Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

SIMULATION EXPERIMENTS: BETTER DATA, NOT JUST BIG DATA

Susan M. Sanchez

Naval Postgraduate School
Operations Research Department
1411 Cunningham Rd.
Monterey, CA 93943-5219, USA

ABSTRACT

Data mining tools have been around for several decades, but the term "big data" has only recently captured widespread attention. Numerous success stories have been promulgated as organizations have sifted through massive volumes of data to find interesting patterns that are, in turn, transformed into actionable information. Yet a key drawback to the big data paradigm is that it relies on observational data—limiting the types of insights that can be gained. The simulation world is different. A "data farming" metaphor captures the notion of purposeful data generation from simulation models. Large-scale designed experiments let us grow the simulation output efficiently and effectively. We can explore massive input spaces, uncover interesting features of complex simulation response surfaces, and explicitly identify cause-and-effect relationships. With this new mindset, we can achieve quantum leaps in the breadth, depth, and timeliness of the insights yielded by simulation models.

1 INTRODUCTION

There is no universally accepted definition for the term "big data." Many have argued that we have always had big data, whenever we had more available than we could analyze in a timely fashion with existing tools. Running a single multiple regression in the 1940's was very difficult, since matrix multiplication and inversion for even a small problem (say, a handful of independent variables and a few hundred observations) is a non-trivial task when it must be done by hand. From that perspective, big data can be viewed as any data set that pushes against the limits of currently available technology.

What can be done with big data? Most people immediately think of data mining, a term that is ubiquitous in the literature. The concept of data farming is less well-known, but has been used in the defense community over the past decade. At the Simulation Experiments & Efficient Designs (SEED) Center for Data Farming at the Naval Postgraduate School, we describe the differences between these metaphors as follows.

Miners seek valuable nuggets of ore buried in the earth, but have no control over what's out there or how hard it is to extract the nuggets from their surroundings. As they take samples from the earth, they gather more information about the underlying geology. Similarly, data miners seek to uncover valuable nuggets of information buried within massive amounts of data. They may look at "all" the "big data" available at a particular point in time, but they typically have no control over what data are out there, nor how hard it will be to separate the useful information from the rest of the data. Data mining techniques use statistical and graphical measures to try to identify interesting correlations or clusters in the data set.

Farmers cultivate the land to maximize their yield. They manipulate the environment to their advantage, by using irrigation, pest control, crop rotation, fertilizer, and more. Small-scale designed experiments let them determine whether these treatments are effective. Similarly, data farmers manipulate simulation models to advantage—but using large-scale designed experimentation. This allows them to learn more about the simulation model's behavior in a structured way. In this fashion, they "grow" data from their models, but

in a manner that facilitates identifying the useful information. For large-scale simulation experiments, this often results in big data sets. Yet, although the data sets are big, they are far smaller than what would be needed to gain insights if the results were observational (i.e., obtained using *ad hoc* or randomly generated combinations of factor settings). The data is also better, because it lets us identify root cause-and-effect relationships between the simulation model input factors, and the simulation output.

We review issues of correlation and causation in Section 2. In Section 3, we describe the '3 V's' of big data, indicate how they carry over to the simulation world, and introduce the '3 F's' of data farming. Section 4 contains a bit more background on designing experiments. In Section 5 we provide an overview of some designs we have found particularly useful. We conclude with a brief discussion of future research.

2 REVISITING CAUSATION AND CORRELATION

The simplest way of establishing cause-and-effect is via an experiment. Let's suppose that we're looking at a stochastic system, so our observation of Y comes with some error. An experiment involving a single factor X involves changing X and observing whether or not there is a change in the response Y. We're trying to uncover the ground truth about the relationship between X and Y over a range of interest, as shown in (1). After collecting data, we will end up estimating this via some metamodels, as shown in (2). The subscripts on $\hat{g}$ and $\hat{\varepsilon}$ show these models were estimated from data obtained using design D.

$$Y = g(X) + \varepsilon(X) \tag{1}$$

$$\hat{Y} = \hat{g}_D(X) + \hat{\varepsilon}_D(X) \tag{2}$$

Three important concepts in the design of experiments are: *control*, *randomization*, and *replication*. For real-world experiments, you exercise control over the situation by deciding which values of X are of interest. You also decide how to control for everything else that isn't of interest, perhaps by holding it constant or using a control group for comparison purposes. Randomization is used to guard against hidden or uncontrollable sources of bias. For example, if you are measuring the miles per gallon of different vehicles by using a single driver, randomizing the order in which they are driven will remove any systematic bias due to fatigue. With replication you collect multiple observations to assess the magnitude of the variability associated with Y, so you can construct confidence intervals or conduct significance tests.

