Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

EXPLORING THE WORLD OF AGENT-BASED SIMULATIONS: SIMPLE MODELS, COMPLEX ANALYSES

Susan M. Sanchez

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

Thomas W. Lucas

Operations Research Department
Naval Postgraduate School
Monterey, CA 93943-5219, U.S.A.

ABSTRACT

Agent-based simulations are models where multiple entities sense and stochastically respond to conditions in their local environments, mimicking complex large-scale system behavior. We provide an overview of some important issues in the modeling and analysis of agent-based systems. Examples are drawn from a range of fields: biological modeling, sociological modeling, and industrial applications, though we focus on recent results for a variety of military applications. Based on our experiences with various agent-based models, we describe issues that simulation analysts should be aware of when embarking on agent-based model development. We also describe a number of tools (both graphical and analytical) that we have found particularly useful for analyzing these types of simulation models. We conclude with a discussion of areas in need of further investigation.

1 INTRODUCTION

What is an agent-based simulation (ABS)? While definitions vary, we use this term to mean a simulation made up of agents, objects or entities that behave autonomously. These agents are aware of (and interact with) their local environment through simple internal rules for decision-making, movement, and action. ABS has been proposed for many situations involving a large number of heterogeneous individuals, such as vehicles and pedestrians in traffic, people in crowds, artificial characters in computer games, agents in financial markets, and humans and machines on battlefields. The aggregate behavior of the simulated system is the result of the dense interaction of the relatively simple behaviors of the individual simulated agents.

ABSs have been used for different purposes. One is as an efficient means of graphically portraying behavior that seems realistic. Flocks and schools have been used as examples of robust self-organizing systems in the literature of parallel and distributed computing systems for quite some time (Kleinrock 1985, Reynolds 1987).

Another reason for employing ABS is to leverage simulation's advantages in cost and time relative to many real-world experiments. For example, Dudenhoeffer, Bruemmer, and Davis (2001) use an ABS model to examine the ability of a human operator to coordinate and interact with large-scale robotic forces. ABS is attractive because the technology, cost, and time limitations prohibit extensive live testing—even though live testing is preferred.

Sometimes the focus of ABS development is on the modeling aspects. Mason and Moffat (2001) develop object-oriented tools in C++ for implementing command-and-control in military simulations. The "proof of principle" is their ability to implement command agents representing each of 12 different roles in a simulation of a services-assisted noncombatant evacuation operation.

An emerging area of interest is that of *creating* or *defining* behavior. Dickie (2002) considers simple rules for controlling unmanned autonomous vehicles involved in a search-and-detection mission. In such cases, the way a simulation is designed, analyzed, and used can be markedly different. Mizuta and Yamagata (2001) describe another ABS meant to create behavior. Their model of greenhouse gas emissions trading is intended to help establish efficient rules for governing this developing international market. Erlenbruch (2002) explores tactical concepts in German peacekeeping operations by examining the impact of authorizing soldiers to use different types of actions when dealing with a variety of civilian behaviors.

Our interest in agent-based modeling arose from work we are doing with the United States Marine Corps' Project Albert (Marine Corps Combat Development Command 2002, see also Horne and Leonardi 2001, Horne and Johnson 2002). This is an unclassified, international effort to provide both the research and technological infrastructure for examining new technologies, and provide a mechanism for transferring this knowledge and skills from the analysts to the decision-makers. Agent-based modeling is a cornerstone of Project Albert's efforts because of the strong interest in so-called *intangibles*: human characteristics such as trust, unit cohesion, fatigue, morale, leadership, aggressiveness, fear, compassion, and so forth. Decision-makers might seek answers to questions such as: What factors affect the ability to quickly complete the mission with minimum casualties? Intangibles, as well as equipment, tactics, and personnel, are thought to play an important role in determining the success of many operations. Applications currently under investigation include small-unit military operations, reconnaissance, peacekeeping, convoy protection, food distribution, counterterrorism, and minesweeping.

In Section 2 we describe a few key modeling aspects of agent-based simulations. In Section 3 we discuss why the analysis of these relatively simple simulations can, nonetheless, be quite complex. We also describe some effective approaches for systematically exploring these models, using examples from recent studies that illustrate some of the design and analysis techniques we have found particularly useful. In Section 4 we conclude with a discussion of issues related to ABS modeling and analysis that merit further investigation.

2 SIMPLE MODELS

Consider the process of people leaving a stadium after a major sports event. Scripting the paths for a large number of individual objects would be tedious at best, and attempts to make global changes to the models would be difficult. In contrast, an agent (in this case, a person) may be given very simple rules such as:

- Try to move toward the closest exit gate.
- If there are too many people in front of you, try moving to the left or right.
- If you've waited a certain amount of time without getting closer, try moving away from the crowd.
- Try to stay close to others in your group of family and friends.

As another example, Reynolds (1987) describes three basic behaviors for his notional birds (called "boids"):

- collision avoidance,
- velocity matching, and
- flock centering.

While these rules are simple to list, coding them in a reasonable manner can sometimes be tricky. Once the rules have been coded, however, it is easy to *populate* an ABS with either a small or a large group of agents. As we later discuss, it may or may not be easy to *run* large-scale ABSs.

Agents can be programmed to evolve or learn during the course of the simulation run. For example, they might have a set of 10 possible rules they could use. Over time, they could assess how well the different rules are working. The appearance of "learning" can take place by an agent updating its probability of taking certain actions (or updating the weights it assigns for different rules) because of their perceived past effectiveness. For example, an increase in student enrollment and a concurrent decrease in parking spaces on our campus meant that during the first few weeks of the academic term, parking was extremely difficult to find between 8:30 and 10:30 in the morning. After a few weeks, people had changed their behaviors: some arrived earlier to assure they could park close to their building(s); some arrived later when spaces opened up as morning classes were completed and others headed home or out to lunch; others parked off campus and walked in to avoid searching for parking spaces; and many began biking or (in the case of students) using the shuttle between campus and base housing. Similarly, with many agents in a model one can simulate this behavior. Agents that begin the simulation as identical entities may end up exhibiting quite different behavior.

