View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Calhoun, Institutional Archive of the Naval Postgraduate School

Calhoun
Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

Faculty and Researcher Publications Faculty and Researcher Publications

2005

# A user's guide to the brave new world of designing simulation experiment

Kleijnen, J.P.C.

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),

DUDLEY
KNOX
LIBRARY

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

*INFORMS Journal on Computing*
Vol. 17, No. 3, Summer 2005, pp. 263–289
ISSN 1091-9856 | EISSN 1526-5528 | 05 | 1703 | 0263

DOI 10.1287/ijoc.1050.0136

© 2005 INFORMS

## State-of-the-Art Review

# A User's Guide to the Brave New World of Designing Simulation Experiments

Jack P. C. Kleijnen
Department of Information Systems and Management/Center for Economic Research (CentER),
Tilburg University (UvT), Postbox 90153, 5000 LE Tilburg, The Netherlands, kleijnen@uvt.nl

Susan M. Sanchez
Operations Research Department and the Graduate School of Business and Public Policy,
Naval Postgraduate School, Monterey, California 93943-5219, USA, ssanchez@nps.edu

Thomas W. Lucas
Operations Research Department, Naval Postgraduate School, Monterey, California 93943-5219, USA,
twlucas@nps.edu

Thomas M. Cioppa
U.S. Army Training and Doctrine Command Analysis Center, Naval Postgraduate School,
PO Box 8692, Monterey, California 93943-0692, USA, thomas.cioppa@us.army.mil

Many simulation practitioners can get more from their analyses by using the statistical theory on design of experiments (DOE) developed specifically for exploring computer models. We discuss a toolkit of designs for simulators with limited DOE expertise who want to select a design and an appropriate analysis for their experiments. Furthermore, we provide a research agenda listing problems in the design of simulation experiments—as opposed to real-world experiments—that require more investigation. We consider three types of practical problems: (1) developing a basic understanding of a particular simulation model or system, (2) finding robust decisions or policies as opposed to so-called optimal solutions, and (3) comparing the merits of various decisions or policies. Our discussion emphasizes aspects that are typical for simulation, such as having many more factors than in real-world experiments, and the sequential nature of the data collection. Because the same problem type may be addressed through different design types, we discuss quality attributes of designs, such as the ease of design construction, the flexibility for analysis, and efficiency considerations. Moreover, the selection of the design type depends on the metamodel (response surface) that the analysts tentatively assume; for example, complicated metamodels require more simulation runs. We present several procedures to validate the metamodel estimated from a specific design, and we summarize a case study illustrating several of our major themes. We conclude with a discussion of areas that merit more work to achieve the potential benefits—either via new research or incorporation into standard simulation or statistical packages.

*Key words*: simulation; design of experiments; metamodels; Latin hypercube; sequential bifurcation; robust design
*History*: Accepted by W. David Kelton, Editor-in-Chief, acting as Area Editor; received January 2003; revised March 2004, August 2004, January 2005; accepted January 2005.

## 1. Introduction

Design of experiments (DOE) has a rich history, with many theoretical developments and practical applications in a variety of fields. Success stories abound in agriculture, clinical trials, industrial product design, and many other areas. Yet, despite the impact DOE has had on other fields and the wealth of experimental designs that appear in the literature, we feel DOE is not used as widely or effectively in the practice of simulation as it should be. We suggest several possible explanations for this.

One reason may be that few simulation analysts have been convinced of the benefits of DOE. Instead of using even a simple experimental design, many analysts end up making runs for only a single system specification, or they vary a handful of the many potential factors one at a time. Their efforts are focused on building—rather than analyzing—the model. DOE benefits include achieving gains (e.g., improving performance instead of a trial-and-error approach to finding a good solution) and avoiding losses (e.g., obtaining an "optimal" result with respect to one specific setting may lead to disastrous results when implemented). Unfortunately, few simulation practitioners seem to be aware of the additional insights that can be gleaned by effective use of designs.

A second reason may be that DOE research is often found in specialty journals seldom read by simulation analysts. Many results improve efficiency or guard against bias, whereas the bigger picture—namely, the setting for which this class of designs is most appropriate—may not be clear to an audience more familiar with simulation modeling.

The primary reason, in our opinion, is that most designs were originally developed for real-world experimentation and have been subsequently adapted for use in simulation studies, rather than developed specifically for simulation settings. Classic DOE textbooks (e.g., Box et al. 1978, Box and Draper 1987, Montgomery 2000, or Myers and Montgomery 2002) focus not on the needs of simulation analysts, but on the practical constraints and implementation issues when conducting real-world experiments. Comprehensive simulation textbooks (Law and Kelton 2000, Banks et al. 2005) cover a broad range of topics and provide detailed lists of references, but they demonstrate DOE by using it on a few simple test problems that do not stretch the reader's mental framework as to the depth and breadth of insights possible. Since DOE and general simulation books familiarize analysts with only a small subset of potential designs and applications, analysts are likely to force their problems to fit a particular design instead of identifying the design that best meets their needs.

Our goal is to bring together (i) a discussion of the issues that analysts *should* be aware of as they prepare to code, collect, and analyze output from a simulation model, and (ii) a guide for selecting appropriate designs. In particular, we contend that analysts must consider:

- the types of questions;
- characteristics of their simulation setting;
- characteristics of, and constraints imposed on, the simulation data collection and analysis; and
- the need to convey the results effectively.

These issues seem straightforward, but fundamental problems related to designing simulation experiments are all-too-often overlooked. We discuss these more fully later, focusing on the practical benefits of DOE. A design suited to a particular application is much better than trial and error or a simple, small design. Consequently, practitioners should be open to the notion that DOE is a useful and necessary part of analysis of complex simulation.

This is not a tutorial on the details for implementing specific designs, nor do we present a historical development of DOE and simulation. Instead, we provide an overview of the wide variety of situations that simulation analysts might face, the benefits and drawbacks of various designs in these contexts, and references. We want to change the mindset of simulation analysts and researchers to consider DOE as an integral part of any simulation project.

This overview is based on our experience through contacts with many simulation users and researchers over the last few decades. Where we disagree with current practice or theory, we present both sides. Despite the wide variety of designs that are available in the literature and, in some cases, statistical or simulation packages, we identify some situations where needs are still unmet. Our goal is to motivate more research to address these deficiencies.

In this paper we use concepts and terminology from simulation and statistics. Many readers may be familiar with simulation—and its DOE aspects—at the level of a textbook such as Law and Kelton (2000, p. xix), who state that a "second course in simulation for graduate students" should cover their Chapter 12 on "experimental design, sensitivity analysis, and optimization." For the reader familiar with only some of these ideas, we introduce brief definitions and explanations. For a refresher or an overview of simulation experiments, see Nakayama (2003) or Kelton and Barton (2003).

In §2 we describe how designing simulation experiments differs from designing experiments on real-world systems. Specifically, we address questions that simulation analysts or clients should ask. We also describe a number of other characteristics of simulation settings that cannot easily be handled through more traditional methods, and we provide examples to motivate the need for designs that cover a wide range of simulation settings. In §3 we discuss some characteristics of designs, including effectiveness. In §4 we describe several classes of designs, and assess their strengths and weaknesses for various simulation settings and their design characteristics. In §5 we describe ways to check the design's assumptions. In §6 we present a small case study, and in §7 we conclude with areas that merit additional work to achieve the potential benefits—either via new research or via incorporation into simulation or statistical software packages. An Online Supplement to this paper on the journal's website provides more details.

## 2. Why Is DOE for Simulation so Different?

We begin with some terminology. An *input* or a *parameter* in simulation is referred to as a *factor* in DOE. A factor can be either qualitative or quantitative. For example, in a queueing simulation, queue discipline can be either LIFO (last in, first out) or FIFO (first in, first out), so this is a qualitative factor. The number of servers is a discrete quantitative factor, while the rate for an exponential distribution used to model customer inter-arrival times is a continuous quantitative factor. Each factor can be set to two or more values, called *factor levels*, typically coded numerically for analysis purposes. A *scenario* or *design point* is a combination of levels for all factors. We consider stochastic simulations, so *replicates* mean that different pseudo-random numbers (PRNs) are used to simulate the same scenario. Unless otherwise specified, we assume that replicates use nonoverlapping PRN streams, so outputs across replicates are independently identically distributed (IID)—as most statistical methods assume. The output stream from a single replicate is a time series, which generally has auto-correlated observations; for example, the basic single-server model gives auto-correlated individual waiting times (if one customer must wait a long time, then the next customer is also apt to wait a long time).

Of course, the simulation is itself a model of some real-world (or prospective) system, process, or entity. We can view the simulation code as a *black box* that implicitly transforms inputs (such as factor-level settings and PRNs) into outputs. A *metamodel* (or *response surface*, *auxiliary model*, *emulator*, etc.) is a model or approximation of this implicit Input/Output (I/O) function that characterizes the relationship between inputs and outputs in much simpler terms than the full simulation. When a simulation experiment is conducted and most or all of the factors are quantitative, a common metamodeling technique is that of polynomial regression. We assume that the I/O relationship is composed of a deterministic (predictable) component, which is a polynomial function of the input factors, and a stochastic component that captures the (typically additive) error or randomness in the response. In a *first-order* or *main-effects* model, the deterministic component takes the form $f(X_1, X_2, \ldots, X_k) = \sum_{i=1}^{k} \beta_i X_i$, where the $X_i$ are the factor-level settings and the $\beta_i$ are estimated from the data. A *second-order* model could also include the quadratic effects $\sum_{i=1}^{k} \beta_{i,i} X_i^2$ and the two-way interactions $\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{i,j} X_i X_j$. Higher-order models might also be defined, but are harder to interpret.

We emphasize the following chicken-and-egg problem. Once a design is specified and simulated, metamodel parameters can be estimated. On the other hand, the types of metamodels that the analyst desires to investigate should guide the selection of an appropriate design.

DOE developed to generate and analyze data efficiently from real-world experimentation. In simulation, with its advances in computing power, we are not bound by some of the constraints that characterize real-world experiments. This is both an opportunity and a challenge. It is an opportunity to gain much more insight into how systems behave, and so provide assistance and information to decision makers that might differ dramatically (in terms of its quantity and nature) from that obtainable using more traditional methods. It is a challenge because it may require a new mindset. Indeed, we argue that the way simulation experiments should be approached is now fundamentally different from the way that real-world experiments involving, say, human subjects, should be approached.

To illustrate the difference between classic DOE and simulation DOE, consider the classic bias-minimizing designs. For example, Donohue et al. (1993), assuming a first-order metamodel but allowing for possible bias caused by second-order effects, derive designs that minimize that bias. We argue that such designs are relevant in real-world experiments but not in simulation. In the former cases, analysts must often select a design that is executed in "one shot" (say, one growing season in agriculture). In contrast, the data are collected sequentially in most simulation experiments, so analysts may start with a design for a first-order metamodel, then test (validate) the adequacy of that model, then augment their design to one that allows the estimation of second-order effects only if necessary (see, e.g., Sanchez et al. 1998, Kleijnen and Sargent 2000).

### 2.1. Asking Appropriate Questions

The importance of identifying the "right" problem before constructing a simulation and conducting the analysis is well known. For example, Law and Kelton (2000) state that the first step in a simulation study is to formulate the problem and plan the study. This step includes the project manager stating the problem of interest, and the analysts specifying the overall study objectives, specific questions to be answered, performance measures that will be used to evaluate the efficacy of different system configurations, system configurations to be modeled, and the time and resources required for the study. They go on to say that experimental design, sensitivity analysis, and optimization deal with situations in which there is "... less structure in the goal of the simulation study: we may want to find out which of possibly many parameters and structural assumptions have the greatest effect on a performance measure, or which set of model specifications appear to lead to optimal performance." (Law and Kelton 2000, p. 622).

We recommend an even broader view since we find that the most common type of question concerns an a priori single specific performance measure (typically a mean) that analysts then try to estimate or optimize. Instead, our starting point is a set of three basic goals that simulation analysts and their clients may have:

- *developing a basic understanding* of a particular simulation model or system,

- *finding robust* decisions or policies, and
- *comparing* the merits of various decisions or policies.

**2.1.1. Developing a Basic Understanding.** The first goal covers a wide range of questions. We use this phrase rather than "testing hypotheses about factor effects" for the following reason. At one extreme, we may develop a simulation to gain insight into situations where the underlying mechanisms are not well understood, and where real-world data are limited or even nonexistent. At the other extreme, we may perform a detailed analysis of a verified and validated simulation model.

As an example of the first situation, Dr. Alfred Brandstein posed the question "When and how should command and control be centralized or decentralized?" when he was Chief Scientist of the Marine Corps (Brandstein 1999). We do not know enough about the human mind to program a model for how decisions are really made by an individual—let alone a group of people! Yet, ignoring these types of questions because they are "too hard" or "inappropriate" for operations research is unacceptable. Our profession's roots are in finding ways to address difficult, interdisciplinary problems.

Addressing new problems often requires new simulation models. We find that making DOE an integral part of the model development process is useful in several ways. DOE can uncover detailed insight into the model's behavior, cause the modeling team to discuss in detail the implications of various model assumptions, help frame questions when the analysts may not know ahead of time what questions should be asked, challenge or confirm expectations about the direction and relative importance of factor effects, and even uncover problems in the program logic. These situations are seldom described in the literature, particularly as they relate to problems in programming logic or modeling assumptions. To illustrate these benefits, we provide some anecdotal evidence from recent workshops and related research on the use of agent-based models for military decision making (also see Horne and Leonardi 2001; Sanchez and Lucas 2002; Horne and Johnson 2002, 2003; Cioppa et al. 2004). Clearly, the benefits of incorporating DOE into the model-development process also apply to other types of simulation models.

Wan (2002) uncovers details about how a modeling platform behaves when simple terrain features are added. While familiarizing himself with the modeling platform, he sets up a skirmish within a corridor and uses simple experimental designs to generate data. Wan initially expects that "barriers" prohibit movement and provide protection from fire, as in earlier agent-based combat simulations. Instead, he uncovers instances where an enemy agent circles around the corridor and then exchanges fire with agents behind the front lines. Discussion with the software developer confirms that these barriers prohibit movement but not fire, behaving as ditches rather than walls. This is a low-probability event in Wan's eventual scenario and illustrates how DOE can uncover details about model behavior that might not be revealed without a broad-based investigation of factor effects.

Gill and Grieger (2003) run several experiments to examine movement rules in time-step, agent-based modeling platforms. Their results led to a discussion of the implications of how various platforms implement an agent's movement when, e.g., the agent's propensity for moving toward a goal is "twice as high" as its propensity for avoiding enemy contact. This can affect the development of new scenarios by directing modeling efforts toward choosing appropriate weights for these propensities in different contexts. Without this investigation, scenario developers and analysts might all use the same phrase to describe movement behavior, but have different internal views of its meaning.

When analysts may not know ahead of time what questions to ask, DOE can help. For example, analysis of a model of a skirmish involving guerrilla forces attacking a conventional force reveals that the most important determinants of losses on both sides are factors associated with the guerrillas' stealth and mobility (Lucas et al. 2003). The scenario was initially set up to explore the merits of conventional force tactics, movement, and squad strength, but the findings suggest that the defenders may gain more in terms of survivability and lethality by improving their ability to detect terrorists than by increasing their firepower.

