Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

SIMULATION OPTIMIZATION: A PANEL ON THE STATE OF THE ART IN RESEARCH AND PRACTICE

Michael C. Fu

Robert H. Smith School of Business
University of Maryland
College Park, MD 20742, USA

Güzin Bayraksan

Department of Integrated Systems Engineering
The Ohio State University
Columbus, OH 43210, USA

Shane G. Henderson

School of Operations Research & Information Eng.
Cornell University
Ithaca, NY 14853, USA

Barry L. Nelson

Dept. of Industrial Eng. & Management Sciences
Northwestern University
Evanston, IL 60208, USA

Warren B. Powell

Dept. of Operations Research & Financial Eng.
Princeton University
Princeton, NJ 08544, USA

Ilya O. Ryzhov

Robert H. Smith School of Business
University of Maryland
College Park, MD 20742, USA

Ben Thengvall

OptTek Systems
2241 Seventeenth Street
Boulder, CO 80302, USA

ABSTRACT

The goal of this panel was to discuss the state of the art in simulation optimization research and practice. The participants included representation from both academia and industry, where the latter was represented by participation from a leading software provider of optimization tools for simulation. This paper begins with a short introduction to simulation optimization, and then presents a list of specific questions that served as a basis for discussion during the panel discussion. Each of the panelists was given an opportunity to provide their preliminary thoughts on simulation optimization for this paper, and the remainder of the paper summarizes those points, ranging from assessments of the field from their perspective to initial reactions to some of the posed questions. Finally, one of the panelists who has worked on an online testbed of simulation optimization problems for the research community was asked to provide an update of the status of the Web site.

978-1-4799-7486-3/14/$31.00 ©2014 IEEE

1 INTRODUCTION

From the preface of the *Handbook of Simulation* (Fu 2014): "Arguably, the two most powerful operations research/management science (OR/MS) techniques are simulation and optimization (where simulation refers to stochastic simulation, whereby there is randomness in the system, also known as Monte Carlo simulation). Both approaches were propelled forward by the advent of the digital computer over half a century ago, leading up to the present golden age when both routinely address complex large-scale real-world problems and both are implemented in a large variety of computer software packages. However, combining the two techniques is a more recent development, and software effectively integrating the two is relatively limited; thus, simulation optimization remains an exciting and fertile area of research." The objective of this panel was to discuss the state of the art in simulation optimization research and practice.

The panel consisted of four researchers from academia (Güzin Bayraksan, Shane Henderson, Barry Nelson, and Warren Powell) and two representatives from leading software providers of optimization tools for simulation. Michael Fu and Ilya Ryzhov, co-chairs of the simulation optimization track for WSC 2014, served as moderators for the panel. The handbook (Fu 2014) provides more details on the state of the art of simulation optimization, including a survey of the most well-established approaches and a sampling of recent research advances in theory/methodology. Shorter introductions to the area can be found in introductory and advanced tutorials in the annual Winter Simulation Conference (WSC) Proceedings, including most recently (Chau, Fu, Qu, and Ryzhov 2014).

Optimization problems have traditionally been classified according to the decision variables, the primary dichotomy being discrete versus continuous. For discrete, the variables could be ordered or unordered (categorical, the simplest being binary decisions). Another dichotomy involves the size of the feasible region: finite versus infinite for discrete problems, bounded versus unbounded for continuous problems. Among the dominant methodologies used in simulation optimization are the following: statistical ranking and selection, efficient simulation budget allocation methods, response surface methodology, random search methods, robust metaheuristics, Bayesian methods, stochastic approximation, and sample average approximation (also known as sample path optimization, stochastic counterpart).

The remainder of this paper contains the list of questions to be considered by the panel. This is followed by some overarching themes summarized by some of the panelists, and initial reactions by several of the other panelists to the questions. Finally, the last section contains a brief update on the Simopt.org testbed of simulation optimization problems, provided by Shane Henderson.

2 FOOD FOR THOUGHT

The following list of questions served as a basis of discussion for the panel.

- *What are some of the present and future challenges and opportunities in simulation optimization research?*
- *What challenges and opportunities do you see in integrating simulation optimization with practice?*
- *What challenges and opportunities do you see in the future of simulation optimization software?*
- *What are the greatest advances you've seen recently in simulation optimization software?*
- *Do you see any new approaches or existing algorithms from research that you think could be fruitfully implemented in future commercial software (that are not already implemented)?*
- *What do you feel are the greatest barriers in going from research to commercial software? And what are some ways to overcome these barriers?*

3 PANELIST STATEMENTS

This section provides initial thoughts on simulation optimization provided by each of the panelists prior to the conference. Two of the panelists responded directly to the questions posed in the previous section, whereas the other three panelists provided more general statements.