3 COMPLEX ANALYSES

Why do we feel that analyzing these simple models is a complex task? There are several reasons. First and foremost, it requires a different frame of mind than we are used to for the analysis of, e.g., manufacturing simulation.

Sacks et al. (1989) state that "The three primary objectives of computer experiments are: (i) *predicting* the response at untried inputs, (ii) *optimizing* a function of the input parameters, and (iii) *calibrating* the computer code to physical data." Unfortunately, for many agent-based models we cannot credibly do any of these! For example, disaster relief efforts are thankfully not an every day occurrence. When they do happen there are only a few factors we might be able to *manipulate*, such as distributing food from a single convoy or scattering several smaller distribution sites over a larger area. We cannot "control" the fear, hunger, or aggressiveness of people seeking food or attempting to evacuate an area after a natural disaster. Ethical implications of experimenting on human subjects also must be considered. So, while we may be able to collect anecdotal evidence on *what* happened, we may not be able to measure—either during the incident or after the fact—any of the intangible factors that might tell us *why* it happened. There may be no possibility of collecting sufficient data even to calibrate our ABS, let alone credibly predict or optimize.

Instead, we assert that in many situations the most relevant analysis is *searching for insights* or *gaining a basic understanding* of the ABS. This is discussed in more detail by Kleijnen et al. (2002), along with two other potential goals: *finding robust configurations* (e.g., systems, decisions, or policies), and *comparing configurations*. Insights we might hope to glean relate to identifying important factors and their interactions, as well as finding regions, ranges, and thresholds where interesting things happen.

Table 1: The Experimental Environment

Traditional DOE Assumptions	Agent-based Model Characteristics
Small or moderate number of factors	Large number of factors
Linear or low-order effects	Non-linear, non-polynomial behavior
Sparse effects	Many substantial effects
Negligible higher-order interactions	Substantial higher-order interactions
Homogeneous errors	Heterogeneous errors
Normally distributed errors	Various error distributions
Black box model	Substantial expertise exists
Univariate response	Many performance measures of interest

There is certainly a need for a systematic, scientific approach to analyzing ABSs. Statistical design of experiments (DOE) has been very beneficial in both real-world and simulation settings, so we can seek to exploit DOE concepts for investigating agent-based models. However, there are differences—some obvious and some more subtle—that mean a straightforward application of traditional DOE methods may not adequately address the questions of interest. Table 1 lists some common assumptions for traditional DOE approaches, as well as characteristics that we feel portray the environment for many ABS studies. In short, while ABS models are often much smaller and simpler than other types of simulation models, their environment can be quite complex.

3.1 Implementing Simple Rules

Whenever a new ABS is developed, it is tempting to begin immediately exploring for insights. However, our experience suggests that the analyst should begin by performing runs for some very simple scenarios. This is part of the process of debugging the logic of the code. For example, we established symmetrical situations as part of learning to use an early version of a time-step ABS modeling platform. Two lines of opposing forces were put in place, and both the Red and Blue agents were given identical behaviors. In 100 independently seeded runs of the simulation, the Blue side always won! It turned out that the internal model logic kept a list of all potential actions. At the beginning of each time step, it processed this list in order. Blue agents were at the top of the list, so they always got to "go first" and so could eliminate Red agents before any Red shots were fired. This undesirable behavior had not been noticed by the developers (during the model development process) or other users, who had been creating small but more "interesting" scenarios to find out the modeling capabilities. The software designers solved this sequencing problem by randomizing the order in which the events were processed. However, if this had gone unchallenged, then for certain (and perhaps large) portions of the response surface, an analyst might mistakenly attribute Blue success to the use of particular tactics, rather than being an artifact of the model.

Another aspect that can be problematic is the use of generic descriptions for specific rules or actions. Sometimes these may have different interpretations, meaning that the analyst may think they understand the consequences of a particular rule or action, but the program logic implements things differently. As an example, consider an agent's movement decision.

Gill and Shi (2002) discuss difficulties that can arise in coding movement within ABSs. In what follows suppose there are only two different types of agents—B Blue agents and R Red agents—and a single flag positioned at the Blue agents' final goal. Now suppose the user is allowed to change weights which correspond to propensities for Blue to move toward or away from Red agents (with weight W_R) and the Flag (with weight W_F). Let the possible weights range between -100 and $+100$ with default values of zero. In the default case, the Blue agents have no impetus to head in any particular direction, so their movement patterns will be random. Let Blue's distance from the Flag be $D_F = 15$ units, with five Red agents (at an average distance of $\overline{D}_R = 5$ units) placed in between. If the user sets the weights to $W_R = -10$ and $W_F = +20$, what *conceptual model* might they have? One possibility is that Blue is twice as likely to move toward the Flag than away from the Red agents (treated as a single group), since $W_F/W_R = -2$. Alternatively, perhaps the total weight to Red is $(W_R \times R)/\overline{D}_R = -10$ and that to the Flag is $W_F/D_F = +1.33$. Then one could argue the agent would be about seven times more likely to move away from the enemy than toward the flag. Gill and Shi (2002) compare the movement penalty function used in MANA (Lauren and Stephen 2001) to an alternative general formula that makes use of relative (rather than absolute) distance, and partial cumulative (rather than average) weighting. Let Z_{new} denote the direction and magnitude of the resulting movement, W_B denote the weight for movement toward other Blue agents, and α and r denote tuning constants between 0 and 1, inclusive. These two movement penalty functions are given in equations (1) and (2), respectively:

$$Z_{new} = \frac{W_R}{100R}\left(\sum_{i=1}^{E}\frac{D_{i,new}+(100-D_{i,old})}{100}\right) + \frac{W_F}{100}\left(\frac{D_{F,new}+(100-D_{F,old})}{100}\right) \quad (1)$$

$$Z_{new} = \frac{W_R}{R^\alpha}\sum_{i=1}^{R}\left(\frac{D_{i,new}-D_{i,old}}{D_{i,old}}\right)^r + \frac{W_B}{B^\alpha}\sum_{j=1}^{B}\left(\frac{D_{j,new}-D_{j,old}}{D_{j,old}}\right)^r + W_F\left(\frac{D_{F,new}-D_{F,old}}{D_{F,old}}\right)^r. \quad (2)$$

Returning to the conceptual models, if Blue is twice as likely to move toward the Flag as away from Red, one might expect Blue to generally head toward the Flag, but avoid Red by bouncing around to one side or the other. If, on the other hand, Blue is seven times more likely to move away from Red, then one would not expect Blue to head to the target. However, using the MANA penalty function in equation (1) Blue will proceed directly toward the Flag through the group of Red agents. This behavior holds for any $W_F > W_R$ and for any number of Reds. Figure 1 (adapted from Gill and Shi 2002) shows the differences in two performance measures that result from running the same scenario while implementing different movement control logic.

