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

SIMULATION OPTIMIZATION: A REVIEW, NEW DEVELOPMENTS, AND APPLICATIONS

Michael C. Fu
Smith School of Business
University of Maryland
College Park, MD 20742, U.S.A.

Fred W. Glover
Leeds School of Business
University of Colorado
Boulder, CO 80309, U.S.A.

Jay April
OptTek Systems, Inc.
1919 Seventh Street
Boulder, CO 80302, U.S.A.

ABSTRACT

We provide a descriptive review of the main approaches for carrying out simulation optimization, and sample some recent algorithmic and theoretical developments in simulation optimization research. Then we survey some of the software available for simulation languages and spreadsheets, and present several illustrative applications.

1 INTRODUCTION

The advances in computing power and memory over the last decade have opened up the possibility of optimizing simulation models. This development offers one of the most exciting opportunities in simulation, and there are plenty of interesting research problems in the field. The goals of this tutorial include the following:

- to provide a general overview of the primary approaches found in the research literature, and include pointers/references to the state of the art;
- to survey some of the commercial software;
- to illustrate the problems through examples and real-world applications.

The general optimization problem we consider it to find a setting of controllable parameters that minimizes a given objective function, i.e.,

$$\min_{\theta \in \Theta} J(\theta), \tag{1}$$

where $\theta \in \Theta$ represents the (vector of) input variables, $J(\theta)$ is the (scalar) objective function, and Θ is the constraint set, which may be either explicitly given or implicitly defined.

The assumption in the *simulation* optimization setting is that $J(\theta)$ is not available directly, but must be estimated through simulation, e.g., the simulation output provides $\hat{J}(\theta)$, a noisy estimate of $J(\theta)$. The most common form for J is an expectation, e.g.,

$$J(\theta) = E[L(\theta, \omega)],$$

where ω represents a sample path (simulation replication), L is the sample performance measure. Although this form is fairly general (includes probabilities by using indicator functions), it does exclude certain types of performance measures such as the median (and other quantiles) and the mode.

Real-World Example: Call Center Design
The state-of-the-art call centers (sometimes called contact centers) integrate traditional voice operations with both automated response systems (computer account access) and Internet (Web-based) services, often spread over multiple geographically separate sites. Most of these centers handle multiple sources of jobs, including voice, e-mail, fax, and interactive Web, each of which may require a different levels of operator (call agent) training, as well as different priorities, e.g., voice almost always preempting any of the other contact types (except possibly interactive Web). There are also different types of jobs according to the service required, e.g., checking the status of an order versus placing a new order versus requesting live service help. Furthermore, because of individual customer segmentation, there are different classes of customers in terms of priority levels. Designing and operating such a call center includes such problems as selecting the number of operators at each skill level, and determining what routing algorithm and type of queue discipline to use. A trade off must be made between achieving a desired level of customer service and the cost of providing service. An objective function might incorporate costs associated with operations such as agent wages and network utilization, as well as customer service performance metrics such as the probability of waiting more than a certain amount of time. This is just one example of how simulation optimization can be applied to business process management. A similar design problem is considered later in the applications section for a hospital emergency room.

Toy Example: Single-Server Queue
Consider a first-come, first-served (FCFS) single-server queue. A well-studied optimization problem uses the following objective function (cf. Fu 1994):

$$J(\theta) = E[W(\theta)] + c/\theta, \tag{2}$$

where W is the mean time spent in the system, θ is the mean service time of the server (so $1/\theta$ corresponds to the server speed), and c is the cost factor for the server speed (i.e., a higher-skilled worker costs more). Since W is increasing in θ, the objective function quantifies the trade-off between customer service level and cost of providing service. This could be viewed as the simplest possible case of the call center design problem, where there is a single operator whose skill level must be selected. For the special $M/M/1$ queue in steady state, this problem is analytically tractable, and serves as a test case for optimization procedures.

Another Academic Example: Inventory Control
The objective is to minimize a total cost function consisting of ordering, holding, and backlogging or lost sales components. The ordering policy involves two parameters, s and S, corresponding to the re-order level and order-up-to level, respectively. When the inventory level falls below s, an order is placed for an amount that would bring the current level back up to S.

The input variables can be divided into two main types: qualitative and quantitative. In the call center example, both types are present, whereas in the two simpler examples, the input variables are quantitative. Quantitative variables are then either discrete or continuous. Many of the call center variables are inherently discrete, e.g., the number of operators, whereas in the single-server queue and inventory control examples, the input variables θ and (s, S) could be specified as either, depending on the particular problem setting or solution technique being applied. As in deterministic optimization, the approaches to solve these different types of problems can differ greatly.

To get an idea of the particular challenges facing simulation optimization, consider the problem of finding the value of θ that minimizes the objective function in (1) versus the problem of finding the minimum value of the objective function itself. In terms of our notation, this is the difference between finding a setting θ^* that achieves the minimum in (1) versus finding the value of $J(\theta^*)$. A key difference between deterministic optimization and stochastic optimization is that these two problems are not necessarily the same! In deterministic optimization, where J can be easily evaluated, the two problems are essentially identical; however, in the stochastic setting, one or the other may be easier. Furthermore, it may also be the case that only one or the other is the real goal of the modeling exercise. In some cases, selecting the best design is the goal, and the objective function is merely a means towards achieving this end, providing a way to measuring the relative performance of each design, whereas the absolute value of the metric may have little meaning. In other cases, the objective function may have intrinsic meaning, e.g., costs or profits. And in yet other (less frequent) cases, estimating the optimal value itself is the primary goal. An example of this is the pricing of financial derivatives with early exercise opportunities (especially when done primarily for the sake of satisfying regulatory requirements of marking a portfolio to market).

To put it another way, simulation optimization has two major components that vie for computational resources: *search* and *evaluation*. How to balance between the two, i.e., how to best allocate simulation replications, is a large challenge in making simulation optimization practical. To be concrete, one is choosing between simulation replications to get better estimates versus more iterations of the optimization algorithm to more thoroughly explore the search space.

2 APPROACHES

We now briefly describe the main approaches in the simulation literature.

2.1 Ranking & Selection

In the setting where it is assumed that there is a *fixed* set of alternatives — so no search for new candidates is involved — the problem comes down to one of statistical inference, and ranking & selection procedures can be applied. Let the probability of correct selection be denoted by PCS, which we will not define formally here, but intuitively "correct selection" would mean either selecting either the best solution or a solution within some presecified tolerance of the best.

There are two main forms the resulting problem formulations can take:

(i) minimize the number of simulation replications subject to the PCS exceeding a given level.