In simulation experiments, we use these concepts in different ways. In the simulation world, the analyst has total control. Potential factors in simulation experiments include the *input parameters* or *distributional parameters* of a simulation model, whether they are controllable in the real world or not. The analyst also has control over the random number seeds and streams. This means that, unlike in physical experiments where random noise occurs, the results from a simulation experiment are perfectly repeatable, and randomization is not needed to guard against hidden or uncontrollable sources of bias. Replication means you get multiple *experimental units* (runs or batches) to get a sense of the magnitude of the variability associated with Y.

There are certainly times when uncovering correlation is very useful. As Hogan (2014) points out, "Epidemiological studies demonstrated more than a half century ago a strong correlation between smoking and cancer. We still don't understand exactly *how* smoking causes cancer. The discovery of the correlation nonetheless led to anti-smoking campaigns, which have arguably done more to reduce cancer rates over the past few decades than all our advances in testing and treatment." Yet there are also many more times when identifying correlation is not nearly good enough. It's well-known that trolling through mountains of data can yield spurious correlations: an excellent discussion of how this affects published research results appears in The Economist (2013). If big data sets reveal a correlation between two variables X and Y, the ground truth can be one of four basic situations: (i) changes in X cause changes in Y; (ii) changes in Y cause changes in X; (iii) changes in X and Y are both the result of changes in other, potentially unknown or unobservable factors; or (iv) this is a spurious correlation. One drawback of big observational data is that we have no real way of testing these results without moving to a different environment. Common sense can help eliminate some spurious correlations (NPR 2014; Vigen 2014), but one of the tenets of big

data is that it is not repeatable. Correlations that appear strong at one point in time and weak at another might mean that the underlying system has changed.

In their recent book, Mayer-Schönberger and Cukier (2013) claim that "the need for sampling is an artifact of a period of information scarcity" and that big data "leads society to abandon its time-honored preference for causality, and in many instances tap the benefits of correlation." But if the goal of analysis is to yield better outcomes via controlling or influencing the inputs, what could be more important than causality? .

3 CHARACTERIZING BIGNESS

Why are so many either extolling or lamenting big data's focus on correlation rather than causation? I suggest it is because their view of big data is observational. This is not the case for those using simulation. Using the data farming metaphor, we grow the data for analysis, rather than mine existing data. However, just as a different mindset is needed for dealing with "big" observational data," so a different mindset is needed for generating and dealing with "big" simulation data.

3.1 The 3 (or more) V's of Big Data

Big data is typically characterized by "3 V's" — volume, velocity, and variety (Laney 2001). These have all increased at an astonishing rate during the last decade, with much of the data being generated, captured from, or stored on the internet. Volume refers to the amount of data, and this is enormous. As of June 2014, IBM states that "90% of the data in the world today has been created in the last two years alone" (IBM. 2014), but that may be out of date by the time these proceedings are published. Some have advocated a look at more V's (including veracity, validity, and volatility, viability, value, and victory), but others argue that these are derived from context-specific analytics while the original 3 V's capture the "intrinsic, definitional big data properties essence of big data" (Grimes 2013). Regardless, the nature of the V's determine the analysis tools (and decisions) that can be applied.

The 3 (or more) V's have a different flavor in simulation than in other big data situations. Velocity and volume are partially controlled by the analyst, who determines how to run the simulation (e.g., on a single core or on a high-performance computing cluster), how much data to output (e.g., aggregate statistics at the end-of-run, batch statistics, or full time-series output) for each performance measure, and the number of performance measures to study. Generated output can have a variety of types, but the variety does not include many of the problems that we find with observational data (e.g., incompatible data formats, inconsistent data semantics). In the near future I anticipate that many simulators may make use of big data tools for tying their models to real-time or near real-time data sets for model input (Elmegreen, Sanchez, and Szalay 2014). With regard to some of the "wanna V's", verification and validation have long been cornerstones of effective simulation practice. A structured V&V process means the simulation output should not suffer from lack of veracity. Of course, the question of validation—whether the simulation model's behavior is sufficiently close to that of the real-world problem of interest for decision-making purposes—is always with us.

3.2 The 3 F's of Data Farming

Moving beyond the 3 V's of observational big data, in the arena of large-scale simulation experiments we can focus on the 3 F's of inferential big data: *factors, features*, and *flexibility*. All these should be "big" when viewed with a simulation mindset.