This issue is particularly important when we are trying to model intangibles. For example, a user might intend to represent aggressive Blue behavior, cautious Blue behavior, or unit cohesion among Blue agents by specifying

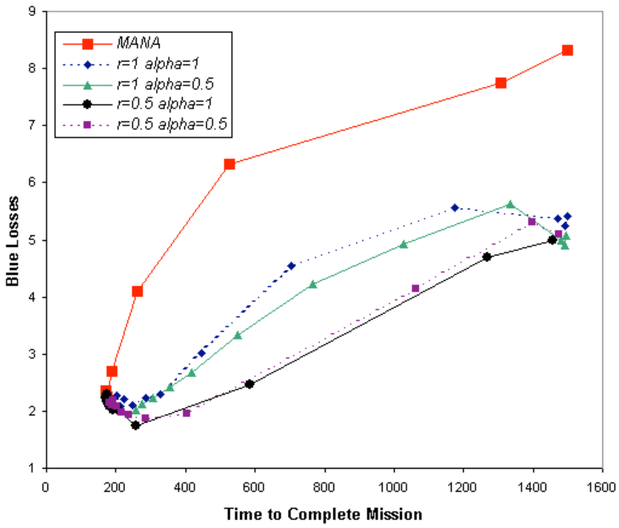

Figure 1: Blue Losses vs. Time to Complete Mission for Various Penalty Functions Determining Movement

$W_F > |W_R|$, $W_F < |W_R|$ or $W_B > |W_R|$, respectively. If there is not a clear understanding of the agent behavior that results from such weights, the behavioral labels can be very misleading. These are by no means the only possible movement algorithms. For example, if the penalties were mapped into a probability distribution function, then simulated annealing could be used to generate movement for agents that 'learn' over time.

3.2 Collecting Data Effectively

The first entry in Table 1 rates special mention—the number of factors involved. Many real-world experiments deal with no more a handful of factors (e.g., five), and rarely are more than 10 investigated at the same time. In constrast, even for the relatively simple models described in Section 2, it is not uncommon to have tens or even hundreds of factors. A model with 100 factors, each able to take on only one of two possible values, still has $2^{100} \approx 10^{30}$ (i.e., more than a trillion trillion) potential combinations of the factor levels! Despite advances in high-speed computing, it is impossible to perform a brute-force analysis of all combinations. Since even millions of runs constitute a sparse sample in a high-dimensional space, we must collect our data intelligently. Trial and error is notoriously risky and inefficient, and relying on visual results of one (or several) runs is dangerous. Along with constraints on computing power and time, we may also be limited in our ability to assimilate large amounts of data.

One way to address the large number of factors is to partition them into classes that will be examined with designs of various resolutions. We discuss only a few designs here. See Lucas et al. (2002) and Kleijnen et al. (2002) for for other designs, references, and additional discussion.

Gridded designs are straightforward, and probably the easiest to explain to someone unfamiliar with the concepts of DOE and statistical analysis. If all k factors have the same number of categories (m) these are called m^k *factorial* designs. The grids need not be identical: we could have, e.g., a $2^{k_1}3^{k_2}$ design that varied k_1 factors across two levels and k_2 factors over three levels, where $k_1+k_2 = k$. Gridded designs are easy to generate, but the exponential growth in the number of scenarios is problematic. For large experiments this can be overwhelming. A 10^9 factorial requires one billion runs per replication. One can argue that using this much data to generate a response surface reflects tremendous inefficiency rather than effective use of computational power.

Low resolution designs can be used to mitigate this exponential explosion in data requirements. However, this efficiency comes at a cost. The analyst must forego the ability to investigate some (or all) higher-order interactions and/or non-linear characteristics of the surface. (Later experiments can be conducted to confirm or refute the validity of such assumptions.) The simplest low resolution designs are called *fractional factorials*. If there are k factors each with m levels, then the minimum number of runs required for a linear metamodel is m^p where p is the smallest integer satisfying $m^p > k$. For details on these and other low resolution designs, see a DOE text such as Box, Hunter, and Hunter (1978); Chapter 12 of Law and Kelton (2000) also has a discussion of several basic designs.

Group screening designs, such as the sequential bifurcation (SB) method proposed by Bettonvil and Kleijnen (1997), are other ways of efficiently reducing a long list of potential factors to a short list of important factors. These designs do require the analyst to make more assumptions about the underlying response surface. For example, SB requires the analyst to know the signs of the factor effects, and assumes that a first-order model with negligible errors provides a good approximation of the underlying response. Group screening approaches hold promise for the exploration of ABSs, not only as stand-alone techniques, but also by grouping factors into sets in conjunction with other experimental designs.

Frequency-based (FB) designs are another way of determining factor level settings (Lucas et al. 2002, Wu 2002). Imagine listing the potential scenarios as $t = 1, 2, 3,$ and so forth. The level for factor i (scaled between -1 and $+1$) during scenario t can then be found by setting it to $\sin(2\pi t f_i)$, where f_i is the frequency (in cycles/observation) associated with factor i. Figure 2 displays scatter plots of all pairwise projections for a five-factor FB design, where the oscillation frequencies for factors 1 through 5 are 1/81, 4/81, 10/81, 17/81, and 29/81, respectively. There are 81 design points in total, and this design allows the analyst to estimate all quadratic and two-way interactions without confounding.