(ii) maximize the PCS subject to a given simulation budget constraint;

In case (i), one ensures a level of correct selection, but has little control over how much computation this might entail. This is the traditional statistics ranking & selection approach, and in the simulation setting, Kim and Nelson (2005) overview the state of the art, where multiple comparison procedures can be used to provide

valid confidence intervals, as well; see also Goldsman and Nelson (1998). The books by Bechhofer, Santner, and Goldsman (1995) and Hochberg and Tamhane (1987) contain more general discussion of multiple comparison procedures outside of the simulation setting.

In case (ii), one tries to do the best within a specified computational limit, but a priori one may not have any idea how good the resulting solution will be. This formulation was coined the "optimal computing budget allocation" (OCBA) problem, as first proposed by Chen (1995). Subsequent and related work includes Chick and Inoue (2001ab), Chen et al. (2005), Fu et al. (2005).

Ranking & selection procedures can also used in the following ways that are relevant to simulation optimization: for *screening* a large set of alternatives, i.e., quickly (meaning based on a relatively small amount of replications) eliminating poor performers in order to get a more manageable set of alternatives; for *comparing* among candidate solutions in an iterative algorithm, e.g., deciding whether or not an improvement has been made; for providing some statistical guidance in assessing the quality of the declared best solution versus all the solutions visit. The latter is what Boesel, Nelson and Kim (2003) call "cleaning up" after simulation optimization.

The framework of ordinal optimization (Ho, Sreenivas, and Vakili 1992; Ho et al. 2000) might also be classified under ranking & selection. This approach is based on the observation that in most cases it is much easier to find ordering among candidate solutions than to carry out the estimation procedure for each solution individually, and then try to rank order the solutions. This can be especially true in the simulation setting, where the user has more control, so for instance can use common random numbers to induce positive correlation between estimates of solution performance to dramatically reduce the number of simulation replications required to make a distinction. Intuitively, it is the difference between estimating $J_1 - J_2 = E[L_1 - L_2]$ versus "estimating" $P(J_1 > J_2)$, say using the simple mean based on n simulation replications. Estimating the former using the sample mean is governed by the Monte Carlo convergence rate of $n^{-1/2}$, whereas deciding on the latter based on the sample mean has an *exponential* convergence rate. Using the theory of large deviations from probability, Dai and Chen (1997) explore this exponential rate of convergence in the discrete-event simulation context.

2.2 Response Surface Methodology

Response surface methodology (RSM) has its roots in statistical design of experiments, and its goal is to obtain an approximate functional relationship between the input variables and the output objective function. In design of experiments terminology, these are referred to as the factors and the response, respectively. RSM carried out on the entire domain of interest results in what is called a metamodel (see Barton 2005). The two most common ways of obtaining this representation are regression and neural networks. Once a metamodel is in hand, optimization can be carried out using deterministic optimization procedures. However, when optimization is the focus, a form of sequential RSM is usually employed (Kleijnen 1998), in which a local response surface representation is obtained that guides the sequential search. For example, linear regression could be used to obtain an estimate of the direction of steepest descent. This approach is model free, well-established, and fairly straightforward to apply, but it is not implemented in any of the commercial packages. Until recently, SIMUL8 (`<http://www.SIMUL8.com/optimiz1.htm>`) had employed an optimization algorithm based on a form of sequential RSM using neural networks, but now uses OptQuest instead. The primary drawback seems to be the excessive use of simulation points in one area before exploring other parts of the search space. This can be especially exacerbated when the number of input variables is large. Recently, kriging has been proposed as a possibly more efficient way of carrying out this step (see van Beers and Kleijnen 2003). For more information on RSM procedures for simulation optimization, see Barton (2005) and Kleijnen (1998).

2.3 Gradient-Based Procedures

The gradient-based approach tries to mimic its counterpart in deterministic optimization. In the stochastic setting, the resulting procedures usually take the form of stochastic approximation (SA) algorithms; the book by Kushner and Yin (1997) contains a general discussion of SA outside of simulation. Specifically, given a current best setting of the input variables, a movement is made in the gradient direction, similar to sequential RSM. However, unlike sequential RSM procedures, SA algorithms can be shown to be provably convergent (asymptotically, usually to a local optimum) under appropriate conditions on the gradient estimator and step sizes, and they generally require far less simulations per iteration. Practically speaking, the key to making this approach successful is the quality of the gradient estimator. Fu (2005) surveys the main approaches available for coming up with gradient estimators that can be implemented in simulation. These include "brute-force" finite differences, simultaneous perturbations, perturbation analysis, the likelihood ratio/score func-

tion method, and weak derivatives. For technical details on simultaneous perturbation stochastic approximation (SPSA), see Spall (1992), Fu and Hill (1997), Spall (2003), and <http://www.jhuapl.edu/spsa>. SPSA has two advantages over the other methods: it requires only two simulations per gradient estimate, regardless of the number of input variables, and it can treat the simulation model as a black box, i.e., no knowledge of the workings of the system is required (model free). A one-simulation version of SPSA is also available, but in practice it appears far noisier than the two-simulation version, even accounting for the effort being half as much.

An in-depth study of various gradient-based algorithms for the single-server $M/M/1$ queue example can be found in L'Ecuyer, Giroux, and Glynn (1994); see Fu (1994b) and Fu and Healy (1997) for the (s, S) inventory system. Kapuscinski and Tayur (1999) describe the use of perturbation analysis in a simulation optimization framework for inventory management of a capacitated production-inventory system. This approach was implemented on the worldwide supply chain of Caterpillar (its success reported in a *Fortune* magazine article by Philip Siekman, October 30, 2000: "New Victories in the Supply Chain Revolution"). The primary drawback of the gradient-based approach is that it currently is really only practically applicable to the *continuous* variable case, notwithstanding recent attempts to apply it to discrete-valued variables. Furthermore, estimating direct gradients may require knowledge of the underlying model, and the applicability of such estimators is often highly problem dependent.

2.4 Random Search

In contrast to gradient-based procedures, random search algorithms are targeted primarily at *discrete* input variable problems. They were first developed for deterministic optimization, but have been extended to the stochastic setting. Like gradient-based procedures, they proceed by moving iteratively from a current best setting of the input variables (candidate solution). Instead of using a gradient, however, the next move is probabilistically drawn from the "neighborhood" of the current best. For example, defining the neighborhood of a solution as all of the other solutions and drawing from a uniform distribution (assuming the number of feasible solutions is finite) would give a "pure" random search algorithm. Practically speaking, the success of a particular random search algorithm depends heavily on the defined neighborhood structure. Furthermore, in the stochastic setting, the estimation problem must also be incorporated into the algorithm. Thus, the two features that define an algorithm in the simulation optimization setting are
(a) how the next candidate solution(s) is(are) chosen; and
(b) how to determine which is the current best solution (which is not necessarily the current iterate).
The second feature (b) does not arise in the deterministic setting, because the current best (among visited solutions) is known with certainty, since the objective function values have no estimation noise. However, in the stochastic case, due to the noise in the objective function estimates, there are many possible choices, e.g., the current solution, the solution that has been visited the most often, or the solution that has the best sample mean thus far.

