Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 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 cannot 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 a better understanding of a complex simulation model. Designs 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, then it is 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.

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, or Chapter 12 of Law 2006). However, the environments in which real-world experiments are performed can be quite different from the simulation environment, so a framework specifically geared toward simulation experiments is beneficial.

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.

We focus on setting up single-stage experiments to address the first goal, and touch briefly on the second. Although the examples in this paper are very simple simulation models, the same types of designs have been extremely useful for investigating more complex simulation models in a variety of application areas. For a detailed discussion of the philosophy and tactics of simulation experiments, a more extensive catalog of potential designs (including sequential approaches), and a comprehensive list of references, see Kleijnen et al. (2005). Related topics not covered in this tutorial are ranking and selection or multiple comparison procedures (see, e.g., Goldsman, Kim, and Nelson 2005) or "optimization for simulation" (Fu 2002).

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

2 NUTS AND BOLTS

Some useful tools will help you gain a great deal of information in a short amount of time. This includes the time you spend setting up experiments and consolidating the results, and the computer time spent running your simulation.

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 $G/G/1$ queueing system might have both quantitative fac-

1-4244-1306-0/07/$25.00 ©2007 IEEE

tors (such as the mean inter-arrival and service times) and qualitative factors (such as LIFO, FIFO, or priority class processing, 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, it more often is a run or a batch, yielding an averaged or aggregated output value. The run is the appropriate experimental unit for terminating simulations. If the measure of interest is the time (or number of events) until termination, then the run's output is already in the form of a single number. When runs form the experimental units for non-terminating simulations, and steady-state performance measures are of interest, care must be taken to delete data from the simulation's warm-up period before performing the averaging or aggregation.

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

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

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

where the ε's are independent random errors. 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_1X_2. A *second-order model* includes both quadratic effects and two-way interactions, although it is best to fit this equation after centering the quadratic and interaction terms, as in (2):

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

In general, a *design* is a matrix where every column corresponds to a factor, and the entries within the column are settings for this factor. Each row represents a particular combination of factor levels, and is called a *design point*. If the row entries correspond to the actual settings that will be used, these are called *natural levels*. Coding the levels is a convenient way to characterize a design. Different codes are possible, but for quantitative data the low and high levels are often coded as -1 and $+1$, respectively. Table 1 shows a simple experiment, in both natural and coded levels, that could be conducted on a $G/G/1$ queue.

Table 1: Simple experimental design for a $G/G/1$ queue.

	Natural Levels		Coded Levels	
Design	Interarrival Rate	Service Rate	Interarrival Rate	Service 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	$+1$	$+1$

Each repetition of the whole design matrix is called a *replication*. 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 common types of simulation studies are not well-designed experiments. The first can occur if several people each suggest an "interesting" combination of factor settings, so a handful of design points end up being explored where many levels change simultaneously. Consider an agent-based simulation model of the 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 show that high stealth is of primary importance, another that speed is the key to success, and a third that they are equally important. There is *no way* to resolve these differences of opinion without collecting more data. In statistical terms, the effects of stealth and speed are said to be *confounded*. In practice, simulation models easily have tens or hundreds of potential factors. A handful of haphazardly chosen scenarios, or a trial-and-error approach, can use up a great deal of time without addressing the fundamental questions.

Speed	Stealth	Success?
Low	Low	No
High	High	Yes

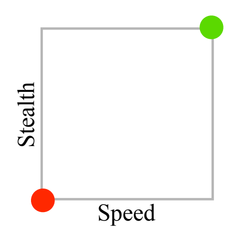

Figure 1: Confounded factor effects for capture-the-flag.

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

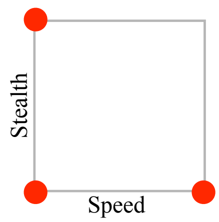

Figure 2: One-at-a-time sampling for capture-the-flag.

However, if all four combinations of speed and stealth (low/low, low/high, high/low, and high/high) are sampled, it is clear that success requires both high speed and high stealth. 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 made far too often in larger studies of more complex models. When only a few variations from a baseline are conducted, there may be many factors that change but a few that 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 set of tasks are performed, in which some tasks must be completed before others can begin. A precedence diagram (Figure 3) represents these relationships graphically. 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. By convention, "Start project" and "End project" tasks are specified so that every task is on at least one path from the beginning to the end.

