

IMPROVING HIT-AND-RUN WITH SINGLE OBSERVATIONS FOR CONTINUOUS SIMULATION OPTIMIZATION

Seksan Kiatsupaibul

Department of Statistics
Chulalongkorn University
Bangkok 10330, Thailand

Robert L. Smith

Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109, USA

Zelda B. Zabinsky

Department of Industrial and Systems Engineering
University of Washington
Seattle, WA 98195, USA

ABSTRACT

Many algorithms for continuous simulation optimization have been proposed, but the question of the number of replications at a specific point is always an issue. In this paper, instead of averaging replications of the objective function at a specific point (e.g., sample average), we average observed function evaluations from neighboring points. The Improving Hit-and-Run algorithm is modified to accommodate averaging in a ball of fixed radius, thus only sampling any point once. The computational results suggest an efficiency with single observations per sample point that simultaneously improves the estimation of the function value and samples closer to the optimum as the algorithm progresses.

1 INTRODUCTION

A computer simulation models a system on a computer by taking a set of system design parameters, say x , along with some random element U and producing an output g that can be viewed as a performance measure of the system. In that sense, a simulation can be described by a function $g : S \times \Omega \rightarrow \mathfrak{R}$ where S is the design space and $(\Omega, \mathcal{A}, \mathbb{P})$ is the probability space that the random element is defined on. We are interested in a case when S is a continuous subset of \mathfrak{R}^d , which is referred to as a continuous simulation. Since the performance produced by one simulation run is *noisy*, we restrict our attention to the expected performance:

$$f(x) = \mathbb{E}[g(x, U)].$$

A continuous simulation optimization problem is written as the following mathematical program

$$\min_{x \in S \subset \mathfrak{R}^d} f(x). \quad (1)$$

We assume the following.

Assumption 1. S is a compact body of \mathfrak{R}^d , i.e., S is bounded and equal to the closure of its interior.

Assumption 2. $f(x)$ is continuous on S .

Even though problem (1) is well defined, $f(x)$ cannot be observed directly. What can be observed is the noisy output $g(x, U)$ from the simulation. Solving program (1) then becomes a statistical estimation

problem as well as an optimization problem. Early approaches in continuous simulation optimization relied on gradient based methods that employed stochastic approximation techniques introduced by Robbins and Monro (1951) and Kiefer and Wolfowitz (1952). A comprehensive treatment of stochastic approximation can be found in, for example, Kushner and Yin (2003).

An alternative approach to continuous simulation optimization is a gradient-free stochastic search. A gradient-free stochastic search samples design points from the design space according to a sequence of probability distributions and performs function evaluations with respect to those sampled designs. Traditionally, it solves the noisy problem by resampling the previously sampled design points, that is, re-evaluating $g(x, U)$ several times at a sampled point x , and taking an average of several of these function evaluations as an estimate of $f(x)$. See, for example, Yakowitz et al. (2000) and Andradóttir and Prudius (2010). Pasupathy and Ghosh (2013) provides an excellent concise overview of the simulation optimization landscape.

In stochastic search for continuous optimization, an exception to a traditional resampling approach is the work by Baumert and Smith (2002) that evaluates only one function evaluation per sampled design. They put a ball around each sampled design point. An estimate of the objective function value at a design point is the average of the function evaluations of all design points within the ball. The noise in the function evaluation of a design point is effectively cancelled by the noises from the function evaluations of other designs surrounding it, removing the necessity of repeating several function evaluations at one sample point. The radius of the ball around each sample design point is controlled to shrink to zero to eliminate the estimation bias, resulting in the name *shrinking ball approach*.

When a simulation run is computationally expensive, this stochastic search approach with single observation per design point can have many advantages. However, constrained by the analysis techniques, the stochastic search by Baumert and Smith (2002) is based on a non-adaptive pure random search paradigm, which is notorious for its inefficiency in searching the design space for optimality. Inspired by the work, we aim at developing an efficient adaptive stochastic search algorithm that performs one function evaluation per sampled design that can solve program (1).

