Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

WORK SMARTER, NOT HARDER: GUIDELINES FOR DESIGNING SIMULATION EXPERIMENTS

Susan M. Sanchez

Operations Research Department and
Graduate School of Business & Public Policy
Naval Postgraduate School
Monterey, CA 93943-5219, U.S.A.

ABSTRACT

We present the basic concepts of experimental design, the types of goals it can address, and why it is such an important and useful tool for simulation. A well-designed experiment allows the analyst to examine many more factors than would otherwise be possible, while providing insights that could not be gleaned from trial-and-error approaches or by sampling factors one at a time. We focus on experiments that can cut down the sampling requirements of some classic designs by orders of magnitude, yet make it possible and practical to develop an understanding of a complex simulation model and gain insights into its behavior. Designs that we have found particularly useful for simulation experiments are illustrated using simple simulation models, and we provide links to other resources for those wishing to learn more. Ideally, this tutorial will leave you excited about experimental designs—and prepared to use them—in your upcoming simulation studies.

1 INTRODUCTION

The process of building, verifying, and validating a simulation model can be arduous, but once it is complete, it's time to have the model work for you. One extremely effective way of accomplishing this is to use experimental designs to help explore your simulation model.

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

The field called Design of Experiments (DOE) has been around for a long time. Many of the classic experimental designs can be used in simulation studies. We discuss a few in this paper to explain the concepts and motivate the use of experimental design (see also Chapter 12 of Law and Kelton 2000). However, the environments in which real-world experiments are performed can be quite different from the simulation environment. Table 1, adapted from Sanchez and Lucas (2002), lists some of the assumptions made in traditional DOE settings, as well as features that characterize many simulation settings.

Three fundamental concepts in DOE are control, replication, and randomization. *Control* means that the experiment is conducted in a systematic manner after explicitly considering potential sources of error, rather than by using a trial-and-error approach. This tutorial should give you a good understanding of controlled experiments. *Replication* can be viewed as a way to gain enough data to achieve narrow confidence intervals and powerful hypothesis tests, or for graphical methods to reveal the important characteristics of your simulation model. In physical experiments, *randomization* provides a probabilistic guard against the possibility of unknown, hidden sources of bias surfacing to create problems with your data.

In this introductory tutorial, we focus on setting up single-stage experiments to address the first goal, and touch briefly on the second. Although some very simple simulation models are used as examples in this paper, the designs we describe have been extremely useful for investigating more complex simulation models in a variety of application areas. For a detailed discussion of the philosophy and tactics of simulation experiments, a more extensive catalog of potential designs (including sequential approaches), and a comprehensive list of references, see Kleijnen et al. (2005).

Table 1: The Experimental Environment

Traditional DOE Assumptions	Simulation Model Characteristics
Small or moderate number of factors	Large number of factors
Linear or low-order effects	Complex response surfaces
Sparse effects	Many substantial effects
Negligible higher-order interactions	Substantial higher-order interactions
Homogeneous errors	Heterogeneous errors
Normally distributed errors	Various error distributions
Univariate response	Many performance measures of interest

Additional examples, software for generating experimental designs, and tips for implementing the experiment once a design has been chosen, will be provided during the tutorial.

We will not cover examples of ranking and selection (R&S) or multiple comparison procedures (MCP), although these are useful approaches for comparing the merits of different policies or qualitatively different systems. We refer the reader instead to Goldsman, Kim, and Nelson (2005) for an overview. We will also not cover techniques for simulation optimization (Fu 2002).

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

2 NUTS AND BOLTS

Our overarching goal is to provide you with some useful tools for gaining a great deal of information in a short amount of time. This includes the time you need to spend to set up the experiments and consolidate the results, as well as the computer time spent for conducting the runs.

2.1 Terminology and Notation

In DOE terms, experimental designs indicate how to vary the settings of *factors* (sometimes called *variables*) to see whether and how they affect the *response*. A *factor* can be qualitative or quantitative. Potential factors in simulation experiments include the *input parameters* or *distributional parameters* of a simulation model. For example, a simple $M/M/1$ queueing system might have some quantitative factors (such as the mean customer inter-arrival time and mean service time), and some qualitative factors (such as LIFO or FIFO processing, priority classes, and preemptive or non-preemptive service rules).

Different types of simulation studies involve different types of *experimental units*. For a Monte Carlo simulation, the experimental unit is a single observation. For discrete-event stochastic simulation studies, the experimental unit more often represents output from a run or a batch that is averaged or aggregated to yield a single output value. The run is the appropriate experimental unit for terminating simulations. If the output of interest is the time until termination, or the number of events prior to termination, then the run's output is already in the form of a single number. When runs form the experimental units for nonterminating simulations, and steady-state performance measures are of interest, care must be taken to delete data from the simulation's warm-up period before performing the averaging or aggregation.

Mathematically, let k denote the number of factors in our experiment, let $X_1, \ldots, X_k$ denote the factors, and let Y denote a response of interest. This is sometimes called a measure of effectiveness (MOE) or measure of performance (MOP). Sometimes graphical methods are the best way to gain insight about Y's, but often we will be interested in constructing *response surface metamodels* that approximate the relationships between the factors and the responses with statistical models (typically regression models).

Unless otherwise stated, we will assume that the X_i's are all quantitative. A *main-effects model* means we assume

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon \qquad (1) \\ &= \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \varepsilon, \end{aligned}$$

where the ε's are independent random errors. Ordinary least squares regression assumes that the ε's are also identically distributed, but the regression coefficients are still unbiased estimators even if the underlying variance is not constant.

"Quadratic effects" means we will include terms like X_1^2 as potential explanatory variables for Y. Similarly, "two-way interactions" are terms like $X_1 X_2$. A *second-order model* includes both quadratic effects and two-way interactions, i.e.,

$$\begin{aligned} Y &= \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{k} \beta_{i,i} X_i^2 \qquad (2) \\ &\quad + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{i,j} X_i X_j + \varepsilon, \end{aligned}$$

although it is best to fit this equation after centering the quadratic and interaction terms, as in (3):

$$\begin{aligned} Y &= \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{k} \beta_{i,i}(X_i - \overline{X}_i)^2 \qquad (3) \\ &+ \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{i,j}(X_i - \overline{X}_i)(X_j - \overline{X}_j) + \varepsilon. \end{aligned}$$

In general, a *design* is a matrix where every columns corresponds to a factor, and the entries within the column correspond to settings for this factor. Each row represents a particular combination of factor levels, and is called a *design point*. If the entries in the rows correspond to the actual settings that will be using for the experiment, these are called *natural levels*. Coding the levels is a convenient way to allow the same basic design to be reused for any experiment involving the same number of factors and the same numbers of levels. Different codes are possible, but a convenient one for quantitative data is to specify the low and high coded levels as -1 and $+1$, respectively. Table 2 shows a simple experiment, in both natural and coded levels, that could be conducted on an $M/M/1$ queue.