Along with the precedence information, we must keep track of the times required to complete the 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 A-E-F-G-H requires 27 days to finish.

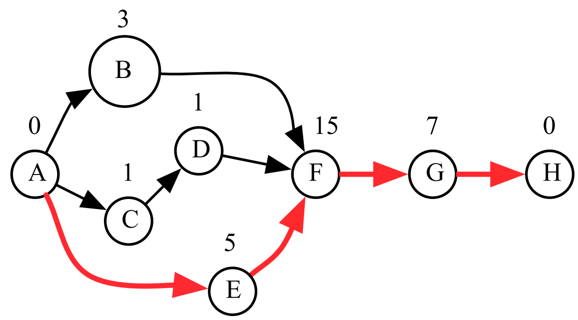

Figure 3: Project management precedence diagram.

A simple technique called PERT (Program Evaluation Review Technique) makes it easy to identify this so-called *critical path* for even larger networks. A probabilistic version of PERT takes into account the variability along the critical path (Hillier and Lieberman 2005). Given the task time means and standard deviations in Table 2 and assuming task times are independent, the mean and variance on path $\mathscr{C}_p = \{$A, E, F, G, H$\}$ (ignoring all other tasks) are $\mu_{tot} = \sum_{i\in\mathscr{C}} \mu_i = 5+15+7 = 27$ and $\sigma^2_{tot} = \sum_{i\in\mathscr{C}} \sigma^2_i = 2^2+3^2+1^2 = 14$. If the individual task time distributions are 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 then estimate the probabilities of completing (or failing to complete) the project by specified points in time.

Table 2: Task time distributional parameters.

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

What might go wrong with these PERT calculations? Sometimes we might not get the full benefit if a task on path $\mathscr{C}_p$ finishes early. For this example, if tasks F and G are expedited, or by chance their completion times are less than expected, this will benefit the project. However, suppose we spend only one day on E but all other tasks take their average times. The "new" critical path (A-B-F-G-H) takes 25 days. We 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 knew more about the task time distributions, we could perform a simple Monte Carlo simulation by randomly generating task completion times based on the task means, standard deviations, and normality assumptions, and using the precedence diagram to determine the project completion time. A frequency distribution of the project completion times, as well the proportion of time each task appears on the critical path, could be built by replicating the experiment. These might provide useful insights to a project manager.

It is rare in practice that we "know" such detail about the inputs. 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 our 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, the project manager and the simulation analyst may try to determine "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. So, consider a more complex project with 26 tasks (AA-ZZ), of which 19 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 3. For now, we retain the normality and independence assumptions for the task times.

Table 3: Low and high factor settings for project management factors.

	Range			Range	
i	μ_i	σ_i	i	μ_i	σ_i
BB	640-660	10-16	QQ	9-39	1-3
EE	1200-1600	50-200	SS	900-1100	0-30
FF	280-320	4-10	TT	280-320	4-10
PP	670-700	0-2			

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 factors or interactions have the greatest influence, or seek robust solutions. Although this 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* 2^k fractional factorials (R5-FFs) 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 for quantitative input factors. Expanding these R5-FFs to central composite designs provides some information about nonlinear behavior in simulation response surfaces.

Factorials (or gridded designs) are perhaps the easiest to discuss: they examine all possible combinations of the factor levels for each of the X_i's. A shorthand notation for the design is m^k, which means k factors are investigated, 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 1 is a $2^1 \times 3^1$ factorial experiment.