Confirming prior expectations can be an important step in establishing face validity for simulation models, but it is also informative when the simulation provides insights that do not match expectations. In early investigations of a model of maneuver through an urban environment where experimentation was limited to investigations of five factors at a time (Lucas et al. 2002), the factors believed by a small group of subject-matter experts to be the most important turn out to have no statistically significant impact. Subsequent experiments provide new insights, indicating that the commanders' propensities to maneuver toward friendly agents and away from enemy agents are critical and that losses are reduced when the commander balances his drive toward the goal with the avoidance of enemy contact. An interaction term reveals that a strong bond between the commander and subordinates can mitigate the negative impacts of so-called *friction* on the battlefield: even if the subordinate agents cannot hear, comprehend, or otherwise act on their commander's orders, their losses are reduced if they stay with him.

Another major benefit of integrating DOE into the model-development process is the ability to uncover problems in the program logic. For example, Wolf (2002, p. 37) mentions anomalous results that, after investigation, are attributed to a modeling artifact. A simple experiment reveals that movement speeds do not behave as the analysts expect since increasing the factor setting from 0 to 1000 (corresponding to speeds of 0 to 10 squares per time step) does not monotonically increase speeds. For example, if the movement setting is 110, then the agent moves two steps 10% of the time but remains in place 90% of the time, for an average speed of only 0.1 squares per time step. Identification of this problem led to its modification in subsequent versions of the software, yet (Wolf 2002) it makes one wonder how often similar problems go undetected in models of complex scenarios.

The benefits of using experimental design during model development are likely to become even more substantial. If the ultimate decision maker needs rapid turnaround on the model development and analysis, this mandates using a modeling platform or reusing previously developed code to put together a model quickly. When code or modeling platforms are used or combined in new ways, some details of the modeling logic may be either poorly documented, or poorly understood by the user.

In exploratory environments like those above, it does not make sense to use the models to estimate factor effects numerically—we seek tendencies rather than values. At the other extreme, suppose we have a model that we are comfortable using for prediction. Then "understanding the system" may result from performing a detailed *sensitivity analysis* of a particular system configuration (i.e., examining the impact of small departures from this configuration). How should we proceed? Searching for effects by varying factors one at a time is an ineffective means of gaining understanding for all but the simplest systems. First, when using this approach it is impossible to identify any interaction effects between two or more factors, where positive interactions imply that factors complement each other, and negative interactions imply that factors are partial substitutes for each other. Second, even when interaction effects are negligible so one-factor-at-a-time sampling provides valid insights into I/O relationships, this can be proven to be an inefficient way to estimate the factor effects. From the outset, the analysts *must* explore factor effects concurrently to understand how their simulation model behaves when its factors are changed.

Between these two extremes of exploratory investigations and prediction generation are situations where we wish to identify a short list of important factors from the long initial list of potential factors. Depending on context, this might lead to a more thorough investigation of this short list via subsequent simulation experiments, a decision to forego adding enhancements or greater detail to aspects of the model that are found to be unimportant (at least over predetermined factor-level ranges), or collection of (additional) real-world data to home in on appropriate values of (say) influential input distributions. Alternatively, simply identifying the most influential factors (and their directional effects on performance) may suffice. Of course, factor importance depends on the *experimental domain* (or *experimental frame*, as Zeigler et al. 2000 call it). For example, oxygen supply is important for missions high in the sky and deep under water, but not on land at sea level. So the clients must supply information on the intended use of the simulation, including realistic ranges of the individual factors and limits on the *admissible scenarios*. This includes realistic combinations of factor values; for example, some factor values must sum to 100%.

**2.1.2. Finding Robust Decisions or Policies.** We discuss robust policies, rather than optimal policies, for a reason. It is certainly true that finding the *optimal* policy for a simulated system is a hot topic, and many methods have been proposed. These methods include heuristic search techniques that treat the simulation model as a black box—such as genetic algorithms, response surface methodology (RSM), simulated annealing, and tabu search—and methods that analyze the simulation model to estimate gradients—such as perturbation analysis and score functions. The latter two techniques can be used in an optimization algorithm such as stochastic approximation. Fu (2002) and Spall (2003) discuss the current research and practice of optimization for simulation. Unfortunately, all these methods implicitly condition on a large number of events or environmental factors. In practice, the future environment is uncertain, so this so-called optimal policy cannot be achieved and may break down completely. Therefore, we wish to find a *robust* policy, that is, one that works well across a broad range of scenarios. Such policies have also been called "satisficing" (Simon 1981).

To illustrate this problem with classic optimization, consider using simulation to explore different layouts for a small factory. The project manager's decision factors relate to the type, number, position, and buffers associated with machines on the factory floor, as well as schemes for prioritizing or expediting orders. This is a prototypical problem often analyzed using a simulation optimization method, but the result of the "optimization" is conditioned on assumptions of specific (typically assumed independent) distributions for order arrivals, order sizes, machine uptimes, downtimes, and service times, and many more input variables. We argue that using the term "optimum" is problematic when the probability of all these assumptions holding in practice—even for a limited time—is effectively zero. Suggesting different possible "optimal" layouts for several potential customer order patterns may be singularly unhelpful since the decision maker cannot control future orders and may not even have good forecasts.

In contrast, a *robust design* approach treats all these assumptions as additional factors when running the experiment. These are considered *noise factors* (rather than *decision factors*) because they are unknown or uncontrollable in the real-world environment. A robust system or policy works well across a *range* of noise conditions that might be experienced, so implementing a robust solution is much less likely to result in unanticipated results. For example, Sanchez et al. (1996) use a multistage sequential approach to evaluate factory layouts and order-dispatching rules in a job shop when the mix of demand for different products is unknown. They find that the "best" factory setup uses neither the commonly used economic order quantity nor the just-in-time dispatching rule for batching orders. Their robust solution yields a 34% smaller loss (in expected squared deviation from the target total time in system) than a solution that optimizes the mean time in the system, and a 10%–22% improvement over solutions that minimize the variance of the time in the system; the robust solution uses no more (and usually fewer) total machines, indicating potential savings in capital expenditures. Another example is Kleijnen and Gaury (2003) who assume a base production planning scenario and compare several solutions to identify which solution is least sensitive to changes in the environment.

This *robust design philosophy* is inspired by Taguchi (1987), who uses simple designs to identify robust product configurations for Toyota. The results improve quality while lowering the cost of automobiles and component systems because the chosen product designs perform well—despite variations in incoming raw material properties, the manufacturing process, and customers' environments. Sanchez (2000) discusses robust design for simulation experiments. Metamodels can suggest scenarios (i.e., new combinations of factor levels) not yet investigated, although the analyst should make confirmatory runs before applying the results.

We do not mean to imply that an optimization approach will necessarily yield a bad answer. An analyst can perform sensitivity analysis on any particular solution, either formally (e.g., applying DOE techniques or invoking mathematical arguments on the nature of the response surface) or informally (e.g., performing some trial-and-error investigations to determine whether small changes in the scenario lead to big changes in the output). If sensitivity analysis around the so-called optimal solution indicates that it still performs well (in an absolute sense) when realistic departures from these assumptions occur, then the optimization algorithm has identified a solution that is likely to perform well in practice. If changes in the environment (e.g., new patterns of customer orders) affect all potential solutions similarly, then the relative merit of particular policies does not change. If factor settings associated with good mean responses are also associated with low response variances, then the optimal solution in terms of mean performance will also be robust. In a recent case study, Kleijnen et al. (2003) derive a solution that minimizes both the expected value and the variance of the output of a supply chain simulation. That solution is controlled by the decision factors; the mean and variance are computed across several environmental scenarios.

Nonetheless, there are situations where optimizing and then performing sensitivity analysis can lead to fundamentally different answers. For example, a military problem of current interest is finding a good strategy for defending a high-value target (courthouse, church, or monument) against a single terrorist. If the analysts condition on the route the terrorist will take approaching the building, then forces will be concentrated along this path. Conversely, if the direction of approach is unknown, then an entirely different strategy (dispersing the protective forces) is much more effective.

**2.1.3. Comparing Decisions or Policies.** We avoid the phrase "making predictions about the performance of various decisions or policies." Comparisons may need to be made across a number of dimensions. Rather than formal statistical methods for testing particular factor effects or estimating a specific performance measure, our goal might be to provide the decision maker with detailed descriptive information. For example, we could present the means, variances, percentiles, and any unusual observations (see the box plots in Law and Kelton 2000) for the distribution functions of the estimators of several performance measures, for each of the systems of interest. These measures can then be reported, along with implementation costs and other considerations not included in the model.

If at least some of the factors are quantitative, and if a performance measure can be clearly stated, then metamodels can describe the I/O relationships via functions of various factor levels. Here, rather than running an experiment to gain insight into how the performance is affected by all the factors, we may focus on a few of immediate interest to the decision maker.

If the analysts wish to compare a fixed small number of *statistical populations* (representing policies or scenarios), ranking and selection procedures (R&S), multiple comparison procedures (MCPs), and multiple ranking procedures (MRP) can be used. There are two basic approaches: (i) how to select, with high probability, the system, decision, or policy that is—for practical purposes—the best of the potential choices; and (ii) how to screen the potential systems, decisions, or policies to obtain a (random-size) subset of "good" ones. Many procedures have been developed specifically to address some of the characteristics of simulation experiments we discuss in §3 (Chick and Inoue 2001, Hsu 1996, Goldsman et al. 2002, Nelson and Goldsman 2001). Some assume that all populations are compared with each other, whereas others assume comparisons with a standard.

**2.1.4. Summary.** The three types of questions we pose differ from those typically suggested in the literature. Sacks et al. (1989) classify problems for simulation analysts as prediction, calibration, and optimization. Kleijnen (1998) distinguishes among global (not local) sensitivity analysis, optimization, and validation of simulation models. (In global sensitivity analysis the simulation inputs vary over the whole experimental area, rather than infinitesimally.) These two classifications are related to the ones we use—for example, global sensitivity analysis can be used as a way of gaining understanding about a problem—but there is not a one-to-one mapping. For certain classes of simulations, such as military operations or hazardous waste disposal, data are extremely limited or nonexistent, so calibrating, optimizing, predicting, and validating may be meaningless goals.

In the best tradition of scientific discovery, simulation experiments can, nonetheless, have a role in supporting the *development* of insights (or theories) in these situations. For example, Helton and Marietta (2000) discuss how to assess the performance of a nuclear waste plant in New Mexico in the next 10,000 years. Obviously, such a model is hard to validate due to a dearth of data and changing conditions. Nevertheless, extensive sensitivity analyses convinced the Environmental Protection Agency of the validity of this model, so it granted a permit to build and exploit this plant. Dewar et al. (1996) also discuss how one can credibly use models that cannot be validated in the simulation of future military conflicts—an area of almost unimaginable complexity. Despite the tremendous amount of uncertainty about potential future conflicts, decisions must be made (such as what equipment to purchase, how to organize units, and how to use future forces) that will affect large sums of money and affect many lives. Since simulations of potential future force-on-force cannot be validated (Hodges 1991), these simulations are used to assist decision makers in gaining insights into extremely complex systems and processes. For example, if the veracity of a given simulation model cannot be ascertained, but the simulation is known to favor one system over another, this knowledge can sometimes be used to make a strong decision. Suppose a warfare simulation is known to be biased in favor of the red side over the blue side (say, the simulation assumes the red force members are all 10 feet tall), yet the blue side always wins in the simulation. Then we can be confident that the blue side will win such a conflict in the real world. This type of reasoning is called an *a fortiori argument* (Hodges and Dewar 1992). Note that if the red side wins in the simulation, we do not know whether this result occurs because the red side is indeed better or because the simulation is biased. An unvalidated simulation model can also be used to generate plausible outcomes if they are consistent with all available information deemed salient. One can easily construct a case in which an action would be avoided if a simulation suggests that a potentially catastrophic outcome is plausible. More typically, high-dimensional explorations of unvalidated models are used to help devise new ideas (i.e., tools for brainstorming) or to trace the consequences of assumptions over a variety of conditions.

The situations above contrast sharply with many simulation experiments in the literature that often assume a thoroughly validated and verified simulation model exists, and that the decision makers have very specific questions about, e.g., the impact on a particular performance measure resulting from changing a small number of factors to specified (new) values. The users might hypothesize the nature and strength of a particular factor effect, and the analysts' charge is to run the simulation model and collect I/O data to test this hypothesis.

### 2.2. The Simulation Setting

We now describe characteristics of simulation settings that call for *nontraditional designs*, drawing on recent practical examples from industrial and military applications for motivation.

**2.2.1. Number of Potential Factors.** In real-world experiments, only a small number of factors are typically varied. Indeed, it is impractical or impossible to attempt to control more than, say, 10 factors; many published experiments deal with fewer than 5. Academic simulations, such as single-server queueing models, are also severely limited in terms of the number of potential input factors. In contrast, a multitude of potential factors exists for simulation models used in practice. For example, the *Map Aware Non-uniform Automata* (MANA) software platform was developed to facilitate construction of simple agent-based models (Lauren and Stephen 2002). The agents' rules for movement are a function of a "personality" or propensities to move based on 10 competing goals. In all, over 20 factors can be modified for each agent for each of 49 personality states, so we are dealing with thousands of factors. Yet this is considered a "simple" modeling platform! Other examples abound. Bettonvil and Kleijnen (1997) describe an ecological case study involving 281 factors. Cioppa (2002) examines 22 factors in an investigation of peace-enforcement operations. Even simple queueing systems can be viewed as having a few dozen factors if the analysts consider arrival rates and distributions that change over time, service distributions, and correlations arising when service times decrease or servers are added as long lines of customers build up.

Good programming avoids fixing the factors at specific numerical values within the code; instead, the code reads factor values so the program can be run for many combinations of values. Of course, the code should check whether these values are admissible; that is, do these combinations fall within the experimental domain? *Such a practice can automatically provide a list of potential factors*. Next, users should confirm whether they indeed wish to experiment with all these factors or whether they wish to fix some factors at nominal (or base) levels a priori. This type of coding helps *unfreeze* the mindset of users who would otherwise be inclined to focus on only a few factors.

**2.2.2. Choice of Performance Measures.** Consider both the type and the number of performance measures. Some problems require only *relative* answers, i.e., whether one policy is better than another. For example, in a study on the search for sea mines, users wanted to know which sonar tilt angle of the sonar is better; see Kleijnen (1995). Conversely, some problems require *absolute* answers. For example, in the same case study, users wanted to know whether the probability of mine detection exceeds a certain threshold before deciding whether to do a mine sweep at all.