Andradóttir and Prudius (2010) provide a general framework for gradient-free stochastic search applying to continuous simulation optimization. However, their framework is still based on a resampling approach. Nevertheless, looking closely, one will find that Andradóttir and Prudius (2010)'s analysis can also expand to a non-resampling approach as long as the objective function evaluations are independent when conditional on the sampled designs. However, such a conditional independence property is too restrictive for an adaptive algorithm to satisfy because, for an adaptive method, it is essential to employ historical objective function evaluations in generating subsequent evaluations, resulting in dependence among the function evaluations.

Improving hit-and-run (IHR) is an efficient adaptive stochastic search for deterministic continuous optimization. IHR is relatively easy to implement. Given a current point, it generates a random direction vector (uniformly distributed on a unit sphere), and then generates a candidate point (uniformly distributed) along the line created by the intersection of the direction vector and the feasible region. If the objective function of the candidate point is better than that of the current point, the candidate point is accepted as the next point. Otherwise, the current point is used again. For more details and other forms, see Zabinsky et al. (1993), Zabinsky and Smith (2013). An advantage to IHR is its computational efficiency. Solving a quadratic program, it has a computational complexity of $O(d^{5/2})$ (Zabinsky et al. (1993)), and so is polynomial in dimension, on average.

In this paper, we adapt IHR to solve a continuous simulation optimization problem by incorporating the shrinking ball approach by Baumert and Smith (2002) to the algorithm. Specifically, we sample the design points from the design space according to IHR. We then estimate the objective function value of a design point by putting a ball around it and estimate its function value by taking an average of the function values of all sampled designed points within the ball. In contrast to Baumert and Smith (2002), instead of shrinking the ball, we keep the size of the ball fixed, which introduces systematic bias, in order to ensure

enough design points land in the ball for statistical estimation. We report numerical performance of the algorithm at different sizes of the ball.

We also contrast IHR for continuous simulation optimization, with its non-adaptive counterpart, namely hit-and-run (HR). Hit-and-run is a Markov chain Monte Carlo sampler that mimics the non-adaptive pure random search, whereas IHR is an adaptive algorithm that is analogous to pure adaptive search (see Zabinsky and Smith (2013) for an overview). The numerical results document the benefit we gain from an adaptive algorithm.

The paper is organized as follows. In Section 2, we compare IHR for continuous simulation optimization and its non-adaptive counterpart HR. Both the adaptive and non-adaptive versions estimate the objective function value using the sample points falling in a ball of fixed radius. In Section 3, we apply the two algorithms to a simulation quadratic program. We then report performance diagnostics of the two algorithms on the noisy problem and compare them to investigate the impact of ball radius and the adaptive versus non-adaptive algorithms. In Section 4, we discuss the results and future research.

2 IMPROVING HIT-AND-RUN FOR CONTINUOUS SIMULATION OPTIMIZATION

We now state the algorithms with single point observation in detail as Algorithm 1, and a non-adaptive counterpart of the algorithm as Algorithm 2. We use the abbreviation IHR for Algorithm 1, although it is a modification of the original Improving Hit-and-Run by averaging the objective function values over the points in the ball associated with current and candidate points. Similarly, we use the abbreviation HR for Algorithm 2, although it is a modification of Hit-and-Run using the averaging over the balls. Algorithm 2 is employed as a benchmark in the performance experiments of Algorithm 1 in Section 3. Both algorithms are derivative-free, and converge in probability to the global minimum of a deterministic problem under mild conditions.

Algorithm 1 (Improving Hit-and-Run for Continuous Simulation Optimization (IHR)). *The parameter of the algorithm is a ball radius r .*

Step 0: Let $n = 1$. Let x_1 be an interior point in S and evaluate $y_1 = g(x_1, u_1)$, where u_1 is an independent sample identically distributed as U . Also, set $S_1 = \{x_1\}$ and $T_1 = \{y_1\}$. Let $x^* = x_1$.