3.2 2^k Factorial Designs (Course Grids)

The simplest factorial design is a 2^k because it requires only two levels for each factor. These can be low and high, often -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's) we get from running our experiment, because estimates of the factors' effects $\hat{\beta}_i$'s and their contribution to the explanatory power (R^2) of the regression metamodel will not depend on what other explanatory terms are 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

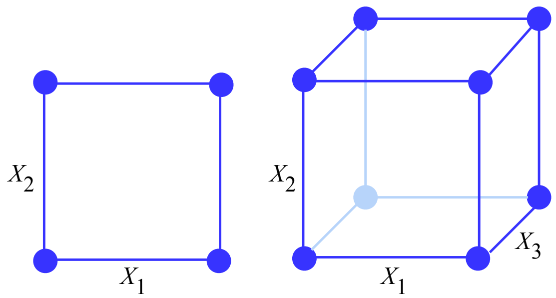

Figure 4: 2^2 and 2^3 factorial designs.

models with main effects and interaction terms. If you must do your analysis in Excel, you will have to manually construct appropriate columns for the interaction terms. When working in coded levels, the interaction columns are found by multiplying the columns for the associated main effects, as Table 4 shows for a 2^3 factorial (the headings indicate the factor numbers.) When working in natural levels, it is best to subtract the means before creating the interaction columns, as in (2). Note that the $\overline{X}$'s in (2) are the average values from the design—they do not necessarily correspond to factor means in the real-world setting.

Table 4: 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

From Table 4, there are seven different terms (three main effects, three two-way interactions, and one three-way interaction) that we could consider estimating from a 2^3 factorial experiment. But 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).

A similar relationship holds as we increase the number of factors k. 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. Adding all these up yields $2^k - 1$ terms plus the intercept. Once again, there will not 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 simple. 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 some higher-order interactions do not exist. In the 2^3 factorial of Table 4, one d.f. would be available for estimating error if the three-way interaction could safely be ignored. We could then fit a second-order regression model to the results. Similarly, if we have data for a single replication of a 2^4 factorial design but can assume there is no four-way interaction we have one d.f. for error; if we can assume there are no three-way or four-way interactions, we have five d.f. for error.

Making simplifying assumptions sounds dangerous, 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 also show up unless you "just happen" to set the low and high levels so the effects cancel. There is also a rule of thumb stating that the magnitudes of two-way interactions are at most about 1/3 the size of main effects, and the magnitudes of three-way interactions are at most about 1/3 the size of the two-way interactions, etc. Whether or not this holds for experiments on simulations of complex systems is not yet certain. We may expect to find stronger interactions in a combat model or a supply chain simulation than when growing potatoes.

Now we revisit the project management example. Suppose we decide to run an experiment where we vary the means for tasks BB, EE, FF, and QQ, and leave all other potential factors (μ_i's and σ_i's) at their middle levels. 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 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 the factors at only two levels (the low and high values of interest) means we have no idea how the simulation behaves for factor combinations in the interior of the experimental region. Finer grids can reveal complexities in the landscape. When each factor has three levels, the convention is to use -1, 0 and 1 (or $-$, 0, and $+$) for the coded levels. Consider the capture-the-flag example once more. Figure 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 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 represent poor results.

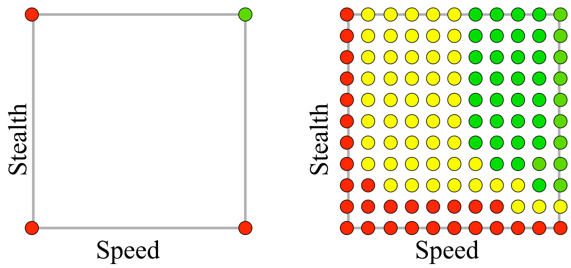

Figure 5: 2^2 and 11^2 factorial experiments for capture-the-flag.

The larger the value of m for an m^k factorial design, the better its space-filling properties. A scatterplot matrix of the design points shows projections of the full design onto each pair of factors. Consider the graph in Figure 6 for a 5^4 factorial. Each subplot has four points in the corners, three 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 corner. The bad news is that the finer grid requires 625 design points instead of 16. Figure 6 provides information about the design, not the output, but once the simulation has been run the task of fitting a regression metamodel to the output is straightforward. Adding columns to your (replicated) design matrix will allow you to match the inputs to the outputs. Start by fitting main-effects metamodels, then see if adding (centered) quadratic terms will improve your metamodel, or explore higher-order terms. Surface plots and contour plots of the average behavior can help you view the results as a function of two factors at a time. These graphical methods allow you to 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 5 shows.

