Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

GREEN SIMULATION OPTIMIZATION USING LIKELIHOOD RATIO ESTIMATORS

David J. Eckman

School of Operations Research and Info Engrg
Cornell University
206 Rhodes Hall
Ithaca, NY 14853, USA

M. Ben Feng

Dept of Statistics and Actuarial Science
University of Waterloo
200 University Ave W
Waterloo, ON N2L 3G1, CANADA

ABSTRACT

Green simulation is the reuse of past simulation outputs to enhance the efficiency of current and future simulation experiments. One natural application of green simulation is in the context of simulation optimization, wherein outputs from past iterations in a search can be reused in subsequent iterations. In this article, we draw attention to challenges that arise when green simulation likelihood ratio estimators are naively employed in simulation optimization. In particular, we show that for searches that identify new designs based on past outputs, outputs in different iterations are conditionally dependent, violating one of the assumptions for the validity of the likelihood ratio estimator. As a result, green simulation likelihood ratio estimators of the objective and gradient can become biased. We demonstrate how this conditional dependence and bias can adversely affect the behavior of gradient-based optimization algorithms.

1 INTRODUCTION

Green simulation was introduced by Feng and Staum (2017) as a new experiment design paradigm that reuses outputs from past simulation experiments to improve the efficiency of future experiments. The appeal of green simulation is that reusing past outputs can provide useful information about future experiments at low computational cost. We consider simulation optimization problems where at each iteration, random simulation outputs are generated and used to estimate the objective, gradient, or any other quantity of interest. In this context, it is natural to apply green simulation to the outputs from previous iterations to improve the quality of the estimation. We identify some challenges in this application of green simulation and explore how they can influence the performance of optimization algorithms. Our main contributions are (1) identifying the violation of a key assumption of Feng and Staum (2017), (2) analyzing the consequences of this violation, and (3) illustrating the consequences in different examples.

Green simulation is motivated by the setting of repeated experiments, wherein experiments are run periodically and the designs in the sequence of experiments change over time (Feng and Staum 2017). This setting arises in practice when simulation is used to support periodic operational decisions, e.g., risk management of financial derivatives, estimation of project completion times, and projection of future stock-outs. In these examples, the designs of the current experiment are the estimates of relevant risk factors, the progress on different project tasks, and the inventory levels, respectively. In the model of Feng and Staum (2017), the designs observed over time are assumed to follow a stochastic process and the current design is assumed to be independent of the outputs from the past experiments. In many practical settings, however, the current design is the result of past operational decisions that were based on the outputs of previous experiments. For example, current inventory levels can be viewed as a consequence of previous reorder decisions that were based on past stock-out projections.

In this article, we study green simulation in settings where the current design depends on the evaluation of past designs. Specifically, we explore green simulation estimators through the lens of simulation optimization, an application proposed by Feng and Staum (2017) as an avenue for future research. In

978-1-5386-6572-5/18/$31.00 ©2018 IEEE

simulation optimization problems, a decision-maker is concerned with identifying an optimal design when the performance of a design must be estimated via stochastic simulation. For the aforementioned inventory example, the decision-maker's goal may be to determine the reorder quantities that minimize some combination of reorder, holding, and stock-out costs.

Many simulation optimization algorithms feature a search process that identifies new designs for evaluation based on the observed or estimated performance of the current design and possibly all past designs. Simulation optimization searches can leverage green simulation estimators in several ways. Firstly, simulation outputs can be reused to estimate the objective function at any or all previously visited designs so as to guide a pattern search or to fit a metamodel. Secondly, green simulation estimates of the gradient can be used in a gradient-based search, such as stochastic approximation. Thirdly, if stochastic constraints are present, green simulation estimators can be used to estimate constraint violations.

If the computational cost of reusing outputs is low, green simulation estimators may be appealing for all of the above purposes. Nonetheless, green simulation estimators at the current design use the *same* outputs from past simulation experiments that were used by the search to identify the current design. Consequently, conditional on the designs visited along a search path, the outputs from past experiments are conditionally dependent. We further show how this conditional dependence can result in biased green simulation estimators of both the objective and the gradient. Through a number of examples, we illustrate how gradient-based searches, specifically stochastic approximation, can be misled by biased estimates of the gradient, leading to slower convergence. Moreover, if green simulation estimators are used in the extreme—with all past replications being reused—this bias can persist throughout a search. Although we focus on gradient-based searches in our examples, we believe that non-gradient-based searches would similarly suffer from the use of biased green simulation estimates of the objective.

A related issue arises in ranking and selection when the designs under consideration are determined by a simulation optimization search (Eckman and Henderson 2018). In this setting, a search may use the estimated ordering of previously explored designs to identify new designs. The primary issue is that simulation outputs that were independent and identically distributed become conditionally dependent given the identities of the explored designs. This conditional dependence can cause the statistical guarantees of ranking-and-selection procedures to be violated when outputs are reused.

Reusing simulation outputs has been studied in different contexts, especially in stochastic simulation metamodeling. For example, the nested simulation methods of Barton et al. (2013) and Xie et al. (2014) use stochastic kriging (Ankenman et al. 2010) to reuse outputs from a few designs to estimate the response surface for other designs. In reinforcement learning, off-policy learning methods reuse outputs to estimate the action-value function of a policy (Precup et al. 2001; Mandel et al. 2014).

One particular implementation of green simulation is the so-called likelihood ratio (LR) method, also known as the score function method. The LR method has been applied in metamodeling, sensitivity analysis, and optimization; see, for example, L'Ecuyer (1990), L'Ecuyer (1993), Rubinstein and Shapiro (1993), Kleijnen and Rubinstein (1996), Glasserman and Xu (2014), and Fu (2015). We focus on green simulation via the LR method, which weights pre-existing outputs by appropriate likelihood ratios to provide an unbiased estimator for the current experiment. This is the main implementation method studied in Feng and Staum (2017). In doing so, we restrict our attention to simulation optimization problems in which the design does not affect any structural parameters of the simulation model. This excludes, for example, simulation-based optimization of dynamic systems where the objective is to find the best facility layout.

A related application of the LR method is importance sampling, a technique commonly used for variance reduction or as an alternative to Markov chain Monte Carlo methods (Bugallo et al. 2017). Importance sampling has also been used in stochastic optimization algorithms, such as stochastic gradient descent (Zhao et al. 2015; Needell et al. 2016). A relevant extension of the importance sampling technique is adaptive importance sampling (AIS), in which the proposal distribution is iteratively updated based on past outputs to produce an estimator with smaller variance. AIS has also been incorporated into simulation optimization algorithms such as simultaneous perturbation stochastic approximation (SPSA) (Medina and Taflanidis 2014; Spall 1998).