Latin Hypercube (LH) designs are efficient and easy to generate (McKay, Beckman, and Conover 1979), and have been coded into many software packages (Sugiyama and Chow 1997). They do not require the analyst to make restrictive assumptions about the response surface and, like FB designs, sample in the interior of the hypercube of factor levels. This *space-filling* behavior allows the analyst to fit complex, and even non-parametric, reponse surface metamodels. Either standard regression packages or other surface-fitting software can be used. Figure 3 illustrates the sampling pattern for a randomly generated LH design. As in Figure 2, scatter plots of all pairwise projections of the combinations of factor levels are shown, but for LH designs the points are uniformly scattered.

We remark that a 3^{5-1} fractional factorial would have the same number of runs, but each projection plot would show only nine combinations of factor levels: one at each corner, one at the center of each side, and only one in the middle. A 9^5 factorial would project regular grids of $9^2 = 81$ points for each sub-plot, and so have space-filling behavior more comparable to the FB and LH designs in Figures 2 and 3. However, it would require 729 times as much data! Note that the FB designs tend to sample less frequently near the centers of the hypercubes and more frequently near the edges, as compared to LH designs. This happens because the sine (or cosine) functions are flatter

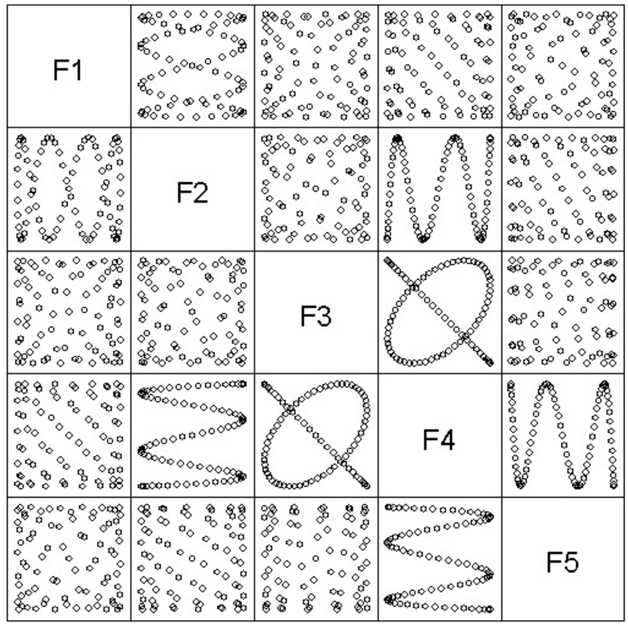

Figure 2: Pairwise Projections of Scaled Factor Levels for a Five-Factor Second-Order Frequency-Based Design

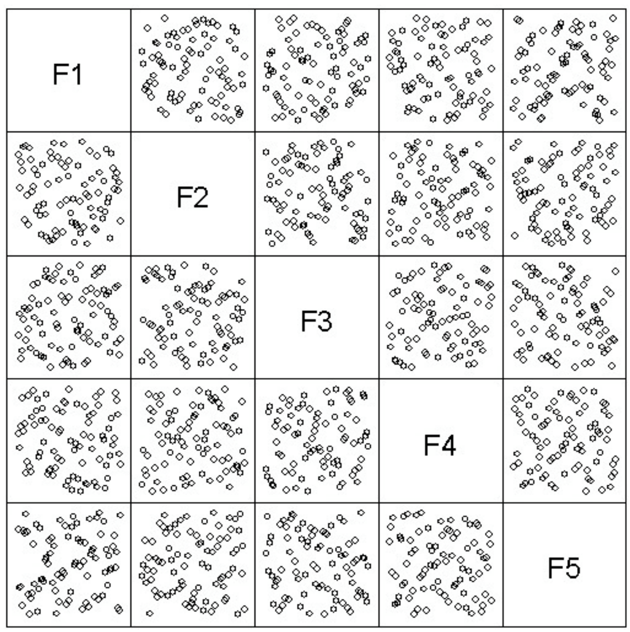

Figure 3: Pairwise Projections of Scaled Factor Levels for a Five-Factor Latin Hypercube Design

near their peaks and valleys, and may make them slightly better for identifying linear (vs. nonlinear) metamodels.

The designs discussed above should be part of an iterative design and analysis process. We tested these on known response surfaces, as well as the examples provided below. The ability to compare against "ground truth" is useful for assessing their strengths and weaknesses.

3.3 Gleaning Insights from Numbers

Numerical summaries can certainly be used to describe subsets of the output data, and regression metamodels can be fit to one or more of the performance measures. These are the most common analytical tools, though other approaches for surface-fitting, such as splines and Kriging, may be better for fitting response surfaces with multiple hill tops, spikes, or thresholds (Cressie 1993, Jin et al. 2001, Van Beers and Kleijnen 2002). Brown (2000) used a five-factor gridded design to examine several intangibles in an ABS motivated by his experiences in Mogadishu, where squads of Blue agents manuever through loosely organized Red forces in an urban environment. At the time, he performed 100 replications of a gridded 5^5 design, but showed that 100 replications of a 2^{5-1} would have been nearly as informative. Wan (2002) used both a full factorial design (with 174,000 runs) and a LH design (with only 4,800 runs) to investigate the effects of human factors on combat outcomes. He found that the LH correctly identified the same important effects that were statistically significant in the model developed from the full factorial.

Cioppa (2002) developed and used nearly-orthogonal LH designs to examine a complex military peace-enforcement operation. He varied 22 factors over 129 levels for each of 100 independently seeded replications of the LH design, and constructed a metamodel of the force exchange ratio as a function of these factors. The results identified the need for maintaining the initiative and speed of execution. This large number of factors meant the experiment could not have been conducted using factorial designs unless the analyst was willing to assume *a priori* that a main-effects metamodel would suffice.