GÜZİN BAYRAKSAN

The problem of optimizing a stochastic system can be approached from two different angles. One is the simulation approach, which relies on a black-box simulation to generate estimates of function values. The other is the optimization approach that aims to model and solve mathematical programs under uncertainty. This presents an opportunity for future research to investigate the commonalities and synergies between the simulation and optimization approaches. Combining both approaches could yield powerful new techniques, allowing us to solve problems that are beyond our capabilities today.

One particular area of research that could benefit from using both approaches in tandem is the rigorous incorporation of risk when optimizing systems. Output from simulation models provide not only average behavior but also variability of the possible outcomes. The output from simulation runs could be used to control risks for a decision maker. Mature tools that deal with risky but rare events already exist in the simulation toolbox but these methods need to be further developed when the aim involves optimizing (minimizing) risk. On the other hand, there is a growing body of optimization literature on how to model and solve mathematical optimization problems for risk-averse decision makers. Managing risk is important in practice, and it could provide many opportunities for methodology development, exploiting the strengths of both simulation and optimization approaches.

Recent developments in the software side are encouraging, with implementations of specific or general algorithms. However, for the success of simulation optimization in practice, software that (i) connects data to models, (ii) simulates and optimizes models using effective solution algorithms, and (iii) provides the users with ways to assess the quality of the obtained solutions need to be further developed. A full-suite stochastic optimization and simulation software would be beneficial to practitioners. One challenge in integrating research into practice is that many practitioners are left wondering which model-method combination is most effective under what conditions. For instance, what types of problems are best tackled by using both simulation and optimization approaches in tandem? More research effort needs to be spent to determine the relative merits and disadvantages of different modeling and solution techniques.

As a final remark, it would be beneficial to enable more interactions between the simulation and optimization communities, software developers, and practitioners. How can this be achieved? This panel is one great example. Workshops for simulation optimization would be most useful. Such workshops have started to be organized and the Winter Simulation Conference has had a steady stream of simulation optimization tracks for a while now. However, further efforts to bring together different parties could prove invaluable for cross-pollination of ideas and bridging research and practice.

SHANE G. HENDERSON (Parallel Computing for Simulation Optimization)

The exponential growth in computer processor speeds that we have enjoyed for several decades appears to be over. Still, computing power is increasing through increases in the number of processors used in a single computer. Multicore computers are now commonplace and look set to be the avenue for future computing-power growth. Accordingly, I think the simulation optimization (SO) community should be directing much of its energy to develop algorithms and analysis techniques that are appropriate for such environments. It is not just a case of "giving one replication to each core and gathering the results." Indeed, there are many algorithmic and statistical complexities involved; see Luo, Hong, Nelson, and Wu (2013) and Ni, Hunter, and Henderson (2013). I should also emphasize that I am thinking of SO problems where the individual simulation replications can be easily completed on a single core, as opposed to the setting in "Parallel and Distributed Simulation" where a single replication requires large numbers of cores to complete.

I first realized that I should be working on devising SO algorithms for parallel computing at a pre-conference workshop organized by Michael Fu and Barry Nelson at the 2010 Winter Simulation Conference where Barry Nelson gave a talk entitled "Simulation Optimization When We Have Massively

Parallel Computing." This was a true "wake-up call" for me, and since then I have directed much of my research effort in this direction.

There is huge scope for devising algorithms for the different parallel environments that exist today, from multicore (and shared memory) desktops, to cloud computing, to high-performance computing. The term "high-performance computing" refers to computing clusters that may have many thousands of cores, linked through communication infrastructure. These environments vary in terms of whether memory is shared or not (it is shared on desktops, but not in the other environments), whether cores are reliable or not (in cloud computing they may not be), and whether communication is fast or not (for example, message passing in high-performance computing environments can take on the order of 10^{-9} seconds). We need different algorithms for all these environments; they are not the same.

We have some "catch up" to do with the physics community which has been using high-performance computing for simulation for decades, e.g., Pawley et al. (1985). As far as I can tell, however, the physics community has primarily focused on Markov chain Monte Carlo problems, so while we have much to learn from that community, there is enormous scope for research.

