Simulating the variability of actual outcomes

SONIA M. GOLTZ

Michigan Technological University, Houghton, Michigan

and

JAMES E. NORTHEY, JR.

*The Options Clearing Corporation, Chicago, Illinois
and Michigan Technological University, Houghton, Michigan*

A method is described for simulating the variability of outcomes encountered in a variety of settings. Outcome values falling within a specified range are randomly generated and presented across subjects and trials, so that, over time, the average values seen by all subjects in a condition are the same. Randomly varying reinforcement dimensions (such as magnitude) over trials for each subject is recommended to enhance both the internal and the external validity of laboratory results. In addition, it can be used to study the effects of the variability or uncertainty of outcome dimensions such as magnitude and duration in investigations of behaviors such as matching, melioration, and behavioral momentum.

Research into reinforcement effects is examining reinforcement's role in increasingly complex situations, such as decision making in organizations (e.g., Goltz, 1992, 1993; Hantula & Crowell, 1994a, 1994b). Complex phenomena are often dependent on an array of highly varying factors, which laboratory researchers attempt to successively isolate. Thus, one or two independent variables of interest are manipulated and other potential influences are controlled, either by holding them constant or by randomly distributing them across conditions. For example, instructions in an experiment are held constant while subjects are randomly assigned to conditions.

In research into reinforcement effects, the former method is often used. Characteristics of reinforcement are held constant across subjects, trials, and conditions (e.g., amount or duration of reinforcement). Our purpose in this paper is to recommend that extraneous reinforcement dimensions be randomly distributed across subjects, conditions, and trials rather than held constant. We propose a method to do so and describe an application of the method. In addition, we suggest that this tool be used to study effects of manipulating the variability of a number of reinforcement dimensions.

VALIDITY CONSIDERATIONS

Internal Validity

Holding certain dimensions of a factor constant while studying the effects of other dimensions could be classified as a form of mono-method bias, a threat to construct

validity, according to Cook and Campbell (1979). As they stated, "Indeed, when all the manipulations are presented the same way . . . then the method is itself an irrelevancy whose influence cannot be dissociated from the influence of the target construct" (p. 66). As Fetterman (1994) pointed out with respect to research into reinforcement effects, while experimenters try to identify one dimension or another as the single controlling factor, both dimensions may affect the behavior. For example, perhaps reinforcement probability affects the response rate of subjects only when the duration of access to reinforcement is held constant or only when it is a certain length. If probability is studied without ever varying duration, however, researchers might believe that only rate of reinforcement contributes to the observed behavior.

To avoid this problem, researchers can systematically vary reinforcement dimensions simultaneously to identify any interactions. But when this is not desirable or feasible, an alternative method is to randomly distribute values of the dimensions not being investigated across subjects, conditions, and trials. When dimensions of outcomes other than the hypothesized independent variable are randomly varied, effects observed cannot be attributed to the combined effects of a particular constant dimension and the manipulated independent variable.

External Validity

Holding dimensions constant can also affect the external validity of research results. As Lewin (1979) noted, "external validity, the generalizability of your results to other settings, is also reduced if you unfortunately hold constant a variable whose different levels have great impact in a second setting" (pp. 113-114). Two concerns are whether the same results would occur with other values of the controlled factor and whether they would occur when the factor was not constant, which can be expected in many settings. For example, the outcomes that people

A previous version of this paper was presented at the 1997 Conference of the Society for Industrial/Organizational Psychology in St. Louis. Correspondence regarding this paper should be addressed to the first author at the School of Business and Economics, Michigan Technological University, 1400 Townsend Avenue, Houghton, MI 49931 (e-mail: smgoltz@mtu.edu).

experience every day are quite variable in their magnitude, sequence, duration, and frequency.

Thus, when one is studying situations in which variability typically occurs, reinforcement dimensions such as magnitude, sequence, duration, and frequency should be randomized within a range of values across subjects, conditions, and trials. This increases the generalizability of the results in two important ways. First, results occur even when a particular dimension is varied *within* subjects. In other words, each individual subject sees a sequence of outcomes varying in the dimension. Second, they occur even though specific outcome patterns experienced are different *across* subjects. The variations within subjects allow one to state with some confidence that the observed effects are likely even in situations in which there is a lot of noise in the feedback individuals experience (e.g., fluctuating stock prices or market share). The variations across subjects allow one to conclude that the observed effects are likely across individuals who have experienced differing patterns of outcomes.