However, constructing metamodels of all the important factors is not the only approach that can be taken. The analyst might be interested in finding a combination of settings for a (perhaps small) group of *decision factors* that yield a robust solution. That is, one which works well over a host of combinations of other uncontrollable factors (Sanchez et al. 1996, Sanchez 2000). In the wake of the *USS Cole* incident in October 2000, a particular concern of the Navy is waterfront force protection—guarding a high-value, in-port asset from attacks from the sea. Childs (2002) built a discrete-event ABS in Java to address this question. In his model, the decision factors were the number of patrol boats, their patrol and intercept speeds, and patrol pattern. Eight patrol boat configurations were pitted against different notional terrorist attacks. The robust approach showed that the patrol pattern and patrol speed were not important, so patrols could be made at low speeds (saving fuel) and in simple patterns. For the factor levels studied, improved protection was associated with more patrol boats and faster intercept speeds. Note that in this example the policy questions related specifically to patrol boat movement characteristics. Thus, realistic movement algorithms were critical.

3.4 Gleaning Insights Visually

Visualization is also extremely helpful—and perhaps better suited for exploratory investigations. Box and whisker plots, bar plots, trellis plots or other small multiples (Tufte 2001), surface and contour graphs, and other graphical methods can provide the analyst with useful information that may not be easy to quantify. We now describe a variety of graphical and exploratory tools that we have found useful.

Regression trees have proven beneficial in understanding and communicating the results of thousands of runs over many factors. Regression trees are more human-readable and can be easier to understand than multiple regression models. Trees simply show the structure in the data. Until a terminal node is reached, the data flowing down the tree encounters one decision at a time (Chambers et al. 1992, Friedman 2002). For example, Figure 4 shows the regression tree for predicting the proportion of Blue casualties in a simulation of a guerrilla attack on Blue forces defending a hilltop position (Ipekci 2002). The data (51,300 responses over 22 factors) to grow the tree were collected using the nearly-orthogonal LH designs of Cioppa (2002). In Figure 4, if the Red stealth is less than 111.5, the number of Red agents is less than 24.5 and the reconnaisance stealth is less than 108.5, Blue takes very low casualties. In other words, given these conditions the guerrillas will not inflict

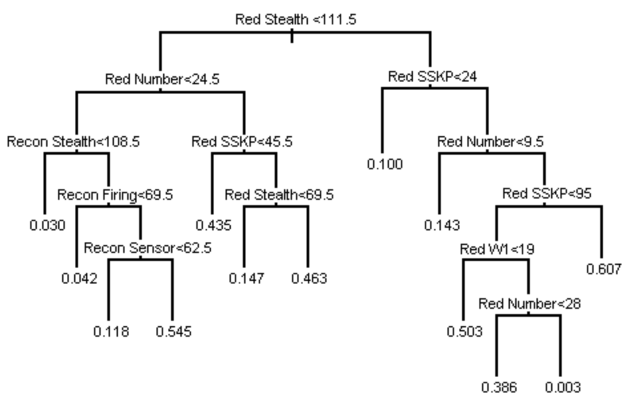

Figure 4: Regression Tree Model of the Proportion of Blue Casualties from an Investigation of Guerilla Combat

many casualties on the Blue force—no matter what values the other parameters take! Furthermore, the relative importance of the variables, using procedures such as MART, can be displayed in a simple bar chart, as in Figure 5.

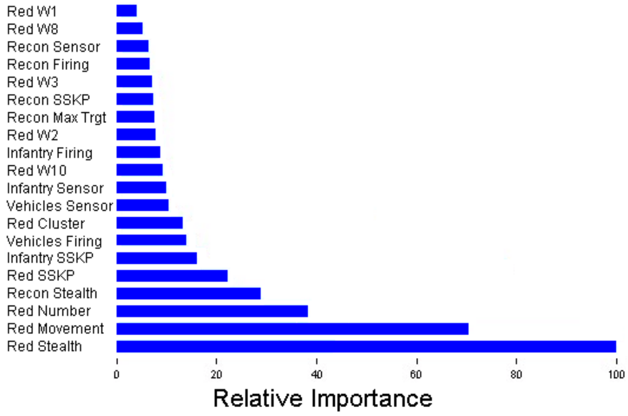

Figure 5: Relative Importance of 22 Factors in an Investigation of Guerilla Combat

Three-dimensional surface plots are readily available in spreadsheet and statistical software packages. In our studies, we have also been making use of the Project Albert Visualization Toolkit (MHPCC 1998, see also Meyer and Johnson 2001) that is being developed specifically to support data farming on Project Albert's suite of agent-based modeling platforms. A screen shot of this visualization tool is shown in Figure 6. The x and y axes can be set to any two of the factors varied during the search. Slider bars for all other factors allow the analyst to quickly scan through and see how the response surfaces change. One can add or delete response surfaces for multiple performance measures; the surfaces themselves can represent performance means, standard deviations, or quantiles. Note that plotting minima and maxima has proven useful for identifying unexpected behaviors—due to unintended consequences of subtle modeling aspects in some cases, and problems related to the computer code in others.

Trellis plots are small multiple plots of various types, including the pairwise projections of factor levels for the FB and LH designs in Figures 2 and 3. We return to the guerrilla combat example (Ipekci 2002), where two performance measures are the proportions of Red and Blue casualties. The trellis plot in Figure 7 displays the relationships after conditioning on the initial number of Red agents in the scenario. In this ABS, the Red side can negate the Blue side's advantage in firepower and number by using 19 to 27 agents in its infiltration. This can be seen from the upper left graph in the trellis plot, where Red inflicts high Blue losses

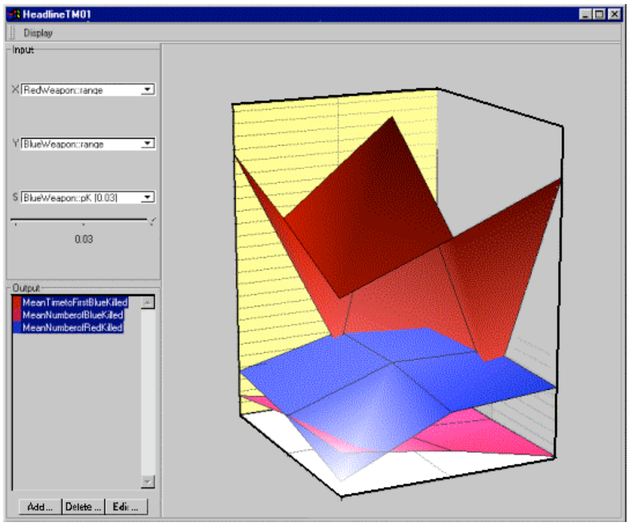

Figure 6: Screenshot Including Multiple Performance Measures from the Project Albert Visualization Toolkit

while suffering low casualties. The Relative Importance graph of Figure 5 shows that Red tactics are important; the trellis plot in Figure 7 provides more insight on why this is so.