Surely if anyone can exploit parallel computing, the fields of simulation and simulation-optimization can!

WARREN B. POWELL (Optimizing simulators: Modeling and the four classes of policies)

Simulation optimization is a well known topic in the simulation community, addressing the problem of *designing* complex systems by finding the best settings of design variables: the size of buffers, the location of facilities, the mix of equipment in a fleet of vehicles. There are many problems where we have to *control* the system over time, which might involve dispatching trucks, scheduling doctors, or scheduling the generation of electricity.

If the problem is deterministic, we might model our problem as a time-staged mathematical program using

$$\min_{x_0,\ldots,x_T} \sum_{t=0}^{T} c_t^T x_t \qquad (1)$$

subject to

$$\begin{aligned} A_0 x_0 &= b_0, & (2)\\ A_t x_t - B_{t-1} x_{t-1} &= b_t, \quad t = 1,\ldots,T, & (3)\\ x_t &\geq 0, \quad t = 1,\ldots,T. & (4) \end{aligned}$$

This modeling vocabulary is used uniformly around the world.

If we introduce random variables, as is uniformly the case with simulation problems, then the academic community splinters in a number of directions with multiple notational systems. A quick literature search turns up names such as Markov decision processes, stochastic programming, optimal control, reinforcement learning, approximate (or adaptive) dynamic programming and decision trees. Terms such as policies, control laws, rules, and strategies are used as if they have well defined definitions and interpretations. Many simulation models imbed rules (assign a job to the shortest queue) without explicitly recognizing that a decision is being made.

The lack of a standardized modeling style represents, in the view of this author, the biggest impediment to the sharing of ideas between communities. We would argue that a simple modeling framework can be used to describe a vast array of sequential decision problems, using five basic components:

- State variables - Widely used but rarely defined, a state variable S_t (x_t in control theory) is the minimally dimensioned function of history that carries the information needed to compute decision functions, cost/reward functions, and transition functions, from time t onward.

- Decision variables - Decisions x_t (or actions a_t or controls u_t) are the variables we want to control over time. While the model should not address how a decision is made, we introduce the notion of a policy which we denote $X^\pi(S_t)$ (or $A^\pi(S_t)$ or $U^\pi(S_t)$) as a function that returns a (feasible) decision $x_t \in \mathscr{X}_t$ ($\mathscr{X}_t$ is given by equations (2) - (4) above). We assume that our challenge is to choose from a family of policies $X^\pi(S_t)$, for $\pi \in \Pi$ where Π is a pre-specified set (discussed below).
- Exogenous information - We let W_t be the set of variables that are first learned at time t from an exogenous source (demands, prices, rainfall, equipment failures).
- Transition function - Known under various names such as the system model, plant model, transfer function, or simply "model," the transition function captures all the dynamics of the system, which we write as $S_{t+1} = S^M(S_t, x_t, W_{t+1})$. In our deterministic problem, the transition function (equation (3)) is viewed as simply another constraint.
- Objective function - We let $C(S_t, x_t)$ be the cost (or contribution/reward) earned at time t. In a deterministic problem (such as (1)-(4)), the problem is to find the best decisions $x_0, \dots, x_T$. In a sequential decision problem, the challenge is to find the best *policy* (or function) for making decisions, which we can write

$$\min_{\pi \in \Pi} \mathbb{E} \sum_{t=0}^{T} C(S_t, X^\pi(S_t)), \tag{5}$$

where $S_{t+1} = S^M(S_t, x_t, W_{t+1})$ and where the expectation is taken over all possible values of $W_1, \dots, W_T$. It is important to emphasize that we may replace the expectation with quantiles (e.g. minimizing the 95th percentile of costs), various risk measures, or the worst case (known as robust optimization).

This modeling framework can be found in books on dynamic programming, but is rarely used in published papers in operations research or computer science. It is more familiar to the control theory community, and yet even this community has not adopted it as a standard archetype with the familiarity of the standard canonical form used in deterministic math programming. In fact, we have been surprised by the almost universal lack of a standard definition for a state variable.

We think that a major reason for the failure to adopt this modeling framework is that the objective function (5) involves a search over "policies" that may be mathematically elegant but, for many, is computationally meaningless. People tend to gravitate toward specific classes of policies that are well defined, and well suited to specific applications.