Table 2: Simple Experimental Design for an $M/M/1$ Queue

	Natural Levels		Coded Levels	
	Interarrival	Service	Interarrival	Service
Design	Rate	Rate	Rate	Rate
Point	λ	μ	λ	μ
1	16	20	-1	-1
2	18	20	$+1$	-1
3	16	22	-1	0
4	18	22	$+1$	0
5	16	24	-1	$+1$
6	18	24	$+1$	$+1$

If we repeat the whole design matrix, this is called a *replication* of the design. Let N be the number of design points, and b be the number of replications. Then the total number of experimental units, whether runs or batches, is $N_{tot} = Nb$.

2.2 Pitfalls to Avoid

Two types of studies are sometimes called "experiments," but they do not fit an example of a well-designed experiment. The first often arises from several people sitting around the table, each of whom has his or her own idea about what constitutes "good" or "interesting" combinations of factor settings. This may lead to the investigation of a handful of design points where many factors change simultaneously. For illustration purposes, consider an agent-based simulation model of the children's game of capture-the-flag, where an agent attempts to sneak up on the other team's flag, grab it, and run away. Suppose that only two design points are used, corresponding to different settings for speed (X_1) and stealth (X_2), with the results in Figure 1. One subject-matter expert might claim these results indicate that high stealth is of primary importance, another might claim that speed is the key factor for success, and a third that they are equally important. There is *no way* to resolve these differences of opinion without collecting more data. In statistical terms, the effects of stealth and speed are said to be *confounded* with each other. In practice, simulation models easily have tens or hundreds of potential factors whose settings can be altered. A handful of haphazardly chosen scenarios, or a trial-and-error approach, can end up using a great deal of time without yielding answers to the fundamental questions of interest.

Speed	Stealth	Success?
Low	Low	No
High	High	Yes

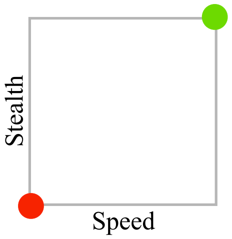

Figure 1: Confounded Factor Effects for Capture-the-Flag

The second type of study that can be problematic occurs when people start with a "baseline" scenario and vary one factor at a time. Revisiting the capture-the-flag example, suppose the baseline corresponds to low stealth and low speed. Varying each factor, in turn, to its high level yields the results of Figure 2. It appears that *neither* factor is important, so someone using the simulation results to decide whether to play the game might just go home instead.

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

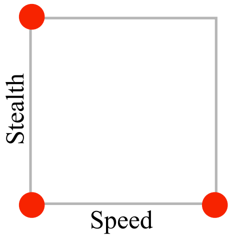

Figure 2: One-at-a-Time Sampling for Capture-the-Flag

However, if all four combinations of speed and stealth (low/low, low/high, high/low, and high/high) were sampled, it would be apparent that success requires both speed and stealth to be at high settings. This means the factors interact—and if there are interactions, one-at-a-time sampling will never uncover them!

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

2.3 Example: Why Projects are Always Late

One well-known problem in operations research is called *project management*. A sequence of tasks are performed where some of the tasks must be completed before others can be started, while others can be worked on concurrently. A precedence diagram (Figure 3) is a graphical way to represent these relationships. The tasks relate to one another in terms of the job completion time. Each node on the diagram corresponds to a task that must be done, and an arrow from node A to node B indicates that task A must be completed before task B can begin. One convention is that we specify "Start project" and "End project" tasks so that every task is on at least one path from the beginning to the end of the project.

In addition to the precedence information, we also need to keep track of the times required to complete the various tasks. The mean completion times appear above the nodes in Figure 3. This graph is so simple that—if all tasks take their average time to complete—the project clearly cannot finish in under 27 days, since the path of A-E-F-G-H requires 27 days to finish.

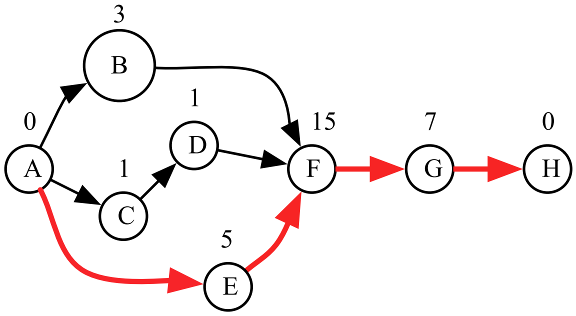

Figure 3: Project Management Precedence Diagram

A simple technique called PERT (Program Evaluation Review Technique) makes it easy to identify this so-called *critical path* for even larger networks. A related approach is CPM (Critical Path Management), which considers how tasks might be cost-effectively expedited. Sometimes these are lumped together and just referred to as PERT/CPM (Hillier and Lieberman 2005). A probabilistic version of PERT takes into account the variability of tasks on the critical path. Consider the task time means and standard deviations shown in Table 3. If the tasks times are assumed to be independent, then the mean and variance on path $\mathcal{C}_p$ ={A, E, F, G, H} (ignoring all other tasks) are

$$\mu_{tot} = \sum_{i \in \mathcal{C}} \mu_i = 5 + 15 + 7 = 27, \text{ and} \tag{4}$$

$$\sigma^2_{tot} = \sum_{i \in \mathcal{C}} \sigma^2_i = 2^2 + 3^2 + 1^2 = 14. \tag{5}$$

If the individual task time distributions nearly normal, or if many tasks lie on the critical path, then the central limit theorem can be invoked. Quantiles from the normal distribution can be used to estimate the probabilities of completing (or failing to complete) the project in a specified time.

Table 3: Task Time Distributional Parameters

Task i	Description	μ_i (days)	σ_i (days)
A	Start	0	0
B		3	0.5
C		1	0.1
D		1	0.2
E		5	2
F		15	3
G		7	1
H	End	0	0

The title of this section is "why projects are always late," so what might go wrong if the calculations in equations (4) and (5) are used? Sometimes you might not get the full benefit if a task on path $\mathcal{C}_p$ finishes early. For this example, if tasks F and G are expedited, or by chance their completion times are less than the expected means, this will benefit the project. However, suppose you happen to spend only one day on Task E but all other tasks take their average times to complete. The "new" critical path will be A-B-F-G-H for a total of 25 days. You will shorten a "bottleneck" task by four days but only save two days on the overall project. PERT/CPM does not account for variations of the critical path itself.