Most procedures (e.g., R&S, MCPs, MRP, and RSM) involve a single quantitative performance measure; the goal is typically to maximize or minimize the expected value of a single simulation output. However, in many simulation applications, it is unrealistic to assume a single measure characterizes the system performance. For example, textbook examples of simple queueing systems often discuss minimizing the average waiting time. In practice, other choices include minimizing the proportion of customers who wait more than a specified time, maximizing the number served within a particular time, improving customer satisfaction by providing information about projected wait time and allowing customers to reschedule, minimizing the errors in processing customer transactions, and balancing workloads across servers. Other examples are the various performance measures in supply-chain management; see Kleijnen and Smits (2003). Consequently, it is restrictive to use a DOE framework suggesting that the appropriate goal of the study should be examining the expected value of a single performance measure.

Taguchi's (1987) robust design approach offers another way to proceed in the case of multiple performance measures. If responses are converted to losses and appropriately scaled, then analysts can construct models of overall expected loss. We prefer to construct separate metamodels for each performance characteristic because it makes it easier to identify *why* certain scenarios exhibit more or less desirable performance than others.

A few researchers use a mathematical-programming framework to analyze multiple simulation outputs, i.e., one output is minimized whereas the remaining outputs should satisfy prefixed constraints (Angün et al. 2002). For example, inventory may be minimized while the service percentage meets a prespecified level.

**2.2.3. Response-Surface Complexity.** Assumptions about the metamodel's complexity are generally broken down into assumptions regarding its deterministic and its stochastic components and often drive the analysis. The standard assumptions in DOE are that the deterministic component can be fit by a polynomial model of the factor levels (perhaps after suitable transformations of the factors or responses) and that the stochastic component can be characterized as additive *white noise*. The latter assumption means that the residuals of the metamodel are normally distributed and IID. In practice, normality may be explained by the central limit theorem, but the IID assumption is violated when the noise has larger variances in subspaces of the experimental area. This is known as *variance heterogeneity* or *heteroscedasticity* and is pervasive in simulation. For example, in queueing problems the intrinsic noise increases dramatically as the traffic load approaches 100% (Cheng and Kleijnen 1999, Kleijnen et al. 2000). Moreover, common random numbers (CRNs) are often used for generating output from several simulation scenarios since they can sharpen the comparison among systems. Unfortunately, CRNs violate the independence assumption.

Good modeling practice means that the analyst should strive to find the simplest metamodel that captures the essential characteristics of the system (Occam's razor). Therefore, we need a suite of design tools, some appropriate for simple response surfaces and others for more complex systems. Simpler metamodels are often easier to justify when only a small number of factors and performance measures are examined, yet interpreting the results may be problematic because the analyst may easily miss important system characteristics. In §4, we describe how some designs allow assessment of the suitability of the estimated metamodel. In principle, we prefer to classify factors into four categories: (i) factors thought to be very important, (ii) factors that might be important, (iii) factors that are thought to be unimportant but are sampled anyway, and (iv) factors that we are quite comfortable in ignoring. Designs that sample differently across these classifications make intuitive sense.

It is increasingly apparent that some systems exhibit highly nonlinear behavior. A system's response surface may be characterized by localized regions where the response differs sharply from the surrounding area (spikes). It may contain thresholds (large, smooth contours in the factor space) where the response is discontinuous, so if the threshold is crossed the response steps up or down. It may contain regions over which the response is chaotic, i.e., extremely sensitive to tiny changes in the input-factor settings so that the output appears impossible to predict. For example, Vinyard and Lucas (2002) make billions of runs and find that chaotic behavior is rampant across many performance measures in a simple deterministic model of combat. Designs that examine only a small number of scenarios are unable to reveal such behavior; instead, the analysts may believe they are facing a simulation model with a large stochastic component.

**2.2.4. Steady-State vs. Terminating Simulations.** Terminating simulations run until a specific event has occurred; for example, we might simulate a single day's operation of a retail establishment. Steady-state simulations have no natural termination point, so they can keep generating data for their analysis. The simulation type has implications on the design and analysis. For terminating simulations, it may be necessary to censor results if we are simulating rare events; see Kleijnen et al. (2001). For steady-state simulations, the initial conditions are often chosen for convenience rather than relevance, e.g., a simulation of a computer network may start with all servers and relay nodes operational and no demands on the system. Here, the simulation output of the *warm-up period* biases the estimated response. The length of the warm-up period affects the total time required for experimentation.

**2.2.5. Inclusion of Simulation-Specific Factors.** Analysts have control over many things during the course of a simulation study (in addition to the factor levels they manipulate and the performance measures they collect). This control includes the maximum run time for terminating simulations. For steady-state simulations this control includes specifying the warm-up period and run length(s), as well as how (if at all) the time-series output is averaged or aggregated into batches. The choice of the number of batches and batch sizes is an important topic of research in itself (e.g., Schmeiser 1982, Steiger et al. 2005, Alexopoulos and Goldsman 2004), and an implicit assumption in many simulation-analysis techniques is that appropriate batch sizes and warm-up periods are used. Other simulation-specific factors that can be controlled include the use of CRNs to facilitate comparisons across alternatives. For example, all potential factory layouts can be subjected to the same pattern of customer orders. Other variance-reduction techniques (VRTs), such as control variates and importance sampling, have been developed for simulation output (see Law and Kelton 2000). Unfortunately, not all designs can easily accommodate these VRTs.

### 2.3. External Concerns and Constraints

We now discuss issues that often play a major role in the implementation of simulation experiments, although they are generally not discussed in the literature.

**2.3.1. Sequential vs. One-shot Data Collection.** In real-world experiments, the basic mindset is often that data should be taken simultaneously unless the design is specifically identified as a sequential design. When samples must be taken sequentially, the experiment is viewed as prone to validity problems. Analysts must therefore randomize the order of sampling to guard against time-related changes in the experimental environment (such as temperature, humidity, consumer confidence, and learning effects) and perform appropriate statistical tests to determine whether the results have been contaminated.

Most simulation experiments are implemented sequentially even if they are not formally analyzed that way. If a small number of design points are explored, this implementation may involve the analysts manually changing factor levels. An approach less prone to data-entry errors involves automatically generating an input file, or series of input files, once a particular design has been chosen. These files may be executed sequentially (and efficiently) in batch mode. Modifying simulations to run in parallel over different computers is possible but not typical. For example, parallelization is being used effectively at the supercomputing clusters of the Maui High Performance Computing Center (http://www.mhpcc.edu) and the Mitre Corporation in Woodbridge, Virginia (http://www.mitre.org). In many cases, parallelization results from manually starting different runs (or sets of runs) on a few computers to cut down on the overall time to complete the data collection. For example, Vonk Noordegraaf et al. (2003) use five PCs to finish their 64 scenarios, each scenario replicated twice, in two weeks. Freely available software, such as that used on literally thousands of PCs as part of the search for extraterrestrial intelligence (http://www.seti-inst.edu), could be used to facilitate parallel data collection for simulation experiments, but this is not yet readily available to simulation analysts in the industrial or academic settings with which we are familiar.

**2.3.2. Premature Termination of the Experiment.** When a simulation takes a nontrivial amount of time to run, analysts may have to terminate the experiment *prematurely* because the computer breaks down, the client gets impatient, preliminary results are needed, etc. This premature termination occurs in many defense simulation projects. It is then better that the analyst organize the list of scenarios so that the output can provide useful information, even if curtailed. For example, consider a simulation where a single input factor (taking the value 1 or 2) defines two systems the decision maker wishes to compare, each run takes one day of CPU time, the design specifies 30 replications of each system, and a single computer is available (so it will take two months to complete the experiment). If all 30 runs for system 1 are conducted before beginning the runs for system 2, it will be impossible to say anything about the relative merits of the systems until the end of day 31. In contrast, an alternating sequence of runs allows preliminary comparisons as early as the end of day 2, and half the data on each system is available by the end of day 30. According to DOE theory the scenarios could be run in any order, but the latter approach is clearly preferable if preliminary results might be requested or the experiment might be terminated early. This idea also applies when runs are conducted on multiple machines or there are multiple input factors, each with multiple levels, provided a long time is needed to complete the experiment.

With this view, even nonsequential designs can be implemented sequentially in ways that are robust to early termination. Some single-stage designs can be viewed as augmentations of simpler designs, so there is a natural way to separate the designs into two or more parts. Clearly, sequential or partially sequential designs have this characteristic: after one stage of sampling the analysts indicate which configuration(s) should be examined next.

**2.3.3. Data Collection Effort.** The increase in computer speeds has caused some analysts to add more details *to* their simulation models. We believe it should spur us to ask more *from* our simulation models.

The traditional concept of a fixed sampling budget (in terms of the number of runs) is unnecessarily restrictive. Even if a single computer is used, the time per run is seldom fixed. Different analysts might use different run lengths and batch sizes. Run times might vary across scenarios because some tend to yield fewer events or have different warm-up periods in steady-state simulations, or lead to early termination for non-steady-state simulations. Implementing a design may be very easy if software is available to generate coded factor levels, next convert them to original factor levels, and then generate input files so the simulation can be run in batch mode. Conversely, if the analysts must edit and recompile the code for each scenario, or make all changes manually through a graphical user interface, implementation time can surpass the run time.

Another way of describing this data-collection effort is by the time required to estimate the metamodel parameters to a certain level of precision. Unfortunately, it is difficult to make generic recommendations using this approach since the time depends on the underlying (heterogeneous) variability. We have recently seen run time vary from less than a second to over 100 hours per scenario on a single processor.

A related issue is the trade-off between the number of design points and the number of replicates per design point. Suppose the total computer time is the same for the two options: one with many replicates per design point, and another with more design points and fewer replicates. The first option enables explicit estimation of response variances that can vary across scenarios. If the primary goal of the study is *finding robust* systems or policies, then some replication at every design point is essential. If the goal is *understanding* the system, this may also include understanding the variance, again mandating replication. However, if the goal is that of *understanding* or *comparing* systems and a constant variance can be assumed, then this constant can be estimated using classic ordinary least squares regression, provided no CRNs are used and the metamodel is correctly specified. Replication is then of less concern and the second option (exploring more scenarios) can be a better way to spend scarce computer time. Note that a single replicate yields an unbiased estimator of the response of a specific scenario. For example, consider a terminating simulation of a bank that closes at 5:00 p.m. The observed maximum queue length during a single day is an unbiased estimator of the true maximum. Of course, simulating more days provides a more precise estimate based on the observed maximum averaged over all simulated days, though it does not change the fact that different scenarios may result in substantially different variances in the daily maximum queue length.

### 2.4. Conveying Results Effectively

The best experiment will come to naught if the results are not communicated properly to the decision maker. We refer back to the three primary goals (developing a basic understanding, identifying robust solutions, and comparing systems). For the first goal, a good analogy is exploratory data analysis. Graphical tools that allow multidimensional visualization of the results may be much more helpful than equations or tables. Useful tools include three-dimensional rotatable plots, contour plots, and trellis plots (Sanchez and Lucas 2002). Regression trees and Bayesian networks have also been effective ways of communicating which factors are most influential on the performance measures (Gentle 2002, Martinez and Martinez 2002). Yet, visualizing simulation results remains a challenge at this stage of simulation experimentation. Tufte (1990) is the seminal reference for excellence in graphical presentation; Meyer and Johnson (2001) describe tools developed specifically for visually exploring large amounts of data from simulation experiments with multiple performance measures.

## 3. Criteria for Evaluating Designs

Once analysts know their situation, the question is: now what? Above we stated that there is no single prototypical situation (in terms of the type of question to be asked, or simulation characteristics) that analysts might face. In this light, it is not surprising that we cannot recommend a specific design. How, then, should analysts choose an appropriate design? While we do not have all the answers, we do attempt to provide some guidance.

In what follows, we use the term *design* to denote a matrix where the columns correspond to the input factors, the entries correspond to (possibly coded) levels for these factors, and each row represents a particular combination of factor levels also called a *design point*. More detail on construction and use of these designs is in this paper's Online Supplement.

Others have listed desirable attributes for designs for experiments with real systems (Box and Draper 1987, Myers and Montgomery 2002). We describe criteria to evaluate designs in simulation settings, and we discuss how they may (or may not) apply directly to the issues described earlier. We will use these criteria in deciding which designs to recommend in §4.

### 3.1. Number of Scenarios

In the literature, a major design attribute is the number of scenarios required to enable estimation of metamodel parameters. A design is called *saturated* if its number of factor combinations (say) $n$ equals the number of metamodel parameters, $q$. For example, if the metamodel is a first-order polynomial in $k$ factors, then $q = k + 1$ (where 1 refers to the grand or overall mean, often denoted by $\beta_0$); a saturated design means $n = k + 1$. Actually, there are several saturated designs for a given metamodel type. For the first-order polynomial in $k$ factors, one saturated design changes one factor at a time, whereas another design is a fractional factorial (see §4 or Box et al. 1978). To choose among different designs, we also consider the following quality attributes.

### 3.2. Orthogonality

A design is said to be *orthogonal* if the columns of the design matrix are orthogonal (i.e., the inner product of any two columns is zero). Orthogonality has long been a desirable criterion for evaluating designs. It simplifies computations. Since the input factors are uncorrelated, it is easier to determine whether to include them in a metamodel (e.g., using regression) and to separate their contributions to the overall metamodel fit. This in turn simplifies interpretation of the results. (Lack of orthogonality, also called *multicollinearity*, implies that the effect estimates have very large standard errors or cannot be computed at all.) Unfortunately, requiring orthogonality also has limitations. In reality, some factor level combinations may not be permissible. For example, in an M/M/1 queue the expected steady-state waiting time is infinite if the arrival rate exceeds the service rate. A complicated application (simulating part of the Rotterdam harbor) with exploding waiting times for the original orthogonal design appears in Kleijnen et al. (1979). In general, forcing orthogonal designs may mean limiting many factors to narrower ranges, or figuring out a way to deal with unstable results for certain scenarios. Unfortunately, in complex models it may not be possible to know a priori which factor-level combinations are problematic.

A design may be orthogonal in the *coded* factor values (such as $-1$ and $+1$) but not in the original factor values. Simulation analysts should be aware of possible scaling effects. Coding all the factor levels can facilitate identification of the most important factors (Box et al. 1978, Bettonvil and Kleijnen 1990).

### 3.3. Efficiency

The design determines the standard errors for the estimated metamodel parameters. The DOE literature uses several criteria (see Kleijnen 1987, p. 335). For example, *A-optimality* means that the sum of these standard errors is minimal. *D-optimality* considers the whole covariance matrix of the estimated parameters (not only the main diagonal) and means that the determinant of this matrix is minimal. *G-optimality* considers the mean squared error of the output predicted through the metamodel (Koehler and Owen 1996).

The criteria above certainly can be (and have been) used to evaluate designs proposed for analyzing simulation experiments. Unfortunately, these criteria require strong a priori assumptions on the metamodels to be fit to the data and the nature of the response (e.g., variance homogeneity). These assumptions are usually violated in simulation. Consequently, these criteria are of little value when there is substantial uncertainty a priori on the nature of the simulation's output. Moreover, focusing on minimizing the number of design points (or maximizing the efficiency for a fixed number of design points) may not be enough to insure "efficient" data collection, at least for steady-state simulations where it does not make much sense to worry about using the most efficient design if one does not also worry about using the smallest run length to achieve the desired goal. In short, efficiency is most critical when the runs are time consuming. Other criteria become more relevant when we are able to gather plenty of data quickly.