2 GREEN LIKELIHOOD RATIO ESTIMATORS IN SIMULATION OPTIMIZATION

In developing green simulation estimators based on the likelihood ratio method, Feng and Staum (2017) consider a setting of repeated experiments in which the same simulation model (logic) is used to evaluate different designs. If the designs represent the distributional parameters for the random variables in the simulation model, outputs from previous designs can be weighted using the likelihood ratio method.

Mathematically, let $x \in \mathscr{X}$ be a given design of interest; for clarity we use lowercase x when the design is fixed and uppercase X when it is stochastic. In the setting of Feng and Staum (2017), the sequence of designs $\{X_k : k = 1,2,\ldots\}$ is regarded as a discrete-time stochastic process on a Polish space $\mathscr{X}$. For a finite sequence of designs, $X_1,\ldots,X_n$, we will refer to X_n as the *current* design and to $X_1,\ldots,X_{n-1}$ as *past* designs. Given X_n, the current simulation experiment samples a random vector Y_n according to the conditional likelihood $f(\cdot;X_n)$. Unless otherwise stated, we assume that for all $x \in \mathscr{X}$, the conditional likelihoods $f(\cdot;x)$ have a common support $\mathscr{Y}$, i.e., Assumption (A4) in Feng and Staum (2017). After the random vector Y_n is simulated, the simulation logic $h(\cdot) : \mathscr{Y} \mapsto \mathbb{R}$ is applied to produce the simulation output $h(Y_n)$.

Conditional on $X_n = x$, the goal of the current experiment is to estimate the expected performance

$$\mu(x) = \mathbb{E}_x[h(Y_n)] = \int_{\mathscr{Y}} h(y) f(y;x) dy, \tag{1}$$

where $\mathbb{E}_x$ denotes expectation under the conditional likelihood $f(\cdot;x)$. In the simulation optimization context, $\mu(x)$ is the objective function value at the iterate x. In (1), the design x is not an argument to the simulation logic $h(\cdot)$, so it affects the expected performance $\mu(x)$ only through the conditional likelihood $f(\cdot;x)$. The model described in (1) is not suitable for all settings; for example, if x represents the number of servers in a queuing system and is a direct input to the system performance measure $h(\cdot)$, then (1) does not apply. Nevertheless, in such settings it may be possible to "push out" the dependence of $h(\cdot)$ on x to the conditional likelihood $f(\cdot;x)$ (Rubinstein 1992).

A common way to estimate $\mu(x)$ is by running r independent replications of the simulation model and then taking the sample average of the outputs, i.e., the *Standard Monte Carlo (SMC)* estimator:

$$\widehat{\mu}_r^{SMC}(x) = \frac{1}{r}\sum_{j=1}^{r} h(Y^{(j)}), \qquad Y^{(j)} \text{ i.i.d.} \sim f(y;x),\ \forall j = 1,\ldots,r. \tag{2}$$

For simplicity, we assume that the number of replications, r, is fixed for all designs, but this is not essential.

We see from (2) that only the outputs in the current experiment are used to estimate $\mu(x)$; outputs from previous experiments, if any, remain unused. This observation motivates the use of the LR method to reuse the outputs from past designs in the current estimate by properly weighting them using likelihood ratios (Feng and Staum 2017). Let $h(\widetilde{Y}^{(j)})$ denote the simulation output of the jth replication from an experiment of design $\widetilde{x}$. Then the LR estimator that uses $h(\widetilde{Y}^{(1)}),\ldots,h(\widetilde{Y}^{(r)})$ to estimate (1) is given by

$$\widehat{\mu}_r^{LR}(x,\widetilde{x}) = \frac{1}{r}\sum_{j=1}^{r} h(\widetilde{Y}^{(j)}) \frac{f\left(\widetilde{Y}^{(j)};x\right)}{f\left(\widetilde{Y}^{(j)};\widetilde{x}\right)}. \tag{3}$$

Mathematically, the LR method is similar to importance sampling, but the two methods differ in their purposes: the LR method is often used to reuse outputs from other designs that are fixed, while importance sampling often tries to find an optimal design $\widetilde{x}$ to minimize the variance of $\widehat{\mu}_r^{LR}(x,\widetilde{x})$. Since $f(\cdot;x)$ and

$f(\cdot;\widetilde{x})$ are assumed to have a common support, one can show that $\widehat{\mu}_r^{LR}(x,\widetilde{x})$ is unbiased for any $x \in \mathscr{X}$:

$$\mathbb{E}_{\widetilde{x}}\left[\widehat{\mu}_r^{LR}(x,\widetilde{x})\right] = \frac{1}{r}\sum_{j=1}^{r}\int_{\mathscr{Y}} h(y)\frac{f(y;x)}{f(y;\widetilde{x})}f(y;\widetilde{x})\,dy = \mathbb{E}_x[h(Y)] = \mu(x).$$

For a given design x, the green simulation *individual likelihood ratio (ILR) estimator* of $\mu(x)$, given previous simulation experiments of designs $X_1,\ldots,X_n$, is

$$\widehat{\mu}_{n,r}^{ILR}(x) = \frac{1}{n}\sum_{k=1}^{n}\left[\frac{1}{r}\sum_{j=1}^{r} h\left(Y_k^{(j)}\right)\frac{f\left(Y_k^{(j)};x\right)}{f\left(Y_k^{(j)};X_k\right)}\right], \quad Y_k^{(j)} \text{ i.i.d.} \sim f(y;X_k),\ \forall j \text{ and } \forall k. \tag{4}$$

We can see from (4) that the ILR estimator is an average of LR estimators $\widehat{\mu}_r^{LR}(x,X_k)$ for $k = 1,\ldots,n$. While the ILR estimator reuses all past outputs, the number of operations needed to compute it grows linearly in the number of iterations since the numerators of the likelihood ratios must be recalculated at each iteration. In this article, we focus on the ILR estimator and refer to Hesterberg (1995), Hesterberg (1988), Owen and Zhou (2000), Veach and Guibas (1995), and Feng and Staum (2017) for other LR estimators that have been studied in the literature.

Among the six assumptions made in Feng and Staum (2017) to justify the use of the LR method in the repeated experiments context, we focus on an assumption that is usually violated by simulation optimization searches but can easily be overlooked. To facilitate further discussion, we paraphrase it using our notation:

Assumption 1 (Assumption (A2) in Feng and Staum (2017)) For any k, $k' \neq k$, j, and j', given X_k, $Y_k^{(j)}|X_k$ is conditionally independent of $X_{k'}$ and $Y_{k'}^{(j')}|X_{k'}$.