Like stochastic approximation algorithms, random search algorithms can generally be shown to be provably convergent (often to a global optimum). For more details on random search methods in simulation, see Andradóttir (2005); for a more general survey on discrete input simulation optimization problems, see Swisher et al. (2001). A recently proposed version of random search that is very promising is Convergent Optimization via Most-Promising-Area Stochastic Search (COMPASS), introduced by Hong and Nelson (2005), which utilizes a unique neighborhood structure and results in a provably convergent algorithm to a locally optimal solution.

2.5 Sample Path Optimization

Sample path optimization (also known as stochastic counterpart, sample average approximation; see Rubinstein and Shapiro 1993) takes many simulations first, and then tries to optimize the resulting estimates. Specifically, if $\tilde{J}_i$ denotes the estimate of J from the ith simulation replication, the sample mean over n replications is given by

$$\hat{J}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \tilde{J}_i(\theta).$$

If each of the $\tilde{J}_i$ are i.i.d. unbiased estimates of J, then by the strong law of large numbers,

$$\hat{J}_n(\theta) \longrightarrow J(\theta) \text{ with probability 1.}$$

The approach then is to optimize, for a sufficiently large n, the deterministic function $\hat{J}_n$, which approximates J. Its key feature, as Robinson (1996) advocates, is that "we can bring to bear the large and powerful array of deterministic [primarily continuous variable] optimization methods that have been developed in the last half-century. In particular, we can deal with problems in which the parameters θ might be subject to complicated constraints, and therefore in which gradient-step methods like stochastic approximation may have difficulty."

In the simulation context, the method of common random numbers is used to provide the same sample paths for $\hat{J}_n(\theta)$ over different values of θ. Furthermore, the availability of derivatives greatly enhances the effectiveness of the approach, as many nonlinear optimization packages require these.

2.6 Metaheuristics

Metaheuristics are methods that guide other procedures (heuristic or truncated exact methods) to enable them to overcome the trap of local optimality for complex optimization problems. Four metaheuristics have primarily been applied with some success to simulation optimization: simulated annealing, genetic algorithms, tabu search and scatter search (occasionally supplemented by a procedure such as neural networks in a forecasting or curve fitting role). Of these, tabu search and scatter search have proved to be by far the most effective, and are at the core of the simulation optimization software that is now most widely used. We briefly sketch the nature of these two approaches below. The general metaheuristics framework of Ólaffson (2005), which looks very much like the deterministic version of random search, also contains discussion of the nested partitions method introduced by Shi and Ólaffson (2000ab) (see also Pinter 1996).

Tabu Search (TS) is distinguished by introducing adaptive memory into metaheuristic search, together with associated strategies for exploiting such memory, equipping it to penetrate complexities that often confound other approaches. Applications of TS span the realms of resource planning, telecommunications, VLSI design, financial analysis, space planning, energy, distribution, molecular engineering, logistics, pattern classification, flexible manufacturing, waste management, mineral exploration, biomedical analysis, environmental conservation and scores of others. A partial indication of the rapid recent growth of TS applications is disclosed by the fact that a Google search on tabu search returns more than 90,000 pages, a figure that has been growing exponentially over the past several years.

Adaptive memory in tabu search involves an attribute-based focus, and depends intimately on the elements of recency, frequency, quality and influence. This catalog disguises a surprising range of alternatives, which arise by differentiating attribute classes over varying regions and spans of time. The TS notion of influence, for example, encompasses changes in structure, feasibility and regionality, and the logical constructions used to interrelate these elements span multiple dimensions, involving distinctions between sequential logic and event driven logic, giving rise to different kinds of memory structures.

The most comprehensive reference for tabu search and its applications is the book by Glover and Laguna, (1997). A new book that gives more recent applications and pseudo-code for creating various implementations is scheduled to appear in 2006.

Scatter Search (SS) has its foundations in proposals from the 1970s that also led to the emergence of tabu search, and the two methods are highly complementary and often used together. SS is an evolutionary (population-based) algorithm that constructs solutions by combining others.

Scatter search is designed to operate on a set of points, called reference points, that constitute good solutions obtained from previous solution efforts. Notably, the basis for defining good includes special criteria such as diversity that purposefully go beyond the objective function value. The approach systematically generates combinations of the reference points to create new points, each of which is mapped into an associated feasible point. The combinations are generalized forms of linear combinations, accompanied by processes to adaptively enforce feasibility conditions, including those of discreteness.

The SS process is organized to (1) capture information not contained separately in the original points, (2) take advantage of auxiliary heuristic solution methods (to evaluate the combinations produced and to actively generate new points), and (3) make dedicated use of strategy instead of randomization to carry out component steps. SS basically consists of five methods: a diversification generation method, an improvement method (often consisting of tabu search), a reference set update method, a subset generation method, and a solution combination method. Applications of SS, like those of TS, have grown dramatically in recent years, and its use in simulation optimization has become the cornerstone of significant advances in the field. The most complete reference on SS is the book by Laguna and Marti (2002).

3 MODEL-BASED METHODS

An approach that looks promising and which has just begun to be explored in the simulation optimization context are *model-based* methods. These are contrasted with what are called *instance-based* approaches, which generate new solutions based only on the current solution (or population of solutions) (cf. Dorigo and Stützle 2004, pp.139-140). The metaheuristics described earlier generally fall into this latter category, with the exception of tabu search, because it uses memory. Model-based methods, on the other hand, are not dependent explicitly on any current set of solutions, but use a *prob-*

Table 1: Optimization for Simulation: Commercial Software Packages

Optimization Package (simulation platform)	Vendor (URL)	Primary Search Strategies
AutoStat (AutoMod)	AutoSimulations, Inc. (www.autosim.com)	evolutionary, genetic algorithms
Evolutionary Optimizer (Extend)	AutoSimulations, Inc. (www.imaginethatinc.com)	evolutionary, genetic algorithms
OptQuest (Arena, Crystal Ball, ProModel, SIMUL8, et al.)	OptTek Systems, Inc. (www.opttek.com)	scatter search, tabu search, neural networks
RISKOptimizer (@RISK)	Palisade Corp. (www.palisade.com)	genetic algorithms
Optimizer (WITNESS)	Lanner Group, Inc. (www.lanner.com/corporate)	simulated annealing, tabu search

ability distribution on the space of solutions to provide an estimate of where the best solutions are located.