Step 1: Increase n to $n + 1$. Perform a hit-and-run step from x^* by sampling a direction v from the uniform distribution on the surface of d -dimensional hypersphere creating a line segment L where

$$L = \{x^* + \lambda v : \lambda \in \mathfrak{R}\} \cap S.$$

Sample x_n uniformly distributed on L . Sample u_n identically distributed as U and independent of $x_1, y_1, \dots, x_{n-1}, y_{n-1}, x_n$. Evaluate the objective function value

$$y_n = g(x_n, u_n).$$

Step 2: Update $S_n = S_{n-1} \cup \{x_n\}$ and $T_n = T_{n-1} \cup \{y_n\}$. For each $x \in S_n$, estimate the objective function value as

$$\hat{f}_n(x) = \frac{\sum_{\{k \leq n: x_k \in B(x, r)\}} y_k}{|\{k \leq n: x_k \in B(x, r)\}|}$$

where $B(x, r)$ is a ball of radius r centered at x . Estimate the optimal value as

$$\hat{f}_n^* = \min_{x \in S_n} \hat{f}_n(x).$$

Select x_n^* from $X_n^* = \{x \in S_n : \hat{f}_n(x) = \hat{f}_n^*\}$ as an optimal solution estimate.

Step 3: Go to Step 1.

Algorithm 2 (Hit-and-Run for Continuous Simulation Optimization (HR)). *The parameter of the algorithm is the ball radius r .*

Step 0: Let $n = 1$. Let x_1 be an interior point in S and evaluate $y_1 = g(x_1, u_1)$, where u_1 is an independent sample identically distributed as U . Also, set $S_1 = \{x_1\}$ and $T_1 = \{y_1\}$.

Step 1: Increase n to $n + 1$. Perform a hit-and-run step from x_{n-1} by sampling a direction v from the uniform distribution on the surface of d -dimensional hypersphere creating a line segment L where

$$L = \{x_{n-1} + \lambda v : \lambda \in \mathfrak{R}\} \cap S.$$

Sample x_n uniformly distributed on L . Sample u_n identically distributed as U and independent of $x_1, y_1, \dots, x_{n-1}, y_{n-1}, x_n$. Evaluate the objective function value

$$y_n = g(x_n, u_n).$$

Step 2: Update $S_n = S_{n-1} \cup \{x_n\}$ and $T_n = T_{n-1} \cup \{y_n\}$. For each $x \in S_n$, estimate the objective function value as

$$\hat{f}_n(x) = \frac{\sum_{\{k \leq n: x_k \in B(x, r)\}} y_k}{|\{k \leq n: x_k \in B(x, r)\}|}$$

where $B(x, r)$ is a ball of radius r centered at x . Estimate the optimal value as

$$\hat{f}_n^* = \min_{x \in S_n} \hat{f}_n(x).$$

Select x_n^* from $X_n^* = \{x \in S_n : \hat{f}_n(x) = \hat{f}_n^*\}$ as an optimal solution estimate.

Step 3: Go to Step 1.

Observe the difference between IHR and HR. IHR performs the hit-and-run step from x^* while HR performs the hit-and-run step from x_{n-1} . Therefore, IHR is an adaptive algorithm while HR is not. The implementation of the hit-and-run step for both algorithms follows Kiatsupaibul, Smith, and Zabinsky (2011). In Step 1, the direction vector is uniformly distributed on a unit hypersphere, and then x_n is uniformly distributed on the line segment formed from the direction vector going through x_{n-1} and the feasible region. In the numerical experiments, the feasible region is a hypersphere, hence the end points of the line segment are easily determined. If the feasible region is such that the end points are not easily determined, the approach in Kiatsupaibul et al. (2011) is to embed the feasible region in a hyperrectangle, and a uniform random point on the line segment within the feasible region can indirectly be sampled by performing rejection steps on the line segment within the hyperrectangle. Each time a point is rejected as being infeasible, the line segment within the hyperrectangle shrinks, until a feasible point is accepted. The accepted sample is the candidate point x_n uniformly distributed on L . For more details on implementation, see also Zabinsky et al. (1993), Zabinsky and Smith (2013).