This assumption holds in the repeated experiments studied in Feng and Staum (2017), where the parameters X_k for $k = 1,2,\ldots$ follows a stochastic process that is independent of the stochasticity of the simulation experiments. In the context of simulation optimization, however, this assumption can be violated because the random sample in each simulation experiment affects the value of the next iterate. This dependence can have an adverse effect on the optimization problem, as we will demonstrate.

Assumption 1 enables the following proof of the unbiasedness of $\widehat{\mu}_{n,r}^{ILR}(x)$, conditional on the identities of the designs $X_1,\ldots,X_n$:

$$\begin{aligned}
\mathbb{E}[\widehat{\mu}_{n,r}^{ILR}(x)|X_1 = x_1,\ldots,X_n = x_n] &= \frac{1}{n}\sum_{k=1}^{n}\left[\frac{1}{r}\sum_{j=1}^{r}\mathbb{E}\left[h\left(Y_k^{(j)}\right)\frac{f\left(Y_k^{(j)};x\right)}{f\left(Y_k^{(j)};X_k\right)}\middle|X_1 = x_1,\ldots,X_n = x_n\right]\right] \\
&= \frac{1}{n}\sum_{k=1}^{n}\left[\frac{1}{r}\sum_{j=1}^{r}\mathbb{E}\left[h\left(Y_k^{(j)}\right)\frac{f\left(Y_k^{(j)};x\right)}{f\left(Y_k^{(j)};X_k\right)}\middle|X_k = x_k\right]\right] = \mu(x),
\end{aligned} \tag{5}$$

where the second equality holds only if Assumption 1 holds. Every term in the ILR estimator, i.e., every LR-weighted simulation output, is conditionally unbiased.

Besides reusing outputs to estimate the objective, the LR method can in some instances produce green simulation estimators of the gradient. Gradient estimation in simulation optimization and sensitivity analysis of stochastic simulation is an important research topic in its own right; general references include L'Ecuyer (1990) and Fu (2015). In this article, we consider the so-called likelihood ratio (LR) or score function (SF) gradient estimator due to its relation to LR estimators for the objective function value. Under certain technical conditions, one can show that based on (1):

$$\nabla_x\mu(x) = \nabla_x\left(\int_{\mathscr{Y}} h(y)f(y;x)\,dy\right) = \int_{\mathscr{Y}} h(y)\nabla_x f(y;x)\,dy = \int_{\mathscr{Y}}\left[h(y)\frac{f(y;x)}{f(y;\widetilde{x})}\nabla_x\log f(y;x)\right]f(y;\widetilde{x})\,dy.$$

Inspired by the ILR estimator (4), the green simulation *ILR gradient estimator* is given by

$$\widehat{\nabla\mu}_{n,r}^{ILR}(x) = \frac{1}{n}\sum_{k=1}^{n}\left[\frac{1}{r}\sum_{j=1}^{r} h\left(Y_k^{(j)}\right)\frac{f\left(Y_k^{(j)};x\right)}{f\left(Y_k^{(j)};X_k\right)}\nabla_x \log f(Y_k^{(j)};x)\right], \quad Y_k^{(j)} \text{ i.i.d.} \sim f(y;X_k)\ \forall j \text{ and } \forall k. \quad (6)$$

Due to the similarities between the ILR estimator (4) and the ILR gradient estimator (6), it is natural—or at least computationally efficient—to use them together in a gradient-based simulation optimization search. However, because of these similarities, the two estimators could be highly correlated, possibly leading to undesirable behavior in the search.

As alluded to above, in the majority of simulation optimization algorithms, the current design X_n is determined by the past random samples $Y_k^{(1)},\ldots,Y_k^{(r)}$ for $k=1,\ldots,n-1$. For example, in many gradient-based searches, simulation outputs generated from experiments of the current design are used to estimate the gradient, and the next design is found by moving in the opposite direction (for minimization problems). Therefore, knowing the current design X_n and previous design X_{n-1} reveals additional information about the previous random outputs $h(Y_{n-1}^{(1)}),\ldots,h(Y_{n-1}^{(r)})$ that were used to estimate the gradient at X_{n-1}. Consequently, for $j=1,\ldots,r$, the random sample $Y_{n-1}^{(j)}|\{X_{n-1},X_n\}$ follows a different distribution than $Y_{n-1}^{(j)}|X_{n-1}$, i.e., conditional on X_{n-1}, $Y_{n-1}^{(j)}|X_{n-1}$ is not conditionally independent of X_n. So Assumption 1 is violated.

To articulate this dependence, denote the search process that identifies the next design by

$$X_n = s(\boldsymbol{X}_{n-1}, h(\boldsymbol{Y}_{n-1})), \quad (7)$$

where $\boldsymbol{X}_{n-1} = \{X_1,\ldots,X_{n-1}\}$ and $h(\boldsymbol{Y}_{n-1}) = \{\{h(Y_1^{(1)}),\ldots,h(Y_1^{(r)})\},\ldots,\{h(Y_{n-1}^{(1)}),\ldots,h(Y_{n-1}^{(r)})\}\}$. In other words, $\boldsymbol{X}_{n-1}$ represents the past designs and $h(\boldsymbol{Y}_{n-1})$ represents the past simulation outputs. In many gradient-based searches, $s(\boldsymbol{X}_{n-1},h(\boldsymbol{Y}_{n-1})) = X_{n-1} - \alpha_{n-1}\widehat{\nabla\mu}(X_{n-1})$, where $\{\alpha_k : k=1,2,\ldots\}$ is a gain sequence and the gradient estimate $\widehat{\nabla\mu}(X_{n-1})$ is a function of the past outputs $h(\boldsymbol{Y}_{n-1})$. Although not explicitly accounted for in (7), $s(\cdot)$ may also take random inputs, for instance when the next design is determined stochastically. Examples include simulated annealing and model-based stochastic search methods; see, for example, Kirkpatrick et al. (1983), Van Laarhoven and Aarts (1987), Aarts and Korst (1989), Zhou and Chen (2013), and Hu (2015).

In the simulation optimization context, the conditional dependence introduced by the search process $s(\cdot)$ and the consequent violation of Assumption 1 imply that the second equality in (5) no longer holds. Therefore the ILR estimator of the objective, $\widehat{\mu}_{n,r}^{ILR}(x)$, is conditionally biased given the identities of the designs $X_1,\ldots,X_n$. For similar reasons, the ILR estimator of the gradient, $\widehat{\nabla\mu}_{n,r}^{ILR}(x)$, is also conditionally biased. In Section 3, we illustrate different adverse effects of using these biased green simulation estimators in simulation optimization problems.