The following are some examples:

- **Swarm Intelligence**. This approach is perhaps best known under the name of "Ant Colony Optimization," because it uses ant behavior (group cooperation and use of pheromone updates and evaporation) as a paradigm for its probabilistic workings. Because there is memory involved in the mechanisms, like tabu search, it is not instance-based; see Dorigo and Stützle (2004) for more details.
- **Estimation of Distribution Algorithms (EDAs)**. The goal of this approach is to progressively improve a probability distribution on the solution space based on *samples* generated from the current distribution. The crudest form of this would utlize all samples generated to a certain point, hence the use of memory, but in practical implementation, parameterization of the distribution is generally employed, and the parameters are updated based on the samples; see Larrañaga and Lozano (2002) for more details.
- **Cross-Entropy (CE) Method**. This approach grew out of a procedure to find an optimal importance sampling measure by projecting a parameterized probability distribution, using cross entropy to measure the distance from the optimum measure. Like EDAs, samples are taken that are used to update the parameter values for the distribution. Taking the optimal measure as a point mass at the solution optimum of an optimization problem, the procedure can be applied in that context; see De Boer et al. (2005), Rubinstein and Kroese (2004), and `<http://www.cemethod.org>` for more details.
- **Model Reference Adaptive Search**. As in EDAs, this approach updates a parameterized probability distribution, and like the CE method, it also uses the cross-entropy measure to project a parameterized distribution. However, the particular projection used relies on a *stochastic sequence* of reference distributions rather than a *single fixed* reference distribution (the final optimal measure) as in the CE method, and this results in very different performance in practice. Furthermore, stronger theoretical convergence results can be established; see Hu, Fu, and Marcus (2005abc) for details.

4 SOFTWARE

Table 1 surveys a few simulation optimization software packages (either plug-ins or integrated) currently available, and summarizes their search strategies. Comparing with Table 1 in Fu (2002), one observes that ProModel and SIMUL8 have both migrated to OptQuest from their previous simulation optimization packages (SimRunner and OPTIMIZ, respectively).

5 APPLICATIONS

Applications of optimization technology are quite diverse; they cover a broad surface of business activities. To illustrate, the user of simulation and other business or industry evaluation models may want to know:

- What is the most effective factory layout?

- What is the safest equipment replacement policy?
- What is the most cost effective inventory policy?
- What is the best workforce allocation?
- What is the most productive operating schedule?
- What is the best investment portfolio?

The answers to such questions require a painstaking examination of multiple scenarios, where each scenario in turn requires the implementation of an appropriate simulation or evaluation model to determine the consequences for costs, profits and risks. The critical "missing component" is to disclose which decision scenarios are the ones that should be investigated – and still more completely, to identify good scenarios automatically by a search process designed to find the best set of decisions. This is the core problem for simulation optimization in a practical setting. The following descriptions provide a sampling of uses of the technology that enable solutions to be identified efficiently.

5.1 Project Portfolio Management

For project portfolio management, OptFolio is a software tool being implemented in several markets including Petroleum and Energy, IT Governance, and Pharmaceuticals. The following example demonstrates the versatility of OptFolio as a simulation optimization tool.

Among many other types of initiatives, the Pharmaceutical Industry uses project portfolio optimization to manage investments in new drug development. A pharmaceutical company that is developing a new breakthrough drug is faced with the possibility that the drug may not do what it was intended to do, or have serious side effects that make it commercially infeasible. Thus, these projects have a considerable degree of uncertainty related to the probability of success. Relatively recently, an options-pricing approach, called "real options" has been proposed to model such uncertainties. Initial feedback has indicated that an obstacle to its market penetration is that it is difficult to understand and use; furthermore, there are no research results that illustrate performance that rivals the algorithmic approach underlying OptFolio.

The following example is based on data provided by Decision Strategies, Inc., a consulting firm with numerous clients in the Pharmaceutical Industry. The data consists of twenty potential projects in drug development having rather long horizons – 5 to 20 years – and the pro-forma information is given as triangular distributions for both per-period net contribution and investment. The models use a probability of success index – from 0% to 100% – applied in such a way that, if the project fails during a simulation trial, then the investments are realized, but the net contribution of the project is not. In this way, the system can be used to model premature project terminations providing a simple, understandable alternative to real options. In this example, we examined five cases, the first two representing approaches commonly used in practice but, as we have found, turn out to be significantly inferior to the ones using the OptFolio software.

Case 1: Simple Ranking of Projects
Projects were ranked in this approach according to a specific objective criterion, a process often adopted in currently available Project Portfolio Management tools in order to select projects under a budgetary constraint. In this case the following objective measure was selected:

$$R = \frac{PV(Revenues)}{PV(Expenses)}.$$

Employing the customary design, the 20 projects were ranked in descending order according to this measure, and projects were added to the final portfolio as long as the budget constraint was not violated. This procedure resulted in a portfolio with the following statistics:

$$\mu_{NPV} = 7342, \quad \sigma_{NPV} = 2472, \quad q_{.05} = 3216,$$

where $q_{.05}$ denotes the 5th percentile (quantile), i.e., $P(NPV \geq q_p) = p$.

In this case, 15 projects were selected in the final portfolio. What follows is a discussion of how using OptFolio can help improve these results.

Case 2: Traditional Markowitz Approach
The decision was to determine participation levels [0,1] in each project with the objective of maximizing the expected NPV of the portfolio while keeping the standard deviation of the NPV below a specified threshold of 1000. An investment budget was also imposed on the portfolio, where B_i denotes the budget in period i.

Maximize μ_{NPV} subject to

$$\sigma_{NPV} \leq 1000, B_1 \leq 125, B_2 \leq 140, B_3 \leq 160.$$

This formulation resulted in a portfolio with the following statistics:

$$\mu_{NPV} = 4140, \quad \sigma_{NPV} = 1000, \quad q_{.05} = 2432.$$

We performed this traditional mean-variance case to provide a basis for comparison for the subsequent cases. An empirical histogram for the optimal portfolio is shown in Figure 1.

Case 3: Risk Controlled by 5th Percentile
The decision was to determine participation levels [0,1] in each project with the objective of maximizing the expected NPV of the portfolio while keeping the 5th percentile of NPV above the value determined in Case 2

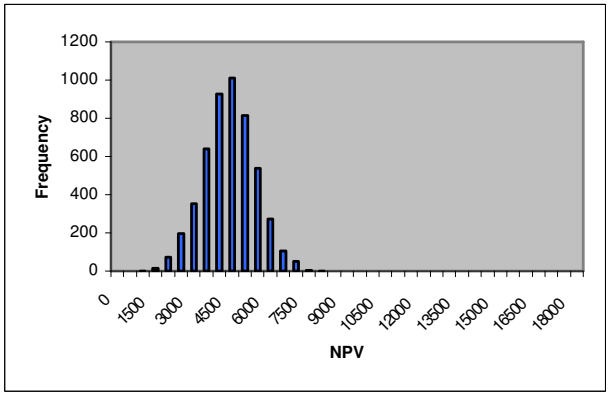

Figure 1: Mean-Variance Portfolio

(2432), keeping the same investment budget constraints on the portfolio.