Limitations of Using Constant Values:

An Example

One complex, multiply determined situation that has received recent attention in research is the persistence or escalation of commitment, the tendency of decision makers to increase allocations to a course of action following a setback (see Staw, 1996, for a review). A factor found to affect this behavior is the decision maker's history of outcomes for previous allocations (Goltz, 1992, 1993; Hantula & Crowell, 1994a, 1994b), with a history of irregular or variable reinforcement predicting greater persistence and escalation than a history of fixed or continuous reinforcement.

In most studies of reinforcement effects on escalation, amounts of outcomes have been kept constant across trials and subjects within a condition. Thus, for example, in two experiments Goltz (1992) simulated the task of stock investing. Each subject was asked to "manage" \$10,000 by periodically deciding how much to invest in a "stock" alternative. Subjects in the large-magnitude condition always received \$30 per \$100 invested in the stock when investment in the stock resulted in a gain and always lost \$10 (per \$100 invested in the stock) when investment in the stock resulted in a loss. Small-magnitude subjects saw the same amount on loss trials, but always received \$10 per \$100 invested in the "stock" alternative on gain trials. Although the Goltz (1993) experiment used a different scenario from the two in Goltz (1992), subjects still saw constant amounts for gains and constant amounts for losses.

This use of constant values calls for caution in interpreting results. Constant values might make the situation appear artificial to subjects and result in artificial responses. Subjects in a new decision-making situation have been found to respond as if they have previously experienced variably occurring reinforcement (Goltz, 1993, Experiment 2). It is probable that they also have experienced variability in terms of other reinforcement dimen-

sions. Therefore, research that randomizes outcome dimensions within a range of values simulates subjects' experiences in other settings better, and the results are less likely to be questioned because of threats to internal and external validity. For this reason, a randomization method was developed and used in subsequent studies.

OUTCOME GENERATION TOOLS

Pseudo-Random Number Generators

Software subroutines that generate random values can be designed to simulate the particular type of variability of concern (e.g., magnitude, duration, delay). The subroutine can be written to vary outcomes within constraints (e.g., a certain range and mean) determined by the experimenter. In fact, random numbers are used to simulate stochastic processes in computerized models of business, economic, social, biological, physical, and chemical systems over time (Maisel & Gnugnoli, 1972). It also makes sense to use them in experiments on psychological and behavioral processes such as decision makers' responses to varying outcome patterns.

These subroutines are termed "pseudo-random number generators" because the numbers generated are obtained from strictly determined calculations but meet the tests for random properties (Maisel & Gnugnoli, 1972). For example, generators using the linear congruential method contain a starting value (X_0), often called a "seed." To begin the sequence, X_0 is run through a linear equation (e.g., it is multiplied by another value; then this resulting number is added to a constant; etc). To obtain the next number in the sequence, this generated number replaces X_0 . Sequences generated in this way are not really random; all have a repeating cycle. However, the equation is usually written to maximize the length of the cycle and to generate values meeting properties expected of random numbers, such as that they be equally distributed if they are obtained from a uniform distribution.

Pseudo-random number generators are available in most software libraries, such as Visual Basic, Visual C++, and JAVA. Table 1 highlights functions available in these and other libraries for generating random data from various distributions. For programming pseudo-random number calculations, a good resource is Knuth's (1981) *The Art of Computer Programming: Semi-Numerical Algorithms*. He presents algorithms for the linear congruential method as well as for other methods of generating random numbers, such as a quadratic method and an additive method. He also provides an algorithm for the technique of "shuffling" that can be used in combination with the linear congruential method.

VI and VR Schedule Generating Programs

Computer programs have been written and used specifically for generating variable interval (VI) and variable ratio (VR) schedules of reinforcement (see, e.g., Chayer-Farrell & Freedman, 1987; Hale, Myerson, & Miezin, 1982; Hantula, 1991). They are used to deliver re-

Table 1
Functions Available in Common Software Libraries for Generating Random Samples From Various Distributions

Probability Distribution	Method or Technique	Comments
Microsoft Excel 6.0 or later		
Excel provides several worksheet functions that can be used to generate random samples.		
Uniform(0,1)	=RAND()	Where: lower - Cell reference for lower bound of the distribution upper - Cell reference for the upper bound of the distribution Note: This provides a continuous distribution. The result can be rounded to the nearest integer for a discrete distribution.
Uniform(lower, upper)	=(upper-lower)*RAND()+lower	
StandardNormal()	=NORMSINV(RAND())	
Normal(mean,stdev)	=NORMINV(RAND(),mean,stdev)	Where: mean - Cell reference or value for the arithmetic mean of the normal distribution stdev - Cell reference or value for the standard deviation of the normal distribution
Visual Basic 3.0 or later		
Uniform(0,1)	result = rnd	Listing 1
Uniform(lower, upper)	result =(upper-lower)*rnd + lower	
StandardNormal()	Listing 1	
ANSI C and ANSI C++		
The Microsoft Visual C++ compiler (version 4.2) was used for the following examples.		
Uniform(0,1)	srand(seedvalue); result = rand()/double(RAND_MAX);	Note: The use of rand() from the standard C library is not recommended for large samples as this is normally implemented as a 16 bit integer method. Some implementations of C provide a drand48() function which is recommended over the rand() method.
Uniform(lower, upper)	result = (Upper-Lower)*(rand()/double(RAND_MAX) + Lower;	
StandardNormal()	Listing 2	
Java 1.0		
The Symantec Café version of Java was used for the following examples. Note that Java is an interpreted language and as such provides meager performance at this time, especially for numerically intense calculations.		
Uniform(0,1)	result = math.random();	Listing 3
Uniform(lower, upper)	result = (Upper - Lower) *math.random() + Lower;	
StandardNormal()	Listing 3	