In our work, we have identified four fundamental classes of policies that span all the algorithmic strategies that we have encountered. These are

- Policy function approximations (PFAs) - These are analytic functions that map a state to an action. These functions may be a lookup table (e.g., turn left at a particular intersection), a rule such as "admit a customer to the queue if it is less than θ," a linear model such as $X^\pi(S_t|\theta) = \theta_0 + \theta_1 S_t + \theta_2 S_t^2$, or a neural network.
- Cost function approximation (CFAs) - A myopic policy might be written

$$X^\pi(S_t) = arg\,min_{x_t \in \mathscr{X}_t} C(S_t, x_t).$$

Now imagine replacing $C(S_t, x_t)$ with an approximation that we denote $\bar{C}^\pi(S_t, x_t|\theta)$ where π captures the nature of the functional approximation, and θ is a set of tunable parameters.
- Policy based on a value function approximation (VFAs) - These policies might be written

$$X^\pi(S_t) = arg\,min_{x_t} \big(C(S_t, x_t) + \mathbb{E}\overline{V}_{t+1}(S_{t+1}|\theta) \big).$$

Here, we might use a linear approximation for the value function

$$\overline{V}_{t+1}(S_{t+1}|\theta) = \sum_{f \in \mathscr{F}} \theta_f \phi_f(S_{t+1}).$$

This is the policy most often associated with dynamic programming.

- Lookahead policies - The simplest lookahead policy optimizes over a horizon $t,\ldots,t+H$ using point forecasts of the future. The field of stochastic programming favors a stochastic lookahead model, which involves solving an approximation of the original model (5) using devices such as Monte Carlo sampling, stage aggregation, temporal aggregation, and dimensionality reduction (ignoring variables).

Each of these policies represents a class of function, combined with a set of tunable parameters θ, which can be tuned using the objective function (5), which generally has to be written as a simulator.

BARRY L. NELSON

1. *What are some of the present and future challenges and opportunities in simulation optimization research?*

   The top three on my list are exploiting high-performance parallel computing, dealing with multiple objectives (including stochastic constraints), and developing a framework for robust optimization in our setting.

2. *What challenges and opportunities do you see in integrating simulation optimization with practice?*

   I have claimed that we do simulation for three reasons: feasibility ("We have a plan, will it work?"); sensitivity ("We're not really sure about everything, so how much does it matter?"); and optimization ("What are the good options and how good are they?"). My conclusion is that that sounds a lot like mathematical programming. But when we teach people how to do simulation we do not think of it as mathematical programming. If we did, then I suspect we would see more use of simulation optimization, and probably more demand for simulation optimization research to provide the kinds of answers people expect from mathematical programming. Our challenge is changing the way we think about simulation so that it is less about modeling and more about decision support.

3. *What challenges and opportunities do you see in the future of simulation optimization software?*

   The deterministic optimization community thinks in terms of solvers that have strengths for particular classes of problems. This has been hugely important. In simulation optimization research we also tend to develop theory and methods tailored to specific classes of problems. Yet for some reason on the commercial software side we have expected *one* solver to handle *all* simulation optimization problems we might have. This is very limiting. We have a real opportunity to improve what we do if we can somehow move toward the "solver" mentality in software.

4. *What are the greatest advances you've seen recently in simulation optimization software?*

   The advances I have noticed on the commercial software side have been in making the solver easier to use and expanding the classes of problems it can address. As I said above, I am not sure having a single solver address an even wider class of problems is such a good direction. I am sure there have also been internal changes to make these optimizers more robust and efficient. I have also seen more willingness by researchers to talk to and work with the simulation software vendors to translate their ideas into commercial products, and that is a good thing.

5. *Do you see any new approaches or existing algorithms from research that you think could be fruitfully implemented in future commercial software (that are not already implemented)?*

   Ranking and selection in high-performance parallel computing environments is going to happen—actually already is happening to a limited extent—and for once I think the research support is going to be there right when it is commercially needed. This one is a no-brainer: we are good at simulation optimization problems for which we can exhaust the feasible solutions and high-performance parallel computing is going to move more problems into that class. But if you do ranking and selection stupidly in a parallel environment you can be inefficient (which costs money if you are paying for the CPU's) and also make mistakes.