Neural networks, in combination with visualization techniques, have proven useful in identifying interesting subregions in our simulation models. In a study assessing the impact of information systems and procedures on battle outcomes, Pee (2002) found that the Blue force can ensure a positive outcome if it can control two of its process latencies—regardless of the values of the nine other factors examined (see Figure 8). The data for this analysis were

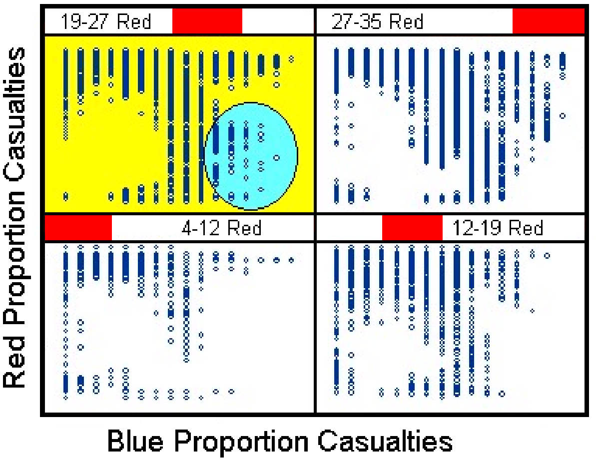

Figure 7: Trellis Plot of the Proportions of Red vs. Blue Casualties, Conditioned on the Number of Red Agents

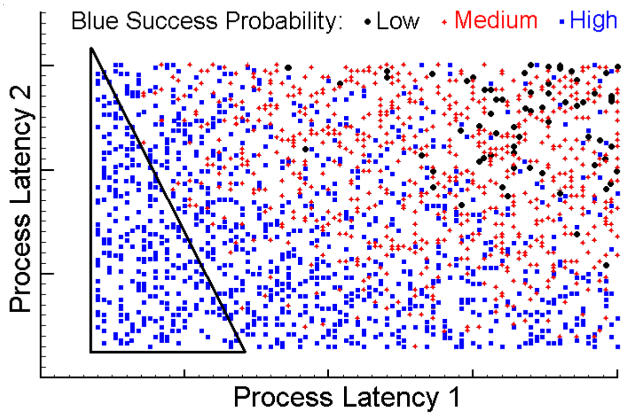

Figure 8: Triangular Region Depicting Outcomes Always Favorable to Blue

generated by 2002 runs (22 sets of Latin hypercubes consisting of 91 uniformly distributed points for each factor) of a Latin hypercube involving 11 factors. This region was found by focusing in on those variables deemed important by the neural network in the data mining software Clementine (SPSS Institute 2001).

Contour plots, or two-dimensional projections of a performance measure, are also useful. Vinyard and Lucas (2001) performed billions of runs on a well-known deterministic combat model (Dewar, Gillogly, and Juncosa 1996). The two-dimensional graph in Figure 9 is one example of the surprising performance that can result. In this figure, the x and y axes represent the initial size of the Red and Blue forces, respectively. One would expect that increasing the initial strength of one side (while holding that of the other side constant) would have a step function effect. This is true in some instances. The horizontal line at a Blue inital force level $B_0 = 800$ shows a single change from Blue winning to Red winning once the initial Red strength crosses a threshold. However, the line at $B_0 = 450$ shows an oscillation of winners (as R_0 increases) over an extended range. These results were determined by a gridded sample of 69,451 points (Vinyard 2001). If this graph is any indication of the subspaces that exist in larger models, then it is easy to see that extreme non-monotonicity might go unnoticed, even when it exists, if samples are taken at only a few intererior points. In larger models the dimensionality of the phase space is incomprehensively vast. Based on the factor level ranges chosen, the analyst may be exploring the model in regions associated with purely monotonic responses. However, it is also possible that they are teetering on the edges of non-monotonic regions like that pictured in Figure 9. Palmore (1996) showed that the non-monotonicity is caused by the chaotic battle trace. These results caused quite a stir in the Defense Modeling and Simulation world.

Figure 9: Winning as a Function of Initial Force Strengths in the Deterministic Dewar Model

This is but one example of chaotic behavior in the model. Figure 10 shows four other subspaces of performance measures. Of the nine subspaces investigated, five exhibit pervasive non-monotonicity, with it showing up in over 80% of the surfaces examined. Although this model is not agent-based, we provide it as an example to illustrate the dangers of assuming that simple *performance* will result from a model that may be simple to program.

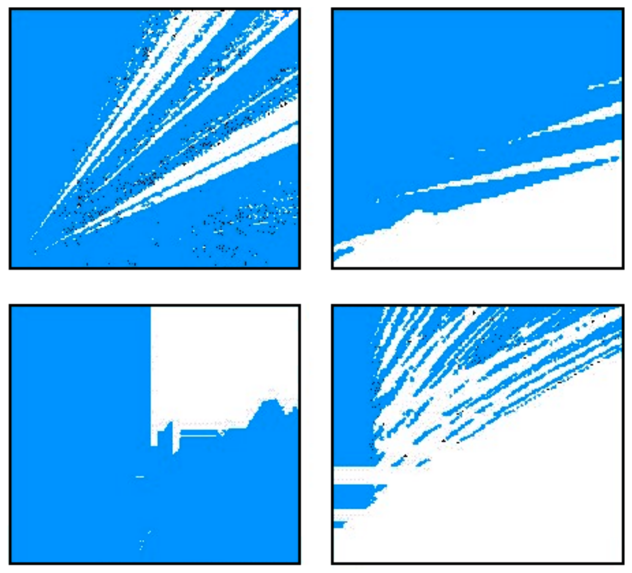

Figure 10: A Maelstrom of Non-Monotonicity in Four Additional Performance Subspaces of the Dewar Model

4 DISCUSSION

ABS is an interesting problem domain for those interested in either modeling or analysis methodology. From the modeling perspective, more work is needed to bring discrete-event tools to the table. Most of the examples we have seen involve time-step models. These have inherent limitations. The results can change dramatically with a different choice of the time-step. It can also be computationally inefficient ($O(n^2)$) to run such models, particularly if the model logic requires every agent to compare their location and/or communicate with every other agent at each time-step.