3 NUMERICAL ILLUSTRATIONS

We present three numerical examples, in increasing order of complexity, to illustrate the adverse effects of naively applying green simulation ILR estimators in simulation optimization. The first example shows how conditional bias can appear in the ILR estimator of the objective after only two iterations of a search and how it could theoretically be eliminated. The second and third examples highlight how the trajectories of gradient-based searches behave when ILR estimators are used by the search. In these examples, we observe that the conditional bias of the ILR estimators of the objective and the gradient can be reduced by taking more replications at each iteration.

3.1 Green Simulation ILR Estimators in a Two-Iteration Search

We first consider two iterations of a simple search to demonstrate some of the deficiencies of ILR estimators when used in simulation optimization. This example focuses on the dependence among simulation outputs that is introduced by the search; discussions about the underlying optimization problem are omitted. To simplify the discussion that follows, we assume that the simulation logic returns the scalar input Y as its output, i.e., $h(Y) = Y$.

Suppose that the search starts at an initial design $x_1 = 1/2$ and generates $Y_1 \sim \text{Bernoulli}(x_1)$. Based on Y_1, the search chooses the second design to be either $x_2 = 1/3$ or $x_2 = 2/3$—using the decision rule illustrated in Figure 1—and then generates $Y_2 \sim \text{Bernoulli}(x_2)$.

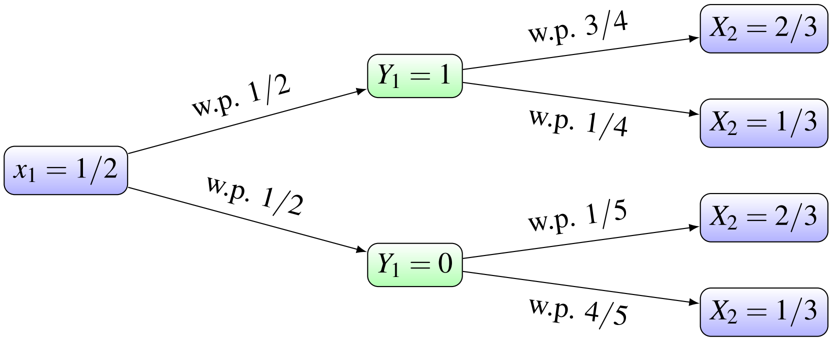


Figure 1: Decision rule for identifying second design X_2 based on Y_1.

For a given second design x_2, the true objective value is $\mu(x_2) = x_2$ and the green simulation ILR estimator of the objective is

$$\widehat{\mu}_{2,1}^{ILR}(x_2) = \frac{1}{2} Y_1 \frac{f(Y_1;x_2)}{f(Y_1;x_1)} + \frac{1}{2} Y_2,$$

where $f(y;x) = x^y(1-x)^{1-y}$. Given x_1 and x_2, the conditional expectation of the ILR estimator of the objective at the second design is

$$\begin{aligned}\mathbb{E}\left[\hat{\mu}_{2,1}^{ILR}(X_2) | X_1 = x_1, X_2 = x_2\right] &= \frac{1}{2}\mathbb{E}\left[Y_1 \frac{f(Y_1;x_2)}{f(Y_1;x_2)}\middle| X_1 = x_1, X_2 = x_2\right] + \frac{1}{2}\mathbb{E}[Y_2 | X_1 = x_1, X_2 = x_2] \\ &= \frac{1}{2}\left[\sum_{y_1 \in \{0,1\}} y_1 \frac{f(y_1;x_2)}{f(y_1;x_1)} g_1(y_1;x_1,x_2)\right] + \frac{1}{2}\mathbb{E}[Y_2 | X_2 = x_2]. \qquad (8)\end{aligned}$$

where $g_1(y_1;x_1,x_2) = \mathbb{P}(Y_1 = y_1 | X_1 = x_1, X_2 = x_2)$. Since x_1 is fixed, Bayes' rule yields

$$g_1(y_1;x_1,x_2) = \frac{\mathbb{P}(X_2 = x_2 | Y_1 = y_1, X_1 = x_1)\mathbb{P}(Y_1 = y_1 | X_1 = x_1)}{\sum_{y_1 \in \{0,1\}} \mathbb{P}(X_2 = x_2 | Y_1 = y_1, X_1 = x_1)\mathbb{P}(Y_1 = y_1 | X_1 = x_1)}. \qquad (9)$$

The probabilities shown in Figure 1 can be substituted into (9) to check that

$$g_1\left(0; \frac{1}{2}, \frac{1}{3}\right) = \frac{16}{21}, \quad g_1\left(1; \frac{1}{2}, \frac{1}{3}\right) = \frac{5}{21}, \quad g_1\left(0; \frac{1}{2}, \frac{2}{3}\right) = \frac{4}{19}, \quad \text{and} \quad g_1\left(1; \frac{1}{2}, \frac{2}{3}\right) = \frac{15}{19},$$

and hence

$$\mathbb{E}\left[\widehat{\mu}_{2,1}^{ILR}(X_2) \middle| X_1 = \frac{1}{2}, X_2 = \frac{1}{3}\right] = \frac{31}{126} < \frac{1}{3} \quad \text{and} \quad \mathbb{E}\left[\widehat{\mu}_{2,1}^{ILR}(X_2) \middle| X_1 = \frac{1}{2}, X_2 = \frac{2}{3}\right] = \frac{49}{57} > \frac{2}{3}.$$

The ILR estimator is therefore biased low when $X_2 = 1/3$ and biased high when $X_2 = 2/3$. Furthermore, the ILR estimator is also *unconditionally* biased in this example. From Figure 1, it can be seen that $X_2 = 1/3$ with probability $21/40$ and $X_2 = 2/3$ with probability $19/40$, and so

$$\mathbb{E}[\hat{\mu}_{2,1}^{ILR}(X_2)] = \left(\frac{31}{126}\right)\frac{21}{40} + \left(\frac{49}{57}\right)\frac{19}{40} = \frac{43}{80}, \quad \text{whereas} \quad \mathbb{E}[\mu(X_2)] = \left(\frac{1}{3}\right)\frac{21}{40} + \left(\frac{2}{3}\right)\frac{19}{40} = \frac{59}{120}.$$

From (8), it can be inferred that the conditional (and unconditional) bias of the ILR estimator would disappear if the denominator in the likelihood ratio were replaced with the conditional likelihood $g_1(y_1;x_1,x_2)$. Unfortunately, the derivation of this conditional likelihood in (9) implicitly requires knowledge of the distribution of the simulation outputs $h(Y)$. If this knowledge were available, it would be possible to estimate $\mu(x)$ via numerical integration and avoid the need to simulate altogether.

