security Applications of Multi-agent Simulations". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson et al., 138–149. Piscataway, New Jersey: IEEE.

MacCalman, A. D., H. Vieira Jr, and T. W. Lucas. 2018. "Second-Order Nearly Orthogonal Latin Hypercubes for Exploring Stochastic Simulations". *Journal of Simulation,* 11(2):137–150.

Markoff, J. 2008. "Military Supercomputer Sets Record". *New York Times*, June 9.

Marlow, D., S. M. Sanchez, and P. J. Sanchez. 2015. "Testing Aircraft Fleet Management Policies Using Simulation Experimental Design". In *Proceedings of MODSIM2015, 21st International Congress on Modelling and Simulation*, edited by T. Weber et al., November 29$^{\text{th}}$–December 4$^{\text{th}}$, Broadbeach, Queensland, Australia, 917–923.

Ryan, T. P. 2007. *Modern Experimental Design*. Hoboken, New Jersey: Wiley.

Sanchez, S. M. 2000. "Robust Design: Seeking the Best of all Possible Worlds". In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines et al., 69–76. Piscataway, New Jersey: IEEE.

Sanchez, S. M. 2015. "Simulation Experiments: Better Data, Not Just Big Data". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz et al., 800–811. Piscataway, New Jersey: IEEE.

Sanchez, S. M., and T. W. Lucas. 2002. "Exploring the World of Agent-Based Simulation: Simple Models, Complex Analyses". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yucësan et al., 116–126. Piscataway, New Jersey: IEEE.

Sanchez, S. M., T. W. Lucas, P. J. Sanchez, C. J. Nannini, and H. Wan. 2012. "Designs for Large-Scale Simulation Experiments, with Applications to Defense and Homeland Security". In *Design and Analysis of Experiments, Volume 3: Special Designs and Applications*, edited by K. Hinkelmann, 413–441. New York: John Wiley & Sons.

Sanchez, S. M., and P. J. Sanchez. 2005. "Very Large Fractional Factorial and Central Composite Designs". *ACM Transactions on Modeling and Computer Simulation* 15(4):362–377.

Sanchez, S. M., and P. J. Sanchez. 2017. "Better Big Data via Data Farming Experiments". In *Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences*, edited by A. Tolk et al., 159–179. Cham, Switzerland: Springer International Publishing.

Sanchez, S. M., and H. Wan. 2015. "Work Smarter, Not Harder: A Tutorial on Designing and Conducting Simulation Experiments". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz et al., 1795–1809. Piscataway, New Jersey: IEEE.

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

SEED Center for Data Farming 2018. Available via http://harvest.nps.edu, accessed July 9, 2018.

Simonite, T. 2018. "The US Again Has the World's Most Powerful Supercomputer". *WIRED*. https://www.wired.com/story/the-us-again-has-worlds-most-powerful-supercomputer/?mbid=social_twitter_onsiteshare, accessed July 10, 2018.

Taguchi, G. 1987. *System of Experimental Design*, Volume 1 and 2. White Plains, New York: UNIPUB/Krauss International.

Vieira, H., S. M. Sanchez, K. H. K. Kienitz, and M. C. N. Belderrain. 2013. "Efficient, Nearly Orthogonal-and-Balanced, Mixed Designs: An Effective Way to Conduct Trade-off Analyses via Simulation". *Journal of Simulation* 7(4):264–275.

Vinyard, W., and T. W. Lucas. 2002. "Exploring Combat Models for Non-Monotonicities and Remedies". *PHALANX* 35(1):36–38.

Xing, D.-D., H. Wan, M. Y. Zhu, S. M. Sanchez, and T. Kaymal. 2013. "Simulation Screening Experiments using Lasso-optimal Supersaturated Design: A Maritime Operations Application". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy et al., 497–508. Piscataway, New Jersey: IEEE.

AUTHOR BIOGRAPHIES

SUSAN M. SANCHEZ is a Professor of Operations Research at the Naval Postgraduate School, and Co-Director of the Simulation Experiments & Efficient Designs (SEED) Center for Data Farming. She also holds a joint appointment in the Graduate School of Business & Public Policy. She has been an active member of the simulation community for many years, and has been recognized as a Titan of Simulation and an INFORMS Fellow. Her web page is http://faculty.nps.edu/smsanche/ and her email is ssanchez@nps.edu.

PAUL J. SÁNCHEZ is on the faculty of the Operations Research Department at the Naval Postgraduate School, and a member of the Simulation Experiments & Efficient Designs (SEED) Center for Data Farming. He has an SB in Economics from MIT, and MS and PhD degrees in Operations Research from Cornell University. His research interests include design of experiments, simulation output analysis, and object-oriented modeling. He actually enjoys programming. His web page is http://faculty.nps.edu/pjsanche/ and his email is pjsanche@nps.edu.

HONG WAN is an Associate Professor in the School of Industrial Engineering at Purdue University. Her research interests include design and analysis of simulation experiments; simulation optimization; simulation of manufacturing, healthcare and financial systems; quality control; and applied statistics. She has taught a variety of courses and is a member of INFORMS and ASA. She currently serves as the associate editor of ACM Transactions on Modeling and Computer Simulation. Her email address is hwan@purdue.edu and her web page is http://web.ics.purdue.edu/~hwan.