inforcers after an interval during which a response has occurred or after a certain number of responses have occurred. The number of responses or duration of the interval are varied around the mean interval or response to reinforcement ratio desired by the researchers.

However, pseudo-random number generators offer more flexibility than do these programs, most of which are based on Fleshler and Hoffman's (1962) equation (e.g., Hantula, 1991). Their equation improves on earlier programs by avoiding the generation of values that resulted in a correlation between the probability of reinforcement and the time since reinforcement. But it results in a negatively skewed distribution of values with a large range. For instance, Fleshler and Hoffman provide a progression of 12 intervals generated by their equation for VI 30 sec. The resulting numbers range from 1.287 to 104.547, with eight numbers falling under the mean and four numbers falling over the mean. The largest number is thrice the mean value. Although this distribution of

values is appropriate for certain uses (see discussion in Catania & Reynolds, 1968), it is likely to be inappropriate for other uses.

When other distributions are desired or when the experimenter wishes to control the range of values, the pseudo-random number generator can be used. For example, one could create a progression of intervals from a uniform distribution of values that averages to be 30 sec with the specification that all values fall within the range from 5 to 55 sec. Alternatively, one could selectively change this mean, range, and/or distribution. Tests can be performed on the sequence generated by the pseudo-random number generator to ensure that numbers do not deviate from what would be expected from a distribution of randomly generated numbers. For example, if the pseudo-random number generator is chosen over the Fleshler-Hoffman (1962) equation, and if it is important to ensure that values are not correlated with time since reinforcement, tests can be performed to determine

whether the values generated by the pseudo-random generator are auto-correlated.

ISSUES TO CONSIDER IN GENERATING PSEUDO-RANDOM NUMBERS

Considering the Sample Size of Numbers

A number of computer languages provide a method for generating a pseudo-random number based on a uniform distribution. For speed and simplicity, most commercially available subroutines use the linear congruent method for calculating this random value. Depending on the constants selected for this method, the random number generator may repeat the sequence after a large number of samples. Thus, these random number generators are adequate for smaller samples of values, but when large samples are needed, the values that the system-supplied routine generates should be analyzed in order to make sure that they meet tests of randomness. We recommend that this analysis be performed when greater than 30,000 random numbers are needed. When tests indicate that the system-supplied values are inadequate, one can refer to Press, Teukolsky, Vetterling, and Flannery (1992), who provide C functions based on Knuth's (1981) guidelines for generating random numbers for very large samples.

Selecting and Generating the Distribution

Selecting the distribution. The distribution used in most commercially available programs is uniform; that is, all the values of an outcome within the specified range have an equal chance of occurring. Using a uniform distribution may be sufficient for many applications of the pseudo-random number generator. In particular, it is useful for bolstering internal validity by demonstrating that effects observed cannot be attributed to the combined effects of a particular constant dimension and the manipulated independent variable. In addition, it is useful for extending external validity by demonstrating that an effect can be found across several values of a dimension, since all values have an equal chance of occurring.

At times, though, researchers may wish to simulate actual outcomes more closely. They may want to be able to show that effects observed will occur even in the presence of other dimensions that, in their natural state, tend to vary within particular ranges and distributions. In these instances, researchers should design the randomizer so that it generates the distribution that tends to occur in the field. Ideally, one would first assess the naturally occurring distribution, then design the experiment to simulate it. Stock market changes, for example, have been found to follow a logarithmic normal distribution (Hull, 1993); experiments simulating stock market changes should therefore generate random values using the logarithmic normal distribution. When it is not possible to ascertain what the naturally occurring distribution is, one might use the normal distribution, since many types of outcomes are likely to be normally distributed.