Factors refers to a broad view of the inputs (or functions of inputs) that, if varied, affect the simulation and should be manipulated to increase our understanding of the simulation responses. A "big factor" view includes many aspects. Clearly, a large number of factors may be of interest: in fact, one can argue that if we didn't feel an input was important, then we would not have included it in our simulation model. But the big factor view also means that factors may vary over wide ranges, rather than limited ranges. They

may be of different types—qualitative, discrete, or continuous—rather than homogenous. Factors can be further broken down into decision factors that can be controlled in the real-world settings; noise factors that are difficult or impossible to control in reality, but can be controlled during the simulation experiment; and artificial (simulation-specific) factors—such as run length, warm-up period, batch size, or random number seed—that may not have an analogy in the real world, but can influence the way we conduct our simulation experiments and the results we obtain.

Features refers to the simulation responses. A "big feature" view includes many aspects. We may be interested in multiple responses, rather than limiting ourselves to a single one. For stochastic simulation models, our responses may have complex variance structures. It's worth mentioning that this characteristic, as pervasive and accepted as it is in simulation, is still the exception rather than the rule for traditional designed experiments. As with the factors, the responses may be of different types. We may be interested in short-term, transient behavior for one response, at the same time we're interested in the long-run quantiles of another response. We may also be searching for different types of features in the response surface landscapes. For example, in trade-off analyses, maxima and minima are often much less interesting than so-called "knees in the curve," where we find that further changes in one or more factors lead to diminishing (or increasing) returns. Other interesting features include thresholds where responses change suddenly, such as a model entity experiencing an abrupt shift from usually losing to usually winning; broad, flat regions that indicate we have a solution that is robust to uncertainties in the underlying factors over certain ranges; and Pareto sets containing those alternatives that cannot be completely dominated (in terms of desirable multivariate responses) by any other alternatives in this set.

Flexibility represents the fact that we may need to be able to answer many questions from our experiments, even if we don't know *a priori* all the questions that might be asked. A "big flexibility" view includes many aspects. Being able to choose among a broad variety of metamodeling, data mining, and graphical analysis tools is far more likely to yield success than picking a restrictive design that is only suitable for one particular approach or model. One way of gaining flexibility is to have a space-filling design: this term has typically been used for designs involving continuous-valued factors. The analogy for discrete-valued or qualitative factors is to have balanced or nearly-balanced designs. Note that in simulation, we can have additional flexibility built into how we store and retrieve our data. As long as we store the design points and random number seeds, we have the option of "growing" new data from our original experiment by rerunning our simulation and printing out additional input, if needed. For example, if we initially print out only end-of-run summaries and then find out that strange things have happened in a handful of the thousands of runs we've conducted, we can rerun that handful and inspect (or animate) the entire sequence of events to better understand what has transpired.

3.3 Fast Data Farming Environments

If inferential simulation data sets are already on hand, we can identify whether they are characterized by enough factors, features, and flexibility to address our questions. But if we are preparing to grow our data, then the timeliness of getting the results is also important.

First, let me state that efficient design of experiments is *absolutely required* for large-scale simulation experiments. In June 2008, a supercomputer called the "Roadrunner" was unveiled. It was assembled from components originally designed for the video game industry, it cost $133 million, and was capable of doing a petaflop (a quadrillion operations per second). The New York Times 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). Four years later, IBM's "Sequoia" supercomputer was the new world leader, with 16 petaflop capability. Yet let's 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 2.5 million years on the Sequoia, and over 40 million years on the Roadrunner, even if each

simulation run consisted of a single machine instruction! (Sanchez, Sanchez, and Wan 2014). Efficient design of experiments can break this curse of dimensionality at a tiny fraction of the hardware cost. Recent breakthroughs provide designs that can be used to explore 100 or more factors, for models that take a more reasonable minute to run, in times ranging from a few hours on a single processor, to a few minutes or days on a computing cluster. We will describe some useful designs in Section 5.1.

So, if you use designed experiments, is that enough to say that your turnaround time is fast? Not necessarily. For example, if changing the factor levels in your simulation model can only be accomplished through a GUI, then the bottleneck is often the analyst's time, rather than the computational time. Manually changing all the factor settings is a time-consuming and error-prone process. If you are facing a very short deadline, your only alternative may to use a small design and double-check or triple-check your input settings. But as we discuss in Section 5.3, setting up a data farming environment is extremely worthwhile. Automating the run generation process immediately expands your capability for growing data that doesn't suffer from input errors, and paves the way for running your model on a cluster.