### 3.4. Space Filling and Bias Protection

Conceptually, space-filling designs sample not only at the edges of the hypercube that defines the experimental area, but also in the interior. A design with good space-filling properties means that analysts need not make many assumptions about the nature of the response surface. Space-filling designs currently provide the best way of exploring surfaces where we do not expect to have smooth metamodels. They are particularly useful for fitting nonparametric models, such as locally weighted regressions. These designs, especially Latin hypercube sampling (LHS), have been applied when fitting Kriging models (see §4) and neural networks (Alam et al. 2004). Detection of thresholds is discussed by Watson and Barnes (1995), who propose a sequential design procedure.

Space-filling designs also provide flexibility when estimating a large number of linear and nonlinear effects, as well as interactions, and so provide general bias protection when fitting metamodels of specific forms. Other designs do not have good space-filling properties but still protect against specific violations of model complexity assumptions. These include the designs of resolution 3, 4, and 5 below. We also refer to Sasena et al. (2002) and Kleijnen and van Beers (2004), who develop customized (but not space-filling) designs where sequentially selected scenarios are driven by the specific simulation application at hand.

### 3.5. Ability to Handle Constraints on Factor-Level Combinations

In some situations (for example, chemical experiments) factor values must add up to 100%. The classic DOE literature presents *mixture* designs for these situations (Montgomery 2000). Many designs exist for exploring experimental regions (i.e., permissible combinations of design points) that are either hypercubes or spheres. In simulation experiments, restricting factor values to realistic combinations may complicate the design process dramatically. This is an area seriously in need of more research. Sanchez et al. (2001) propose elliptical designs, motivated by observational economic data. In many queueing situations, certain combinations of factor settings give unstable outputs (Kleijnen et al. 1979, Sanchez et al. 2005). Until designs that can handle such situations are available, visual presentation of the results—and exploratory data analysis—may be the most appropriate ways of determining whether these situations exist.

### 3.6. Ease of Design Construction and Analysis

Designs should be easy to construct if they are to be used in practice. Nonetheless, some designs are useful even if difficult to generate, so we do not rule out the use of tabulated designs, particularly if they are incorporated into software packages. The major statistical software packages include some experimental-design generation methods. Ideally, design software should be readily available for many platforms. One example is WebDOE, which helps users to design their experiments with deterministic simulation models, offering a library of classical designs through an easy-to-use Web interface (Crary Group 2004).

The *analysis* is also easy if software is available for many platforms. Regression software is abundant, so the most common analysis tool is readily available and need not be discussed further. Newer surface-fitting methods are also available, including Kriging, neural nets, radial basis functions, splines, support-vector regression, and wavelets; see Clarke et al. (2005) and Antioniadis and Pham (1998). These are metamodel construction methods that can be applied to data collected using a variety of experimental designs and may do a better job fitting certain complex response surfaces. Because we have some experience with Kriging, which has established a track record in deterministic simulation—and this metamodel is not yet much applied in random simulation—we briefly explain the basic approach (see van Beers and Kleijnen 2003, 2004).

*Kriging* is named after the South African mining engineer D. G. Krige, who developed his technique while searching for gold. It is an interpolation method that predicts unknown values of a random function or random process; see the classic Kriging textbook of Cressie (1993) or the excellent text by Santner et al. (2003). More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. These weights depend on the distances between the input for which the output is to be predicted and the inputs already simulated. Kriging assumes that *the closer the input scenarios are, the more positively correlated the outputs are*. This is modeled through the correlogram or the related variogram.

The optimal Kriging weights vary with the input value for which output is to be predicted, whereas linear regression uses one estimated metamodel for all input values.

If analysts are interested in the I/O behavior within a local area, then a low-order polynomial may be an adequate metamodel. However, for an experimental area that is global (not local), Kleijnen and van Beers (2005) demonstrate that a low-order polynomial gives poor predictions compared with a Kriging metamodel. Giunta and Watson (1998) also compare Kriging with polynomial metamodels. Jin et al. (2001) compare Kriging with polynomial metamodels, splines, and neural nets. More recently, van Beers and Kleijnen (2003) apply Kriging to stochastic simulation; Jin et al. (2002) discuss the accuracy of Kriging and other metamodels under a sequential sampling approach.

Note that in *deterministic simulation*, Kriging has an important advantage over linear regression analysis: Kriging gives predicted values at observed input values that are exactly equal to the simulated output values. Deterministic simulations are used for computer-aided engineering in the development of airplanes, automobiles, computer chips, computer monitors, etc.; see the pioneering article of Sacks et al. (1989), and Simpson et al. (2001) for an update. Lophaven et al. (2002) have developed a MATLAB toolbox for Kriging approximations to computer models, but the commercially supported software products currently available (such as the Kriging software in S-Plus) are intended for real-world data, and so limited to three dimensions.

In theory, if a design is used to generate multiple outputs they will be accounted for in the analysis. For example, *multivariate regression analysis* may be applied. Each output is usually analyzed individually in practice. For linear regression analysis, Khuri (1996) proves that this suffices if all outputs are generated by the same design. The same design is indeed used when running the simulation and observing multiple outputs.

## 4. Design Toolkit: What Works and When

Now that we have identified several characteristics of simulation settings and designs, it is time to match them together. Consider Figure 1, in which we chart some designs according to two dimensions that together describe the simulation setting. The horizontal axis represents a continuum from simple to complex response surfaces. Since the metamodel complexity depends on both the deterministic and stochastic components, there is not a unique mapping. We list some of the assumptions along the axis to inform the users about the types of metamodels that can be fit. The vertical axis loosely represents the number of factors. So the lower left represents simple response surfaces with only a handful of factors, that is, the traditional DOE setting with Plackett-Burman designs developed in the 1940s, etc. The upper right represents very complex response surfaces with many fac-

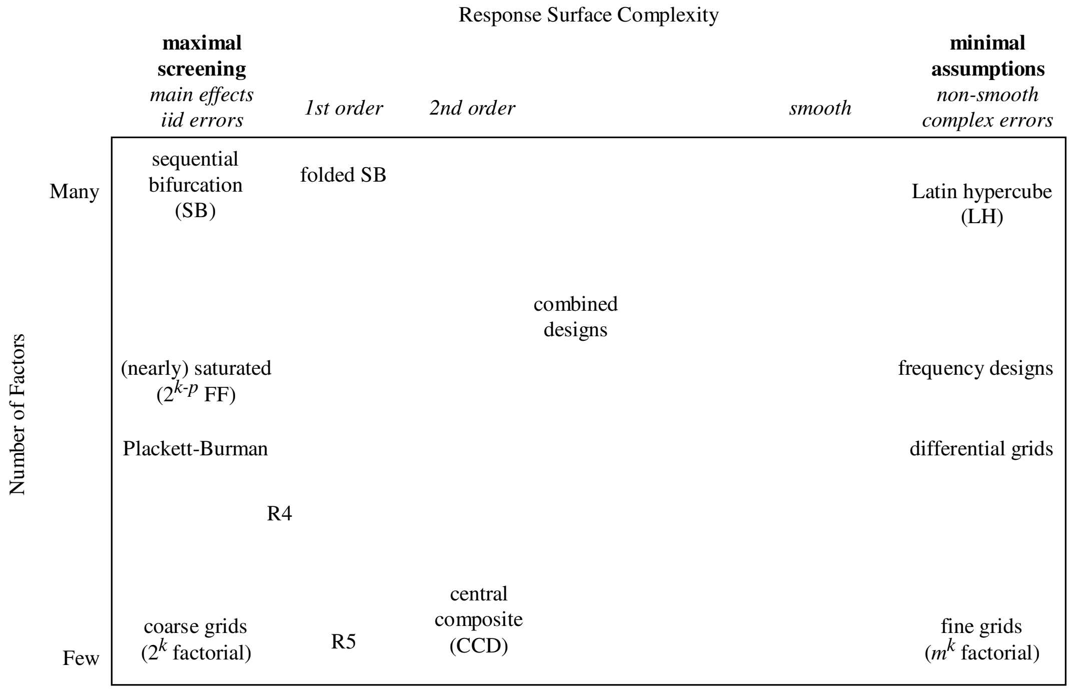

**Figure 1** Recommended Designs According to the Number of Factors and System Complexity Assumptions

tors. We do not present a comprehensive list of all available designs, but rather describe those that seem most promising and are either readily available or fairly easy to generate.

We hope to change the mindset of those who might otherwise begin experimentation by focusing on a small number of factors, so we advocate using designs near the top of this figure. In this way, analysts can look broadly across the factors in the simulation study. The analyst willing to make simplifying assumptions can start from the left of the figure, which will tend to reduce the initial data-collection effort. (Of course, whenever assumptions are introduced, their validity should be checked later.) Alternatively, the analyst can start from the upper right of the figure for an initial experiment if little is known about the nature of the response. Employing CRNs or other VRTs can make certain procedures more efficient, and perhaps allow the analyst to handle more factors or make fewer assumptions for a given computational effort. In our experience, VRTs other than CRNs seldom give dramatic efficiency gains—except for rare-event simulations—but others report they have found VRTs quite effective in large-scale simulations.

If this initial experiment does not completely address the main goal, then preliminary results can be used to design new experiments (augmenting the current data) to focus on factors or regions that appear most interesting. This corresponds to moving south in Figure 1 to focus on the short list of factors selected after the initial experiment while holding the remaining factors to only a few configurations. If assumptions about the response-surface complexity are made for the initial experiment, then moving south-east is beneficial to check their validity. If few assumptions are made and the initial analysis indicates that the response surface is not very complex, then moving south-west allows the analyst to take advantage of highly efficient designs in subsequent experiments.

We now provide brief descriptions of the designs in Figure 1 and their characteristics, along with references for details. A few sample designs are given in the Online Supplement (also see Kleijnen 2005).

### 4.1. Gridded or Factorial Designs

Factorial designs are easy to explain to someone unfamiliar with classic DOE. A popular type of factorial is a $2^k$ design, which examines each of $k$ factors at two levels, and simulates all resulting combinations. Then it is possible to fit a metamodel including *all* interactions, not only between pairs of factors, but also among triplets, etc.

Considering more complex metamodels (i.e., moving to the right in Figure 1), the analysts may use finer grids. Three levels per factor result in $3^k$ designs; in general, $m$ levels per factor result in $m^k$ designs. When there are more than a few factors, analysts may use different grids for different groups of factors, employing finer grids for those factors thought to be important to enable them to either view nonlinearities in the response surface or test the linearity assumption. Unfortunately, the number of scenarios $n$ grows exponentially when $k$ increases, so factorial designs are notoriously inefficient when more than a handful of factors are involved. Nevertheless, these designs are an important tool since they are easy to generate, plot, and analyze. Hence, whenever individual run times are minimal, the benefit of detailed information about the nature of the response surface may easily outweigh the additional computation time relative to the more efficient designs we discuss next.

### 4.2. Resolution 3 (R3) and Resolution 4 (R4) Designs

A design's resolution determines the complexity of metamodels that can be fit, with higher-resolution designs allowing more complex models. Specifically, "a design of resolution $R$ is one in which no $p$-factor effect is confounded with any other effect containing less than $R-p$ factors" (Box et al. 1978, p. 385). Two effects are confounded when they cannot be separately estimated. For metamodels with main effects only (i.e., first-order metamodels with no interaction terms), it can be proved that the most efficient designs are R3 designs, provided the white noise assumption holds. If $k+1$ is a power of two, R3 designs are fractional factorial designs, denoted as $2^{k-p}$ designs where the total number of design points is $2^{k-p}$. If $k+1$ is not a power of two but is a multiple of four, then R3 designs are tabulated as Plackett-Burman designs. See any DOE textbook for details (e.g., Box et al. 1978).

If *interactions* are assumed to be present but users are mainly interested in estimating first-order effects, then R4 designs are appropriate. These designs give unbiased estimators of main effects even if two-factor interactions are present. They can be easily constructed through the *fold-over* procedure, i.e., after executing the R3 design, run the mirror design that reverses each high and low value in a specific factor's column. In other words, proceed in two stages by first running an R3 design and then augmenting it to an R4 design. (See also the RSM designs in Donohue et al. 1993.)

Even if the white-noise assumption does not hold, classic designs produce unbiased estimators of the metamodel parameters, although not necessarily with minimum standard errors. If we account for analysts' time and energy, then these designs seem acceptable. Clearly, R3 designs (which use all scenarios to estimate all effects) give smaller standard errors for the estimated first-order effects than the popular practice of *changing one factor at a time* (which use only two scenarios per effect).

### 4.3. Resolution 5 (R5) Designs

If users are also interested in the individual two-factor interactions, then an R5 design is needed. Few $2^{k-p}$ R5 designs are saturated. Saturated designs include those of Rechtschaffner (1967), discussed by Kleijnen (1987, pp. 310–311) and applied by Kleijnen and Pala (1999). R5 designs require $O(k^2)$ factor combinations, so are less attractive if individual runs are time consuming. If an R4 design suggests that certain factors are unimportant, then computing requirements can be reduced by limiting the R5 design to fewer factors. The $2^{k-p}$ designs can be looked up in tables (Box et al. 1978, Kleijnen 1974–1975, 1987, or Myers and Montgomery 2002), but they are relatively easy to construct and therefore can be automated.

Fractional factorial designs (including R3, R4, and R5 designs) meet classic optimality criteria such as D-optimality for specific metamodels. Other designs that satisfy these criteria are derived in *optimal design* theory, pioneered by Kiefer (1959) and Fedorov (1972); see also Pukelsheim (1993) or Spall (2003). These so-called optimal designs typically lack the simple geometric patterns of classic designs, and are too complicated for many practitioners.

### 4.4. Central Composite Designs (CCD)

A second-order metamodel includes purely quadratic effects in addition to main effects and two-factor interactions. This means that the response functions need not be monotonic. Best known designs for this case are CCD, with five values per factor. These values are coded as $0, \pm 1, \pm c$, with $c \neq 0, 1$. It is possible to determine an optimal value of $c$ if the white-noise assumption holds. Since this assumption does not hold for most simulation experiments, we do not worry too much about the choice of $c$ except to suggest that analysts choose an intermediate value for better space filling. Details on CCD can be found in any DOE textbook (Box et al. 1978, Box and Draper 1987, Montgomery 2000, or Myers and Montgomery 2002).

Actually, estimation of quadratic effects requires no more than three factor levels, so to save computer time analysts may again use *saturated* designs, which implies $n = 1 + k + k(k-1)/2 + k$, namely, one overall mean, $k$ main effects, $k(k-1)/2$ interactions, and $k$ purely quadratic effects. Kleijnen (1987, pp. 314–316) discusses several saturated design types, including so-called simplex designs and fractional $3^k$ designs. Kleijnen and Pala (1999) apply simple saturated designs; see also Batmaz and Tunali (2003).