Generating the distribution. While most computer languages and spreadsheet programs provide some sort of

method for generating a uniform distribution, few provide methods to generate random numbers from a normal distribution. One of the few exceptions to this is Microsoft Excel, which provides a rich set of worksheet functions for generating random samples from various probability distributions, including the normal distribution. Other products, such as @RISK and Crystal Ball, can be used to generate random samples. To generate a sample from a normal distribution, the technique usually employed is to generate a random number from a uniform distribution and then transform this value into the desired distribution—in this case, the normal (Maisei & Gnugnoli, 1972). An algorithm often used for generating a normal distribution from a random sample is provided in the National Bureau of Standards' *Handbook of Mathematical Functions* (Abramowitz & Stegun, 1964). An implementation of the inverse normal distribution for use in generating a standard normal sample is shown for Visual Basic, C++, and Java, in Listings 1–3, respectively. An alternative approach using two random samples is provided by Press et al. (1992). Knuth (1981) also provides algorithms for generating alternative distributions.

Assessing the "Randomness" of Generated Values

At times it is important to determine the extent to which a sequence is sufficiently random, as, for example, when a large sample of numbers is needed. As Press et al. (1992) note, what is random enough for one application may not be random enough for another. For instance, one application may require that values not be correlated, whereas another application may require that the values be equally distributed. Therefore, to determine the randomness of various dimensions of a sequence, there are 10 different statistical tests that can be performed on the sequence, as Knuth (1981) describes. Knuth presents both equations and computer algorithms for the more complicated of the 10 tests.

Many of these tests use the chi-square statistic to compare expected frequencies of events with observed frequencies. The *frequency test*, for example, assesses whether numbers occur significantly more or less frequently than would be expected, on the basis of characteristics of the distribution. In a uniform distribution, for instance, numbers should be equally distributed. The *serial test* assesses whether pairs or trios of successive numbers occur more or less frequently than would be expected. The *poker test* examines groups of five successive integers for patterns and tests whether these patterns occur more or less frequently than would be expected. Statistics other than chi-square are more appropriate for examining other aspects of the randomness of numbers. The *serial correlation test*, for example, correlates a set of values with their immediate successors. Numbers sequentially generated should not be correlated with each other.

Knuth (1981) considers tests for the distribution of numbers and serial correlations to be the two weakest tests, in that nearly all number generators pass these tests. Of the 10 tests, Knuth suggests that the strongest tests are the *run test* and the *collision test*. The run test examines the length of sequential number segments that are

increasing or decreasing. Since adjacent runs are not independent, the statistic V is used instead of chi-square. This test can indicate whether the constant used in a linear congruential generator is not large enough. As the numbers in a sequence reach infinity, run lengths become normally distributed. When a constant is not large enough, runs will be longer than normal. This test is especially recommended for use with computerized random number generators because most of them, such as the C subroutine library, use the linear congruential method.

The collision test is also recommended for detecting deficiencies of computerized generators. The collision test can be used to test the randomness of values generated by a random number generator simultaneously on more than one dimension. Knuth (1981) describes it in this way: "Suppose we have m urns and throw n balls at random into those urns, where m is much greater than n . Most of the balls will land in urns that were previously empty, but if a ball falls into an urn that already contains at least one ball we say that a 'collision' has occurred. The collision test counts the number of collisions, and a generator passes this test if it doesn't induce too many or too few collisions" (p. 69). This test is recommended when the number of categories (e.g., range of possible values) is larger than the number of observations (e.g., actual values generated). It uses the expected number of observations per category, which will be a fraction when there are more categories than observations, to calculate the expected number of collisions per category (urns in the previous example). Expected collisions are then summed across categories, and this value is compared with the observed number of collisions.

APPLYING THE NUMBER GENERATOR: AN EXAMPLE

A recent experiment illustrates the use of the pseudo-random number generator to simulate the variability of outcomes found in organizations (Goltz, in press, Experiment 1). This study examined behavioral momentum as an explanation for escalation in an organizational context. As in the Goltz (1993) experiment, subjects performed a computerized task, playing the role of a company decision maker trying to increase market share. The subjects were asked to make a series of decisions concerning the amount of additional money to allocate to research and development during the course of 43 "quarters" over the period of several "years." Immediately following each subject's allocation for a quarter, information was presented concerning any increases or decreases in market share. During the first 24 quarters ("acquisition"), a variable schedule of reinforcement was presented in terms of irregularly occurring increases and decreases in market share. Continuous decreases in market share for investing in research and development ("extinction") were presented during the last 19 quarters.