Is a small design always preferable? For simulation experiments, the answer is a resounding 'No!' If your cluster has 1000 cores and your simulation takes a fixed amount of CPU time to complete, it will take no longer to conduct 1000 runs than to conduct a single run; if so, calling a design with 10 design points "faster" than one with 1000 designs points is both wrong and counterproductive! It is better to gather enough data, via larger designs and multiple replications, to be able to explore the simulation's performance without resorting to lots of simplifying assumptions. The bottom line is that a large-scale simulation is fast if you get the kind of data you need in time to act on it.

4 ADDITIONAL BACKGROUND AND MOTIVATION

The field of Design of Experiments (DOE) has been around for a long time. Many of the classic experimental designs could be used in simulation studies—but that is not our "big data" view. The context for real-world experiments can be much more constrained than for simulations in terms of costs, number of factors, time required, ability to replicate, ability to automate, etc., so a framework specifically oriented toward simulation experiments is beneficial. Before discussing useful designs, we provide a brief refresher on some DOE basics.

4.1 Definitions and Notation

One of the first things an analyst must do to design a good experiment is identify the factors. In DOE parlance, *factors* are the input (or independent) variables that are thought to potentially have some impact on *responses* (i.e., experimental outputs). In general, an experiment might have many factors, each of which might be assigned a variety of values, called *levels* of the factor in DOE.

To identify appropriate designs, it is often useful to classify the factors along several dimensions. Designs can be *quantitative* or *qualitative*. Quantitative factors naturally take on numerical values; qualitative factors do not (though they might be assigned numeric codings). Quantitative factors can be *discrete* or *continuous*. Discrete factors can have levels only at certain separated values such as non-negative integers, while continuous factors can assume any real value, perhaps within some range. Factors can be *binary* or *non-binary*. Binary factors are naturally constrained to just two levels; non-binary factors could take on more than two values, but might still be tested at only two levels, typically "low" and "high." Finally, it's often useful to classify factors as *controllable (decision)* or *uncontrollable (noise)*. Determining whether or not a factor is easily controllable in the real-world setting can affect how the analyst designs the experiment and interprets the results—even though all factors can be manipulated and controlled in a simulation experiment.

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). *Dynamic* simulations explicitly model the evolution of state over time—

shuffling the output order would destroy information—while *static* simulations generate replications without any implied ordering. Dynamic simulations can also be characterized as *terminating* or *non-terminating*, depending on whether the stopping conditions are clearly defined in terms of the state vs. whether we could legitimately continue running the model if we chose to do so.

Let $X_1, \ldots, X_k$ denote the k factors in our experiment, and $Y_1, \ldots, Y_p$ denote the responses of interest. Sometimes graphical methods are the best way to gain insight about the Y's, but often we are interested in constructing analytic *response surface metamodels* that approximate the relationships between the factors and the responses, typically in the form of regression models. We will assume that the reader is familiar with multiple regression techniques.

A *design* is a matrix where every column corresponds to a factor, and each row describes a particular combination of factor levels. Each unique combination of factor levels is called a *design point*. If the row entries correspond to the actual settings that will be used, these are called *natural levels*. Alternatively, a standardized coding of the levels is a convenient way to characterize a design. For quantitative data, the low and high levels are often encoded as -1 and $+1$, respectively, for arithmetic convenience. Let n_d be the number of design points. Each repetition of the entire design matrix is called a *replication*. We generally assume that the replications are performed independently.

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, more often it is an entire run or a batch of observations from within a run, yielding an averaged or aggregated output value. Since simulation programs are computer programs their state variables must be initialized, often to a convenient but non-representative state. If steady-state performance measures are of interest, you must allow the model to "warm up" before performing any averaging or aggregation. Details may be found in a simulation text, such as Law (2014) or Kelton, Smith, and Sturrock (2011).

4.2 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, but factors need not correspond directly to inputs. For example, suppose two inputs are the mean times μ_1 and μ_2 required for an agent to process messages from class 1 and class 2, respectively, where class 1 is considered more complex than class 2. Varying μ_1 and μ_2 independently may either result in unrealistic situations where $\mu_1 < \mu_2$, or require the analyst to use narrow factor ranges. Instead, we could use μ_1 as one factor to represent the capabilities of the agent, and vary the "traffic intensity" ratio μ_2/μ_1 over a range of interesting values to represent the relative difference in message complexity.