If we actually knew the true task time means and standard deviations, we could do a simple Monte Carlo simulation. Each replication would involve generating completion times for each task based on the task means, standard deviations, and normality assumptions, and using the precedence diagram to determine the time needed to complete the entire project. A frequency distribution of the total project completion time, as well the proportion of time each task appears on the critical path, could be built by replicating the experiment. These, in turn, might provide useful insights to a project manager.

It is rare in practice that we "know" such detail about the inputs to the simulation model. A validated simulation model should reflect the essential characteristics of the real-world system, but the very act of modeling means that simplifying assumptions will be made. For this project management example, we have implicitly assumed independence among the task times, specific distributions for the task time variability (normal), as well as specific parameters for these distributions (the μ_i's and σ_i's). Instead, suppose the project manager and the simulation analyst have determined what they consider reasonable low and high values for the task means and standard deviations.

Real-world projects often have many more tasks and more complicated precedence structures than that of of Figure 3. Consider a more complex project where there are 26 tasks (A-Z). Suppose that 19 of these tasks are considered to have deterministic task times, ranging from 100 minutes to 1,000 minutes. Information about the low and high levels for the task time distributional parameters for the other seven tasks are provided in Table 4. For now, we retain the assumptions of normality and independence for the task times.

Table 4: Low and High Factor Settings for Project Management Factors

Task i	Range for μ_i	Range for σ_i
B	640 – 660	10 – 16
E	1,200 – 1,600	50 – 200
F	280 – 320	4 – 10
P	670 – 700	0 – 2
Q	9 – 39	1 – 3
S	900 – 1,100	0 – 30
T	280 – 320	4 – 10

In the next sections, we show how treating some or all of these as factors in well-designed experiments allows us to explore the system, gain insights about which of the factors or interactions have the most influence on the response, or seek robust solutions. Although the project management example is a terminating simulation, the designs can also be used for truncated runs or batches when exploring a steady-state system simulation.

3 USEFUL DESIGNS

3.1 What Works When

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

Factorials (or gridded designs) are perhaps the easiest to discuss: they examine all possible combinations of the factor levels for each of the X_i's. A shorthand notation for the design is m^k, which means k factors are investigated, each at m levels, in a total of m^k design points. We can write designs where different sets of factors are investigated at different numbers of levels as, e.g., $m_1^{k_1} \times m_2^{k_2}$. These are sometimes called *crossed* designs. For example, the design in Table 2 is a $2^1 \times 3^1$ factorial experiment.

3.2 2^k Factorial Designs (Course Grids)

The most commonly used factorial design is a 2^k because it requires only two levels for each factor. These can be low and high, often -1 and $+1$ (or $-$ and $+$). 2^k designs are very easy to construct. Start by calculating the number of rows $N = 2^k$. The first column alternates -1 and $+1$, the second column alternates -1 and $+1$ in groups of 2, the third column alternates in groups of 4, and so forth by powers of 2. If you are using a spreadsheet, you can easily move from a design for k factors to a design for $k+1$ factors by copying the 2^k design, pasting it below to obtain a $2^k \times k$ matrix, and then adding a column for factor $k+1$ with the first 2^k values set to -1 and the second set of 2^k values set to $+1$. Conceptually, 2^k factorial designs sample at the corners of a hypercube defined by the factors' low and high settings. Figure 4 shows examples for 2^2 and 2^3 designs. Envisioning a 2^4 or larger design is left to the reader.

Factorial designs have several nice properties. They let us examine more than one factor at a time, so they can be used to identify important interaction effects. They are also *orthogonal* designs: the pairwise correlation between any two columns (factors) is equal to zero. This simplifies the analysis of the output (Y values) we get from running our experiment, because estimates of the factors' effects $\hat{\beta}_i$'s) and their contribution to the explanatory power (R^2)

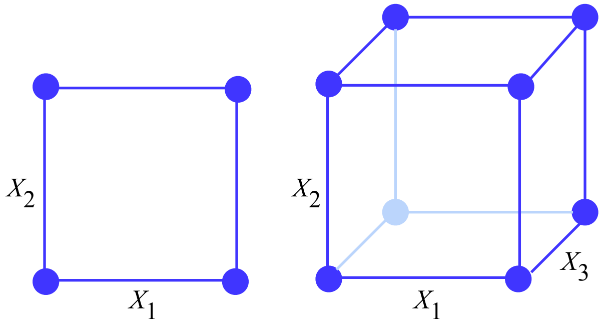

Figure 4: 2^2 and 2^3 Factorial Designs

of the regression metamodel will not depend on what other explanatory terms are placed in the regression metamodel.

Any statistical software package (e.g., JMP, Minitab, SAS, S-plus, SPSS, etc.) will allow you to to fit regression models with interaction terms, as well as main effects. If you must do your analysis in Excel, you will have to manually construct the appropriate columns for the interaction terms. When working in coded levels, the interaction columns are found by simply multiplying the columns for the associated main effects, as Table 5 shows for a 2^3 factorial. (To save a little room on the headings, I've left out the X's and just given the factor numbers.) When working in natural levels, it is best to subtract the means from each of the factors before creating the interaction columns. For example, the explanatory term corresponding to the X_1X_2 interaction should be the column of values $(X_1 - \overline{X}_1)(X_2 - \overline{X}_2)$. Note that the $\overline{X}$'s used in these equations are the average values from the design; they do not necessarily correspond to factor means in the real-world setting under investigation.

Table 5: Terms for a 2^3 Factorial Design

Design	Term						
Point	1	2	3	1,2	1,3	2,3	1,2,3
1	−1	−1	−1	+1	+1	+1	−1
2	+1	−1	−1	−1	−1	+1	+1
3	−1	+1	−1	−1	+1	−1	+1
4	+1	+1	−1	+1	−1	−1	−1
5	−1	−1	+1	+1	−1	−1	+1
6	+1	−1	+1	−1	+1	−1	−1
7	−1	+1	+1	−1	−1	+1	−1
8	+1	+1	+1	+1	+1	+1	+1

Table 5 shows that there are seven different terms (three main effects, two two-way interactions, and one three-way interaction) that we could consider estimating from a 2^3 factorial experiment. Of course, since we also want to estimate the intercept (overall mean), that means there are eight things we could try to estimate from eight data points. That will not work—we will always need at least one degree of freedom (d.f.) for estimating error (and preferably, a few more).

If we increase the number of factors k, we find a similar relationship. In general, there will be k main effects, (k choose 2) two-way interactions, (k choose 3) three-way interactions, and so forth, up to a single k-way interaction. If we add all these up, we get a total of $2^k - 1$ terms plus the intercept. Once again, we won't be able to estimate everything because there won't be any d.f. left over for error.