3.2 Minimizing a Constant Objective Function with Bernoulli Random Samples

We now study a more optimization-oriented example and explore how the conditional dependence and bias of the ILR estimators are manifested throughout a search.

Let $x \in \mathscr{X} = [0,1]$ be the design at any iteration and let $Y|x \sim \text{Bernoulli}(x)$ with support $\mathscr{Y} = \{0,1\}$. In addition, the simulation logic is a constant function given by $h(0) = h(1) = 1$. As a result, the objective function is also a constant, i.e., $\mu(x) = \mathbb{E}_x[h(Y)] = 1$ for all $x \in \mathscr{X}$. We consider minimizing $\mu(x)$ over $\mathscr{X}$ using a gradient-based search. Admittedly, this problem is trivial and does not require any (simulation) optimization. Nevertheless, it provides valuable insights about the use of green simulation ILR estimators in simulation optimization.

Since the simulation logic is constant, the SMC estimator for the objective is always accurate, i.e., unbiased with zero variance. Furthermore, any gradient estimate based on finite differences or infinitesimal perturbation analysis (IPA) will be zero with probability 1, hence the iterates will not move away from the initial design. In contrast, the likelihood ratios in the ILR estimators of the objective and the gradient depend on the random outputs at each iteration, resulting in non-zero variance for both estimators. As we will see, using ILR estimators in a gradient-based search affects the search's behavior in surprising ways.

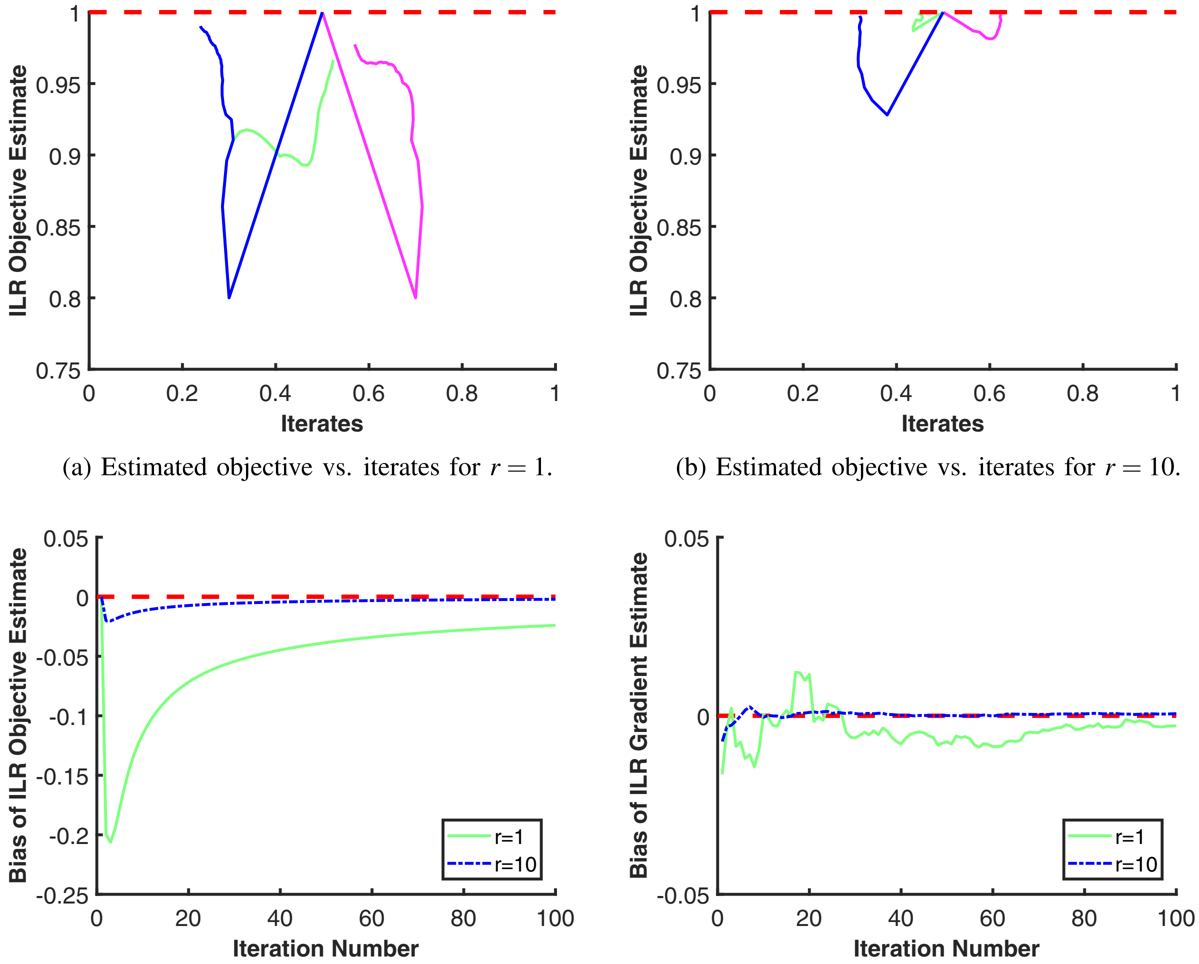
For any $x \in \mathscr{X}$, the probability mass function (pmf) of the Bernoulli distribution is $f(y;x) = x^y(1-x)^{1-y}$ for $y \in \mathscr{Y}$. The likelihood ratio is then $\frac{f(y;x)}{f(y;\tilde{x})} = \frac{x}{\tilde{x}}\mathbf{1}\{y=1\} + \frac{1-x}{1-\tilde{x}}\mathbf{1}\{y=0\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. In addition, the score function is $\nabla_x \log f(y;x) = \frac{\mathbf{1}\{y=1\}}{x} - \frac{\mathbf{1}\{y=0\}}{1-x}$. As a result, the green simulation ILR estimators for $\mu(x)$ and its gradient are given by

$$\widehat{\mu}_{n,r}^{ILR}(x) = \frac{1}{nr}\sum_{k=1}^{n}\left[\sum_{j:Y_k^{(j)}=1}\frac{x}{X_k} + \sum_{j:Y_k^{(j)}=0}\frac{1-x}{1-X_k}\right], \quad \widehat{\nabla\mu}_{n,r}^{ILR}(x) = \frac{1}{nr}\sum_{k=1}^{n}\left[\sum_{j:Y_k^{(j)}=1}\frac{1}{X_k} - \sum_{j:Y_k^{(j)}=0}\frac{1}{1-X_k}\right]. \quad (10)$$

For illustration, we test a stochastic approximation algorithm and initialize it in the center of the design space, i.e., $X_1 = 1/2$ with probability 1. At the kth iterate X_k, the objective value is estimated by $\widehat{\mu}_{k,r}^{ILR}(X_k)$ and the next design is $X_{k+1} = \max\{\min\{X_k - \alpha_k \widehat{\nabla\mu}_{k,r}^{ILR}(X_k), 1\}, 0\}$, where α_k is the step size or learning rate at each iteration; we consider a sequence of diminishing step sizes $\alpha_k = 0.1/k$ in this example.