4.3 Setting Appropriate Goals

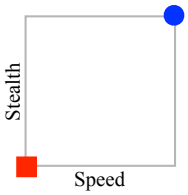
For over a decade, the SEED Center has advocated three basic goals of large-scale simulation experiments: (i) *developing a basic understanding* of a particular simulation model or system; (ii) *finding robust* decisions or policies; and (iii) *comparing the merits* of various decisions or policies (see, e.g., Sanchez and Lucas 2002; Kleijnen et al. 2005). We have stated that these require a different mindset than other goals that are often mentioned for computer experiments, namely: (i) constructing accurate *predictions*; (ii) *calibrating* the simulation to real-world data, or (iii) *optimizing* a system (see, e.g., Santner, Williams, and Notz 2003). Casting our results in a big data framework helps to clarify why we choose different goals: we believe that if simulation is used to model complex problems, it's unlikely that decision-makers will be interested only in the answer to a single, narrow question about that system. Developing an understanding—including

identifying important factors and interesting features—allows us to address a much richer range of questions. Similarly, we believe robust (resilient) alternatives found by seeking those that perform well over a variety of noise factor settings may be much more useful in practice than alternatives found by optimizing a single performance measure while implicitly holding noise factors constant. Finally, in trade-off analyses it may be much more useful to identify the existence of "knees in the curve," where we start to see increasing or decreasing rates of return, than to numerically predict exactly where these inflection points fall (Vieira Jr et al. 2013). The interest in more complex questions may increase even more rapidly in the future (Elmegreen, Sanchez, and Szalay 2014).

4.4 Pitfalls to Avoid

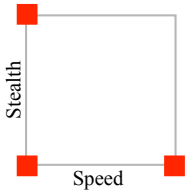
Once common pitfall of simulation studies is to perform scenario oriented experiments, where putting the focus on pre-selected "interesting" combinations of factor settings results in exploring a handful of design points where many factors are changed simultaneously. Consider an agent-based simulation 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. A blue circle indicates 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 at different design points. In statistical terms, the effects of stealth and speed are said to be *confounded*.

Speed	Stealth	Success?
Low	Low	No
High	High	Yes



a. Confounded effects

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



b. One-at-a-time sampling effects

Figure 1: Two poor designs for capture-the-flag.

The second type of problematic study 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 results to choose a team would not know how (or if) to proceed. But if we combine the results of Figures 1a and 1b, it is clear that success requires both high speed and high stealth—those two factors have an *interaction*. If there are interactions, one-at-a-time sampling will *never* uncover them.

The pitfalls of using a poor design seem obvious for this toy problem. Imagine how they are exacerbated for realistic simulation models with dozens, hundreds, or thousands of potential factors! This underscores the importance of growing big simulation data in a smart way, so that it will immediately be able to provide clearer pictures of the interesting features of the response surfaces, as well as useful insights about causal relationships between the factors and the responses.

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/or limit the number of levels for each factor. As discussed in Section 3.3, 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. We shouldn't forgo factors, features, and flexibility to focus exclusively on a myopic view of what makes an experiment fast.

5 USEFUL DATA FARMING TOOLS

How do we run simulation experiments that meet the 3 F's of factors, features, flexibility in a way that is sufficiently fast? First and foremost, we use a designed experiment capable of meeting these requirements given our computing resources and the time frame available for making the decision. We first provide an overview of some useful designs. We then briefly describe some potential analysis methods and implementation tips, and provide reference sources for finding out more detail.

5.1 Portfolio of Potential Designs for Large-scale Simulation Experiments

Sanchez and Wan (2012) created a "consumer report" chart that provides guidance to those interested in conducting large-scale simulation experiments. An updated version of this chart is kept on the SEED Center web pages at http://harvest.nps.edu. It characterizes designs in terms of their factors, features, and flexibility; gives notes with additional guidance; provides citations for the source papers; and highlights designs that we've found to be good starting points. We briefly describe some highlighted designs. For all designs we discuss, software or spreadsheets can be be freely downloaded.

If the experiment will investigate a mix of continuous-valued and discrete-valued factors, the nearly orthogonal-and-balanced (NOAB) designs of Vieira Jr et al. (2013) are quite flexible. Code for constructing these designs is available, though it can take a long time to find a good design for large problems. Consequently, we have preconstructed a NOAB with $n_d = 512$ that can be easily customized to handle up to $k = 300$ quantitative factors, including 100 continuous-valued factors and 20 discrete-valued factors with each of 2, 3, ..., 11 levels. Space-filling designs, like this one, are capable of revealing many features, and quite flexible in how they can be used for analysis.