Maximize μ_{NPV} subject to

$$q_{.05} \geq 2432, \quad B_1 \leq 125, \quad B_2 \leq 140, \quad B_3 \leq 160.$$

This case has replaced standard deviation with the 5th percentile for risk containment, which is an intuitive way to control catastrophic risk (Value at Risk or VaR in traditional finance terminology). The resulting portfolio has the following attributes:

$$\mu_{NPV} = 7520, \quad \sigma_{NPV} = 2550, \quad q_{.05} = 3294.$$

By using the 5th percentile as a measure of risk, we were able to almost double the expected return compared to the solution found in Case 2, and improved on the simple ranking solution. Additionally, as previously discussed, the 5th percentile provides a more intuitive measure of risk, i.e., there is a 95% chance that the portfolio will achieve a NPV of 3294 or higher. The NPV distribution is shown in Figure 2. It is interesting to note that this solution has more variability but is focused on the upside of the distribution. By focusing on the 5th percentile rather than standard deviation, a superior solution was created.

Case 4: Maximizing Probability of Success

The decision was to determine participation levels [0,1] in each project with the objective of maximizing the probability of meeting or exceeding the mean NPV found in Case 2, keeping the same investment budget constraints on the portfolio.

Maximize Prob(NPV $\geq$ 4140) subject to

$$B_1 \leq 125, \quad B_2 \leq 140, \quad B_3 \leq 160.$$

This case focuses on maximizing the chance of obtaining a goal and essentially combines performance and risk

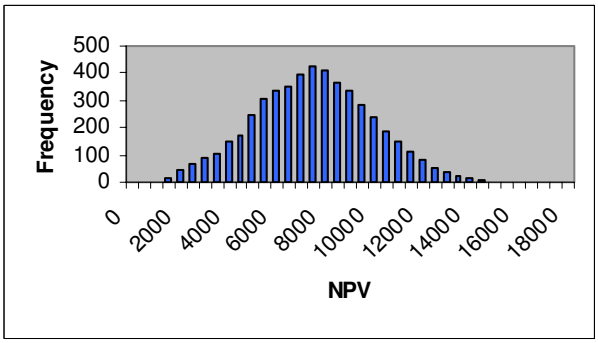

Figure 2: 5th Percentile Portfolio

containment into one metric. The resulting portfolio has the following attributes:

$$\mu_{NPV} = 7461, \quad \sigma_{NPV} = 2430, \quad q_{.05} = 3366.$$

This portfolio has a 91% chance of achieving/exceeding the NPV goal of 4140, representing a significant improvement over the Case 2 portfolio, where the probability was only 50%. The NPV distribution is shown in Figure 3.

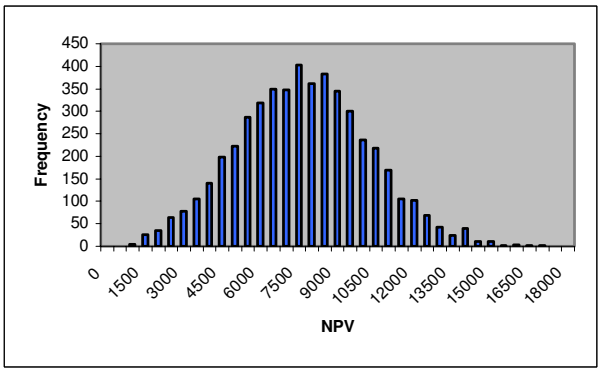

Figure 3: 5th Percentile Portfolio

Case 5: All-or-Nothing

In many real-world settings, these types of projects have all-or-nothing participation levels, whereas in the Case 4 solution, most of the optimal participation levels found were fractional. Under the same investment budget constraints on the portfolio, Case 5 modified the Case 4 constraints to allow only 0 or 1 participation levels, i.e., a project must utilize 100% participation or be excluded from the portfolio.

Maximize Prob(NPV $\geq$ 4140) subject to

$$B_1 \leq 125, \ B_2 \leq 140, \ B_3 \leq 160, \ \text{Participations} \in \{0, 1\}.$$

The resulting portfolio has the following attributes:

$$\mu_{NPV} = 7472, \quad \sigma_{NPV} = 2503, \quad q_{.05} = 3323.$$

In spite of the participation restriction, this portfolio also has a 91% chance of exceeding an NPV of 4140, and has a high expected return. In this case, as in Case 1, 15 out of the 20 projects were selected in the final portfolio, but the expected returns are higher. These cases illustrate the benefits of using alternative measures for risk. Not only are percentiles and probabilities more intuitive for the decision-maker, but they also produce solutions with better financial metrics. The OptFolio system can also be used to optimize performance metrics such as Internal Rate of Return (IRR) and Payback Period.

The illustrated analyses can be applied very effectively for complex, as well as simple, sets of projects, where different measures of risk and return can produce improvements over the traditional Markowitz (mean-variance) approach, as well as over simple project ranking approaches. The flexibility to choose various measures and statistics, both as objective performance measures as well as constraints, is a major advantage to using a simulation optimization approach as embedded in OptFolio. The user is given an ability to select better ways of modeling and controlling risk, while aligning the outcomes to specific corporate goals.

OptFolio also provides ways to define special relationships that often arise between and among projects. Correlations can be defined between the revenues and/or expenses of two projects. In addition, the user can define projects that are mutually exclusive, or dependent. For example, in some cases, selecting Project A implies selecting Project B; such a definition can easily be done in OptFolio.

Portfolio analysis tools are designed to aid senior management in the development and analysis of project portfolio strategies, by giving them the capability to assess the impact on the corporation of various investment decisions. To date, commercial portfolio optimization packages are relatively inflexible and are often not able to answer the key questions asked by senior management. As a result of the simulation optimization capabilities embodied in OptFolio, new techniques are made available that increase the flexibility of portfolio optimization tools and deepen the types of portfolio analysis that can be carried out.

5.2 Business Process Management

When changes are proposed to business processes in order to improve performance, important advantages can result by evaluating the projected improvements using simulation, and then determining an optimal set of changes using simulation optimization. In this case it becomes possible to examine and quantify the sensitivity of making the changes on the ultimate objectives to reduce the risk of actual implementation. Changes may entail adding, deleting, and modifying processes, process times, resources required, schedules, work rates within processes, skill levels, and budgets. Performance objectives may include throughput, costs, inventories, cycle times, resource and capital utilization, start-up times, cash flow, and waste. In the context of business process management and improvement, simulation can be thought of as a way to understand and communicate the uncertainty related to making the changes while optimization provides the way to manage that uncertainty.