6. *What do you feel are the greatest barriers in going from research to commercial software? And what are some ways to overcome these barriers?*

   Leon McGinnis of Georgia Tech, a manufacturing guy and frequent WSC participant, once said to me, "The problem with you guys in simulation is that you can't agree on what a model is!" This has made it hard for us to move to a many-solver environment, and that is a big barrier. I have to applaud OptTek Systems, Inc. in that they have been willing to do the dirty work to customize OptQuest® for so many simulation modeling platforms. But that is not a very scalable solution. As I said above, the big step is in thinking about simulation differently, less as queueing theory and more as mathematical programming.

BEN THENGVALL

1. *What are some of the present and future challenges and opportunities in simulation optimization research?*

   When our firm OptTek Systems began providing the commercial simulation optimization library OptQuest 20 years ago, the concept of simulation optimization was not widely understood nor embraced. Much has changed in the last two decades, and a majority of commercial discrete event, Monte Carlo, and agent-based simulation frameworks now embed a simulation optimization capability.

   There are still many opportunities for improvement though and we are currently active in researching better approaches for multiple objective optimization, dynamically adjusting optimization techniques based on model characteristics, and analysis and visualization of simulation optimization results to provide easily interpretable insights for analysts.

2. *What challenges and opportunities do you see in integrating simulation optimization with practice?*

   Even though simulation optimization capabilities are embedded in most commercial simulation packages today, it is used in only a fraction of simulation studies. A great deal of time and effort in simulation studies is expended modeling and validating the real-world system in the simulation tool. However, analysts rarely retrieve all of the knowledge and insights that the completed model may yield. Often sponsors of a study limit their questions to a few discrete variations on a base case, or a simple parametric analysis on one or two variables. Running only a handful of alternate simulation runs generally does not answer all of the relevant questions. Analysts are limited to these simple explorations, because they do not have the time and training to perform more complex analyses manually and they do not understand how to utilize embedded optimization tools to automate these analyses and retrieve all of the knowledge and insights available from their models.

   As a community we need to better educate young simulation analysts about the power of combining simulation and optimization. We also need to do more work in our user interfaces to make it easy and intuitive to 1) identify simulation inputs to vary and outputs to collect, 2) define constraints and objectives, 3) monitor optimization progress, and 4) view and interpret simulation optimization results.

3. *What are the greatest advances you've seen recently in simulation optimization software?*

   An area that we have been investing in the last couple years is pairing an analysis capability with our simulation optimization offerings. After solving a simulation optimization problem you have a dataset of 100s or 1000s of pairs of simulation inputs and simulation outputs. We have traditionally been concerned with what is the optimal set of simulation inputs to achieve our objective, or perhaps with finding an efficient frontier of options that provide optimal tradeoffs between multiple objectives. What we have heard from our customers though is that in addition to optimal solutions, they are also keenly interested in a characterization of the system they are modeling.

The questions most commonly asked to characterize a system are which simulation input values are most influential on the simulation outputs of interest, are there relationships that define how subsets of simulation inputs interact with simulation outputs, and are there good and bad regions of the simulation input space relative to simulation outputs of interest. Identifying the most influential simulation inputs allows analysts to hone their search to the most promising inputs and understanding good and bad regions of the input space rather than just providing point solutions allows decision makers greater flexibility in design and also provides a measure of robustness for a proposed course of action.

We are incorporating a series of statistical and data mining techniques in our software that automatically execute following a simulation optimization run to provide influential variable and good and bad region analysis. At times these analysis results confirm an analysts expectations. When analysis results yield surprises they either provide unexpected insights into the underlying system being modeled or lead the analyst to an error in logic or modeling that needs to be addressed.

4. *What do you feel are the greatest barriers in going from research to commercial software? And what are some ways to overcome these barriers?*

   The market is very pragmatic; commercial software only survives if analysts in the field understand it and are able to derive value from it. Some research is too specialized or too arcane for general commercial software, but there is much that has been done and is being done in the research community that could be quite beneficial to business and government users. Those of us that straddle the research and commercial spheres need to do a better job of articulating the benefits of this research and packaging simulation, optimization, and data analysis tools in ways that simulation analysts without advanced degrees can understand and use.

4 SIMULATION OPTIMIZATION TESTBED

The idea of a testbed of simulation optimization problems was suggested during the WSC panel in 2000, so it is only fitting that we conclude with an update on that testbed that has been developed under the leadership of Raghu Pasupathy and Shane Henderson, and housed at http://www.simopt.org.