### 4.5. Sequential Bifurcation (SB)

In practice, there are situations with many factors but few important ones. In such cases, a main-effects model—possibly augmented with two-factor interactions—may suffice. Moreover, users may be able to specify the sign (or direction) of each potential main effect. In these situations, the individual factors can be aggregated into groups such that individual main effects will not cancel out. *Group screening* can be very effective at identifying important factors. A practical and efficient group screening procedure is SB. For example, in an ecological case study, 281 factors are screened after only 77 factor combinations are simulated, resulting in only 15 important factors; see Bettonvil and Kleijnen (1997). If interactions might be important, SB still gives unbiased estimators of the main effects, provided the number of combinations is doubled (similar to the fold-over principle for R3 and R4 designs discussed above). If allowed to run to completion, SB will keep subdividing factor groups unless the estimated aggregate effect for a group is either insignificant or negative, or it identifies individually significant factors. However, SB can be stopped at any stage, and it will still provide upper bounds for aggregated effects, as well as estimates of any individual effects already identified. The most important factor is identified first, then the next most important factor, and so on. Consequently, SB is robust to premature termination of the experiment.

Bettonvil and Kleijnen (1997) discuss SB for deterministic simulations. Cheng (1997) extends the method for stochastic simulations. Kleijnen et al. (2005) also discuss SB for random simulations, including a supply-chain case study. Wan et al. (2003) propose a modification, called controlled sequential bifurcation, and provide proof of its performance under heterogeneous-variance assumptions. Other screening techniques with less restrictive metamodels are discussed by Campolongo et al. (2000), Dean and Lewis (2004), Holcomb et al. (2000a, b), Lin (1995), and Trocine and Malone (2001). Their performance relative to SB needs further research.

### 4.6. Latin Hypercube Sampling (LHS)

For situations involving a relatively large number of factors, McKay et al. (1979) proposed LHS. Let $k$ still define the number of factors, let $n$ denote the number of design points desired ($n \geq k$), and define $n$ levels per factor. Each column of the design matrix is a random permutation of the factor levels. LHS is so straightforward that it is incorporated in popular add-on software (such as @Risk) for spreadsheet simulation; see Sugiyama and Chow (1997).

LHS designs have good space-filling properties—particularly if several LHS designs are appended—so they are efficient ways of exploring unknown, but potentially complicated response surfaces with many quantitative factors. For LHS in Kriging (which assumes smooth metamodels, possibly with many local hilltops) see Koehler and Owen (1996), Morris and Mitchell (1995), Simpson et al. (2001), Pacheco et al. (2003), or Santner et al. (2003).

There are numerous variants of basic LHS. Assuming a linear metamodel, Ye (1998) developed an algorithm for orthogonal LHS. Cioppa (2002) extended the number of factors that can be examined in orthogonal LHS within a fixed number of runs. Moreover, he found that by giving up a small amount of orthogonality (allowing pairwise correlations between the design columns less than 0.03), the analysts can dramatically increase the space-filling property of these designs. His LHS designs are not easy to generate, but are tabulated (Cioppa and Lucas 2005) and thus useful in situations where the total number of runs is limited (perhaps because individual simulation runs are time consuming).

### 4.7. Frequency-Based Designs

For quantitative factors, a frequency-based approach makes each factor oscillate sinusoidally between its lowest and highest value at a unique and carefully chosen frequency. If the simulation is coded so that factors can be oscillated during the course of a run (called the *signal run*), then comparisons can be made to the *noise run* where all factors are held at nominal levels. This approach has been advocated as a screening tool for identifying important metamodel terms; see Schruben and Cogliano (1987) and Sanchez and Buss (1987).

More recently, frequency-based designs have been used to set factor levels for scenarios *externally*. That is, factor levels remain constant during the course of the simulation run, but they change from run to run; see Lucas et al. (2002) or Sanchez and Wu (2003). These designs have reasonably good space-filling properties. Moreover, there is a natural gradation in the granularity of sampling. Factors oscillated at low frequencies are sampled at many levels, whereas factors oscillated at high frequencies are sampled at fewer levels. This property may help analysts design an experiment to be robust to early termination, for example, by choosing higher oscillation frequencies for those factors believed a priori to be most important. By carefully choosing the oscillation frequencies, it is possible to use the results to fit second- and third-order metamodels. The designs are relatively easy to construct and to implement (Jacobson et al. 1991, Morrice and Bardhan 1995, Saltelli et al. 1999, or Sanchez and Wu 2003).

### 4.8. Crossed and Combined Array Designs

Selecting designs for finding *robust* solutions falls naturally into the upper middle portion of Figure 1. While there may be many factors, the analysts are interested in a metamodel that captures the impact of the *decision* factors only. So their metamodel (while it may be complex) does not require estimation of all factor and interaction effects. Actually, the *noise* factors enter into the metamodel via their impact on the *variability* of the response for a particular combination of decision-factor levels. This clear division of factors suggests that the analysts sample the two sets differently—for example, by *crossing* a high-resolution design for the decision factors with a lower resolution design for the noise factors. Crossing means that each combination of decision-factor values is simulated for each environmental scenario, which is defined by the combination of values of the environmental factors. These environmental scenarios enable estimation of the mean and variance of the simulation response per combination of decision factor values. Instead of a crossed design, the analyst may use a combined (or combined array) design (Shoemaker et al. 1991, Myers et al. 1992). In a combined design, a single design matrix (such as a factorial) is used with columns divided among parameters and noise factors. As Myers et al. (1992) suggest, this can lead to a great reduction in the data-collection effort since the only interactions that need to be estimated are those involving two decision factors. Sanchez et al. (1996) apply both crossed and combined designs to explore a job-shop simulation model. In §6 we illustrate a crossed design to identify robust decision-factor settings in a small case study; the design is provided in the Online Supplement.

Many types of designs have been used in this context. Taguchi (1987) proposes a particular class of orthogonal designs, but these are intended for factory experiments and are limited to main-effects models, which we find too restrictive for simulation environments. Ramberg et al. (1991) use a sequential approach, beginning with a $2^{k-p}$ design augmented with a center point for the decision factors, and a saturated or nearly saturated factorial for the noise factors. Moeeni et al. (1997) use three levels (varied across runs) per decision factor and frequency-based oscillation (varied within a run) for 35 noise factors. Cabrera-Rios et al. (2002) propose three levels per decision factor and two levels per environmental factor. If the number of decision factors is not too large, then the analysts may cross a CCD for the decision factors with LHS for the noise factors; see the case study in Kleijnen et al. (2005). If the number of decision factors is large, then orthogonal or nearly orthogonal LHS may be a good design for the decision factors. In short, crossed designs are easy to generate, and the two subdesigns can be chosen to achieve the characteristics (space-filling, orthogonality, efficiency) that are most pertinent to the problem at hand.

Crossed designs can be exploited in situations other than robustness studies. Lucas et al. (1997) give an example of group screening within a fractional factorial design crossed with LHS. Lucas et al. (2002) discuss the benefits of combining multiple designs after classifying factors into several groups based on their anticipated impact. This allows analysts much more flexibility than simply putting each factor into (or leaving it out of) the experiment.

### 4.9. Summary

We have presented several design options for simulation experiments involving either a few or many factors. If runs are extremely time consuming, then analysts can reduce the computational effort by making assumptions about the nature of the response surface. These assumptions can be checked after the runs are completed, as we describe in §5. We contrast this approach to arbitrarily limiting the number of factors. Indeed, if the analysts change only a few factors while keeping all other factors constant, then the conclusions of the study may be extremely limited.

We have not attempted to list all designs that have been proposed for simulation experiments. For example, we have not placed any simulation optimization methods in Figure 1, although *optimization* can be viewed as a means of comparing systems under very specific conditions. Our goal is to suggest some designs that analysts can readily use.

## 5. Checking the Assumptions

Whichever design is used, sound practice means checking assumptions. With designs from the right of Figure 1, few assumptions are made about the nature of the response surface. In the process of fitting a metamodel, analysts determine what (if any) assumptions are reasonable. If they start in the upper left corner of Figure 1, then the experiment is likely used to screen the factors and identify a short list as the focus of subsequent experimentation. If so, there are likely to be fewer assumptions during the next stages of experimentation. If they start from the lower left (as traditional DOE does), then it may be essential to confirm that the resulting metamodel is sufficient, or to augment it appropriately.

One check has the *signs* of the estimated effects evaluated by experts on the real system being simulated. For example, does a decreased traffic rate (resulting from adding or training servers) indeed reduce the average waiting time? Another example is the case study by Kleijnen (1995) on a sonar simulation, in which naval experts evaluate the signs of the metamodel effects; because all signs are accepted, the underlying simulation model is considered to be "valid." In general, checking the signs may be particularly applicable when the goal of the simulation study is general understanding rather than prediction, as for the agent-based models discussed earlier. Sometimes intuition is wrong and needs to be challenged. For example, Smith and Sanchez (2003) describe a forecasting project where the model of losses (incurred for certain groups of loans) had the "wrong" signs. Examination of the detailed files confirmed that their patterns differed from the vast majority of loans and revealed why, so that the model ended up providing new—and valid—insights to experts. Another example is the ecological case study that Bettonvil and Kleijnen (1997) use to demonstrate SB: the resulting short list of factors includes some that the ecological experts had not expected to be important.

Another check *compares* the metamodel predictions to the simulation outputs for one or more new scenarios (which might be selected through a small LHS design). If the results do not differ significantly, the metamodel is considered acceptable (see any textbook on linear models such as Kutner et al. 2004, also Kleijnen et al. 1998). Kleijnen and Sargent (2000) discuss how to use output from initial simulation experiments to test the metamodel constructed from other scenarios in subsequent experiments. They refer to this as *validating metamodels*, not to be confused with validating a simulation model.

The assumption of normal IID errors can be examined via residual analysis (if regression is used to fit the metamodels), or by taking additional replications at a few design points. Tunali and Batmaz (2000) investigate procedures for validating this and other assumptions for least-squares metamodel estimation.

Note that *higher-order interactions* are notoriously difficult to explain to users; nevertheless, traditional DOE routinely estimates and tests these interactions. One solution *transforms* the original inputs or outputs of the simulation model, to simplify the metamodel. For example, replacing two individual factors by their ratio may help in queueing simulations where the arrival and the service rates are combined into the traffic rate; in combat models the relative strength may provide a better metamodel than the individual absolute strengths of the two combatants. Furthermore, logarithmic transformations of inputs and outputs may provide a better-fitting metamodel in queueing problems; see Kleijnen and Van Groenendaal (1992).

Unfortunately, it may be difficult or impossible to transform individual *factors* to achieve simple metamodels, particularly when multiple performance measures are collected. One option might be to transform certain *responses*. We have observed instances where a transformation serendipitously yields responses of direct interest to the decision maker (such as the differences in, rather than magnitudes of, sensor ranges), while allowing the analyst to fit simpler models in the transformed spaces.

Transformations of responses are sometimes applied to satisfy the assumptions underlying the statistical analysis, rather than to construct performance measures of interest to the decision maker or to simplify the form of the metamodel. This may require a specialized analysis approach. For example, Irizarry et al. (2003) develop the so-called MLE-Delta method after finding that constructing metamodels and back-transforming the results may yield highly biased point and confidence interval estimates of the original (untransformed) responses. Often, simpler methods that do not rely on transformation suffice, particularly when we seek insights rather then predictions. For example, classic ordinary least squares (OLS) assumes normally and independently distributed simulation responses with constant variances across different scenarios, but the resulting estimators are unbiased even when the variances are not constant, and the usual test statistic is known to be very insensitive to departures from the normality assumption. This means a readily available tool can be used for analysis. A simple method that accommodates variance heterogeneity and CRNs (but does not rely on transformation) replicates the design (say) $m$ times, estimates the metamodel from each replication, and computes confidence intervals for the metamodel's parameters and predictions from these $m$ replicates using a Student $t$ statistic with $m-1$ degrees of freedom (Kleijnen 2005, Appendix A). We reiterate that if substantial variance heterogeneity is present, it should be characterized and conveyed directly to the decision maker.

Most practical simulation models have (say) $w$ multiple outputs, or a multivariate response in statistical terminology. Fortunately, in the case of linear regression modeling, the OLS estimators computed for the $w$ individual responses are identical to the generalized least squares estimators whenever the multivariate response is generated by a single design (Rao 1967).

Even with careful thought and planning, it is rare that the results from a single experiment are so comprehensive that the simulation model and its metamodel(s) need never be revisited. In practice, results from experiments often need to be modified, i.e., expanded or thrown out to obtain more detailed information on the simulation performance for a smaller region of the factor combinations. These modifications are determined in large part by the expertise of the analysts. This illustrates a need for semiautomatic methods for suggesting design refinements, which can be tricky. For example, suppose the analysts have built a response-surface model that accurately characterizes simulation performance over a particular region of the factor space. Over time, the external environment changes so that the initial factor-level combinations are no longer of primary interest, and therefore additional experiments are conducted. When is it appropriate to use a global metamodel (with data from all experiments) instead of focusing on several local metamodels (over smaller ranges)? This question merits additional research.

## 6. Case Study: Humanitarian Assistance Operations

Clearly, no single investigation will use all experimental designs described in §4 even though they represent only a subset of possible designs. To illustrate a number of the points made in the paper, we now present a small case summary of an investigation of an agent-based model of humanitarian assistance operations in urban environments. We use a scenario developed by Wolf (2003) but expand the investigation to illustrate the central points of this paper more fully. Additional details are provided in the Online Supplement.

Wolf (2003) examines a humanitarian assistance operation implemented using the MANA software platform (Lauren and Stephen 2002). A convoy with a security escort follows a given route to the most southern of two food-distribution sites in an urban environment. The convoy enters from the northeast corner, traveling west, and then turns south toward its final destination. Initially, the northern civilians make their way to the northern site, while the southern civilians move toward the southern site. As the northern civilians sense the trucks passing by, they speed up and try to follow the trucks. A lone aggressor searches for the convoy, provides harassing fire, and then runs away. The security escort returns fire if it identifies the aggressor, while the convoy responds by speeding up and driving out of the area. Once it reaches the southern site, the convoy begins distributing food. The simulation runs for a fixed time that represents a single day's operation; initial conditions differ across runs due to random initial placement of agents within a defined border. The output measures include the numbers of northern and southern civilians fed, whether or not the aggressor is killed, and whether one of the convoy trucks is destroyed.

Several of the goals for this initial investigation (Wolf 2003, Wolf et al. 2003) are consistent with those in §2.1. One goal is to see whether gaining a better *understanding* of the model's behavior offers general insights to those interested in using agent-based models for humanitarian assistance operations. A second goal is to determine whether a *robust strategy* exists for the convoy; that is, are there choices that improve its ability to feed people over a broad range of environments? If no robust strategy emerges, policies can be *compared* within a few environments to see if appropriate strategies can be identified for more limited ranges of external conditions.