The following example is based on a model provided by CACI, and simulated on SIMPROCESS. Consider the operation of an emergency room (ER) in a hospital. Figure 4 shows a high-level view of the overall process, which begins when a patient arrives through the doors of the ER, and ends when a patient is either released from the ER or admitted into the hospital for further treatment. Upon arrival, patients sign in, receive an assessment of their condition, and are transferred to an ER. Depending on their assessment, patients then go through various alternatives involving a registration process and a treatment process, before being released or admitted into the hospital.

Patients arrive either on their own or in an ambulance, according to some arrival process. Arriving patients are classified into different levels, according to their condition, with Level 1 patients being more critical than Level 2 and Level 3 patients.

Level 1 patients are taken to an ER immediately upon arrival. Once in the room, they undergo their treatment. Finally, they complete the registration process before being either released or admitted into the hospital for further treatment.

Level 2 and Level 3 patients must first sign in with an Administrative Clerk. After their condition is then assessed by a Triage Nurse, and then they are taken to an ER. Once in the room, Level 2 and 3 patients, must first complete their registration, then go on to receive their treatment, and, finally, they are either released or admitted into the hospital for further treatment.

After undergoing the various activities involved in registration and treatment, 90% of all patients are released from the ER, while the remaining 10% are admitted into the hospital for further treatment. The final release/hospital admission process consists of the following activities: 1. In case of release, either a nurse or a PCT fills out the release papers (whoever is available first). 2. In case of admission into the hospital, an Administrative Clerk fills out the patients admission papers. The patient must then wait for a hospital bed to become available. The time until a bed is available is handled by an empirical probability distribution. Finally, the patient is transferred to the hospital bed.

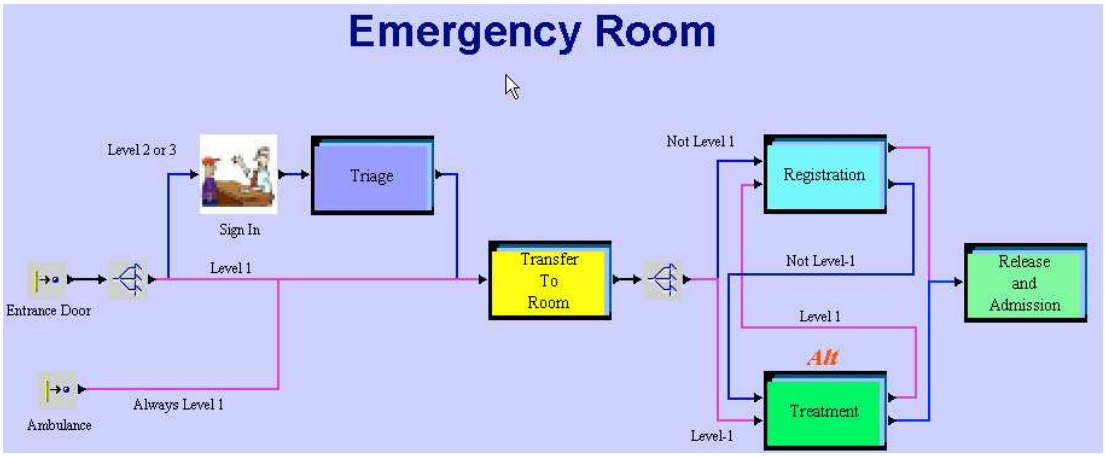

Figure 4: High-level Process View

The following illustrates a simple instance of this process that is actually taken from a real world application. In this instance, due to cost and layout considerations, hospital administrators have determined that the staffing level must not exceed 7 nurses, 3 physicians, 4 PCTs and 4 Administrative Clerks. Furthermore, the ER has 20 rooms available; however, using fewer rooms would be beneficial, since the additional space could be used more profitably by other departments in the hospital. The hospital wants to find the configuration of the above resources that minimizes the total asset cost. The asset cost includes the staffs hourly wages and the fixed cost of each ER used. We must also make sure that, on average, Level 1 patients do not spend more than 2.4 hours in the ER. This can be formulated as an optimization problem, as follows:

Minimize Expected Total Asset Cost
subject to the following constraints:
Average Level 1 Cycle Time $\leq$ 2.4 hours,
Nurses $\leq$ 7,
Physicians $\leq$ 3,
PCTs $\leq$ 4,
Admin. Clerks $\leq$ 4,
ERs $\leq$ 20.

This is a relatively unimposing problem in terms of size: five variables and six constraints. However, if we were to rely solely on simulation to solve this problem, even after the hospital administrators have narrowed down our choices to the above limits, we would have to perform 7x3x4x4x20=6,720 experiments. If we want a sample size of, say, at least 30 runs per trial solution in order to obtain the desired level of precision, then each experiment would take about 2 minutes, based on a Dell Dimension 8100 with a 1.7GHz Intel Pentium 4 processor. This means that a complete enumeration of all possible solutions would take approximately 13,400 minutes, or about 70 working days. This is obviously too long a duration for finding a solution.

In order to solve this problem in a reasonable amount of time, we called upon the OptQuest optimization technology integrated with SIMPROCESS. As a base case we used the upper resource limits provided by hospital administrators, to get a reasonably good initial solution. This configuration yielded an Expected Total Asset Cost of $36,840, and a Level 1 patient cycle time of 1.91 hours.

Once we set up the problem in OptQuest, we ran it for 100 iterations (experiments), and 5 runs per iteration (each run simulates 5 days of the ER operation). Given these parameters, the best solution, found at iteration 21 was: 4 nurses, 2 physicians, 3 PCTs, 3 administrative clerks, and 12 ERs.

The Expected Total Asset Cost for this configuration came out to $25,250 (a 31% improvement over the base case), and the average Level 1 patient cycle time was 2.17 hours. The time to run all 100 iterations was approximately 28 minutes.

After obtaining this solution, we redesigned some features of the current model to improve the cycle time of Level 1 patients even further. In the redesigned model, we assume that Level 1 patients can go through the treatment process and the registration process in parallel. That is, we assume that while the patient is undergoing treatment, the registration process is being done by a surrogate or whoever is accompanying the patient. If the patients condition is very critical, than someone else can provide the registration data; however, if the patients condition allows it, then the patient can provide the registration data during treatment.