An Update on SimOpt (Shane)

Raghu Pasupathy and I continue to develop SimOpt (Pasupathy and Henderson 2011, Pasupathy and Henderson 2006, Pasupathy and Henderson 2011), a repository of simulation optimization (SO) test problems. Our development of SimOpt was motivated by the excellent survey (Fu 2002) and the commentary (Glynn 2002). The goals of SimOpt are as follows.

- It is typically difficult to develop an understanding of SO algorithms through analytical results. Such results are usually limited to establishing convergence to a local or global optimum as the computational effort grows without bound, or to results that explain the behavior of an algorithm once it reaches a neighborhood of the optimal solution. These results are interesting and important, but they are usually only relevant in “asymptopia” (a wonderful word that I first heard from Peter Glynn), when the computational effort expended is massive, and probably beyond what we can achieve. SimOpt provides a suite of test problems on which we can run SO algorithms and compare how they perform, thereby helping drive innovation on algorithm design.

- As SimOpt grows, problem *classes* may become apparent, for which specialized algorithms can be developed and applied. I hope that one day SO software will be able to recognize structure in models automatically, and select an SO algorithm that has been tailored for such structure. I also hope that one day SO users will be aware of the benefits of problem structure and build their models accordingly, much as the field of deterministic optimization exploits specialized algorithms for specialized structures. Examples of such structured classes are linear programs, second-order cone programs, semidefinite programs, smooth nonlinear optimization problems and so forth.

A side benefit of such a testbed is that as users become more sophisticated in their understanding and use of SO algorithms, they will also realize the limitations of such methods. Even finding a local minimum of a continuous function through access to only noisy function evaluations is a great challenge, and we must avoid "over promising" on what such algorithms can achieve.

The repository now contains 49 problems, of which 43 have been coded. Almost all of the problems are coded in Matlab, for ease of sharing and prototyping. Current work involves verifying the code that has been implemented, and adding additional problems. An important development we are excited about is the addition of SO *algorithms* to SimOpt, as described in Pasupathy and Henderson (2011). When complete, this will allow us to compare multiple algorithms on multiple problems with a single command, thereby greatly facilitating our ability to benchmark algorithms. A key ongoing challenge is ensuring that algorithms and example problems can be seamlessly integrated when they are implemented in different languages.

The problems are available for use, and indeed have been used, by a number of researchers. We hope that this promising initial usage will continue to grow and that people will share their experiences through the wiki pages on the site. You can help further the SimOpt initiative by downloading and using the problems, and by creating new problems based on your own research and practical experience. Creating a new problem is not trivial. Aren't Ph.D. students a blessing? An alternative is to send us information that is sufficient for us to create the problem. This information could be as simple as a pointer to a paper. We are also keen to add SO algorithms to the site, so do please send us your SO code. Ideally the code would be in Matlab, although we are working to be able to handle other languages by enclosing your code in a Matlab "wrapper" that interfaces your code with the Matlab-encoded problems.

ACKNOWLEDGMENTS

Güzin Bayraksan's work was partially supported by National Science Foundation Grant CMMI-1345626 Shane Henderson's work was partially supported by National Science Foundation Grant CMMI-1200315. Warren Powell's work was partially supported by AFOSR grant contract FA9550-11-1-0172. Michael Fu's work was partially supported by the National Science Foundation (NSF) under Grants CMMI-0856256 and EECS-0901543, and by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-10-1-0340.

REFERENCES

Chau, M., M. C. Fu, H. Qu, and I. O. Ryzhov. 2014. "Simulation Optimization: An Overview and Recent Developments". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller. Piscataway NJ: IEEE.

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

Fu, M. C. 2014. *Handbook of Simulation Optimization*. Springer.

Glynn, P. W. 2002. "Additional perspectives on simulation for optimization". *INFORMS Journal on Computing* 14:220–222.

Luo, J., J. L. Hong, B. L. Nelson, and Y. Wu. 2013. "Fully Sequential Procedures for Large-Scale Ranking-and-Selection Problems in Parallel Computing Environments". *Working Paper*.

Ni, E. C., S. R. Hunter, and S. G. Henderson. 2013. "Ranking and selection in a high performance computing environment". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 833–845. Piscataway NJ: IEEE.

Pasupathy, R., and S. G. Henderson. 2006. "A testbed of simulation-optimization problems". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 255–263. Piscataway NJ: IEEE.

Pasupathy, R., and S. G. Henderson 2011. "SimOpt". Accessed May 9, 2014. http://www.simopt.org.