In this experiment, the pseudo-random number generator was used in three ways: (1) to present different rates

of reinforcement using a variable schedule; (2) to randomize, with a specified range and mean, the magnitude of the increases that subjects experienced during acquisition; and (3) to randomize, with a specified range and mean, the magnitude of the decreases that subjects experienced (both during acquisition and extinction). The first use is similar to what has been done frequently in operant research; values were not generated using Fleshler and Hoffman's (1962) equation, however. The latter two uses of the randomizer served the purpose of decreasing the artificiality of the task and, as a consequence, of increasing the external validity of the results. Thus, these two uses are described here in more detail.

Magnitude of increases experienced by subjects was manipulated in this experiment to create three conditions: *small*, *medium*, and *large*. This variable could have been manipulated by presenting, for each particular condition, constant amounts of increases for all trials and subjects, as in the Goltz (1992, 1993) experiments. For example, the small-, medium-, and large-magnitude subjects could have been presented with constant increases of .05%, .15%, and .25%, respectively. However, subjects' responses to unchanging magnitudes of increases may not be the same responses that would occur to market share increases that vary, a situation that is likely to occur in a variety of organizational contexts. Thus, a different approach was taken in the experiment. Amounts of gains were individually generated for each trial for each subject using a pseudo-random number generation subroutine written in the ARAGO programming language. (ARAGO is a derivation of the dBASE III language. Since most readers are likely to be unfamiliar with this language, the listing is not provided here. For a program listing, contact the first author.) Subjects in the large-magnitude condition experienced an average increase of .25% on trials that resulted in an increase in market share (range: .21%–.29%). Subjects in the medium-magnitude condition experienced an average increase of .15% (range: .11%–.19%), and subjects in the small-magnitude condition experienced an average increase of .05% (range: .01%–.09%) on trials that resulted in an increase. Thus, the specific pattern of amounts of gains seen by each subject was different, although all subjects in each condition experienced the same average increase. Amounts varied within a constant range across conditions (.09%) and were generated from a uniform distribution.

Unlike magnitude of increases, magnitude of decreases was not an independent variable in the experiment. However, in order to reduce artificiality, the decreases were also presented using a pseudo-random number generator. Amounts of decreases in market share averaged to be .15% in all conditions during both acquisition and extinction. The actual decreases presented to subjects ranged from .11% to .19% and were also generated using a uniform distribution. As with the increases, these amounts were individually generated for each subject and trial, so the specific sequences of the magnitudes of decreases subjects experienced differed across individuals.

Results of repeated measures analyses of variance indicated that the magnitude manipulation significantly affected allocations both during acquisition [$F(2,227) = 5.41, p < .01, \eta^2 = .05$] and during extinction [$F(2,227) = 5.01, p < .01, \eta^2 = .04$]. Therefore, use of this approach to control dimensions of reinforcement helped increase the generalizability of results of the study in two important ways. Significant effects of magnitude were found for both experimental phases even when the actual amounts of increases and decreases seen by subjects varied *within* subjects across trials and when patterns of outcomes varied *across* subjects. This allowed the experimenter to conclude with some degree of confidence that the effects are likely to occur even in situations in which there is a lot of noise in the feedback that individuals experience for allocation decisions, such as when amounts of increases or decreases in outcomes fluctuate. In addition, these results are likely to occur even when the particular patterns of outcomes experienced differ across individuals, which is possible in a variety of organizational settings. These conclusions could not have been made with as much confidence had magnitudes of increases and decreases been held constant across trials and subjects in the experiment.

In the example presented here, the variability of outcome magnitude was simulated. However, the pseudo-random number generator can be used to simulate other characteristics of reinforcers that tend to vary in many settings, such as the delay or duration of reinforcement. For example, when the reinforcer in an experiment is access to a game or a break after hard work on a task, the duration and delay of access could be varied around a mean to simulate other situations. Agents allowing access to reinforcers in other settings may not always be in a position to immediately provide the access to the game or break or to closely monitor the amount of time someone has been on break or playing a game. Thus delay and access time are likely to vary within a range such as 5–30 min.

USING THE PSEUDO-RANDOM NUMBER GENERATOR TO STUDY EFFECTS OF VARIABILITY

Although to this point we have focused on the need for researchers to vary dimensions of outcomes to increase internal and external validity, this technique for randomizing dimensions of outcomes has other potential applications. In particular, it can be used to study responses to the degree of variability of a number of outcome dimensions. This line of investigation would be valuable in terms of both extending research on reinforcement effects and understanding implications of using randomly varied values rather than constant values in experiments.