So, what do people do with a factorial design? One possibility is to *replicate* the design to get more d.f. for error. Estimating eight effects from eight observations (experimental units) is not possible, but estimating eight effects from 16 observations is easy. Replication also makes it easier to detect smaller effects by reducing the underlying standard errors associated with the β's.

Another option is to *make simplifying assumptions.* The most common approach is to assume that some higher-order interactions don't exist. In the 2^3 factorial of Table 5, one d.f. would be available for estimating error if the three-way interaction could safely be ignored. We could then fit a second-order regression model to the results. Similarly, if we generated data for a single replication of a 2^4 factorial design but could assume there was no four-way interaction we would have one d.f. for error; if we could assume there were no three-way or four-way interactions, we would have five d.f. for error.

Making simplifying assumptions sounds like a potentially dangerous thing to do, but it is often a good approach. Over the years, statisticians conducting field experiments have found that often, if there are interactions present, the main effects will also show up unless you "just happened" to set the low and high levels so everything cancelled out. There's also a rule of thumb stating that the magnitudes of two-way interactions are at most about 1/3 the size of main effects, and the magnitudes of three-way interactions are at most about 1/3 the size of the two-way interactions, etc. Whether or not this holds for experiments on simulations of complex systems is not yet certain. We may expect to find stronger interactions in a combat model or a supply chain simulation than when growing potatoes.

Now, let's return to project management. Suppose we decide to run an experiment where we vary the means for tasks B, E, F, and M, and leave all other potential factors (μ_i's and σ_i's) at their middle levels. The actual design, in both coded and natural levels, appears in Table 6. With four factors, there are 16 runs and 15 effects (four main effects, six two-way interactions, four three-way interactions, and one four-way interaction). We could estimate all but one of these effects from single replication of the experiment, or all these effects if two or more replications are made.

Once one or more replications of this basic design are conducted, and the resulting response Y is analyzed, we

Table 6: 2^4 Factorial Design for Project Management

Design	Coded Levels				Natural Levels			
Point	B	E	F	Q	B	E	F	Q
1	−	−	−	−	640	1200	280	9
2	+	−	−	−	660	1200	280	9
3	−	+	−	−	640	1600	280	9
4	+	+	−	−	660	1600	280	9
5	−	−	+	−	640	1200	320	9
6	+	−	+	−	660	1200	320	9
7	−	+	+	−	640	1600	320	9
8	+	+	+	−	660	1600	320	9
9	−	−	−	+	640	1200	280	39
10	+	−	−	+	660	1200	280	39
11	−	+	−	+	640	1600	280	39
12	+	+	−	+	660	1600	280	39
13	−	−	+	+	640	1200	320	39
14	+	−	+	+	660	1200	320	39
15	−	+	+	+	640	1600	320	39
16	+	+	+	+	660	1600	320	39

can build regression models or use graphical methods to estimate various factor and interaction effects.

3.3 m^k Factorial Designs (Finer Grids)

Examining each of factors at only two levels (the low and high values of interest) means you have no idea how the simulation behaves for factor combinations in the interior of the experimental region. Finer grids can reveal complexities in the landscape. When each factor has three levels, the convention is to use -1, 0 and 1 (or −, 0, and +) for the coded levels. Consider the capture-the-flag example once more. Figure 5 shows the (notional) results of two experiments: a 2^2 factorial (on the left) and an 11^2 factorial (on the right). For the 2^2 factorial, all that can be said about the factors is that when speed and stealth are both high, the agent is successful. Much more information is conveyed by the 11^2 factorial: here we see that if the agent can achieve a minimal level of stealth, then speed is more important. In both subgraphs the green circles—including the upper right-hand corner—represent good results, the light yellow circles in the middle represent mixed results, and the red circles on the left-hand side and bottom of the plot represent poor results.

The larger the value of m for an m^k factorial design, the better the space-filling properties of the design. A scatterplot matrix of the design points shows pairwise projections of the full design onto each pair of factors, and can be a useful way to show the design's space-filling characteristics. Consider the graph in Figure 6 that corresponds to a 5^4 factorial design. Each subplot has four points in the corners, four additional points along each edge, and nine points in the interior. The corresponding subplots for a 2^4 factorial would each reveal only four points, one at each quarter. The bad news is that the finer grid requires 625 design points instead of 16.

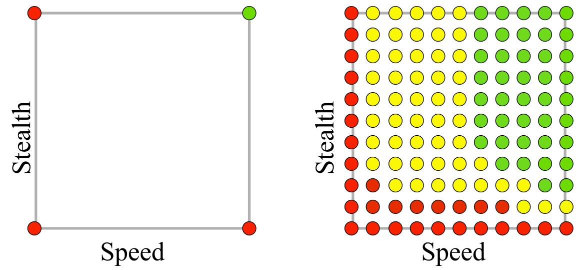

Figure 5: 2^2 and 11^2 Factorial Experiments for Capture-the-Flag

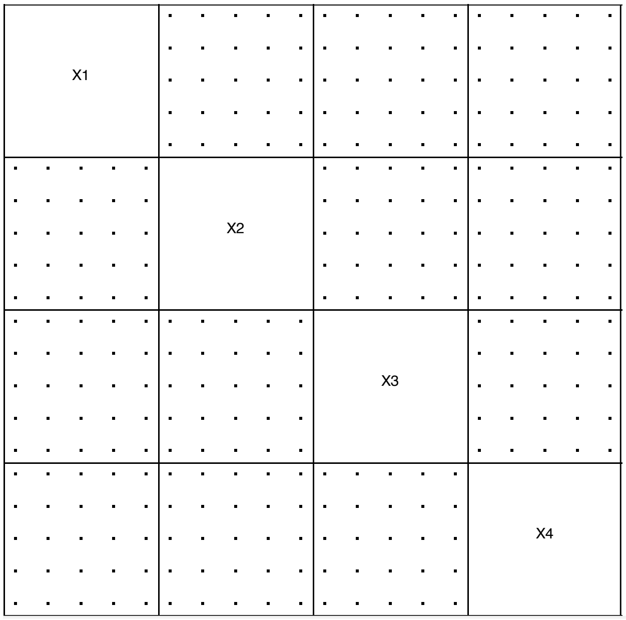

Figure 6: Scatterplot Matrix for a 5^4 Factorial Design

Table 7 shows just the design, not the results, but fitting regression models to the output data is again straightforward. Take care that if your statistical software doesn't automatically center the interaction terms when it's time to fit the model, you do this manually. You can see if adding (centered) quadratic terms will improve your metamodel, or explore higher-order terms. Surface plots and contour plots of the average behavior may be nice ways of looking at the results as a function of two factors at a time. These graphical methods mean you can focus on interesting features of the response surface landscape (such as thresholds, peaks, or flat regions) without assuming a specific form for the regression model. Regression trees, interaction plots, contour plots, and parallel plots are also useful for exploring the data. Examples can be found in Sanchez and Lucas

(2002); Cioppa, Lucas, and Sanchez (2004); or Kleijnen et al. (2005).