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

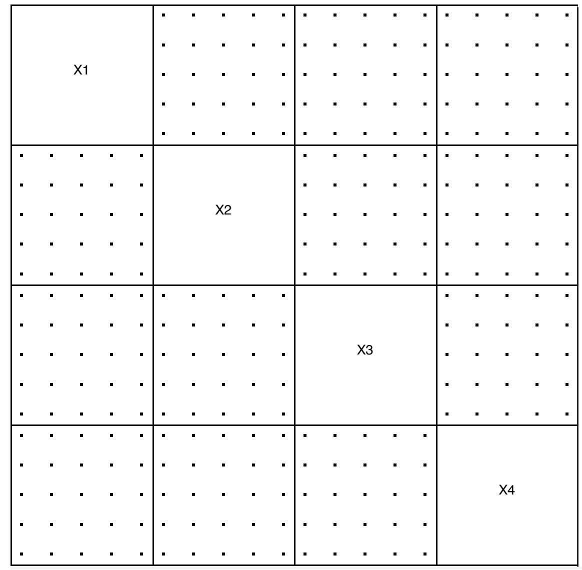

Figure 6: Scatterplot matrix for a 5^4 factorial design.

Table 5: 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*	9.5×10^9	1,048,576
40	*think of it!*	9.1×10^{21}	1.0×10^9

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. Our convention for factor levels in LH designs is that 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 of 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. $N = 121$ for 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		

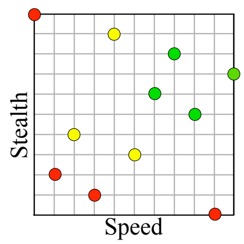

Figure 7: Random Latin hypercube for capture-the-flag.

The benefits of LH sampling are greatest for large k. The smallest LH designs are square, so the number of design points grows linearly with k. Suppose our simulation runs in one second. Each replication of a 40-factor experiment would take under a minute using an LH design, but over 348 centuries using a 2^{40} factorial design.

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 is to generate many random LH designs and then choose a good one. Alternatively, Cioppa and Lucas (2007) construct nearly orthogonal Latin hypercube (NOLH) designs that have good space-filling and orthogonality properties for small or moderate k. Table 6 lists the number of design points for NOLHs with $k \leq 29$. These are dramatically less than the numbers for gridded designs in Table 5.

Table 6: 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

A scatterplot matrix of a NOLH for 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.

Consider the project management simulation once more. Instead of limiting the study to the four factors representing the mean completion times for tasks BB, EE, FF, and QQ, we could instead examine all seven means in a NOLH design with 17 design points. Alternatively, we could vary four

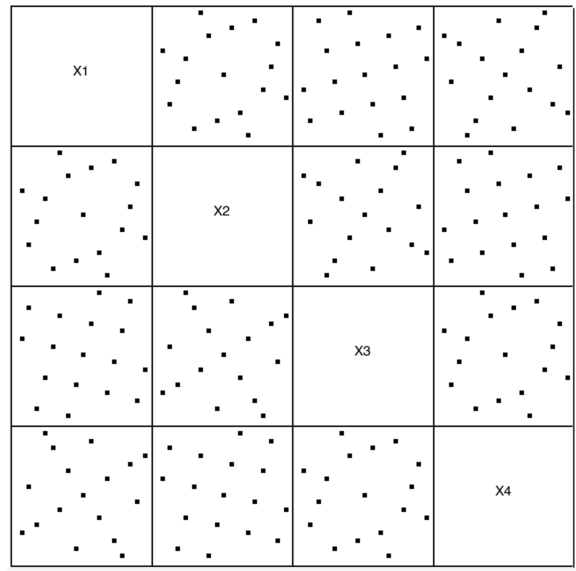

Figure 8: Scatterplot matrix for a nearly orthogonal Latin hypercube design with four factors in 17 runs.

means and four standard deviations in a NOLH design with 33 design points, or all seven means and all seven standard deviations in a NOLH with 65 design points.