Extending Research on Reinforcement Effects

In research on reinforcement effects such as matching (Rachlin, 1973), behavioral momentum (Nevin, 1988), and

melioration (Herrnstein, 1982), it has been found that the rate of reinforcement plays a large role in response patterns. The matching law (Herrnstein, 1970) captures this relationship:

$$\frac{P_A}{P_A + P_B} = \frac{R_A}{R_A + R_B}$$

(where P refers to number of responses, R refers to number of reinforcers, and A and B refer to two alternatives). Reinforcement rate is a function of dimensions such as amount, duration, probability, and delay. Recent research investigations into the variability of some of these dimensions have revealed some interesting effects. Pseudo-random number generators, which can be used to control outcome ranges, means, and distributions, could be used to investigate effects of the variability of reinforcement dimensions more rigorously.

For example, a number of studies suggest that subjects exhibit a preference for uncertain reinforcement over certain reinforcement, such as a preference for variable over fixed delays of delivery of reinforcement (Fantino, 1967; Herrnstein, 1964; Rider, 1983) and a preference for uncertain large over certain small reinforcers (Young, 1981). This preference for uncertainty is significant because Mellon and Shull (1986) and others (Grace & Nevin, 1997) found that outcome preferences in choice procedures are predictive of resistance to change in multiple schedules. The bulk of research to date that has been conducted on variability has concerned effects of variable sequence and variable delay. The pseudo-random number generator could be used to more systematically examine reinforcement preferences for various types of uncertainty (e.g., varying means, ranges, distributions) for these and other reinforcement dimensions (such as the amount of food or the duration during which food is available in pigeon experiments). Researchers could examine, for example, whether responses are the same when two distributions of outcomes have the same form (e.g., normal, uniform, or bimodal) but different ranges. Another direction for future research might be to use the pseudo-random generator to simultaneously manipulate the degree of variability (e.g., constant, moderately variable, highly variable) of two or more dimensions of reinforcement (e.g., delay and amount or delay and probability) to examine whether variability effects are additive or interactive.

Along similar lines, researchers interested in human decision making in organizational and economic contexts could use the pseudo-random number generator to study effects of changing the level of uncertainty of decision-making outcomes. Uncertainty, a direct result of the variability with which outcomes occur, is a topic discussed frequently in this literature. For example, economists consider it important for investors to look at both the rate of return and the risk of the investment. They measure uncertainty (risk) by assessing the dispersion of

returns on investments around a mean (Hirshleifer & Glazer, 1992). In addition, the uncertainty of a decision situation, called "equivocality," has been discussed as a factor involved in the tendency of decision makers to escalate (Bowen, 1989; Bragger, Bragger, Hantula, & Kirnan, in press).

Understanding Effects of Simulating Outcome Variability in Research

The need to study effects of variability is important as well from the standpoint of using the pseudo-random number generator to increase the internal and external validity of studies. Using variable values, rather than constant values, of outcomes or reinforcers may do more than extend generalizability. It may also change the nature of the effects obtained. In particular, the aforementioned research indicating a preference for variability and a relationship between preference and resistance to change suggests that increased variability may increase the persistence of responses.

In the second experiment reported by Goltz (in press) this possibility was examined by manipulating the variability of the magnitude of outcomes. The task was the same as in the first experiment. However, while all subjects saw average increases of .15% and average decreases of .15% for every million dollars invested in research and development, some subjects saw no variability in that number (no-variability condition), others saw numbers ranging from .13% to .17% (small-variability condition), and still others saw numbers ranging from .11% to .19% (large-variability condition). As predicted, large-variability subjects were more persistent in their responses during extinction. This generated an effect size of $\eta^2 = .05$, which was greater than the effect size found for rate of reinforcement ($\eta^2 = .04$), also manipulated in this experiment. Owing to low power, however, significance testing did not detect this variability effect [$F(2,98) = 2.42, p = .09$].

Results of this study and future research into variability effects will serve to point out which past studies using constant values might be limited in their generalizability to other situations varying in those values. When variability effects are found for a dimension, caution should be taken in interpreting past studies using constant values for that dimension. Also, future studies should be designed to vary that dimension to increase the generalizability of results. When variability effects are not found for a dimension, the use of the pseudo-random number generator is not needed for external validity reasons, but may still be retained for internal validity considerations.

CONCLUSION

In summary, this paper proposes the use of the pseudo-random number generator to simulate the variability of outcomes that occurs in many settings. This tool has a

number of potential applications. Randomly varying reinforcement dimensions (such as magnitude) over trials for each subject can be used for methodological reasons, such as to avoid mono-method bias and to extend the external validity of results. This use increases the credibility of laboratory research results. Further, the tool can be used to facilitate systematic research on effects of variability or uncertainty, topics that have interested researchers in several different fields.

This technique is powerful because it can be used to vary a wide variety of outcome dimensions in order to study a number of different behaviors. Thus far, for example, the majority of research on the matching law has concerned effects of rate of reinforcement as manipulated through delay or probability. The random number generator has the potential to facilitate increased research on other reinforcement parameters that are likely to be involved in effects such as matching, melioration, and momentum.