Despite the greater detail provided, and the ease of interpreting the results, fine grids are not good experimental designs for more than a handful of factors because of their massive data requirements. Even 2^k designs have this problem, as Table 7 shows.

Table 7: Data Requirements for Factorial Designs

No. of factors	10^k factorial	5^k factorial	2^k factorial
1	10	5	2
2	$10^2 = 100$	$5^2 = 25$	$2^2 = 4$
3	$10^3 = 1,000$	$5^3 = 125$	$2^3 = 8$
5	100,000	3,125	32
10	10 billion	9,765,625	1,024
20	*don't even think of it!*	95 trillion	1,048,576
40		9100 trillion trillion	1 trillion

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

3.4 Latin Hypercube Designs

Latin hypercube (LH) sampling provides a flexible way of constructing efficient designs for quantitative factors. They have some of the space-filling properties of factorial designs with fine grids, but require orders of magnitude less sampling. Once again, let k denote the number of factors, and let $N \geq k$ denote the number of design points. We will use a different coding for factor levels in LH designs. The low and high levels for factor X_i are coded as 1 and N, respectively, and the set of coded factor levels are $\{1, 2, \ldots, N\}$.

For a random LH design, each column is randomly permuted. In one replication, each of the k factors will be sampled exactly once at each of its N levels. Table 7 shows an example of a random LH for $k = 2$ and $N = 11$. Using this experimental design for our capture-the-flag simulation yields the results of Figure 7. Compare this design to those of Figure 5. Unlike the 2^2 factorial design, the LH design provides some information about what happens in the center of the experimental region. We do not get the same detailed information that the 11^2 provides about the boundaries between regions poor, fair, and good performance, but we do find that success occurs when both speed and stealth are high, that high stealth and moderate speed yield mixed results, and that if either speed or stealth is low the agent is unsuccessful. This happens with a fraction of the sampling cost ($N = 11$ vs. $N = 121$ of the 11^2 factorial design.

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

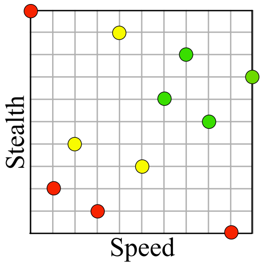

Figure 7: Random Latin Hypercube Design for Capture-the-Flag

The benefits of LH sampling become most apparent as k increases. The smallest LH designs are square, with $N = k$, so the number of design points grows linearly with k rather than exponentially. This means that 40 factors can be investigated in as few as 40 design points, rather than the over 1,000,000,000,000 required for a 2^{40} factorial experiment. Suppose our simulation runs in one second—with a LH design we could complete a replication of the experiment in under a minute, while the 2^{40} factorial design would require over 348 centuries of CPU time for each replication.

Random LH designs have good orthogonality properties if N is much larger than k, but for smaller designs some factors might have high pairwise correlations. One approach often taken is to randomly generate many LH designs and then choose a good one. Alternatively, Cioppa and Lucas (2005) have developed tables of so-called nearly orthogonal Latin hypercube (NOLH) designs that have good space-filling and orthogonality properties for small or moderate k. A scatterplot matrix of a NOLH that analyzes four factors in 17 design points is shown in Figure 8. The two-dimensional space-filling behavior compares favorably with that of the 5^4 design (requiring 625 design points) of Figure 6.

The number of design points required for investigating $k \leq 29$ factors are provided in Table 8. These are dramatically less than the design points for gridded designs shown in Table 7.

Table 8: Data Requirements for Nearly Orthogonal Latin Hypercube Designs

No. of Factors	No. of Design Points
2–7	17
8–11	33
12–16	65
17–22	129
23–29	257

Consider the project management simulation once again. Instead of limiting the study to the four factors representing the mean completion times for tasks B, E, F,

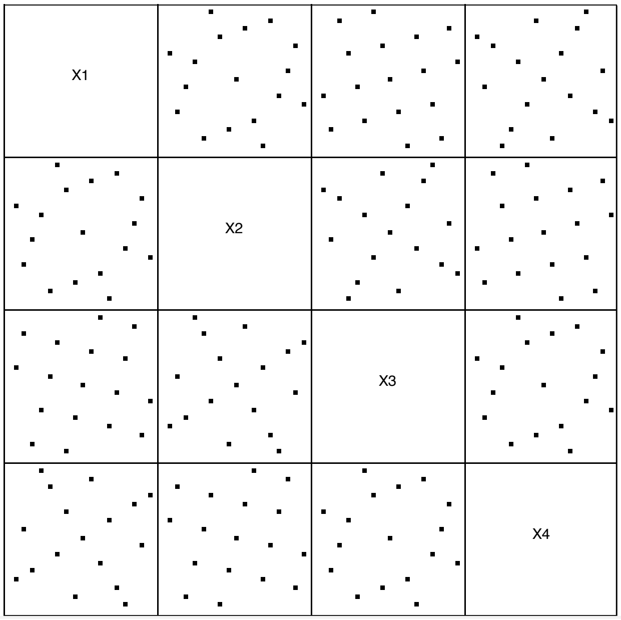

Figure 8: Scatterplot Matrix for a Nearly Orthogonal Latin Hypercube Design with Four Factors in 17 Runs

and Q, we could instead examine all seven means in a NOLH design with 17 design points, as Table 9 shows. Alternatively, we could vary four means and four standard deviations in a NOLH design with 33 design points, or all seven means and all seven standard deviations in a single design with 65 design points.

Table 9: NOLH Design for Seven Factors in 17 Runs for Project Management Simulation (Natural Levels)

Design Point	B	E	F	P	Q	S	T
1	646	1600	313	681	17	1088	303
2	641	1300	315	687	9	963	305
3	643	1375	283	678	28	1063	320
4	644	1450	293	700	26	925	310
5	655	1575	298	674	18	900	313
6	660	1325	295	694	11	1050	315
7	653	1275	320	679	35	988	318
8	651	1550	310	698	33	1025	308
9	650	1400	300	685	24	1000	300
10	654	1200	288	689	32	913	298
11	659	1500	285	683	39	1038	295
12	658	1425	318	693	20	938	280
13	656	1350	308	670	22	1075	290
14	645	1225	303	696	30	1100	288
15	640	1475	305	676	37	950	285
16	648	1525	280	691	13	1013	283
17	649	1250	290	672	15	975	293

Replicating the design will allow us to determine whether or not a constant error variance is a reasonable characterization of the simulation's performance, and is highly recommended. If we have the time and budget for even more sampling, then several Latin hypercubes can be stacked to obtain a larger design with better space-filling properties. Examples for agent-based simulation models appear in Allen, Buss and Sanchez (2004), Wolf et al. (2003), and Kleijnen et al. (2005).