Replicating the design allows us to determine whether or not a constant error variance is a reasonable characterization of the simulation's performance, and is highly recommended. If we have the time and budget for even more sampling, then several Latin hypercubes can be stacked to obtain a larger design with better space-filling properties. Examples for agent-based simulations appear in Allen, Buss and Sanchez (2004), Kleijnen et al. (2005), or Cioppa and Lucas (2006).

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. If we are willing to assume that some high-order interactions are not important, we can cut down (perhaps dramatically) the number of runs required. This will be illustrated using a 2^k factorial, but the same ideas hold for other situations. Consider the 2^3 design in Table 1, and suppose that we are willing to assume that no interactions exist. We could call the $X_1X_2X_3$ 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.

Better yet, as long as we are assuming no interactions, we can squeeze a few more factors into the study. Take Table 4, which shows all the interaction patterns for a 2^3 factorial, and substitute in a new factor for each interaction term. The resulting design (Table 7) is called a 2^{7-4} fractional factorial, because 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 do not have a factor X_7) then we can examine six factors in only eight runs. If we take $b = 2$ replications, we can examine seven factors in only 16 runs.

Table 7: Terms for a 2^{7-4} fractional factorial design.

Des. Pt.	X_1	X_2	X_3	X_4 (1,2)	X_5 (1,3)	X_6 (2,3)	X_7 (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.

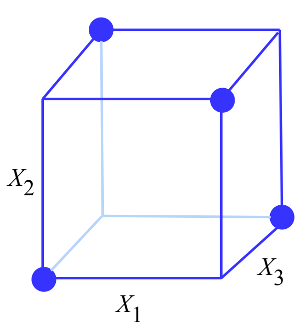

Figure 9: 2^{3-1} fractional factorial.

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 ignore them when 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 find a very efficient R5 fractional factorial for more than about a dozen factors. The largest R5 fractional factorial in Montgomery (2000) is a 2^{10-3}; the largest in Box, Hunter, and Hunter (1978) and NIST/Sematech (2005) is 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 factorials—a short 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.

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 $(0,0,\ldots,0)$, the first pair of star points are $(-c,0,\ldots,0)$ and $(c,0,\ldots,0)$; the second pair of star points are $(0,-c,0,\ldots,0)$ and $(0,+c,0,\ldots,0)$, and so on. A graphical depiction of a CCD for three factors

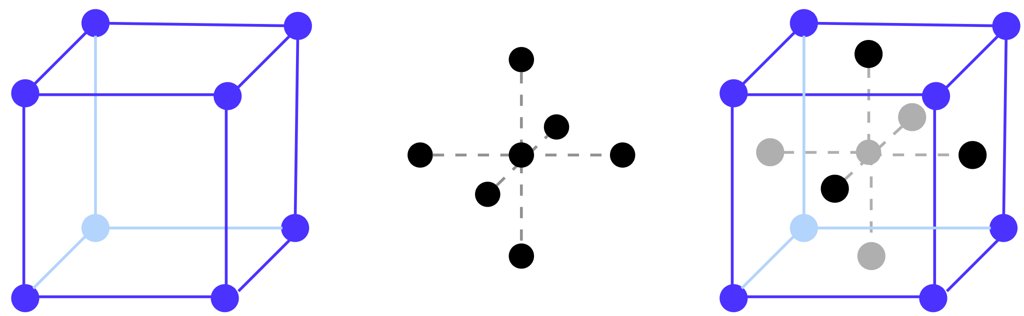

Figure 10: Construction of central composite designs.

appears in Figure 10. If $c = 1$ the star 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 8, 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 8: Data requirements for 3-level designs.

		CCD	3^k
k	No. of Terms	No. of Design Pts	No. of Design Pts
2	5	10	9
5	20	28	243
10	65	152	59,049
30	495	2,110	2.1×10^{14}
70	2,555	16,526	2.5×10^{33}
120	7,380	33,010	1.8×10^{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: *decision factors*, 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. A third group is sometimes added, consisting of *simulation-specific factors* like 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. Redefining the performance measure to reflect the trade-off between a good mean and a small variance is one approach. 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, the expectation is taken across the noise space.