REFERENCES

- ABRAMOWITZ, M., & STEGUN, I. A. (1964). *Handbook of mathematical functions with formulas, graphs and mathematical tables*. Washington, DC: U.S. Department of Commerce.
- BOWEN, M. G. (1989). The escalation of phenomenon reconsidered: Decision dilemmas or decision errors? *Academy of Management Review*, *12*, 52-66.
- BRAGGER, J. D., BRAGGER, D., HANTULA, D. A., & KIRNAN, J. (in press). Hysteresis and uncertainty: The effect of uncertainty on delays to exit decisions. *Organizational Behavior & Human Decision Processes*.
- CATANIA, A. C., & REYNOLDS, G. S. (1968). A quantitative analysis of the responding maintained by interval schedules of reinforcement. *Journal of the Experimental Analysis of Behavior*, *11*, 327-383.
- CHAYER-FARRELL, L., & FREEDMAN, N. L. (1987). CORE: Computer-controlled operant reinforcement. *Behavior Research Methods, Instruments, & Computers*, *19*, 319-326.
- COOK, T. D., & CAMPBELL, D. T. (1979). *Quasi-experimentation: Design and analysis issues for field settings*. Boston: Houghton Mifflin.
- FANTINO, E. (1967). Preference for mixed- versus fixed-ratios. *Journal of the Experimental Analysis of Behavior*, *10*, 35-43.
- FETTERMAN, J. G. (1994, May). *An experimental synthesis of timing and counting processes*. Paper presented at the Conference of the Society for the Quantitative Analyses of Behavior.
- FLESHLER, M., & HOFFMAN, H. S. (1962). A progression for generating variable-interval schedules. *Journal of the Experimental Analysis of Behavior*, *5*, 529-530.
- GOLTZ, S. M. (1992). A sequential learning analysis of continued investments of organizational resources in nonperforming courses of action. *Journal of Applied Behavior Analysis*, *25*, 561-574.
- GOLTZ, S. M. (1993). Examining the joint roles of responsibility and reinforcement history in recommitment. *Decision Sciences*, *24*, 977-994.
- GOLTZ, S. M. (in press). Can't stop on a dime: The roles of matching and momentum in persistence of commitment. *Journal of Organizational Behavior Management*.
- GRACE, R. C., & NEVIN, J. A. (1997). On the relation between preference and resistance to change. *Journal of the Experimental Analysis of Behavior*, *67*, 43-65.
- HALE, S. S., MYERSON, J., & MIEZIN, F. M. (1982). Top-down design for a system to control operant choice experiments. *Behavior Research Methods & Instrumentation*, *14*, 272-275.
- HANTULA, D. A. (1991). A BASIC program for generating values for

- variable interval schedules of reinforcement. *Journal of Applied Behavior Analysis*, **24**, 799-801.
- HANTULA, D. A., & CROWELL, C. R. (1994a). Behavioral contrast in a two-option analogue task of financial decision making. *Journal of Applied Behavior Analysis*, **27**, 607-617.
- HANTULA, D. A., & CROWELL, C. R. (1994b). Intermittent reinforcement and escalation processes in sequential decision making: A replication and theoretical analysis. *Journal of Organizational Behavior Management*, **14**, 7-36.
- HERRNSTEIN, R. J. (1964). Aperiodicity as a factor in choice. *Journal of the Experimental Analysis of Behavior*, **7**, 179-182.
- HERRNSTEIN, R. J. (1970). On the law of effect. *Journal of the Experimental Analysis of Behavior*, **13**, 243-266.
- HERRNSTEIN, R. J. (1982). Melioration as behavioral dynamism. In M. L. Commons, R. J. Herrnstein, & H. Rachlin (Eds.), *Quantitative analyses of behavior: Vol. 2. Matching and maximizing accounts* (pp. 433-458). Cambridge, MA: Ballinger.
- HIRSHLEIFER, J., & GLAZER, A. (1992). *Price theory and applications*. Englewood Cliffs, NJ: Prentice-Hall.
- HULL, J. C. (1993). *Options, futures, and other derivative securities* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.
- KNUTH, D. (1981). *The art of programming: Semi-numerical algorithms* (2nd ed., Vol. 2). Reading, MA: Addison-Wesley.
- LEWIN, M. (1979). *Understanding psychological research*. New York: Wiley.
- MAISEL, H., & GNUGNOLI, G. (1972). *Simulation of discrete stochastic systems*. Chicago, IL: Science Research Associates.
- MELLON, R. C., & SHULL, R. L. (1986). Resistance to change produced by access to fixed-delay vs. variable delay terminal links. *Journal of the Experimental Analysis of Behavior*, **46**, 79-92.
- NEVIN, J. A. (1988). Behavioral momentum and the partial reinforcement effect. *Psychological Bulletin*, **108**, 44-56.
- PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., & FLANNERY, B. P. (1992). *Numerical recipes in C: The art of scientific computing* (2nd ed.). New York: Cambridge University Press.
- RACHLIN, H. (1973). Contrast and matching. *Psychological Review*, **80**, 217-234.
- RIDER, D. P. (1983). Preference for mixed versus constant delays of reinforcement: Effect of probability of the short, mixed delay. *Journal of the Experimental Analysis of Behavior*, **39**, 257-266.
- STAW, B. M. (1996). The escalation of commitment: An update and appraisal. In Z. Shapira (Ed.), *Organizational decision making* (pp. 191-215). New York: Cambridge University Press.
- YOUNG, J. S. (1981). Discrete-trial choice in pigeons: Effects of reinforcer magnitude. *Journal of the Experimental Analysis of Behavior*, **35**, 23-29.