3.5 2^{k-p} Resolution 5 Fractional Factorial Designs

While Latin hypercubes are very flexible, they are not the only designs useful for simulation experiments involving many factors. Sometimes many factors take on only a few levels. Traffic at both rush-hour and off-peak times might be of interest. We might have a few types of equipment that could be used to manufacture a particular part, or a few different rules for handling tasks of different priorities. A project manager might be able to expedite a specific task. LH designs work best when most factors have many levels.

Instead, we can consider varations of gridded designs. As long as we are willing to assume that some high-order interactions aren't important, then we can cut down (perhaps dramatically) on the number of runs that are required for a factorial experiment. This will be illustrated using a 2^k factorial, but the same ideas hold for other situations. Consider the 2^3 design in Table 2, and suppose that we are willing to assume that there are NO interactions. It turns out that we could call this column X_4, and investigate four factors in $2^3 = 8$ runs instead of four factors in 16 runs! This is called a 2^{4-1} fractional factorial. The design shows up in Table 10: we would be able to estimate (or test) four different factors in eight runs.

Table 10: 2^{4-1} Fractional Factorial Design

Design Point	X_1	X_2	X_3	X_4
1	-1	-1	-1	-1
2	$+1$	-1	-1	$+1$
3	-1	$+1$	-1	$+1$
4	$+1$	$+1$	-1	-1
5	-1	-1	$+1$	$+1$
6	$+1$	-1	$+1$	-1
7	-1	$+1$	$+1$	-1
8	$+1$	$+1$	$+1$	$+1$

Better yet, as long as we are assuming no interactions, we could squeeze a few more factors into the study. Take Table 5, which showed all the interaction patterns for a 2^3 factorial, and substitute in a new factor for any interaction term.

For example, the design in Table 11 is called a 2^{7-4} fractional factorial, since the base design varies seven factors in only $2^{7-4} = 8$ runs instead of $2^7 = 128$ runs! X_4 uses the column that would correspond to an X_1X_2 interaction, X_5 uses the column that would correspond to an X_1X_3 interaction, and similarly for X_6 and X_7. The design is said to be *saturated* since we cannot squeeze in any other factors. If we ignore the last column completely (i.e., we don't have a factor X_7) then we can examine six factors in only eight runs. If we take $b = 2$ replications of this experiment, we can examine seven factors in only 16 runs.

Table 11: Terms for a 2^{7-4} Fractional Factorial Design

Des.	X_1	X_2	X_3	X_4	X_5	X_6	X_7
Pt.				(1,2)	(1,3)	(2,3)	(1,2,3)
1	-1	-1	-1	$+1$	$+1$	$+1$	-1
2	$+1$	-1	-1	-1	-1	$+1$	$+1$
3	-1	$+1$	-1	-1	$+1$	-1	$+1$
4	$+1$	$+1$	-1	$+1$	-1	-1	-1
5	-1	-1	$+1$	$+1$	-1	-1	$+1$
6	$+1$	-1	$+1$	-1	$+1$	-1	-1
7	-1	$+1$	$+1$	-1	-1	$+1$	-1
8	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$

Graphically, fractional factorial designs sample at a carefully-chosen fraction of the corner points on the hypercube. Figure 9 shows the sampling for a 2^{3-1} factorial design, i.e., investigating three factors, each at two levels, in only $2^{3-1} = 4$ runs. There are two points on each of the left and right faces of the cube, and each of these faces has one instance of X_2 at each level and one instance of X_3 at each level, so we can isolate the effect for factor X_1. Similarly, averaging the results for the front and back faces allows us to estimate the effect for factor X_2, and averaging the results for the top and bottom faces allows us to estimate the effect for factor X_3.

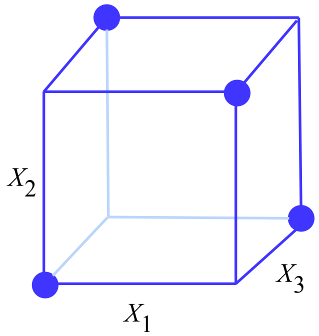

Figure 9: 2^{3-1} Fractional Factorial

Saturated or nearly-saturated fractional factorials are very efficient (relative to full factorial designs) when there are many factors. For example, 64 runs could be used for a single replication of a design involving 63 factors, or two replications of a design involving 32 factors. Saturated or nearly saturated fractional factorials are also very easy to construct. However, these designs will not do a good job of revealing the underlying structure of the response surface if there truly are strong interactions but we have ignored them in setting up the experiment. A compromise is to use R5 fractional factorials. These allow two-way interactions to be explored but can require many fewer design points.

It is easy to create a 2^{k-1} factorial (called a *half fraction*) by setting up the first 2^{k-1} columns as if we just had $k-1$ factors, and then constructing a column for the last factor by taking the interaction (product) of the first $k-1$ columns. Except for the special cases when $k \leq 4$, we will also be able to estimate two-way interactions with the 2^{k-1} designs. Unfortunately, a half-fraction is still inefficient if k is large. Until recently it was difficult to construct a very efficient R5 fractional factorial for more than about a dozen factors. For example, the largest R5 fractional factorial in Montgomery (2000) is a 2^{10-3}, while Box, Hunter, and Hunter (1978) and NIST/Sematech (2005) provide a 2^{11-4}. Sanchez and Sanchez (2005) recently developed a method, based on discrete-valued Walsh functions, for rapidly constructing very large R5 fractional factorial designs—a simple Java program generates designs up to a $2^{120-105}$ in under a minute. These allow all main effects and two-way interactions to be fit, and may be more useful for simulation analysts than saturated or nearly-saturated designs. The sizes of the resulting designs are given in Table 12.