Pasupathy, R., and S. G. Henderson. 2011. "SimOpt : A library of simulation optimization problems". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 4080–4090. Piscataway NJ: IEEE.

Pawley, G., K. Bowler, R. Kenway, and D. Wallace. 1985. "Concurrency and parallelism in MC and MD simulations in physics simulations in physics". *Computer Physics Communications* 37:251—260.

AUTHOR BIOGRAPHIES

MICHAEL C. FU is Ralph J. Tyser Professor of Management Science in the Robert H. Smith School of Business, with a joint appointment in the Institute for Systems Research and affiliate faculty appointment in the Department of Electrical and Computer Engineering, all at the University of Maryland. He holds a Ph.D. in applied math from Harvard and degrees in math and EECS from MIT. His research interests include simulation optimization and applied probability, with applications in supply chain management and financial engineering. He served as WSC 2011 Program Chair, NSF Operations Research Program Director, *Management Science* Stochastic Models and Simulation Department Editor, and *Operations Research* Simulation Area Editor. He is a Fellow of **INFORMS** and **IEEE**. His email address is mfu@umd.edu.

GÜZİN BAYRAKSAN is an Associate Professor in the Department of Integrated Systems Engineering at the Ohio State University. She received her Ph.D. in Operations Research and Industrial Engineering from the University of Texas at Austin. She is the recipient of 2012 NSF CAREER award, 2012 Five Star Faculty Award (UA), and the 2008 INFORMS best case study award. Her research interests are in stochastic optimization, particularly Monte Carlo simulation-based and data-driven methods for stochastic programming with applications to water resources management. Her email address is bayraksan.1@osu.edu and her web page is www.ise.osu.edu/ISEFaculty/bayraksan.

SHANE G. HENDERSON is a professor in the School of Operations Research and Information Engineering at Cornell University. He holds a B.Sc. (Hons) from the University of Auckland and an M.S. (Statistics) and Ph.D. (Operations Research) from Stanford University. His research interests include discrete-event simulation and simulation optimization, and he has worked for some time with emergency services. He is an associate editor for both *Management Science* and *Stochastic Systems*, and he co-edited the Proceedings of the 2007 Winter Simulation Conference. His email address is sgh9@cornell.edu and his web page is http://people.orie.cornell.edu/∼shane.

BARRY L. NELSON is the Walter P. Murphy Professor and Chair of the Department of Industrial Engineering and Management Sciences at Northwestern University. He holds a BA in Mathematics and Computer Science from DePauw University and MS and PhD in Industrial Engineering from Purdue University. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems, and he is the author of *Foundations and Methods of Stochastic Simulation: A First Course*, from Springer. He is a Fellow of **INFORMS** and **IIE**. His email address is nelsonb@northwestern.edu.

WARREN B. POWELL is a professor in the Department of Operations Research and Financial Engineering at Princeton University, where he has taught since 1981. He holds a B.S. in engineering from Princeton and am M.S. and Ph.D. in Civil Engineering from MIT. His research specializes in computational stochastic

optimization, with applications in energy, transportation, health and finance. He has authored/coauthored over 190 publications and two books. He founded and directs CASTLE Labs and the Princeton Laboratory for Energy Systems Analysis (PENSA), focusing on the solution of stochastic optimization problems that arise in a wide range of applications in energy, transportation, finance and health. He is a Fellow of **INFORMS**. His email address is powell@princeton.edu and his web page is http://www.castlelab.princeton.edu.

ILYA O. RYZHOV is an Assistant Professor in the Robert H. Smith School of Business at the University of Maryland. He received a Ph.D. in Operations Research and Financial Engineering from Princeton University. His research deals with optimal learning and the broader area of stochastic optimization, with applications in disaster relief, energy, and revenue management. He was a recipient of WSC's Best Theoretical Paper Award in 2012. His work has appeared in *Operations Research*, and he is a co-author of the book *Optimal Learning*, published in 2012 by John Wiley & Sons. His email address is iryzhov@rhsmith.umd.edu.

BEN THENGVALL is Vice President of Government Solutions at OptTek Systems, Inc. He holds a M.S.E. and Ph.D. in Operations Research and Industrial Engineering from the University of Texas at Austin. He has extensive experience in the commercial and government spheres designing and implementing pragmatic software solutions to solve complex real-world problems. His primary interests and areas of application are metaheuristics, mathematical modeling, and simulation optimization and analysis. His email address is thengvall@opttek.com.