While computational complexity is an issue in ABS, there is no evidence that the complexity of natural flocks is bounded. If so there would be a "sharp upper bound on the size of natural flocks when individual birds become overloaded by the complexity of their navigation tasks" (Reynolds 1987). This suggests that ABS modelers may be able to develop constructs and algorithms that are not unduly complex in terms of computation, as well as from a modeling perspective. For example, flocking algorithms that are $O(n^2)$ (where n is the number of agents) are not suitable for large flocks. The Java-based Simkit libraries have implemented motion and sensing algorithms that make discrete-event models (particularly event-graph models) more scalable (Buss and Sanchez 2002).

The question of random number generation also may rear its head again. We are used to thinking of random number streams as being "very long." However, if we are making literally millions of runs, where each run might perform millions of random draws, the question as to whether or not random number generators have sufficient cycle lengths is open.

Finally, much (though not all) of the literature describes the development of specific ABS models, rather than ABS modeling platforms. This may be either a benefit or a drawback. It can be difficult to come up with generic reusable agents because of differences in exactly what behaviors should be modeled, and how. To the extent that some prewrapped agents can be put together, this allows for the rapid development and deployment of ABSs for new scenarios. This, in fact, is one of the goals of Project Albert. On the other hand, different models may have slightly (or even markedly) different characteristics and/or decision rules. In this case, testing out insights on multiple modeling platforms can be beneficial in determining whether the insights are real or functions of the platform-specific modeling and implementation assumptions (Brandstein 1999). In the long run, this might help answer questions about how best to implement simple rules to achieve certain types of behaviors.

Most of the designs that simulation practioners are familiar with evolved from traditional DOE methods developed for situations involving only a handful of factors and a nominal amount of experimental units. Unfortunately, many of these traditional designs do not scale well, and are inefficient for exploring ABS models. Nonetheless, the dramatic increase in computing power makes it is feasible to run millions of experiments on simple ABS models. Exploring this new world requires a different mindset. We have touched on a few approaches that we have found useful, but there is ample room for those with interests in simulation methodology to develop additional tools and techniques that are effective, efficient, and easy to use.

ACKNOWLEDGMENTS

This work was supported by a grant from the Marine Corps Combat Development Command as part of the Project Albert effort. We wish to thank the past and present students who have contributed directly or indirectly to this endeavor: Lloyd Brown, Tom Cioppa, Alistair Dickie, Tom Erlenbruch, Arif Ipekci, Eng Yau Pee, Bill Vinyard, Szu Ching Wan, and Hsin-Fu Wu.

REFERENCES

Bettonvil, B. and J. P. C. Kleijnen. 1997. Searching for important factors in simulation models with many factors: sequential bifurcation. *European Journal of Operational Research* 96(1): 180–194.

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: John Wiley & Sons.

Brandstein, A. 1999. Operational synthesis: applying science to military science. *PHALANX* 32(4): 1, 30–31.

Brown, L. P. 2000. Agent Based Simulation as an Exploratory Tool in the Study of the Human Dimension of Combat. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via `<http://library.nps.navy.mil/uhtbin/hyperion-image/00_Brown.pdf>` [accessed August 13, 2002].

Buss, A. H. and P. J. Sanchez. 2002. Building complex models with LEGOs (listener event graph objects). *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers.

Chambers, J. M, Hastie, and J. Trevor. 1992. *Statistical Models in S*. Boca Raton, Florida: CRC Press.

Childs, M. D. 2002. An Exploratory analysis of waterfront force protection measures using simulation. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via `<http://theses.nps.navy.mil/`

Thesis_02Mar_Childs.pdf> [accessed August 13, 2002].

Cioppa, T. M. 2002. Efficient nearly orthogonal and space-filling experimental designs for high-dimensional complex models. Doctoral dissertation, Operations Research Department, Naval Postgraduate School, Monterey, California.

Cressie, N. A. C. 1993. *Statistics for spatial data*. Revised ed. New York: John Wiley & Sons.

Dewar, J., J. Gillogly, and M. Juncosa. 1996. Non-monotonicity, chaos, and combat models. *Military Operations Research* 2(2): 37–49.

Dickie, A. 2002. Modeling robot swarms using agent-based simulation. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via <http://theses.nps.navy.mil/02Jun_Dickie.pdf> [accessed August 13, 2002].

Dudenhoeffer, D. D., D. J. Bruemmer, and M. L. Davis. 2001. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 730–739. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers. Available online via <http://www.informs-cs.org/wsc01papers/096.pdf> [accessed August 13, 2002].

Erlenbruch, T. 2002. Agent-based simulation of German peacekeeping operations for units up to platoon level. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via <http://library.nps.navy.mil/uhtbin/hyperion-image/02Mar_Erlenbruch.pdf> [accessed August 13, 2002].

Friedman, J. H. 2002. Getting Started with MART in R. Stanford University, April 2002. Available online via <http://www-stat.stanford.edu/~jhf/r-mart/tutorial/tutorial.pdf> [accessed August 13, 2002].

Gill, A. and P. Shi. 2002. Movement algorithm verification and issues in joint concept development and experimentation. In *Proceedings of the 5th Project Albert International Workshop*. Available online via <http://www.projectalbert.org/Documents/Albert/conferences/fifth_workshop/germany_main_frame.html> [accessed August 13, 2002].

Horne, G. and M. Leonardi, eds. 2001. *Maneuver Warfare Science 2001*. Quantico, Virginia: Defense Automated Printing Service.