Table 12: Data Requirements for Efficient 2^{k-p} R5 Fractional Factorial Designs

	No. of		No. of
k	Design Points	k	Design Points
1	$2^1 = 2$	18-21	$2^9 = 512$
2	$2^2 = 4$	22-29	$2^{10} = 1,024$
3	$2^3 = 8$	30-38	$2^{11} = 2,048$
4-5	$2^4 = 16$	39-52	$2^{12} = 4,096$
6	$2^5 = 32$	53-69	$2^{13} = 8,192$
7-8	$2^6 = 64$	70-92	$2^{14} = 16,384$
9-11	$2^7 = 128$	93-120	$2^{15} = 32,768$
12-17	$2^8 = 256$		

3.6 Central Composite Designs

Because 2^k factorials or fractional factorials sample each factor at only two levels, they are very efficient at identifying slopes for main effects or two-way interactions. Unfortunately, sampling at only two levels means the analyst has no idea about what happens to the simulation's response in the middle of the factor ranges. Going to a 3^k factorial would let us estimate quadratic effects, but it takes quite a bit more data—especially if k is large!

Another classic design that lets the analyst estimate all full second-order models (i.e., main effects, two-way interactions, and quadratic effects) is called a central composite design (CCD). Start with a 2^k factorial or R5 2^{k-p}

fractional factorial design. Then add a center point and two "star points" for each of the factors. In the coded designs, if -1 and $+1$ are the low and high levels, respectively, then the center point occurs at $(0,0,...,0)$, the first pair of star points are $(-c,0,...,0)$ and $(c,0,...,0)$; the second pair of star points are $(0,-c,0,...,0)$ and $(0,+c,0,...,0)$, and so on. A graphical depiction of a CCD for three factors appears in Figure 10. If $c=1$ the start points will be on the face of the cube, but other values of c are possible.

Although the CCD adds more star points when there are more factors, using a fractional factorial as the basic design means the CCD has dramatically fewer design points than a 3^k factorial design for the same number of factors. The additional requirements are $O(k)$. Some examples are given in Table 13, using the efficient R5 fractional factorials of Sanchez and Sanchez (2005) as the base designs for the CCDs. Once again, it is clear that a brute force approach is impossible when k is large, but efficient experimental designs allow the analyst to conduct an experiment.

Table 13: Data Requirements for 3-Level Designs

		Central Composite	3^k Factorial
k	No. of Terms	No. of Design Pts	No. of Design Pts
2	5	10	9
3	9	16	27
4	14	26	81
5	20	28	243
6	27	46	729
7	35	80	2,187
8	44	82	6,561
9	54	150	19,683
10	65	152	59,049
30	495	2,110	2.1E+14
70	2,555	16,526	2.5E+33
120	7,380	33,010	1.8E+57

3.7 Crossed and Combined Designs

So far, we have discussed designs for the first of the stated goals: *developing a basic understanding* of a particular model or system. The second goal was that of *finding robust* decisions or policies. A robust design approach (Taguchi 1987, Sanchez 2000) means that the factors are classified into two groups: *decisionfactors*, which represent factors that are controllable in the real world setting the simulation models; and *noise factors*, which are uncontrollable or controllable only at great cost in the real world, but potentially affect the system's performance. Sometimes a third group is added, consisting of *simulation-specific factors* such as the choices of random number streams, batch sizes, run lengths, and more.

The robust design philosophy means that the decision should not be based solely on mean performance and how close it is to a user-specified target value, but also on the performance variability. One way of accomplishing this is to redefine the performance measure to reflect the trade-off between a good mean and a small variance. Alternatives that often provide more guidance to the decision-maker are to examine the response mean and response variability at each design point separately, or to the fit separate models of the response mean and response variability. Regardless, working with the expected performance means that expectation is taken across the noise space.

One way this can be accomplished is by constructing a big design with columns for all of the decision and noise factors, referred to as a *combined design* (Sanchez et al. 1996). For example, suppose the decision factors are the means and standard deviations for tasks B, E, F, and Q in the project management scenario, perhaps because different workers, equipment, or procedures could be used. Further, suppose the noise factors are the means and standard deviations of tasks P, S, and T. This total of 14 factors could be examined using a NOLH with 65 design points or a CCD with 119 design points (replicated as needed). Examining the results in terms that involve only the decisions factors will yield insight into whether or not specific decision-factor combinations are robust to uncontrollable sources of variation.

Another design choice requires more sampling but may be easier to justify to decision-makers. Two basic designs are chosen—one for the decision factors, and another for the noise factors. They need not be the same type of design. A *crossed design* is then constructed by running each of the noise factor design points for each of the decision factor design points. Table 14 shows a portion of the design obtained by crossing a NOLH with 33 design points (for the decision factors) with a NOLH with 17 design points (for the noise factors) for the project management simulation. The base design has a total of $33 \times 17 = 561$ runs.

Whether the goal is to develop a basic understanding of the model, or to identify robust settings for decision factors, crossed designs can also be useful when a few factors take on a handful of discrete qualitative or quantitative levels. The capture-the-flag simulation could be run in dusk or night settings, e.g., by crossing a 2^1 design for time of day with an 11^2 design for speed and stealth. The project management simulation could be run by crossing a combined design for the 14 task time means and standard deviations with a 3^3 design that varies the task time distributions (normal, uniform, and symmetric triangular) for three of the tasks.

4 DISCUSSION

Designs like the ones described in this paper have assisted the U.S. military and over five allied countries in a series

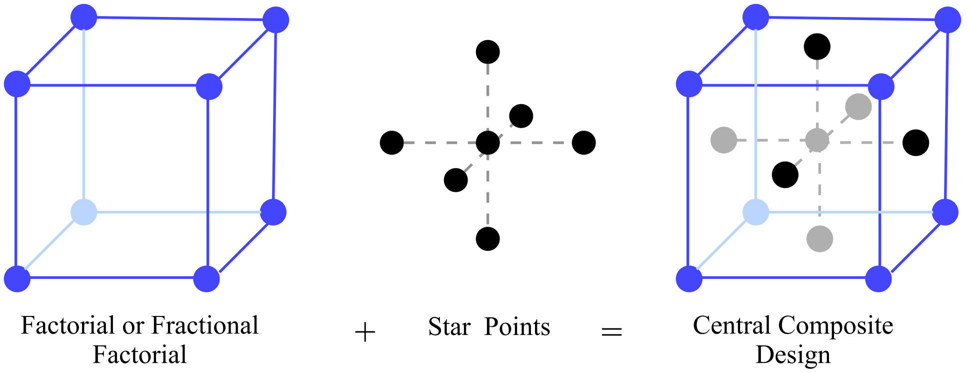

Figure 10: Construction of Central Composite Designs

Table 14: Crossed Design for Project Management Simulation

Crossed	Decision Factors					Noise Factors				
Design Point	Design Point	μ_B	μ_E	$\cdots$	σ_Q	Design Point	μ_P	μ_S	$\cdots$	σ_T
1	1	680	1238	$\cdots$	2.4	1	679	1100	$\cdots$	9.6
2	1	680	1238	$\cdots$	2.4	2	672	950	$\cdots$	5.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
17	1	680	1238	$\cdots$	2.4	17	683	925	$\cdots$	6.3
18	2	676	1600	$\cdots$	1.9	1	679	1100	$\cdots$	9.6
19	2	676	1600	$\cdots$	1.9	2	672	950	$\cdots$	5.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
34	2	676	1600	$\cdots$	1.9	17	683	925	$\cdots$	6.3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
544	33	648	1350	$\cdots$	1.5	1	679	1100	$\cdots$	9.6
545	33	648	1350	$\cdots$	1.5	2	672	950	$\cdots$	5.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
561	33	648	1350	$\cdots$	1.5	17	683	925	$\cdots$	6.3