Figure 5 shows the model with this change. By optimizing the model that incorporates this change, we now

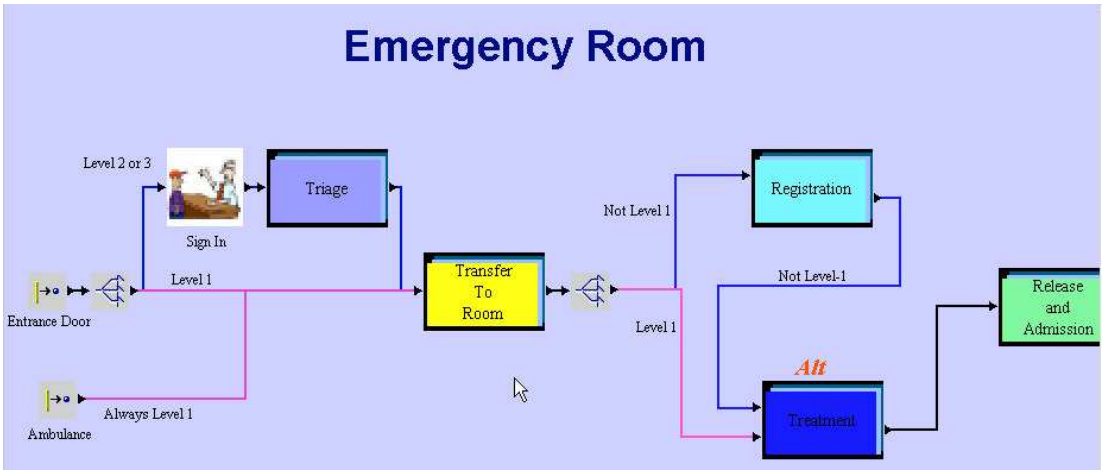

Figure 5: Proposed Process

obtain an average Level 1 patient cycle time of 1.98 (a 12% improvement).

The new solution had 4 nurses, 2 physicians, 2 PCTs, 2 administrative clerks, and 9 ERs, yielding an Expected Total Asset Cost of $24,574, and an average Level 1 patient cycle time of 1.94 hours. By using simulation optimization, we were able to find a very high quality solution in less than 30 minutes.

6 CONCLUSIONS

In addition to chapters in the *Handbook on Operations Research and Management Science: Simulation* volume cited already, more technical details on simulation optimization techniques can be found in the chapter by Andradóttir (1998) and the review paper by Fu (1994), whereas the feature article by Fu (2002) explores deeper research versus practice issues. Previous volumes of these Winter Simulation Conference proceedings also provide good current sources (e.g., April et al. 2003, 2004). Other books that treat simulation optimization in some technical depth include Rubinstein and Shapiro (1993), Fu and Hu (1997), Pflug (1997), Spall (2003).

Note that the "model" in model-based approaches is a probability distribution on the solution space, as opposed to modeling the response surface itself; the input variables are the same in both cases. Is there some way of combining the two approaches? One seeming advantage of the probabilistic approach is that it applies equally well to both the continuous and discrete case. The key in both cases to practical implementation is parameterization! For example, neural networks and regression are used in the former case, whereas the natural exponential family works well in the latter case.

Relatively little research has been done on multi-response simulation optimization, or for that matter, with random constraints, i.e., where the constraints themselves must be estimated. Most of the commercial software packages, however, do allow multiple responses (combining by using a weighting) and explicit inequality constraints on output performance measures, but in the latter case, there is usually not provided any statistical estimate as to how likely the constraint is actually being violated (just a confidence interval on the performance measure itself).

To summarize, here are some key issues in simulation optimization algorithms:

- neighborhood definition;
- mechanism for exploration/sampling (search), especially how previously generated (sampled) solutions are incorporated;
- determining which candidate solution(s) to declare the best (or "good"); statistical statements?
- the computational burden of each function estimate (obtained through simulation replications) relative to search (the optimization algorithm).

The first two issues are not specific to the stochastic setting of simulation optimization, but their effectiveness depends intimately on the last issue. For example, defining the neighborhood as a very large region may lead to theoretical global convergence, but it may not lead to very efficient search, especially if simulation is expensive. The model-based algorithms allow a large neighborhood, but can also allow search in a localized manner by the way the model (probability distribution) is constructed and updated.

ACKNOWLEDGMENTS

Michael C. Fu was supported in part by the National Science Foundation under Grant DMI-0323220, and by the Air Force of Scientific Research under Grant FA95500410210.

REFERENCES

Andradóttir, S. 1998. Simulation optimization. Chapter 9 in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, ed. J. Banks. New York: John Wiley & Sons.

Andradóttir, S. 2005. An overview of simulation optimization via random search. Chapter 21 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier.

April, J., F. Glover, J.P. Kelly, M. Laguna. 2003. Practical introduction to simulation optimization. In *Proceedings of the 2003 Winter Simulation Conference*, eds. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice. 71-78. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

April, J., F. Glover, J.P. Kelly, M. Laguna. 2004. New advances and applications for marrying simulation and optimization. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 255-260. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Barton, R. 2005. Response surface methodology. Chapter 19 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier.

Bechhofer, R.E., T.J. Santner, and D.M. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*, New York: John Wiley & Sons.

Boesel, J., B.L. Nelson, and S.-H. Kim. 2003. Using ranking and selection to 'clean up' after simulation optimization. *Operations Research* 51: 814-825.

Chen, C. H. 1995. An Effective Approach to Smartly Allocate Computing Budget for Discrete Event Simulation. *Proceedings of the 34th IEEE Conference on Decision and Control*, 2598-2605.

Chen, C.H., J. Lin, E. Yücesan, S.E. Chick. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems: Theory and Applications* **10** 251-270.

Chen, C.H., E. Yücesan, L. Dai, H.C. Chen. 2005. Efficient computation of optimal budget allocation for discrete event simulation experiment. *IIE Transactions* forthcoming.

Chen, H.C., C.H. Chen, and E. Yucesan. 2000. Computing efforts allocation for ordinal optimization and discrete event simulation. *IEEE Transactions on Automatic Control* 45: 960-964.

Chick, S.E., K. Inoue. 2001a. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research* **49** 732-743.

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

Dai, L., C. Chen. 1997. Rate of convergence for ordinal comparison of dependent simulations in discrete event dynamic systems. *Journal of Optimization Theory and Applications* **94** 29-54.

De Boer, P.-T., D.P. Kroese, S. Mannor, and R.Y. Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals of Operation Research* 134 (1): 19-67.

Dorigo, M. and T. Stützle. 2004. *Ant Colony Optimization.* Cambridge, MA: MIT Press.

Fu, M.C. 1994a. Optimization via simulation: A review. *Annals of Operations Research* 53: 199-248.

Fu, M.C., 1994b. Sample path derivatives for (s, S) inventory systems, *Operations Research* 42, 351-364.

Fu, M.C. 2002. Optimization for simulation: Theory vs. Practice (Feature Article). *INFORMS Journal on Computing* 14 (3): 192-215, 2002.

Fu, M.C. 2005. Gradient estimation. Chapter 19 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier.

Fu, M.C. and K.J. Healy. 1997. Techniques for simulation optimization: An experimental study on an (s, S) inventory system. *IIE Transactions* 29 (3): 191-199.