If k is very large, and the factors are of mixed types, there may not be a single design available that provides enough flexibility. In this case, *crossing* two or more designs, where each provides settings for different sets of factors, is an option. Constructing separate designs for decision factors and for noise factors can also be useful, because it allows an "apples to apples" comparison of the results obtained at different decision factor design points without the use of metamodels. However, because n_d for a crossed design is the product of the individual n_d's, you may want to use relatively small individual designs. Smaller designs may also be desirable if the deadline is short, the simulation takes a long time to run, and you do not have access to a high-performance computing cluster. For continuous factors, nearly orthogonal Latin hypercube (NOLH) designs, such as those in Cioppa and Lucas (2007) (see also Joseph and Hung 2008) or the nearly saturated S-NOLH designs of Hernandez, Lucas, and Carlyle (2012), are a good option. These space-filling designs are suitable for identifying many and varied response features, are flexible enough to accommodate many different analysis approaches, and the near-orthogonality has some advantages for fitting regression metamodels. The NOLH designs are suitable for up to $k = 29$ factors in $n_d = 257$ design points. The spreadsheet of S-NOLH designs include more flexible combinations of k and n_d for $k \leq 63$, and GAMS code is available for constructing other S-NOLHs.

For binary factors, the very large resolution 5 fractional factorials (R5FF) of Sanchez and Sanchez (2005) are capable of simultaneously estimating all main-effects and all second-order interactions for up to 120 factors in $2^{120-105}$ design points. Extending these to central composite designs allows them to be used as screening designs for quantitative factors, and determine whether any factors have nonlinear effects. The software makes it possible to construct these second-order designs for up to 443 factors in $2^{443-423}$ design points (just over one million runs).

For studies involving thousands of factors, we currently advocate adaptive sequential screening designs, such as the FFCSB-X procedures of Sanchez, Wan, and Lucas (2009) or the Hybrid procedure of Shen, Wan, and Sanchez (2009). Both of these use group screening to efficiently identify important main effects without assuming that two-way interactions are negligible, and without requiring the analyst to have *a priori* knowledge of the factor effect signs. Once the number of factors has been reduced from a few thousand to a few hundred, further experiments can reveal interesting features of the response surface.

Those already familiar with DOE may notice that we omit many of the classic designs that are highlighted in DOE texts and used to advantage for small-scale experiments. We do so because either they require so many assumptions about the response surface that they sacrifice features and flexibility (e.g., only permit main-effect metamodels), they are intended for use with a handful of factors, or the design construction method is computational infeasible for large k.

5.2 Analysis Approaches

Clearly, the design you choose will impact the types of analyses you can conduct. One common type of metamodel involves polynomial models. We find the main effects model too restrictive, but a good starting point for a single response Y is often a model that can include second-order effects, such as quadratic terms and two-way interactions. For some designs, such as the CCDs of Sanchez and Sanchez (2005), it is possible to simultaneously estimate all first-order and second-order terms. Also, MacCalman, Vieira Jr, and Lucas (2014) constructed second-order Latin hypercubes for a modest number of factors ($k \leq 12$).

Alternatively, space-filling designs based on orthogonal or nearly-orthogonal Latin hypercubes are much more efficient, growing as $O(k)$ or $O(k^2)$, rather than the $O(2^k)$ required to simultaneously estimate all 2^k potential effects. A variety of polynomial metamodels can be fit, from first-order models, models with higher-order terms involving a subset of factors, up to an (albeit ridiculous) $(n_d - 1)$-degree polynomial involving a single factor. Stepwise regression or some other automated method can help determine the subset of terms that are most important. Note that in a big data world, many statistically significant terms may, nonetheless, be eliminated from the metamodel because they are not deemed to be of practical importance

A second approach we find to be quite useful is the data mining approach of partition trees, also called classification and regression trees (CART). Partition trees employ a binning and averaging process to successively split a large group of heterogeneous data into two smaller groups of data, where each leaf in a split is individually more homogeneous, but the difference between the leaves is large.

Another metamodeling approach of interest is kriging, or stochastic kriging. This metamodeling approach is quite flexible in terms of the model form; kriging has been heavily used for deterministic computer experiments, and also adapted for stochastic simulation experiments (see, e.g., van Beers and Kleijnen 2003; Ankenman, Nelson, and Staum 2010). However, the analysis approach is computationally quite intensive, so kriging model-fitting is typically conducted for experiments involving a small number of factors and a small number of design points. Adaptive sequential designs are often used, so that each new design point is chosen to reduce the uncertainty associate with model predictions at untested points as much as possible (Kleijnen and van Beers 2004). The community of those designing and analyzing deterministic computer experiments has thus focused on features and a one-at-a-time view of fastness, although some recent work in batch sequential designs (Loeppky et al. 2010; Duan et al. 2014), may provide additional flexibility for other metamodeling and analysis approaches if the data become too large for kriging software to handle. Brantley et al. (2013) describe sequential methods for allocating a fixed computing budget across design points to efficiently identify terms in polynomial metamodels.