We first run the above algorithm with 1 replication per iteration for 100 iterations. Figure 2a depicts the *estimated* objective value on the vertical axis and the iterates on the horizontal axis for three macroreplications of the algorithm. That is, the horizontal changes on a given curve describe the trajectory of the search path while the vertical changes show how the ILR estimate of the objective at the current iterate changes throughout the algorithm. The search trajectories shown in Figure 2a tend to be fairly smooth because the ILR estimate of the gradient only changes by a single term from iteration to iteration. In contrast, if LR/SF gradient estimates based on outputs at only the current design were used in a stochastic approximation algorithm, the search trajectory would be expected to oscillate back and forth from iteration to iteration.

Another important observation from Figure 2a is that the ILR estimator underestimates the objective throughout the algorithm, with the negative bias being most significant during the first few iterations. This



(a) Estimated objective vs. iterates for $r = 1$.

(b) Estimated objective vs. iterates for $r = 10$.

(c) Bias in objective vs. iteration.

(d) Bias in gradient vs. iteration.

Figure 2: Minimizing a constant objective function using stochastic approximation with ILR estimators.

negative bias can be most easily understood by considering the first two iterations of the algorithm. From (10), $\widehat{\nabla\mu}_{1,1}^{ILR}(X_1)$ is positive when $Y_1 = 1$ and negative when $Y_1 = 0$. A positive estimate of the gradient at the first design means that the second design will be to the left of X_1, hence the term X_2/X_1 appearing in $\widehat{\mu}_{2,1}^{ILR}(X_2)$ will be less than 1. Therefore the ILR estimator of the objective at this value of X_2 will be biased low. Similar reasoning shows that the ILR estimator of the objective is also biased low when the gradient estimate at the first design is negative.

We run the same experiment with 10 replications per iteration and plot the estimated objective function value versus the iterates for three macroreplications in Figure 2b. Again, the ILR estimator underestimates the objective function value, especially early in the algorithm, and the search trajectories are smooth.

We further run 10,000 macroreplications of the algorithm for $r = 1$ and $r = 10$ to assess the average bias of the ILR estimator of the objective, as depicted in Figure 2c. The average bias plotted in Figure 2c is the *unconditional* bias, as opposed to the conditional bias—given the identities of the designs—discussed in Section 2. As suggested by the curves in Figures 2a and 2b, the negative bias is most significant in the first few iterations and persists for over 100 iterations. Figure 2c also shows that the magnitude of the bias is smaller for $r = 10$ than for $r = 1$. This suggests that increasing the number of replications per iteration is a possible remedy for reducing, but not eliminating, the bias of the ILR estimator of the objective.

The reason that the bias can be reduced by increasing the number of replications can be seen by again considering the first two iterations. For $r = 1$, a single output from the first design appears in $\widehat{\mu}_{2,1}^{ILR}(X_2)$ as either X_2/X_1 or $(1-X_2)/(1-X_1)$, both of are less than 1, conditional on X_2. However, when $r = 10$, there are 10 terms of the form X_2/X_1 or $(1-X_2)/(1-X_1)$, the average of which is now closer to 1, conditional on X_2. In other words, with more replications per iteration, the average of the weighted outputs from the past designs tends to be less extreme.

We also plot the average bias of the ILR estimator of the gradient for $r = 1$ and $r = 10$ in Figure 2d. Given the symmetry of the problem, the ILR gradient estimate is likely unconditionally unbiased in this example, as indicated by the closeness of the two curves to the true value of zero.

3.3 Minimizing a Quadratic Objective Function with Normally Distributed Random Samples

In this example, we consider a simple yet meaningful simulation optimization problem that involves minimizing a quadratic objective function with normally distributed random samples. This example is intended to resemble the situation in which a search is approaching the local optimum of a smooth objective function. We observe that the conditional dependence and bias of the ILR estimators of the gradient can inhibit the convergence of a gradient-based search.

For any design $x \in \mathbb{R}$, the corresponding random sample is normally distributed with mean x and known fixed variance σ^2, i.e., $Y|x \sim N(x, \sigma^2)$. The simulation logic $h : \mathscr{Y} \to \mathbb{R}$ is the quadratic function $h(y) = y^2$ and the objective function is $\mu(x) = \mathbb{E}[h(Y)|x] = \mathbb{E}[Y^2|x] = \sigma^2 + x^2$ with a global minimizer $x^* = 0$. We consider solving this problem with a gradient-based algorithm, by simulating random samples $Y_n^{(1)}, \ldots, Y_n^{(r)}$ at each iterate X_n and then estimating the objective and gradient at the current iterate via the ILR estimators.

For a sampling design $\widetilde{x}$ and target design x, the likelihood ratio is $f(y;x)/f(y;\widetilde{x}) = \exp\left(\frac{\widetilde{x}^2 - 2y(\widetilde{x}-x) - x^2}{2\sigma^2}\right)$. Given the random samples $Y_k^{(j)}$, $j = 1, \ldots, r$, at designs $X_1, \ldots, X_n$, the ILR estimators for the objective and the gradient at a design x are