Forty factors are chosen for exploration. This means from the outset that efficient experimental designs are needed to allow a broad look across the factors. The factors are all quantitative, and involve an agent's propensities for movement under different environmental conditions, and its ability to sense, communicate, and interact with other agents. The 15 convoy and security factors are considered decision factors since they reflect actions or capabilities of the Marines. Attributes and behaviors of the civilians (14 factors) and the aggressor (11 factors) characterize a variety of environments in which the food-distribution operation could take place. Few assumptions are made about the nature of the response surface, so the experimental setting falls in the upper right corner of the design space in Figure 1. By appending 16 square random Latin hypercubes (each involving 40 design points), Wolf (2003) examines the impact of simultaneously changing the specified values of these 40 factors. The final experiment (with 50 replications at each of $16 \times 40 = 640$ design points) requires 32,000 simulation runs, but even this relatively large number of runs can be completed in 7.5 hours on a computing cluster.

For exploratory purposes, we begin by averaging the responses at each design point and graphically assessing the results. Rather than summarizing the analysis in Wolf (2003), we present examples of types of graphs that we have found particularly useful for developing an understanding of the system behavior. Unfortunately, static black-and-white pictures cannot fully portray the insights gained from using these multicolor graphs interactively, but we briefly describe the dynamic aspects in the text.

Figure 2 is a *mean diamonds* plot of the average number fed as a function of the communication distance, where this distance indicates how close the civilians must be to the convoy to see or hear it passing by, and how close they must be to other civilians to share this information. This plot was constructed using JMP IN® (SAS Institute 2004), but other statistical packages can produce similar plots. The plotted points are the average number of civilians fed for each of the 640 design points. Each diamond represents a 95% confidence interval for the mean response associated with specific communication distances, i.e., a set of "reasonable" values for the underlying mean response. The diamond's upper and lower points correspond to the upper and lower confidence interval limits, and the center line is the mean. The circles on the right portion of the graph correspond to simultaneous 95% confidence intervals for all pairwise comparisons. These arise from the Tukey-Kramer honestly significant difference (HSD) test, which is an exact test if the sample sizes are the same and a conservative test if they are unequal; see Hayter (1984). Clicking on one of these circles highlights (in boldface) the communication distance in question, and displays the set of all communication distances whose average responses are not significantly different by unitalicizing the labels and displaying both the circles and labels in red. We box these to aid the reader viewing the plot in black and white. This type of graph is useful because it allows the analyst to check the overall direction of main effects, as well as explore whether there are any interesting patterns, clusters, or outliers among the means. We have uncovered software logic problems using similar plots.

Once the data have been screened for extreme outliers, regression models can be used to identify factors and interactions that seem to play important roles in determining the response. Wolf (2003) fits several models, balancing model simplicity against the explanatory power. He examines how the convoy

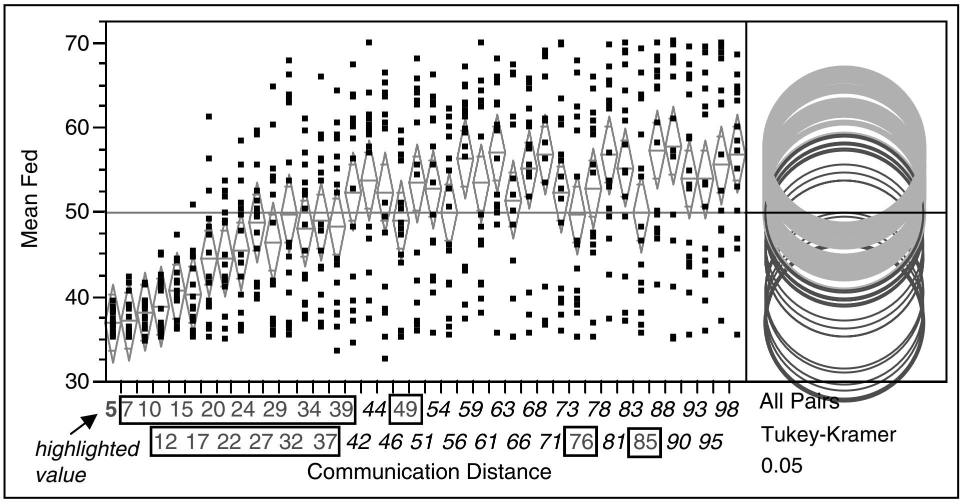

**Figure 2** Mean Diamonds Plot of Average Number Fed vs. Communication Distance

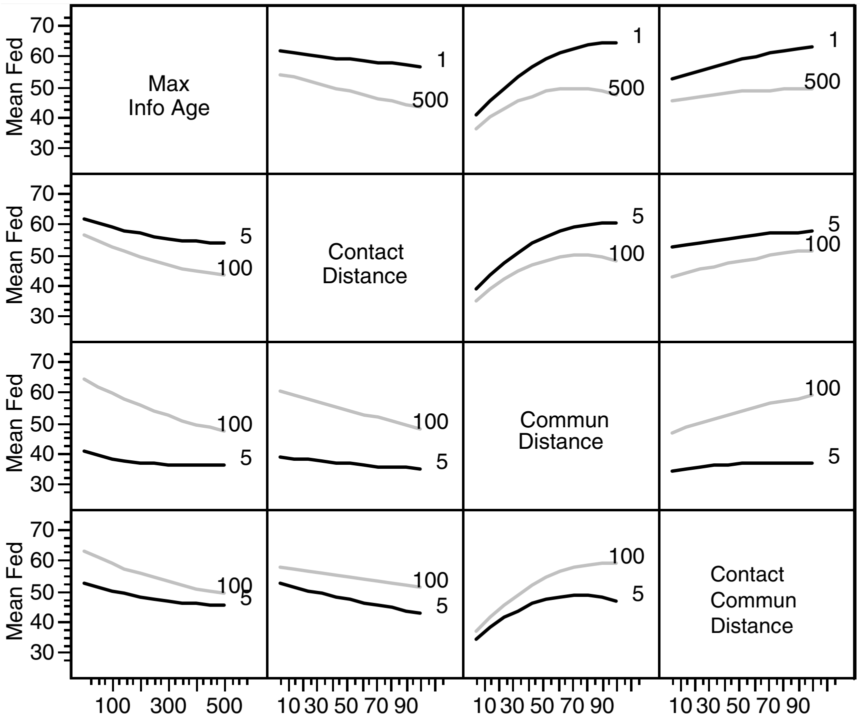

**Figure 3** **Interaction Profiles for Humanitarian Assistance Study**

agents can affect the success of the food-distribution operation by specifying appropriate values for factors under their control, but no robust strategy emerges since he finds that the convoy's decision factors have very little impact on performance. The model is improved substantially if communication distance is used as a surrogate for the convoy broadcasting its presence while traveling and reclassified as a decision factor.

Interpreting the impact of model terms is often difficult from the regression output, particularly when interactions and quadratic effects are present or the factors have very different scales. Interaction profiles, such as those in Figure 3, are useful graphical summaries. The tiny subplots depict the four significant interactions in a model involving only communication factors. The factors are the maximum age of information shared among civilians, how close the civilians need to be to the convoy to begin following it (which also determines how close other civilians must be to influence movement), the communication distance (described earlier), and the communication distance once they come into contact with the convoy (which may differ). Curves indicate quadratic effects and dashed lines indicate that interactions are not present. For example, the third box in the top row shows that when information stays current (maximum information age = 1), increasing the communication distance also increases the average number of civilians fed. Increasing communication distance is also valuable—up to a point—even when the shared information may be old (maximum information age = 500), but fewer are fed than when current information is available.

The above graphs and analytic methods all deal with a single performance measure at a time. A graph often used in the social sciences is the *parallel coordinates plot* (Wegman 1990). This displays several variables (input factors and performance measures) simultaneously and connects the (scaled) variable values for each design point with lines. Figure 4 is a parallel coordinates plot for the humanitarian-assistance study. For simplicity, we plot only 4 of the 40 input factors and four performance measures. Because each input factor takes on 40 values in our structured design, the left of the plot is quite dense. The performance measures do not follow regular patterns.

Once again, plots like Figure 4 are most useful when they can be used interactively, when clicking on a particular line will highlight the values of all variables for the corresponding design point. The three design points corresponding to the highest observed convoy losses are highlighted in Figure 4. These design points are also associated with lower losses for the security escort and moderate probabilities of killing the aggressor, but sizeable numbers of civilians are still fed. The highlighted lines also relate the responses to the input factors. For example, the initial communication distances are all high, while no particular pattern is revealed for the contact distance. Ideally, highlighting design points in the plot also highlights them in the data set so interesting subsets

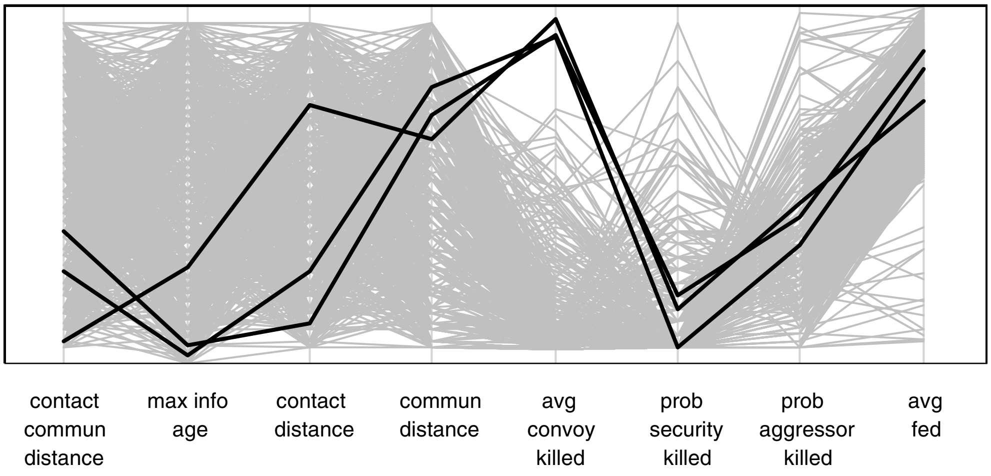

**Figure 4** Parallel Coordinates Plot with Four Factors and Four Performance Measures

of the data can be easily extracted and examined in more detail.

Scatter plots can display how two responses are related to one another, though some care must be taken when the data sets are large. Since multiple points may result in the same pair of response values, a straightforward scatter plot will not necessarily reveal where the majority of the data are clustered. Instead, points can be offset slightly by addition of random noise (sometimes called *jitter*), the point sizes can vary based on the number of data points represented, or points can be plotted using transparent colors so that intense colors correspond to high data densities. (Similar approaches may be valuable for residual plots following regression analysis on very large data sets.) The data can be displayed as a collection of small plots or graphs, e.g., by splitting the data into subsets according to values of two input factors and constructing a scatter plot of two responses for each subset. These collections are also called *small multiples, trellis,* or *tiled plots*.

In all, Wolf (2003) explores several different models based on different sets of potential factors, including models involving the convoy's decision factors, models involving communication factors, and models involving all decision and noise factors. We draw on the results of his exploratory investigation to develop a second experimental design to examine robust strategies more closely. Two of the original 40 factors are dropped (Wolf 2003), and the remaining 38 are divided into three groups: four decision factors (the initial convoy and security movement speeds, along with the northern civilians' communication and contact communication distances), seven environmental factors that show up in at least one of his models, and 27 environmental factors that have little apparent impact on humanitarian assistance operations in the initial experiment. We use a 17-run orthogonal LH design for the four decision factors, and an 8-factor, 33-run nearly-orthogonal LH design for the noise factors. Here we group all 27 not-so-interesting noise factors to form the eighth factor. We then cross the two designs and run 50 independent replications at each of the 561 design points for a total of 28,050 runs (details are provided in the Online Supplement). Our main purpose is to facilitate comparisons in our search for robust solutions by ensuring orthogonality among the decision factors. The near-orthogonality of the noise-factor design also makes it easier to identify strategies that depend on noise factors deemed important in the initial phase of analysis. At the same time, embedding the other noise factors in the nearly-orthogonal design allows us to check the initial conclusion that these factors are less important. We keep the sample size roughly comparable to that of the first experiment so that results can be obtained within one working day.

We use squared-error losses to examine the robustness of the responses. This entails specifying a target $\tau$ that represents the "ideal" response value. In our case, a natural target is the total number of civilians seeking food ($\tau = 70$), although other values are possible. If $\mu_x$ and $\sigma_x^2$ denote the response mean and variance at a particular design point $x$, then the expected loss is proportional to $\sigma_x^2 + (\mu_x - \tau)^2$ (see Ramberg et al. 1991, or Sanchez et al. 1996). We compute the average (scaled) losses for each of the 17 design points. The most robust of these design points has an average loss of 375; its settings correspond to high civilian communication distances with the convoy and security escort traveling at moderately low speeds, and results in feeding 57, on average, which is still far from the highest possible value. Average losses for three other design points are less than 424 (13% larger). The remaining design points are far from robust; their average losses range from 540 to 1170 (44% to 212% larger).

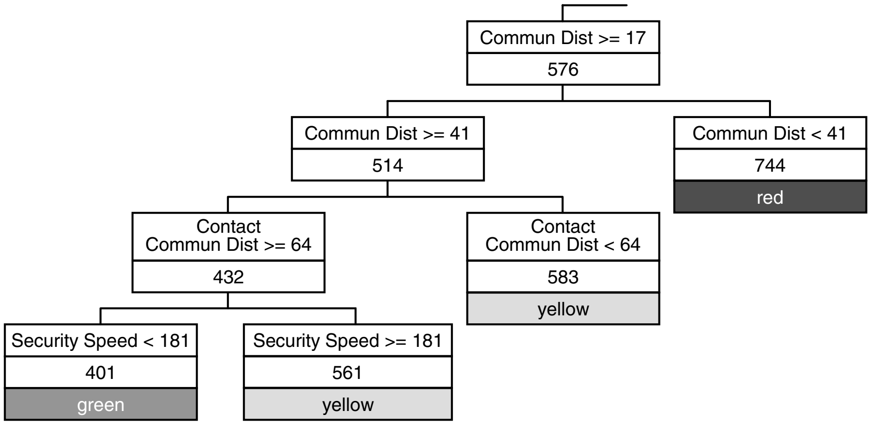

**Figure 5** **Partial Regression Tree**

While the above analysis allows us to compare robustness of specific design points, finding out how the decision factors affect robustness is also of interest. A nonparametric tool we find particularly useful is a regression tree (see Gentle 2002) that recursively partitions the data to provide the most explanatory power for a performance measure of interest. Figure 5 is a portion of the regression tree for the loss (averaged across replications). The first split of the data (not shown) involves communication distance: when this distance is less than 17, the average loss is very high (1158); otherwise the loss is much lower (576). Figure 5 shows the next three splits. The "leaves" at the bottom of the branches in the regression tree denote subsets of the data that are similar in terms of their performance. Additional information, such as the overall $R^2$ value (a measure of the model's explanatory power) and the number of points and response standard deviation associated with each leaf, is available in the computer output. For the tree in Figure 5, $R^2 = 0.29$ (i.e., the tree "explains" 29% of the variability in the response). Constructing a regression tree is an interactive process. Leaves are added until the analyst is satisfied that enough explanatory power is obtained, branches can be pruned to simplify the model, and splits can be forced at certain leaves to examine smaller subsets of the data in more detail. We find it useful to tag the leaves as green, yellow, or red (for good, fair, or poor responses, respectively) when presenting the results.