Listing 1 Visual Basic Implementation to Generate a Standard Normal Deviate

```

'Inverse Normal Distribution
'
Function InverseNormalDist(x As Double) As Double
    Dim k, b1, b2, b3, b4, b5 As Double

    If x >= 0 Then
        k = 1# / (1 + 0.2316419 * x)
        b1 = 0.31938153 * k
        b2 = -0.356563782 * k * k
        b3 = 1.781477937 * k ^ 3
        b4 = -1.821255978 * k ^ 4
        b5 = 1.330274429 * k ^ 5
        InverseNormalDist = 1 - NormalDist(x) * (b1 + b2 + b3 + b4 + b5)
    Else
        InverseNormalDist = 1 - InverseNormalDist(-x)
    End If
Exit Function
End Function
'Normal Function
Function NormalDist(ByRef x As Double) As Double
    NormalDist = (1 / Sqr(2 * 3.14159265359)) * Exp(-(x * x) / 2)
Exit Function
End Function

```

(Continued on next page)

Listing 2
C / C++ Implementation to Generate
a Standard Normal Deviate

```

#include <stdlib.h>
#include <math.h>

//Normal Function
double NormalDist(double x)
{
    return (1.0 / sqrt(2 * 3.14159265359)) * exp(-(x * x) / 2.0);
}
//
//Inverse Normal Distribution
//
double InverseNormalDist(double x)
{
    double k, b1, b2, b3, b4, b5;

    if ( x = 0 )
    {
        k = 1.0 / (1 + 0.2316419 * x);
        b1 = 0.31938153 * k;
        b2 = -0.356563782 * k * k;
        b3 = 1.781477937 * pow(k,3);
        b4 = -1.821255978 * pow(k,4);
        b5 = 1.330274429 * pow(k,5);
        return 1 - NormalDist(x)*(b1+ b2 + b3 + b4 + b5);
    }
    else {
        return 1 - InverseNormalDist(-x);
    }
}

```

Listing 3
Java Implementation to Generate
a Standard Normal Deviate

```

import java.lang.*;

/*
 * imath - Fundamental Mathematical and statistical methods
 * Author: Jim Northey, jnorthey@up.net
 * Date: 2aug96
 *
 */
public class imath
{
    static final double a1 = 0.31938153 ;
    static final double a2 = -0.356563782 ;
    static final double a3 = 1.781477937 ;
    static final double a4 = -1.821255978 ;
    static final double a5 = 1.330274429 ;

    //
    // Inverse Normal Distribution
    //
    // Reference: Handbook of Mathematical Functions With Formulas,
    //           Graphs, and Mathematical Tables
    //           U.S. Dept. of Commerce,
    //           National Bureau of Standards
    //           Applied Mathematical Series No. 55
    //           Milton Abramowitz and Irene A. Stegun, Editors
    //           pp. 930-934
    public static double InverseNormalDist(double x)
    {
        if ( x >= 0.0)
        {
            double k = 1.0 / (1.0 + 0.2316419 * x) ;
            double series = a1*k + a2*k*k + a3*Math.pow(k,3)
                + a4*Math.pow(k,4) + a5*Math.pow(k,5) ;
            return 1 - imath.NormalDist(x) * series ;
        }
        else
        {
            return 1 - imath.InverseNormalDist(-x) ;
        }
    }
    //End InverseNormalDist() method
    //
    //Normal Function
    //
    public static double NormalDist(double x)
    {
        return (1 / Math.sqrt(2.0*Math.PI)) * Math.exp(-(x * x)/2.0);
    }
}
/*end imath class*/

```