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

Ipekci, A. I. 2002. How agent based models can be utilized to explore and exploit non-linearity and intangibles inherent in guerrilla warfare. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via <http://theses.nps.navy.mil/02Jun_Ipekci.pdf> [accessed August 13, 2002].

Jin, R., W. Chen, and T. Simpson. 2001. Comparative studies of metamodeling techniques under multiple modeling criteria. *Journal of Structural Optimization* 23(1): 1–13.

Kleijnen, J. P. C., S. M. Sanchez, T. Lucas, and T. M. Cioppa. A user's guide to the brave new world of designing simulation experiments. Working paper, Department of Information Management/Center for Economic Research (CentER), Tilburg University, Tilburg, the Netherlands.

Kleinrock, L. 1985. Distributed systems. In *Communications of the ACM* 28(11): 1200–1213.

Lauren, M. K. and R. T. Stephen. 2001. *MANA Map Aware Non-Uniform Automata Version 1.0 User's Manual*. Australia: Land Operations Division, Defence Science and Technology Organisation.

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

Lucas, T. W., S. M. Sanchez, L. P. Brown, and W. C. Vinyard. 2002. Better designs for high-dimensional explorations of distillations. In *Manuever Warfare Science 2002*, eds. G. Horne and S. Johnson, 17–46. Quantico, Virginia: USMC Project Albert.

Marine Core Combat Development Command. 2002. MCCDC Project Albert. Available online via <http://www.projectalbert.org> [accessed August 13, 2002].

Mason, C. R. and J. Moffat. 2001. An agent architecture for implementing command and control in military simulations. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 721–729. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers. Available online via <http://www.informs-cs.org/wsc01papers/095.pdf> [accessed August 13, 2002].

McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2): 239–245.

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

MHPCC, 1998. Project Albert VisTool 2.6. Maui, Hawaii: Maui High Performance Computing Center.

Mizuta, H. and Y. Yamagata. 2001. Agent-based simulation and greenhouse gas emissions training. In *Proceedings of the 2001 Winter Simulation Conference*, eds. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 535–540. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers. Available online via `<http://www.informs-cs.org/wsc01papers/070.pdf>` [accessed August 13, 2002].

Palmore, J. 1996. Research on the causes of dynamical instability in combat models. Technical Report 96/95, U. S. Army Construction Engineering Research Laboratories, Champaign, Illinois.

Pee, E. Y. 2002. An exploratory analysis on the effects of information superiority on battle outcomes. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via `<http://library.nps.navy.mil/uhtbin/hyperion-image/02Mar_Pee.pdf>` [accessed August 13, 2002].

Reynolds, C. W. 1987. Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics* 21(4): 25–34.

Sacks, J., W. Welch, T. Mitchell, and H. Wynn. 1989. Design and analysis of computer experiments. *Statistical Science* 4(4): 409–422.

Sanchez, S. M. 2000. Robust design: seeking the best of all possible worlds. In *Proceedings of the 2000 Simulation Conference* ed. J. Joines, R. Barton, and K. Kang, 69–76. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers. Available online via `<http://www.informs-cs.org/wsc00papers/013.pdf>` [accessed August 13, 2002].

Sanchez, S. M., P. J. Sanchez, J. S. Ramberg, and F. Moeeni. 1996. Effective engineering design through simulation. *International Transactions on Operational Research* 3(2): 169–185.

SPSS Inc. 2001. *Clementine 6.0 User's Manual*. Chicago, Illinois: SPSS Inc.

Sugiyama, S. O. and J. W. Chow. 1997. @Risk, Riskview and BestFit. *OR/MS Today* 24(2): 64–66.

Tufte, E. R. 2001. *The Visual Display of Quantitative Information*. 2nd ed. Cheshire, Connecticut: Graphics Press.

Van Beers, W. and J. P. C. Kleijnen. 2002. Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, forthcoming.

Vinyard, W. C. 2001. Reducing non-monotonicity in combat models. Master's Thesis, Naval Postgraduate School, Monterey, California, 2001. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via `<http://theses.nps.navy.mil/Thesis_01sep_Vinyard.pdf>` [accessed August 13, 2002].

Vinyard, W. C. and T. W. Lucas. 2002. Exploring combat models for non-monotonicities and remedies. *PHALANX* 35(1), 19, 36–38.

Wan, S. C. 2002. An exploratory analysis on the effects of human factors on combat outcomes. Master's Thesis, Naval Postgraduate School, Monterey, California, 2001. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California. Available online via `<http://library.nps.navy.mil/uhtbin/hyperion-image/02Mar_Wan.pdf>` [accessed August 13, 2002].

Wu, H.-F. 2002. Spectral analysis and sonification of simulation data generated in a frequency domain experiment. M.S. Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California.

AUTHOR BIOGRAPHIES

SUSAN M. SANCHEZ is a Professor and Associate Chair of Instruction in the Operations Research Department at the Naval Postgraduate School, where she also holds a joint appointment in the Graduate School of Business and Public Policy. She received a 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, and is currently President of the INFORMS College on Simulation. She serves as the Deputy Editor of *Naval Research Logistics* and as the Simulation Area Editor for the *INFORMS Journal on Computing*; she is a former associate editor of *Operations Research*. Her e-mail and web addresses are `<ssanchez@nps.navy.mil>` and `<http://diana.or.nps.navy.mil/~susan>`.

THOMAS W. LUCAS is an Associate Professor of Operations Research at the Naval Postgraduate School. He received a B.S. in Industrial Engineering and Operations Research from Cornell University, an M.S. in Statistics from Michigan State University, and a Ph.D. in Statistics from the University of California at Riverside. He is a member of the ASA, INFORMS MAS, and MORS. His primary research interests are combat analysis, design of simulation experiments, and robust Bayesian statistics. Previously, he worked as a statistician at RAND and as a systems engineer at Hughes Aircraft Company. His e-mail is `<twlucas@nps.navy.mil>` and a detailed CV can be found at `<http://diana.or.nps.navy.mil/~twlucas/CV.htm>`.