Regression trees are a nonparametric approach to fitting a response to a set of data. Multiple regression can be used to suggest alternatives (i.e., combinations of factor levels that have not yet been examined) that might perform even better. Accordingly, we fit second-order models of the average loss involving only the decision factors. Another possibility would be to construct separate models for the response mean and variability (Sanchez et al. 1996). A simplified model involving four main effects, one interaction term, and one quadratic works essentially as well ($R^2 = 0.30$) as a full second-order model ($R^2 = 0.31$). (Residual plots reveal some variance heterogeneity, but the OLS estimators are nonetheless unbiased.) The results suggest that the convoy and security escort should travel slowly and broadcast their locations, particularly once they come into contact with civilians. This combination of factor values is not one of the design points for the decision factors, but the most robust of the 17 decision-factor combinations has the highest average communication distance and moderately low speeds. So, the regression results complement those of the regression tree, and can suggest alternatives whose robustness can easily be checked with a set of confirmation runs.

Finally, since we cross an orthogonal design matrix with a nearly-orthogonal one, we can assess the impact of adding noise (environmental) factor terms to our regression model without worrying about multicollinearity. Adding the significant noise factors and decision-by-noise interactions to the model increases $R^2$ from 0.30 to 0.62. An examination of the signs associated with the noise factors and interactions indicates that setting all factors to their low levels is a favorable environment for the relief efforts, while setting all to their high levels is unfavorable. This could be a first step in adapting the convoy's tactics to suit the environment.

As always, the results must be conveyed to decision makers effectively. We find that regression trees are often easier to explain than regression equations. Three-dimensional surface plots (*landscapes*) are useful and easily understood when gridded data are available. So, after a broad exploration is complete, it may be beneficial to run a gridded experiment involving only four or five factors to facilitate displaying the output. Once again, small multiples can be used to compare and contrast the surfaces when holding the undisplayed factors to different levels, or to compare several performance measures. For example,

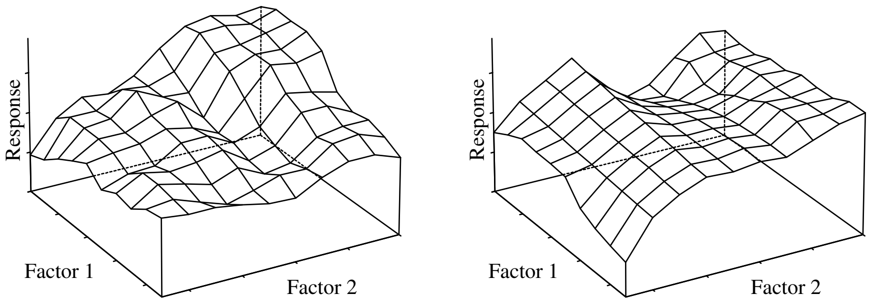

**Figure 6** **Side-by-Side Surface Plots**

each subplot in Figure 6 shows the average response for a performance measure as a function of two input factors, while three other input factors are held at specified levels. It is sometimes informative to plot multiple surfaces on a single graph, such as the lower and upper quartiles for a particular performance measure. For nongridded data, contour plots can provide similar insights. Both surface and contour plots are readily available in spreadsheets and statistical software, though once again there are benefits to interactive graphs. For example, slider bars corresponding to the undisplayed factors' values provide an easy way to see how the landscapes change; some statistical packages have similar profiling tools to see how the fitted surfaces change.

In summary, by using efficient experimental designs coupled with modern graphic and analytic tools, we are able to examine 40 factors at each of 40 levels simultaneously. The setup time is minimal since the factor levels are specified using spreadsheets, and the resulting file is used to update the factor values automatically to run the experiment. The total computational effort for the two experiments (50 replications at 1201 design points) is less than two-thirds the amount required to run a gridded design for any two of the factors at 40 levels ($40^2 = 1600$ design points) or any 11 factors at only two levels ($2^{11} = 2048$). While the results indicate that second-order models suffice for this particular example, our designs require no such assumptions. The graphic and analytic results also provide insights that cannot easily be obtained using trial-and-error or by restricting the list of potential factors.

## 7. Conclusions and Future Research

Our primary goal in writing this paper is to help change the *mindset* of simulation practitioners and researchers. Indeed, we believe that practitioners should view DOE as an integral part of any simulation study, while researchers should move beyond viewing the simulation setting merely as an application area for traditional DOE methods. We advocate thinking first about three potential goals of a simulation experiment, namely, (i) understanding a system, (ii) finding robust solutions, and (iii) comparing two or more systems. We contend that the above goals are often more appropriate than those typically used, namely, testing hypotheses about factor effects, seeking an optimal policy, or making predictions about performance. To illustrate our points, we describe examples from decades of combined experience. We also describe many characteristics of the simulation setting that call for nontraditional designs as part of the simulation analyst's toolkit. In particular, simulation experiments are often characterized by a large number of potential factors, complex response surfaces, time-varying correlated output streams, and multiple performance measures. Analysts also have the opportunity to control simulation-specific factors (such as run lengths, random-number streams, and warm-up periods) that can be exploited for additional design efficiencies. Steady-state simulations offer the possibility for batching output or conducting very long runs that may not have useful analogs in real-world experiments.

Another change in mindset occurs when analysts begin thinking explicitly about sequential experimentation. This has two major implications. First, it means that a sequence of experiments may allow the analyst to gather insights efficiently. Second, even for one-shot experiments, it may be beneficial to sequence the simulation runs appropriately to allow for useful partial information as preliminary results become available or in case the experiment is halted prematurely. We argue that the data-collection effort consists not only of the number and length of the simulation runs, but also the effort required to generate the experimental designs and manage the runs. Emphasizing solely the former may unnecessarily limit the choice of experimental designs. A related idea is the benefit of coding the simulation model in a way that facilitates creating a list of potential factors and subsequently modifying their levels. At the same time, conveying the results effectively remains a challenge for high-dimensional response surfaces.

We discuss several criteria for evaluating designs, and provide guidance on selecting designs suitable for a particular context and using them appropriately. A small case study of a humanitarian assistance operation illustrates several of our major points.

We have listed many problems that require more investigation, resulting in a *research agenda for the design of simulation experiments*. For example, it is important to investigate sequential design and analysis since most computer architectures simulate the scenarios and replicates one after the other. The issue of "robust" instead of "optimal" solutions requires more research. Further work on better matching types of metamodels (and appropriate designs for developing these metamodels) to the characteristics of the simulation setting will continue to be important to analysts and decision makers. Screening designs deserve investigation and application, particularly if they can be incorporated into other designs to reduce the large number of factors at the start of the investigation. Nonsmooth metamodels are needed to represent spikes, thresholds, and chaotic behavior; appropriate designs require more research and software. Multiple outputs might need special designs and analyses for different metamodels—such as Kriging and neural nets—and for evaluating or comparing systems. In addition, approaches that deal effectively with massive data sets, constraints on factor-level combinations, and unstable system configurations are critical if we are to explore large regions of the factor space.

In addition to the research, better software is needed to provide support for appropriate design-and-analysis methods. While gains have been made in recent years—as in visualization software, Kriging, and data-mining tools—there is still much room for improvement. Challenges remain for simulation modelers, software developers, consultants, and analysts. Modelers who use general-purpose software should incorporate sound programming techniques that allow analysts to alter factor levels within input files, rather than burying factor level settings deep inside the code. Simulation-software developers should incorporate experimental-design modules, particularly those involving simulation-specific factors, into their software packages. Software developers should continue developing tools that facilitate experimentation in distributed computing environments. Statistical-software vendors should continue adding design, analysis, and visualization tools that address the three primary goals of simulation experiments. Simulation consultants should consider whether their clients' needs might be best served by incorporating experimental-design approaches. Finally, we challenge simulation researchers and practitioners to continue a dialogue that leads to rapid dissemination of new developments in, and useful applications of, the design and analysis of simulation experiments.

## Acknowledgments

The authors thank the Associate Editor and three referees for their very careful and thorough reviews of the earlier versions of this paper. This work was supported in part by the National Research Council, the U.S. Marine Corps Combat Development Command, and the U.S. Marine Corps Warfighting Laboratory.

## References

Alam, F. M., K. R. McNaught, T. J. Ringrose. 2004. A comparison of experimental designs in the development of a neural network simulation metamodel. *Simulation Modelling: Practice Theory* **12** 559–578.

Alexopoulos, C., D. Goldsman. 2004. To batch or not to batch? *ACM Trans. Model. Comput. Simulation* **14** 76–114.

Angün, E., D. den Hertog, G. Gürkan, J. P. C. Kleijnen. 2002. Response surface methodology revisited. E. Yücesan, C. H. Chen, J. L. Snowdon, J. M. Charnes, eds. *Proc. 2002 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 377–383.

Antioniadis, A., D. T. Pham. 1998. Wavelet regression for random or irregular design. *Computat. Statist. Data Anal.* **28** 353–369.

Banks, J., J. S. Carson, B. L. Nelson, D. M. Nicol. 2005. *Discrete-event Simulation*, 4th ed. Prentice-Hall, Upper Saddle River, NJ.

Batmaz, I., S. Tunali. 2003. Small response surface designs for metamodel estimation. *Eur. J. Oper. Res.* **145** 455–470.

Bettonvil, B., J. P. C. Kleijnen. 1990. Measurement scales and resolution IV designs. *Amer. J. Math. Management Sci.* **10** 309–322.

Bettonvil, B., J. P. C. Kleijnen. 1997. Searching for important factors in simulation models with many factors: Sequential bifurcation. *Eur. J. Oper. Res.* **96** 180–194.

Box, G. E. P., R. Draper. 1987. *Empirical Model-Building with Response Surfaces*. Wiley, New York.

Box, G. E. P., W. G. Hunter, J. S. Hunter. 1978. *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. Wiley, New York.

Brandstein, A. 1999. Operational synthesis: Supporting the maneuver warrior. *Phalanx* **32** 30–31.

Cabrera-Rios, M., C. A. Mount-Campbell, S. A. Irani. 2002. An approach to the design of a manufacturing cell under economic considerations. *Internat. J. Production Econom.* **78** 223–237.

Campolongo, F., J. P. C. Kleijnen, T. Andres. 2000. Screening methods. A. Saltelli, K. Chan, E. M. Scott, eds. *Sensitivity Analysis*. Wiley, New York, 65–89.

Cheng, R. C. H. 1997. Searching for important factors: Sequential bifurcation under uncertainty. S. Andradottir, K. J. Healy, D. H. Withers, B. L. Nelson, eds. *Proc. 1997 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 275–280.

Cheng, R. C. H., J. P. C. Kleijnen. 1999. Improved design of simulation experiments with highly heteroskedastic responses. *Oper. Res.* **47** 762–777.

Chick, S. E., K. Inoue. 2001. New procedures to select the best simulated system using common random numbers. *Management Sci.* **47** 1133–1149.

Cioppa, T. M. 2002. Efficient nearly orthogonal and space-filling experimental designs for high-dimensional complex models. Doctoral dissertation, Operations Research Department, Naval Postgraduate School, Monterey, CA, http://handle.dtic.mil/100.2/ADA406967.

Cioppa, T. M., T. W. Lucas, S. M. Sanchez. 2004. Military applications of agent-based models. R. G. Ingalls, M. D. Rossetti, J. S. Smith, B. A. Peters, eds. *Proc. 2004 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 171–180.

Cioppa, T. M., T. W. Lucas. 2005. Efficient nearly orthogonal and space-filling Latin hypercubes. Working paper, Department of Operations Research, Naval Postgraduate School, Monterey, CA.

Clarke, S. M., J. H. Griebsch, T. W. Simpson. 2005. Analysis of support vector regression for approximation of complex engineering analyses. *ASME J. Mech. Design* **127**.

Crary Group. 2004. WebDOE. http://www.webdoe.cc/.

Cressie, N. A. C. 1993. *Statistics for Spatial Data*. Revised ed. Wiley, New York.

Dean, A. M., S. M. Lewis. 2004. *Screening*. Springer-Verlag, New York.

Dewar, J. A., S. C. Bankes, J. S. Hodges, T. W. Lucas, D. K. Saunders-Newton, P. Vye. 1996. Credible uses of the distributed interactive simulation (DIS) system. MR-607-A, RAND Corporation, Santa Monica, CA, http://www.rand.org/publications/MR/MR607.pdf.

Donohue, J. M., E. C. Houck, R. H. Myers. 1993. Simulation designs and correlation induction for reducing second-order bias in first-order response surfaces. *Oper. Res.* **41** 880–902.

Fedorov, V. V. 1972. *Theory of Optimal Experiments*. Academic Press, New York.

Fu, M. C. 2002. Optimization for simulation: Theory vs. practice. *INFORMS J. Comput.* **14** 192–215.

Gentle, J. E. 2002. *Computational Statistics*. Springer, New York.

Gill, A., D. Grieger. 2003. Comparison of agent based distillation movement algorithms. *Military Oper. Res.* **8** 5–16.

Giunta, A. A., L. T. Watson. 1998. A comparison of approximating modeling techniques: Polynomial versus interpolating models. *7th AIAA/USAF/NASA/ISSMO Sympos. Multidisciplinary Anal. Optim.* St. Louis, MO, AIAA 98–4758, 381–391.

Goldsman, D., S.-H. Kim, W. S. Marshall, B. L. Nelson. 2002. Ranking and selection for steady-state simulation: Procedures and perspectives. *INFORMS J. Comput.* **14** 2–19.

Hayter, A. J. 1984. A proof of the conjecture that the Tukey-Kramer multiple comparisons procedure is conservative. *Ann. Math. Statist.* **12** 61–75.

Helton, J. C., M. G. Marietta, eds. 2000. Special issue: 1996 performance assessment for the waste isolation pilot plant. *Reliability Engrg. Systems Safety* **69** 1–454.

Hodges, J. S. 1991. Six (or so) things you can do with a bad model. *Oper. Res.* **39** 355–365.

Hodges, J. S., J. A. Dewar. 1992. Is it you or your model talking? A framework for model validation. Report R-4114-AF/A/OSD, RAND Corporation, Santa Monica, CA.

Holcomb, D., D. C. Montgomery, W. M. Carlyle. 2000a. Analysis of supersaturated designs. *J. Quality Tech.* **35** 13–27.

Holcomb, D., D. C. Montgomery, W. M. Carlyle. 2000b. Some combinatorial aspects, construction methods, and evaluation criteria for supersaturated designs. *Quality Reliability Engrg. Internat.* **18** 299–304.

Horne, G., S. Johnson, eds. 2002. *Maneuver Warfare Science 2002*. USMC Project Albert, Quantico, VA.

Horne, G., S. Johnson, eds. 2003. *Maneuver Warfare Science 2003*. USMC Project Albert, Quantico, VA.

Horne, G., M. Leonardi, eds. 2001. *Maneuver Warfare Science 2001*. Marine Corps Combat Development Command, Defense Automated Printing Service, Quantico, VA.

Hsu, J. C. 1996. *Multiple Comparisons; Theory and Methods*. Chapman & Hall, London.