Fu, M.C. and S.D. Hill. 1997. Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions* 29 (3): 233-243.

Fu, M.C. and J.Q. Hu, 1997. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications.* Boston: Kluwer Academic Publishers.

Fu, M.C., J.Q. Hu, C.H. Chen, and X. Xiong. 2005 Simulation allocation for determining the best design in the presence of correlated sampling. *INFORMS Journal on Computing* forthcoming.

Glover, F. and M. Laguna. 1997. *Tabu Search.* Boston: Kluwer Academic.

Goldsman, D. and B.L. Nelson. 1998. Comparing systems via simulation. Chapter 8 in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, ed. J. Banks. New York: John Wiley & Sons.

Ho, Y.C., C.G. Cassandras, C.H. Chen, and L.Y. Dai. 2000. Ordinal optimization and simulation. *Journal*

of Operations Research Society 51: 490-500.

Ho, Y.C., R. Sreenivas, and P. Vakili. 1992. Ordinal optimization of DEDS. *Discrete Event Dynamic Systems: Theory and Applications* 2: 61-88.

Hochberg, Y. and A.C. Tamhane. 1987. *Multiple Comparison Procedures*. New York: John Wiley & Sons.

Hu, J., M.C. Fu, and S.I. Marcus. 2005a. A model reference adaptive search algorithm for global optimization. *Operations Research*, submitted. also available at `http://techreports.isr.umd.edu/ARCHIVE/dsp_reportList.php?year=2005&center=ISR`.

Hu, J., M.C. Fu, and S.I. Marcus. 2005b. A model reference adaptive search algorithm for stochastic global optimization, submitted.

Hu, J., M.C. Fu, and S.I. Marcus. 2005c. Simulation optimization using model reference adaptive search. *Proceedings of the 2005 Winter Simulation Conference*.

Kapuscinski, R. and S.R. Tayur. 1999. Optimal policies and simulation based optimization for capacitated production inventory systems. Chapter 2 in *Quantitative Models for Supply Chain Management*, eds. S.R. Tayur, R. Ganeshan, M.J. Magazine. Boston: Kluwer Academic Publishers.

Kim, S.-H., and Nelson, B.L. 2005. Selecting the best system. Chapter 18 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier, 2005.

Kleijnen, J.P.C. 1998. Experimental design for sensitivity analysis, optimization, and validation of simulation models. Chapter 6 in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, ed. J. Banks. New York: John Wiley & Sons.

Kushner, H.J. and G.G. Yin. 1997. *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag.

Laguna, M., and R. Marti. 2002. *Scatter Search*, Boston: Kluwer Academic Publishers.

Larrañaga, P. and J.A. Lozano. 2002. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston: Kluwer Academic Publishers.

L'Ecuyer, N. Giroux, and P. W. Glynn. 1994. Stochastic optimization by simulation: Numerical experiments with the M/M/1 queue in steady-state. *Management Science* 40: 1245-1261.

Ólaffson, S. 2005. Metaheuristics. Chapter 22 in *Handbooks in Operations Research and Management Science: Simulation*, S.G. Henderson and B.L. Nelson, eds., Elsevier.

Pflug, G.C. 1996. *Optimization of Stochastic Models*. Boston: Kluwer Academic Publishers.

Pinter, J.D. 1996. *Global Optimization in Action*. Boston: Kluwer Academic Publishers.

Rubinstein, R. Y. and A. Shapiro. 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. New York: John Wiley & Sons.

Rubinstein, R. Y. and D.P. Kroese. 2004. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. New York: Springer-Verlag.

Shi, L. and S. Ólafsson. 2000. Nested partitioned method for global optimization. *Operations Research* 48: 390-407.

Spall, J.C. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control* 37 (3): 332-341.

Spall, J.C. 2003. *Introduction to Stochastic Search and Optimization*. New York: John Wiley & Sons.

Swisher, J.R., P.D. Hyden, S.H. Jacobson, and L.W. Schruben. 2001. A survey of recent advances in discrete input parameter discrete-event simulation optimization, *IIE Transactions* 36(6): 591-600.

van Beers, W. C. M. and J. P. C. Kleijnen. 2003. Kriging for interpolation in random simulation, *Journal of the Operational Research Society* 54(3): 255-262.

AUTHOR BIOGRAPHIES

MICHAEL C. FU is a Professor in the Robert H. Smith School of Business, with a joint appointment in the Institute for Systems Research and an affiliate appointment in the Department of Electrical and Computer Engineering, all at the University of Maryland. He received degrees in mathematics and EE/CS from MIT, and an M.S. and Ph.D. in applied mathematics from Harvard University. His research interests include simulation methodology and applied probability modeling, particularly with applications towards manufacturing and financial engineering. He teaches courses in simulation, stochastic modeling, computational finance, and supply chain logistics and operations management. In 1995 he received the Allen J. Krowe Award for Teaching Excellence, and was a University of Maryland Distinguished Scholar-Teacher for 2004–2005. He currently serves as Simulation Area Editor of *Operations Research*, and was co-Editor for a 2003 special issue on simulation optimization in the *ACM Transactions on Modeling and Computer Simulation*. He is co-author of the book, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*, which received the INFORMS College on Simulation Outstanding Publication Award in 1998. His e-mail address is ⟨`mfu@rhsmith.umd.edu`⟩.

FRED W. GLOVER is President of OptTek Systems, Inc., and is in charge of algorithmic design and strategic planning initiatives. He currently serves as Comcast Chaired Professor in Systems Science at the University of Colorado. He has authored or co-authored more than 340 published articles and seven books in the fields of mathematical optimization, computer science and artificial intelligence, with particular emphasis on practical applications in industry and government. Dr. Glover is the recipient of the distinguished von Neumann Theory Prize, as well as of numerous other awards and honorary fellowships, including those from the American Association for the Advancement of Science, the NATO Division of Scientific Affairs, the Institute of Management Science, the Operations Research Society, the Decision Sciences Institute, the U.S. Defense Communications Agency, the Energy Research Institute, the American Assembly of Collegiate Schools of Business, Alpha Iota Delta, and the Miller Institute for Basic Research in Science. He also serves on advisory boards for numerous journals and professional organizations. His email address is ⟨glover@OptTek.com⟩.

JAY APRIL is Chief Development Officer of OptTek Systems, Inc. He holds bachelors degrees in philosophy and aeronautical engineering, an MBA, and a Ph.D. in Business Administration (emphasis in operations research and economics). Dr. April has held several executive positions including VP of Business Development and CIO of EG&G subsidiaries, and Director of Business Development at Unisys Corporation. He also held the position of Professor at Colorado State University, heading the Laboratory of Information Sciences in Agriculture. His email address is ⟨april@OptTek.com⟩.