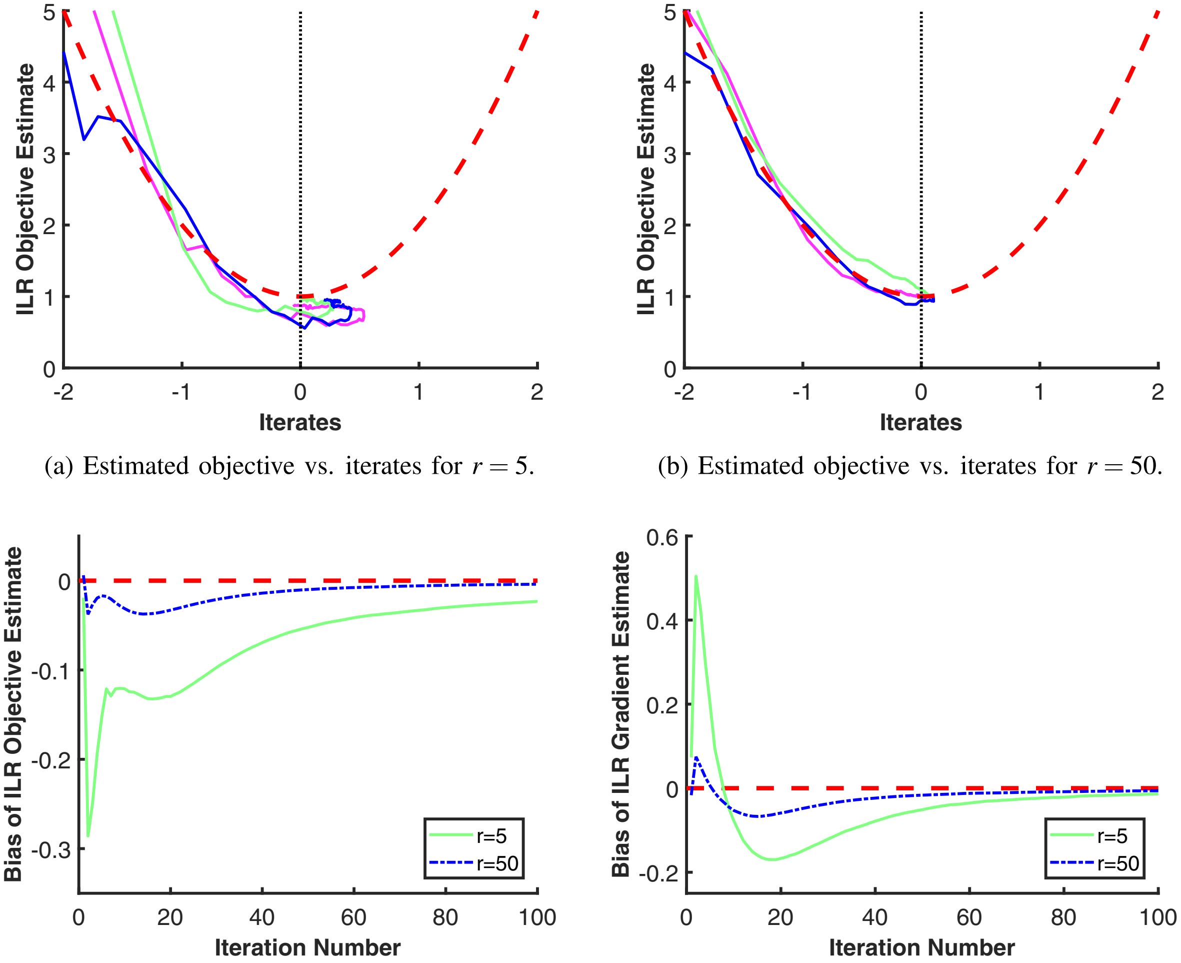
$$\widehat{\mu}_{n,r}^{ILR}(x) = \frac{1}{nr} \sum_{k=1}^{n} \sum_{j=1}^{r} \left(Y_k^{(j)}\right)^2 \exp\left(\frac{X_k^2 - 2Y_k^{(j)}(X_k - x) - x^2}{2\sigma^2}\right), \text{ and}$$

$$\widehat{\nabla\mu}_{n,r}^{ILR}(x) = \frac{1}{nr} \sum_{k=1}^{n} \sum_{j=1}^{r} \left(Y_k^{(j)}\right)^2 \exp\left(\frac{x_k^2 - 2Y_k^{(j)}(X_k - x) - x^2}{2\sigma^2}\right) \left(\frac{Y_k^{(j)} - x}{\sigma^2}\right). \tag{11}$$

At the kth iterate X_k, the next design is $X_{k+1} = X_k - \alpha_k \widehat{\nabla\mu}_{k,r}^{ILR}(X_k)$, where a constant step size $\alpha_k = 0.1$ is used in this example. A reason for using constant step size in this example is to examine whether the ILR gradient estimator accurately estimates the zero gradient at the minimizer $x^* = 0$. For simplicity, we use a fixed variance $\sigma^2 = 1$ and initial design $x_1 = -2$.

We first run the above algorithm with 5 replications per iteration for 100 iterations. Figure 3a shows the estimated objective value versus the iterates for three macroreplications. The search trajectories shown in Figure 3a exhibit a tendency to overshoot the minimizer before looping back. In addition, the ILR estimator appears to underestimate the objective function when the search trajectory first approaches the minimizer. We also run the algorithm with 50 replications per iteration and observe that these issues disappear, as shown in Figure 3b.

In Figure 3c we plot the average (unconditional) bias of the ILR estimator of the objective throughout the search based on 10,000 macroreplications. For both $r = 5$ and $r = 50$, there are two dips in the average bias: one during the first few iterations of the search and another during the iterations at which the search first approaches the minimizer. After the second dip, the bias of the ILR estimator of the objective shrinks. As was the case in Example 2, the magnitude of the bias decreases as the number of replications per iteration increases.



(a) Estimated objective vs. iterates for $r = 5$.

(b) Estimated objective vs. iterates for $r = 50$.

(c) Bias in objective vs. iteration.

(d) Bias in gradient vs. iteration

Figure 3: Minimizing a quadratic objective function using stochastic approximation with ILR estimators.

Figure 3d shows that the ILR gradient estimator initially overestimates the gradient, then underestimates it, before the bias eventually vanishes. For both $r = 5$ and $r = 50$, the peak and valley in Figure 3d coincide with the two valleys in Figure 3c. This suggests that due to the similarity of the ILR estimators of the objective and gradient in (11), the magnitudes of the biases of the two estimators are greatest at the same times, but the directions can be different. The dip in the bias of the ILR gradient estimator explains how the algorithm can overshoot the minimizer because the gradient estimate is negative instead of zero, indicating further improvement to the right.

4 CONCLUSIONS

We have discussed how in a simulation optimization search, past simulation outputs are conditionally dependent given the visited designs, an aspect that may be easily overlooked when trying to leverage green simulation. As a consequence, green simulation estimators of the objective function and gradient are conditionally biased, as demonstrated by our analysis of likelihood ratio estimators. Through several simple examples, we showed how these properties of green simulation estimators can affect the behavior of gradient-based searches by changing the shape of a typical search trajectory and potentially hindering the algorithm's convergence. It remains unknown just how significant these issues could be for practical simulation optimization problems or non-gradient-based searches. For this reason, we advise caution when considering whether to reuse past outputs within simulation optimization searches.