Once the most important factors and interactions have been identified, many other statistical and graphical approaches are also extremely useful at conveying the information to the decision maker.

5.3 Automated Implementation

Even using good designs, the data farming approach includes automating the process of data collection whenever possible. Once you have chosen the design (or design algorithm), you should use a computing script to automatically run the experiments, allocate individual runs to distributed computing assets, and consolidate the output in a form suitable for analysis. This requires some programming expertise and may take a bit more time initially, but the payoff is worthwhile. Software that facilitates this automation for simulations with XML input files is available at the SEED Center web pages.

Another issue to consider when running large-scale experiments is the number of replications. For run-based experiments with long individual run times, it makes sense to iterate through replications of the entire design, rather than iterate through a large number of replications of design point 1, then design point 2, etc. This makes it more likely that you will have useful information if you stop the experiment early, and it allows the analyst to stop once they've seen enough.

5.4 Testing the Results: Another Reason Simulators Have Better Data

In large-scale simulation experiments, if we identify that some factor X belongs in a metamodel for some response Y, the ground truth can be one of four different situations: (i) changes in X (either individually, or in conjunction with other factors) cause changes in Y; (ii) changes in X are partially confounded with changes in high-order interactions of other factors; (iii) changes in X are fully confounded with changes in high-order interactions of other factors; or (iv) we have a false positive effect. Yet, in the simulation world, it is possible to explicitly test the results and determine whether or not our effects are truly important. Our choice of design determines the degree of confounding we are willing to accept *a priori*; this, in conjunction with model confirmation runs or secondary experiments, lets us test to see if the metamodel is sufficiently accurate at previously-untested regions in the factor space. If so, we can feel comfortable using the metamodels, in place of running additional simulations, for quick-turn analysis. If not, we can use new information and additional designs to enhance the metamodels until their performance is satisfactory.

5.5 Reality, and Back Again

All the techniques we've described allow us to get better understanding of our simulation model. In this way, we've found them very useful throughout the simulation process: during model development and model verification, as well as for "production runs." Yet it's important to remember that our experiments are providing us with information about the simulation model's behavior, not necessarily the behavior of the real-world system they attempt to emulate. In cases where it is possible to get some real-world data for validation purposes, we can use our simulation experiments to suggest real-world test cases that may be most informative or interesting. In other situations, where we do not have (and may hope never to have) real-world data, the ability to perform large-scale designed experiments allows us to examine a plethora of "what-if" cases in an efficient and effective manner.

6 FINDING OUT MORE

For more on the philosophy and tactics of designing large-scale 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 Kleijnen et al. (2005), Sanchez and Wan (2012), or Sanchez et al. (2012). Books that discuss DOE for computer or simulation models include Santner, Williams, and Notz (2003); Kleijnen (2007); and Law (2014). Note that their goals for those performing simulation experiments, and hence their design recommendations, may differ from those in this paper.

Software, spreadsheets, and other resources for a broad portfolio of designs, and a host of applications, are available at the SEED Center's website, <http://harvest.nps.edu>.

7 CONCLUDING REMARKS

Elsewhere in these proceedings, I describe some aspects of the intersection between big data, simulation, and decision making that will be of increasing interest in the near future (Elmegreen, Sanchez, and Szalay 2014). Future simulation clients will be interested in big problems that need to be modeled computationally, such as climate change, economics, transportation. They'll be interested in a broad set of questions related to these models, and the data science community will increase the number of decision makers who do not shy away from analytics, including the inferential big data from large-scale simulation experiments.

As we look to the future, our simulation community has an important role to play. A data farming approach—where we design experiments that handle big sets of factors, big sets of features, yet are flexible and fast—allows simulation researchers and practitioners to stake out the area of inferential big data. We have the chance to become recognized as the gold standard for model-based decision making within the big data analytics community, and we should seize this opportunity.

ACKNOWLEDGMENTS

This work was supported in part by the Naval Postgraduate School's Acquisition Research Program and U.S. Marine Corps Expeditionary Energy Office. Portions of this paper are adapted from Sanchez and Wan (2012) and Sanchez, Sanchez, and Wan (2014). Thanks to Paul Sanchez for helpful suggestions.