One way this can be accomplished is by constructing a *combined design* with columns for all of the decision and noise factors (Sanchez et al. 1996). For example, suppose the decision factors are the means and standard deviations for tasks BB, EE, FF, and QQ 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 PP, SS, and TT. 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 (Table 9). 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 9 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 be useful when a few factors take on a handful of discrete levels. The capture-the-flag

Table 9: Crossed design for project management simulation.

Crossed Design Point	Decision Factors Design Point	μ_B	μ_E	$\cdots$	σ_Q	Noise Factors 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
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
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
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
34	2	676	1600	$\cdots$	1.9	17	683	925	$\cdots$	6.3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
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
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
561	33	648	1350	$\cdots$	1.5	17	683	925	$\cdots$	6.3

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 NOLH 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 several allied countries in a series of international data farming workshops (Horne and Meyer 2004; SEED Center for Data Farming 2007). Interdisciplinary teams of officers and analysts develop and explore agent-based simulation models to address questions of current interest to the U.S. military and allies, such as network-centric operations, effective use of unmanned vehicles, peace support operations, and more. Sanchez and Lucas (2002) provide an overview of issues in modeling and analysis aspects of agent-based simulation. 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 discussed in Kleijnen et al. (2005). Lucas et al. (2007) describe several defense and homeland security applications: critical infrastructure protection, non-lethal capabilities in a maritime environment, and emergency first response to a crisis event.

The *SEED Center for Data Farming* at the Naval Postgraduate School was established to advance the use of **S**imulation **E**xperiments & **E**fficient **D**esign. Its web pages (at <harvest.nps.edu>) contain links to several types of resources, including master's theses where simulation experiments have been used, papers and proceedings articles, and some spreadsheet tools and Java software for creating designs.

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. Army Training and Doctrine Command Analysis Center Monterey (TRAC-MTRY) and the Department of Defense's Modeling & Simulation Coordination Office. Earlier versions of this tutorial paper appeared in Sanchez (2005) and Sanchez (2006).

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*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 940–948. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

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. 2007. Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* 49(1): 45–55.

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*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 171–180. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Fu, M. 2002. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14 (3): 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*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 178–187. 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*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 171–180. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

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 (3): 263–289.

Law, A. M. 2006. *Simulation modeling and analysis*. 4th ed. New York: McGraw-Hill.

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

Lucas, T. W., S. M. Sanchez, F. Martinez, L. R. Sickinger, and J. W. Roginski. 2007. Defense and homeland security applications of multi-agent simulations. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, forthcoming. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

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

NIST/SEMATECH. 2005. *e-Handbook of statistical methods*. Available via <www.itl.nist.gov/div898/handbook/> [accessed July 1, 2006].

Sanchez, S. M. 2000. Robust design: Seeking the best of all possible worlds. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 69–76. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sanchez, S. M. 2005. Work smarter, not harder: Guidelines for designing simulation experiments. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 69–82. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sanchez, S. M. 2006. Work smarter, not harder: Guidelines for designing simulation experiments. In *Proceedings of the 2006 Winter Simulation Conference*, ed. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 47-57. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

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*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. Charnes, 116–126. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sanchez, S. M. and P. J. Sanchez. 2005. Very large fractional factorials and central composite designs. *ACM Transactions on Modeling and Computer Simulation* 15 (4): 362–377.

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.

SEED Center for Data Farming. 2007. Simulation experiments & efficient designs [online]. Available via <harvest.nps.edu> [accessed July 15, 2007].

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

AUTHOR BIOGRAPHY

SUSAN M. SANCHEZ is a Professor and Associate Chair for Research in the Operations Research at the Naval Postgraduate School; she also holds a joint appointment in the Graduate School of Business and Public Policy, and is Co-director of NPS' SEED Center for Data Farming. She received her B.S. in Industrial and Operations Engineering from the University of Michigan, and her M.S. and Ph.D. in Operations Research from Cornell University. She is a member of INFORMS, DSI, ASA, and ASQ. She is currently the Vice-President of the WSC Board of Directors, where she represents the American Statistical Association. Her research interests include experimental design, data-intensive statistics, and robust selection. Her email address is <ssanchez@nps.edu> and her web pages can be reached from <harvest.nps.edu>.