Irizarry, M. de los A., M. E. Kuhl, E. K. Lada, S. Subramanian, J. R. Wilson. 2003. Analyzing transformation-based simulation metamodels. *IIE Trans.* **35** 271–283.

Jacobson, S., A. Buss, L. Schruben. 1991. Driving frequency selection for frequency domain simulation experiments. *Oper. Res.* **39** 917–924.

Jin, R., W. Chen, T. Simpson. 2001. Comparative studies of metamodeling techniques under multiple modeling criteria. *J. Structural Optim.* **23** 1–13.

Jin, R., W. Chen, A. Sudjianto. 2002. On sequential sampling for global metamodeling in engineering design. *Proc. DETC '02, ASME 2002 Design Engrg. Tech. Conf. Comput. Inform. Engrg. Conf.* DETC2002/DAC-34092, Montreal, Canada, 1–10.

Kelton, W. D., R. M. Barton. 2003. Experimental design for simulation. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, eds. *Proc. 2003 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 59–65.

Khuri, A. I. 1996. Multiresponse surface methodology. S. Ghosh, C. R. Rao, eds. *Handbook of Statistics*, Volume 13. Elsevier, Amsterdam, The Netherlands.

Kiefer, J. 1959. Optimal experimental designs (with comments). *J. Roy. Statist. Soc. Ser. B* **21** 272–319.

Kleijnen, J. P. C. 1974–1975. *Statistical Techniques in Simulation*, Volumes I, II. Marcel Dekker, Inc., New York.

Kleijnen, J. P. C. 1987. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, Inc., New York.

Kleijnen, J. P. C. 1995. Case study: Statistical validation of simulation models. *Eur. J. Oper. Res.* **87** 21–34.

Kleijnen, J. P. C. 1998. Design for sensitivity analysis, optimization, and validation of simulation models. J. Banks, ed. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. Wiley, New York, 173–223.

Kleijnen, J. P. C. 2005. An overview of the design and analysis of simulation experiments for sensitivity analysis. *Eur. J. Oper. Res.* **164** 287–300.

Kleijnen, J. P. C., E. Gaury. 2003. Short-term robustness of production management systems: A case study. *Eur. J. Oper. Res.* **148** 452–465.

Kleijnen, J. P. C., O. Pala. 1999. Maximizing the simulation output: A competition. *Simulation* **73** 168–173.

Kleijnen, J. P. C., R. G. Sargent. 2000. A methodology for the fitting and validation of metamodels in simulation. *Eur. J. Oper. Res.* **120** 14–29.

Kleijnen, J. P. C., M. T. Smits. 2003. Performance metrics in supply chain management. *J. Oper. Res. Soc.* **54** 507–514.

Kleijnen, J. P. C., W. van Groenendaal. 1992. *Simulation: A Statistical Perspective*. Wiley, Chichester, UK.

Kleijnen, J. P. C., W. C. M. van Beers. 2004. Application-driven sequential designs for simulation experiments: Kriging metamodeling. *J. Oper. Res. Soc.* **55** 876–883.

Kleijnen, J. P. C., W. C. M. van Beers. 2005. Robustness of Kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *Eur. J. Oper. Res.* **165** 826–834.

Kleijnen, J. P. C., B. Bettonvil, F. Persson. 2003. Robust solutions for supply chain management: Simulation and risk analysis of the Ericsson case study. Working paper, Tilburg University, Tilburg, The Netherlands.

Kleijnen, J. P. C., B. Bettonvil, F. Persson. 2005. Finding the important factors in large discrete-event simulation: Sequential bifurcation and its applications. A. M. Dean, S. M. Lewis, eds. *Screening*. Springer-Verlag, New York.

Kleijnen, J. P. C., R. C. H. Cheng, V. B. Melas. 2000. Optimal design of experiments with simulation models of nearly saturated queues. *J. Statist. Planning Inference* **85** 19–26.

Kleijnen, J. P. C., A. J. Feelders, R. C. H. Cheng. 1998. Bootstrapping and validation of metamodels in simulation. D. J. Medeiros, E. F. Watson, J. S. Carson, M. S. Manivannan, eds. *Proc. 1998 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 701–706.

Kleijnen, J. P. C., A. J. van den Burg, R. T. H. van der Ham. 1979. Generalization of simulation results: Practicality of statistical methods. *Eur. J. Oper. Res.* **3** 50–64.

Kleijnen, J. P. C., A. Vonk Noordegraaf, M. Nielen. 2001. Sensitivity analysis of censored output through polynomial, logistic and tobit models: Theory and case study. B. A. Peters, J. S. Smith, D. J. Medeiros, M. W. Rohrer, eds. *Proc. 2001 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 486–491.

Koehler, J. R., A. B. Owen. 1996. Computer experiments. S. Ghosh, C. R. Rao, eds. *Handbook of Statistics*, Volume 13. Elsevier, Amsterdam, The Netherlands, 261–308.

Kutner, M. H., C. J. Nachtsheim, J. Neter, W. Li. 2004. *Applied Linear Statistical Models*, 5th ed. McGraw-Hill/Irwin, Boston, MA.

Lauren, M. K., R. T. Stephen. 2002. Map-aware non-uniform automata—A New Zealand approach to scenario modelling. *J. Battlefield Tech.* **5** 27–31.

Law, A. M., W. D. Kelton. 2000. *Simulation Modeling and Analysis*, 3rd ed. McGraw-Hill, New York.

Lin, D. K. J. 1995. Generating systematic supersaturated designs. *Technometrics* **37** 213–225.

Lophaven, S. N., H. B. Nielsen, J. Sondergaard. 2002. DACE: A Matlab Kriging toolbox, version 2.0.IMM Technical University of Denmark, Lyngby, Denmark, http://www.imm.dtu.dk/~hbn/dace/.

Lucas, T. W., S. C. Bankes, P. Vye. 1997. Improving the analytic contribution of advanced warfighting experiments (AWEs). Documented Briefing DB-207-A. RAND Corporation, Santa Monica, CA.

Lucas, T. W., S. M. Sanchez, L. Brown, W. Vinyard. 2002. Better designs for high-dimensional explorations of distillations. G. Horne, S. Johnson, eds. *Maneuver Warfare Science 2002*. USMC Project Albert, Quantico, VA, 17–46.

Lucas, T. W., S. M. Sanchez, T. M. Cioppa, A. I. Ipekci. 2003. Generating hypotheses on fighting the global war on terrorism. G. Horne, S. Johnson, eds. *Maneuver Warfare Science 2003*. USMC Project Albert, Quantico, VA, 117–137.

Martinez, W. L., A. R. Martinez. 2002. *Computational Statistics Handbook with MATLAB*. Chapman & Hall/CRC, Boca Raton, FL.

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

Moeeni, F., S. M. Sanchez, A. J. Vakharia. 1997. A robust design methodology for Kanban system design. *Internat. J. Production Res.* **35** 2821–2838.

Montgomery, D. C. 2000. *Design and Analysis of Experiments*, 5th ed. Wiley, New York.

Morrice, D. J., I. R. Bardhan. 1995. A weighted least squares approach to computer simulation factor screening. *Oper. Res.* **43** 792–806.

Morris, M. D., T. J. Mitchell. 1995. Exploratory designs for computational experiments. *J. Statist. Planning Inference* **43** 381–402.

Meyer, T., S. Johnson. 2001. Visualization for data farming: A survey of methods. G. Horne, M. Leonardi, eds. *Maneuver Warfare Science 2001*. Marine Corps Combat Development Command, Quantico, VA, 15–30.

Myers, R. H., D. C. Montgomery. 2002. *Response Surface Methodology: Process and Product Optimization using Designed Experiments*, 2nd ed. Wiley, New York.

Myers, R. H., A. I. Khuri, G. Vining. 1992. Response surface alternatives to the Taguchi robust design parameter approach. *Amer. Statist.* **46** 131–139.

Nakayama, M. 2003. Analysis of simulation output. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, eds. *Proc. 2003 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 49–58.

Nelson, B. L., D. Goldsman. 2001. Comparisons with a standard in simulation experiments. *Management Sci.* **47** 449–463.

Pacheco, J., C. Amon, S. Finger. 2003. Incorporating Information from Replications into Bayesian Surrogate Models. *2003 ASME Design Engrg. Tech. Conf.* DETC2003/DTM-48644, Chicago, IL.

Pukelsheim, F. 1993. *Optimal Design of Experiments*. Wiley, New York.

Ramberg, J. S., S. M. Sanchez, P. J. Sanchez, L. J. Hollick. 1991. Designing simulation experiments: Taguchi methods and response surface metamodels. B. L. Nelson, W. D. Kelton, G. M. Clark, eds. *Proc. 1991 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 167–176.

Rao, C. R. 1967. Least squares theory using an estimated dispersion matrix and its application to measurement of signals. *Proc. 5th Berkeley Sympos. Math. Statist. Probab.* I. University of California Press, Berkeley, CA, 355–372.

Rechtschaffner, R. L. 1967. Saturated fractions of $2^n$ and $3^n$ factorial designs. *Technometrics* **9** 569–575.

Sacks, J., W. J. Welch, T. J. Mitchell, H. P. Wynn. 1989. Design and analysis of computer experiments (includes Comments and Rejoinder). *Statist. Sci.* **4** 409–435.

Saltelli, A., S. Tarantola, P. S. Chan. 1999. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics* **41** 39–56.

Sanchez, P. J., A. H. Buss. 1987. A model for frequency domain experiments. A. Thesen, H. Grant, W. D. Kelton, eds. *Proc. 1987 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 424–427.

Sanchez, S. M. 2000. Robust design: Seeking the best of all possible worlds. J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick, eds. *Proc. 2000 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 69–76.

Sanchez, S. M., T. W. Lucas. 2002. Exploring the world of agent-based simulations: Simple models, complex analyses. E. Yücesan, C.-H. Chen, J. L. Snowdon, J. Charnes, eds. *Proc. 2002 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 116–126.

Sanchez, S. M., H.-F. Wu. 2003. Frequency-based designs for terminating simulations: A peace-enforcement application. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, eds. *Proc. 2003 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 952–959.

Sanchez, S. M., P. J. Sanchez, J. S. Ramberg. 1998. A simulation framework for robust system design. B. Wang, ed. *Concurrent Design of Products, Manufacturing Processes and Systems*. Gordon and Breach, New York, 279–314.

Sanchez, S. M., L. D. Smith, E. C. Lawrence. 2005. Tolerance design revisited: Assessing the impact of correlated noise factors. Working paper, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Sanchez, S. M., P. J. Sanchez, J. S. Ramberg, F. Moeeni. 1996. Effective engineering design through simulation. *Internat. Trans. Oper. Res.* **3** 169–185.

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

SAS Institute, Inc. 2004. *JMP IN® Version 5.1 for Windows, Macintosh, and Unix*. Duxbury Thompson Learning, Pacific Grove, CA.

Sasena, M. J., P. Y. Papalambros, P. Goovaerts. 2002. Exploration of metamodeling sampling criteria for constrained global optimization. *Engrg. Optim.* **34** 263–278.

Schmeiser, B. W. 1982. Batch size effects in the analysis of simulation output. *Oper. Res.* **30** 556–568.

Schruben, L. W., V. J. Cogliano. 1987. An experimental procedure for simulation response surface model identification. *Comm. ACM* **30** 716–730.

Shoemaker, A. C., K.-L. Tsui, C. F. J. Wu. 1991. Economical experimentation methods for robust design. *Technometrics* **33** 415–427.

Simon, H. A. 1981. *The Sciences of the Artificial*, 2nd ed. MIT Press, Cambridge, MA.

Simpson, T. W., D. K. J. Lin, W. Chen. 2001. Sampling strategies for computer experiments: Design and analysis. *Internat. J. Reliability Appl.* **2** 209–240.

Smith, L. D., S. M. Sanchez. 2003. Assessment of business potential at retail sites: Empirical findings from a U. S. supermarket chain. *Internat. Rev. Retail, Distribution Consumer Res.* **13** 37–58.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization; Estimation, Simulation, and Control*. Wiley, New York.

Steiger, N. M., E. K. Lada, J. R. Wilson, J. A. Joines, C. Alexopoulos, D. Goldsman. 2005. ASAP3: A batch means procedure for steady-state simulation output analysis. *ACM Trans. Model. Comput. Simulation* **15** 39–73.

Sugiyama, S. O., J. W. Chow. 1997. @Risk, riskview and bestFit. *OR/MS Today* **24** 64–66.

Taguchi, G. 1987. *System of Experimental Designs*, Vol. 1, 2. UNIPUB/Krauss International, White Plains, NY.

Trocine, L., L. C. Malone. 2001. An overview of newer, advanced screening methods for the initial phase in an experimental design. B. A. Peters, J. S. Smith, D. J. Medeiros, M. W. Rohrer, eds. *Proc. 2001 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 169–178.

Tufte, E. R. 1990. *Envisioning Information.* Graphics Press, Cheshire, CT.

Tunali, S., I. Batmaz. 2000. Dealing with the least squares regression assumptions in simulation metamodeling. *Comput. Indust. Engrg.* **38** 307–320.

van Beers, W. C. M., J. P. C. Kleijnen. 2003. Kriging for interpolation in random simulation. *J. Oper. Res. Soc.* **54** 255–262.

van Beers, W. C. M., J. P. C. Kleijnen. 2004. Kriging interpolation in simulation: A survey. R. G. Ingalls, M. D. Rossetti, J. S. Smith, B. A. Peters, eds. *Proc. 2004 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 113–121.

Vinyard, W., T. W. Lucas. 2002. Exploring combat models for nonmonotonicities and remedies. *PHALANX* **35** 19, 36–38.

Vonk Noordegraaf, A., M. Nielen, J. P. C. Kleijnen. 2003. Sensitivity analysis by experimental design and metamodelling: Case study on simulation in national animal disease control. *Eur. J. Oper. Res.* **146** 433–443.

Wan, H., B. Ankenman, B. L. Nelson. 2003. Controlled sequential bifurcation: A new factor-screening method for discrete-event simulation. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, eds. *Proc. 2003 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 565–573.

Wan, S. C. 2002. An exploratory analysis on the effects of human factors on combat outcomes. M. S. thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA, http://handle.dtic.mil/100.2/ADA403526.

Watson, A. G., R. J. Barnes. 1995. Infill sampling criteria to locate extremes. *Math. Geology* **27** 589–608.

Wegman, E. M. 1990. Hyperdimensional data analysis using parallel coordinates. *J. Amer. Statist. Association* **85** 664–674.

Wolf, E. S. 2003. Using agent-based distillations to explore logistics support to urban, humanitarian assistance/disaster relief operations. M.S. thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Wolf, E. S., S. M. Sanchez, N. Goerger, L. Brown. 2003. Using agents to model logistics. Working paper, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Ye, K. Q. 1998. Orthogonal column Latin hypercubes and their application in computer experiments. *J. Amer. Statist. Association* **93** 1430–1439, http://handle.dtic.mil/100.2/ADA418300.

Zeigler, B. P., K. Praehofer, T. G. Kim. 2000. *Theory of Modeling and Simulation*, 2nd ed. Academic Press, San Diego, CA.