REFERENCES

Ankenman, B. E., B. L. Nelson, and J. C. Staum. 2010. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58:371–382.

Brantley, M. W., L. H. Lee, C.-H. Chen, and A. Chen. 2013. "Efficient Simulation Budget Allocation with Regression". *IIE Transactions* 45:291–308.

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

Duan, W., B. E. Ankenman, P. J. Sanchez, and S. M. Sanchez. 2014. "Sliced Full Factorial-Based Latin Hypercube Designs as a Framework for a Batch Sequential Design Algorithm". Working paper, Northwestern University, Dept. of Industrial Engineering and Management Sciences, Evanston, Illinois.

Elmegreen, B. E., S. M. Sanchez, and A. S. Szalay. 2014. "The Future of Computerized Decision Making". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.

Grimes, S. 2013, 7 August. "Big Data: Avoid 'Wanna V' Confusion". *Information Week*.

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.

Hogan, Joe 2014, June 9. "So Far, Big Data is Small Potatoes". Scientific American Blog Network. Available via http://blogs.scientificamerican.com/cross-check/2014/06/09/so-far-big-data-is-small-potatoes/.

IBM. 2014. "Big Data at the Speed of Business". Available via www-01.ibm.com/software/data/bigdata/what-is-big-data.html [accessed 20 June 2014].

Joseph, V. R., and Y. Hung. 2008. "Orthogonal-Maximin Latin Hypercube Designs". *Statistica Sinica* 18:171–186.

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. 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.

Kleijnen, J. P. C., and W. C. M. van Beers. 2004. "Application-Driven Sequential Designs for Simulation Experiments: Kriging Metamodeling". *Journal of the Operational Research Society* 55:876–883.

Laney, D. 2001, 6 February. "3D Data Management: Controlling Data Volume, Velocity, and Variety". In *Application Delivery Strategies*, Number 949: META Group Inc.

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

Loeppky, J. L., L. M. Moore, and B. M. Williams. 2010. "Batch Sequential Designs for Computer Experiments". *Journal of Statistical Planning and Inference* 140:1452–1464.

MacCalman, A. D., H. Vieira Jr, and T. W. Lucas. 2014. "Second Order Nearly Orthogonal Modified Latin Hypercubes for Exploring Models with Multiple Unknown Response Surface Forms". Working paper, Naval Postgraduate School, Monterey, California.

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

Mayer-Schönberger, V., and K. Cukier. 2013. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. 1st ed. New York: Houghton Mifflin Harcourt Publishing Company.

NPR. 2014, 17 May. "Drawing Phony Connections With Mismatched Metrics". Interview, Weekend Edition, NPR News. Available via http://www.npr.org/2014/05/17/313343183/drawing-phoney-connections-with-mismatched-metrics [accessed 20 June 2014].

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, C. Chen, J. L. Snowdon, and J. Charnes, 116–126. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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: Special Designs and Applications* (1st ed.), edited by K. Hinkelmann, Volume 3, Chapter 12, 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., P. J. Sanchez, and H. Wan. 2014. "Simulation Experiments: Better Insights by Design". In *Proceedings of the 2014 Summer Simulation Conference*. San Diego, California: The Society for Modeling & Simulation International.

Sanchez, S. M., and H. Wan. 2012. "Work Smarter, Not Harder: A Tutorial on Designing and Conducting Simulation Experiments". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Urmacher, 1929–1943. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.

Sanchez, S. M., H. Wan, and T. Lucas. 2009. "Two-Phase Screening Procedure for Simulation Experiments". *ACM Transactions on Modeling and Computer Simulation* 19 (2): 1–24.

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

Shen, H., H. Wan, and S. M. Sanchez. 2009. "A Hybrid Method for Simulation Factor Screening". *Naval Research Logistics* 57: 45–57.

The Economist. 2013, 19 October. *Unreliable Research: Trouble at the Lab*. The Economist.

van Beers, W. C. M., and J. P. C. Kleijnen. 2003. "Kriging for Interpolation in Random Simulation". *Journal of the Operational Research Society* 54:255–262.

Vieira Jr, 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.

Vigen, T. "Spurious Correlations". Available via <http://www.tylervigen.com> [accessed 20 June 2014].

AUTHOR BIOGRAPHY

SUSAN M. SANCHEZ is a Professor in 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 a B.S. in Industrial & Operations Engineering from the University of Michigan, and a Ph.D. in Operations Research from Cornell. She has long been active in within the simulation community, including the WSC Board of Directors. Her web page is http://faculty.nps.edu/smsanche/, and her email is ssanchez@nps.edu.