of international workshops as part of the U.S. Marine Corps' *Project Albert* effort (Horne and Meyer 2004). Interdisciplinary teams of officers and analysts develop and explore agent-based simulation models to address questions of current interest to the U.S. military and allies, such as network-centric warfare, effective use of unmanned vehicles, future combat systems, peace support operations, convoy protection, and more. Sanchez and Lucas (2002) provide an overview of issues in modeling and analysis aspects of agent-based simulation. Cioppa, Lucas, and Sanchez (2004) discuss highlights from studies of squad size determination, degraded communications on the battlefield, and unmanned surface vehicles for both information, reconnaissance and surveillance missions and force protection scenarios. A humanitarian assistance scenario is described by Wolf et al. (2003) and Kleijnen et al. (2005). More information about the Project Albert efforts can also be found online at <http://www.projectalbert.org>. The web page for the Simulation Experiments & Efficient Design Laboratory (SEED lab) at the Naval Postgraduate School, at <http://diana.cs.nps.navy.mil/SeedLab>, is another resource. It contains links to numerous masters theses where simulation experiments have been used to explore a variety of questions of interest to military decision makers, as well as some spreadsheet tools and Java software for creating the designs described in this paper. For more on the philosophy and tactics of designing simulation experiments, examples of graphical methods that facilitate gaining insight into the simulation model's performance, and an extensive literature survey, we refer the reader to Kleijnen et al. (2005). This tutorial has touched on a few designs that we have found particularly useful, but other design and analysis techniques exist. Our intent was to open your eyes to the benefits of DOE, and convince you to make your next simulation study a simulation *experiment*.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. Marine Corps' Project Albert.

REFERENCES

Allen, T. E., A. H. Buss, and S. M. Sanchez. 2004. Assessing obstacle location accuracy in the REMUS unmanned underwater vehicle. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 940–948. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available via <http://www.informs-sim.org/wsc04papers/120.pdf> [accessed July 1, 2005].

Box, G. E. P., W. G. Hunter, and J. S. Hunter. 1978. *Statistics for experimenters: An introduction to design, data analysis and model building*. New York: Wiley.

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

Cioppa, T. M., T. W. Lucas, and S. M. Sanchez. 2004. Military applications of agent-based simulations. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 171–180. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available via <http://www.informs-sim.org/wsc04papers/020.pdf> [accessed July 1, 2005].

Fu, M. 2002. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14: 192–215.

Goldsman, D., S.-H. Kim, and B. L. Nelson. 2005. State-of-the-art methods for selecting the best system. In *Proceedings of the 2005 Winter Simulation Conference*, eds. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, forthcoming. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Hillier, F. S. and G. J. Lieberman. 2005. *Introduction to Operations Research*, 8th ed. New York: McGraw-Hill.

Horne, G. E. and T. E. Meyer. 2004. Data farming: Discovering surprise. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 171–180. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available via <http://www.informs-sim.org/wsc04papers/100.pdf> [accessed July 1, 2005].

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 simulation experiments. *INFORMS Journal on Computing* 17: 263–289.

Law, A. M. and W. D. Kelton. 2000. *Simulation modeling and analysis*. 3d. ed. New York: McGraw-Hill.

Marine Corps Warfighting Laboratory. 2005. [online]. Project Albert web pages. Available via <http://www.projectalbert.org> [accessed July 1, 2005]

Montgomery, D. C. 2000. *Design and analysis of experiments*. 5th ed. New York: Wiley.

NIST/SEMATECH. 2005. *e-Handbook of statistical methods*. Available via <http://www.itl.nist.gov/div898/handbook/> [accessed July 1, 2005].

Sanchez, S. M. 2000. Robust design: Seeking the best of all possible worlds. In *Proceedings of the 2000 Winter Simulation Conference*, eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 69–76. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available via <http://www.informs-sim.org/wsc00papers/013.pdf> [accessed July 1, 2005].

Sanchez, S. M and T. W. Lucas. 2002. Exploring the world of agent-based simulations: Simple models, complex analyses. In *Proceedings of the 2002 Winter Simulation Conference*, eds. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. Charnes, 116–126. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available via <http://www.informs-sim.org/wsc02papers/015.pdf> [accessed July 1, 2005].

Sanchez, S. M. and P. J. Sanchez. 2005. Very large fractional factorials and central composite designs. Working paper, Operations Research Department, Naval Postgraduate School, Monterey, California. Available via <http://diana.cs.nps.navy.mil/˜susan/LinkedFiles/fracfac.pdf> [accessed July 1, 2005].

Sanchez, S. M., P. J. Sanchez, J. S. Ramberg, and F. Moeeni. 1996. Effective engineering design through simulation. *International Transactions on Operational Research* 3: 169–185.

SeedLab. 2005. Simulation experiments & efficient designs [online]. Available via <http://diana.cs.nps.navy.mil/SeedLab/> [accessed July 1, 2005].

Taguchi, G. 1987. *System of experimental design, vols. 1 and 2*. White Plains, New York: UNIPUB/Krauss International.

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

<http://diana.cs.nps.navy.mil/˜susan/LinkedFiles/AgentLogistics.pdf> [accessed July 1, 2005].

AUTHOR BIOGRAPHY

SUSAN M. SANCHEZ is a Professor of Operations Research at the Naval Postgraduate School, where she holds a joint appointment in the Graduate School of Business and Public Policy. Her research interests include experimental design, data-intensive statistics, and robust selection. She has a Ph.D. in Operations Research from Cornell University. She has served the simulation community in several roles, including overseeing the transition of the INFORMS College on Simulation to the INFORMS Simulation Society during her term as President of the College. She is currently the Simulation Area Editor for the *INFORMS Journal on Computing* and the ASA representative to the WSC Board of Directors. Her e-mail is <ssanchez@nps.edu> and her web page is <http://www.nps.navy.mil/orfacpag/resumePages/sanchs.htm>.