While common random numbers are frequently used in simulation optimization to reduce the variance of estimated differences in objective values, we believe that they would clash with green simulation estimators. In particular, common random numbers might further exaggerate the bias of the green LR estimators.

Another future research direction is to study how green simulation stochastic kriging estimators can be used in simulation optimization. In stochastic kriging, a metamodel is constructed from the outputs of a fixed set of designs, but in green simulation optimization, the set of designs grows over time as new designs are evaluated and the model is updated. The identities of new designs would depend on the metamodel fitted from the previous outputs, possibly adding a conditional bias to the metamodel.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grant DGE–1650441. We thank Shane Henderson for helpful discussions and three anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions.

REFERENCES

Aarts, E., and J. Korst. 1989. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Chichester, England: John Wiley & Sons Inc.

Ankenman, B., B. L. Nelson, and J. Staum. 2010. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58(2):371–382.

Barton, R. R., B. L. Nelson, and W. Xie. 2013. "Quantifying Input Uncertainty via Simulation Confidence Intervals". *INFORMS Journal on Computing* 26(1):74–87.

Bugallo, M. F., V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. 2017. "Adaptive Importance Sampling: The Past, the Present, and the Future". *IEEE Signal Processing Magazine* 34(4):60–79.

Eckman, D. J., and S. G. Henderson. 2018. "Reusing Search Data in Ranking and Selection: What Could Possibly Go Wrong?". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 28(3):18:1–18:15.

Feng, M., and J. Staum. 2017. "Green Simulation: Reusing the Output of Repeated Experiments". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 27(4):23:1–23:28.

Fu, M. 2015. "Stochastic Gradient Estimation". In *Handbook of Simulation Optimization*, edited by M. Fu, Chapter 5, 105–147. New York, New York: Springer.

Glasserman, P., and X. Xu. 2014. "Robust Risk Measurement and Model Risk". *Quantitative Finance* 14(1):29–58.

Hesterberg, T. 1988. *Advances in Importance Sampling*. Ph. D. thesis, Stanford University, Stanford, California.

Hesterberg, T. 1995. "Weighted Average Importance Sampling and Defensive Mixture Distributions". *Technometrics* 37(2):185–194.

Hu, J. 2015. "Model-Based Stochastic Search Methods". In *Handbook of Simulation Optimization*, edited by M. Fu, Chapter 12, 319–340. New York, New York: Springer.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. "Optimization by Simulated Annealing". *Science* 220(4598):671–680.

Kleijnen, J. P., and R. Y. Rubinstein. 1996. "Optimization and Sensitivity Analysis of Computer Simulation Models by the Score Function Method". *European Journal of Operational Research* 88(3):413–427.

L'Ecuyer, P. 1990. "A Unified View of the IPA, SF, and LR Gradient Estimation Techniques". *Management Science* 36(11):1364–1383.

L'Ecuyer, P. 1993. "Two Approaches for Estimating the Gradient in Functional Form". In *Proceedings of the 1993 Winter Simulation Conference*, edited by G. W. Evans et al., 338–346. Piscataway, New Jersey: IEEE.

Mandel, T., Y. E. Liu, S. Levine, E. Brunskill, and Z. Popović. 2014. "Offline Policy Evaluation Across Representations with Applications to Educational Games". In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, edited by A. Bazzan et al., 1077–1084. Richland, South Carolina: International Foundation for Autonomous Agents and Multiagent Systems.

Medina, J. C., and A. A. Taflanidis. 2014. "Adaptive Importance Sampling for Optimization Under Uncertainty Problems". *Computer Methods in Applied Mechanics and Engineering* 279:133–162.

Needell, D., N. Srebro, and R. Ward. 2016. "Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz Algorithm". *Mathematical Programming* 155(1-2):549–573.

Owen, A., and Y. Zhou. 2000. "Safe and Effective Importance Sampling". *Journal of the American Statistical Association* 95(449):135–143.

Precup, D., R. Sutton, and S. Dasgupta. 2001. "Off-Policy Temporal-Difference Learning with Function Approximation". In *Proceedings of the Eighteenth International Conference on Machine Learning*, edited by C. E. Brodley and A. P. Danyluk, 417–424. San Francisco, California: Morgan Kaufmann Publishers, Inc.

Rubinstein, R. Y. 1992. "Sensitivity Analysis of Discrete Event Systems by the 'Push Out' Method". *Annals of Operations Research* 39(1):229–250.

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

Spall, J. C. 1998. "Implementation of Simultaneous Perturbation Algorithm for Stochastic Optimization". *IEEE Transactions on Aerospace and Electronic Systems* 34(3):817–823.

Van Laarhoven, P. J., and E. H. Aarts. 1987. "Simulated Annealing". In *Simulated Annealing: Theory and Applications*, 7–15. Dordrecht, Netherlands: Springer.

Veach, E., and L. J. Guibas. 1995. "Optimally Combining Sampling Techniques for Monte Carlo Rendering". In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, edited by S. G. Mair and R. Cook, 419–428. New York, New York: ACM.

Xie, W., B. L. Nelson, and R. R. Barton. 2014. "A Bayesian Framework for Quantifying Uncertainty in Stochastic Simulation". *Operations Research* 62(6):1439–1452.

Zhao, H., Z. Yue, Y. Liu, Z. Gao, and Y. Zhang. 2015. "An Efficient Reliability Method Combining Adaptive Importance Sampling and Kriging Metamodel". *Applied Mathematical Modelling* 39(7):1853–1866.

Zhou, E., and X. Chen. 2013. "Sequential Monte Carlo Simulated Annealing". *Journal of Global Optimization* 55(1):101–124.

AUTHOR BIOGRAPHIES

DAVID J. ECKMAN is a Ph.D. candidate in the School of Operations Research and Information Engineering at Cornell University. His research interests deal with simulation optimization with a focus on ranking-and-selection and exploratory multi-armed bandits. His email address is dje88@cornell.edu and his website is https://people.orie.cornell.edu/dje88/.

M. BEN FENG is an assistant professor in actuarial science at the University of Waterloo. He earned his Ph.D. in the Department of Industrial Engineering and Management Sciences at Northwestern University. He is an Associate of the Society of Actuaries (ASA). His research interests include stochastic simulation design and analysis, optimization via simulation, nonlinear optimization, and financial and actuarial applications of simulation and optimization methodologies. His e-mail address is ben.feng@uwaterloo.ca and his website is http://www.math.uwaterloo.ca/~mbfeng/.