3 EXPERIMENTS AND PERFORMANCE DIAGNOSTICS

We apply the two algorithms, parameterized by the radius $r = 0.1$ and $r = 0.01$, on the following spherical program

$$\begin{aligned} \min \quad & \mathbb{E} [\|x\|^2 + (1 + \|x\|^2)U] \\ \text{s.t.} \quad & \|x\| \leq 1 \end{aligned}$$

where $x \in \mathfrak{R}^{10}$, $U \sim \text{Uniform}[-0.1, 0.1]$ and $\|\cdot\|$ is the Euclidean norm. In each experiment with a fixed r , we apply the algorithm to solve the problem 30 times. The initial point for each of the 30 replications is generated according to a uniform distribution on the feasible region. Each time we run the algorithm for 10,000 iterations, and record the following four measurements over the iterations. For $n = 1, 2, \dots, 10,000$,

- the estimate of the optimal value:

$$\hat{f}_n^* = \min_{x \in \mathcal{S}_n} \hat{f}_n(x),$$

- the (true) objective function of a best candidate (optimal solution estimate):

$$f(x_n^*) = \|x_n^*\| \text{ where } x_n^* \in \text{Arg min}_{x \in \mathcal{S}_n} \hat{f}_n(x),$$

- the contributions to the best candidate (the counts of the sample points that contribute to an objective function estimate of the best candidate):

$$c_n^* = |\{k \leq n : x_k \in B(x_n^*, r)\}|$$

- the average noise of the best candidate:

$$\hat{e}_n^* = \frac{\sum_{\{k \leq n: x_k \in B(x_n^*, r)\}} (1 + \|x_k\|^2) U_k}{c_n^*}.$$

Finally, for $r = 0.1$ and 0.01 and for IHR and HR, for each of the four measurements we take the average over the 30 runs at each iteration, and plot them in Figures 1 and 2. Figure 1 shows the four performance diagnostics containing the average over 30 runs, for IHR and HR when $r = 0.1$. Figure 2 shows the performance diagnostics when $r = 0.01$.

4 DISCUSSION

In Panel (a) of Figure 1, when the radius parameters r is 0.1 , the estimates of the optimal value from IHR approach the true value (target) more quickly than those from HR. The optimal value estimates from the IHR and HR contain some biases, with the estimates from IHR containing negative biases. In Panel (b) of Figure 1, the objective functions of best candidates from IHR move closer to the optimal value than those from HR. In both IHR and HR cases, the objective functions of best candidates still contain biases, with those from HR containing a greater magnitude of bias.

In Figure 1, Panel (c) shows the contributions to the best candidate design. Observe that the contributions to the best candidate design for IHR are increasing over iteration. Thus the estimation is improving with iteration, as IHR brings the sampled point closer to the optimum. This suggests a synergy between exploration and estimation as IHR progresses. The reason is that, from a current design point x , the hit-and-run step samples the next point from the region close to x with higher probability than the region far away from x . In IHR, the current design point anchors the best candidate, so the algorithm returns to the best candidates often. As a result, with the hit-and-run steps, points surrounding the best candidates are sampled with higher probability and, hence, provide more contributions to the best candidates. In HR, the current point drifts around the feasible design space and no sampled design points get more refined than the others. For this reason, in Panel (d) of Figure 1, the average noises of the best candidates from IHR are smaller than those from HR in terms of absolute value. Consequently, the estimates of the objective function values from IHR are more precise, even though any single point is only observed once. These indicate the benefits of an adaptive algorithm over a non-adaptive one in terms of speed, accuracy and precision, under this single-observation-per-design setting. Note that, in Panel (d), the average noises of the best candidates are negative, causing negative bias in IHR optimal value estimates in Panel (a). This is understandable because we are minimizing the objective function. The estimates that contain negative errors tend to be lower in value and the corresponding sampled designs tend to get selected as the best candidates.

In Figure 2, when the radius parameters r is 0.01 , we observe the same trend between IHR and HR. However, in Panel (c) of Figure 2, the contributions of the best candidate designs are very low for both

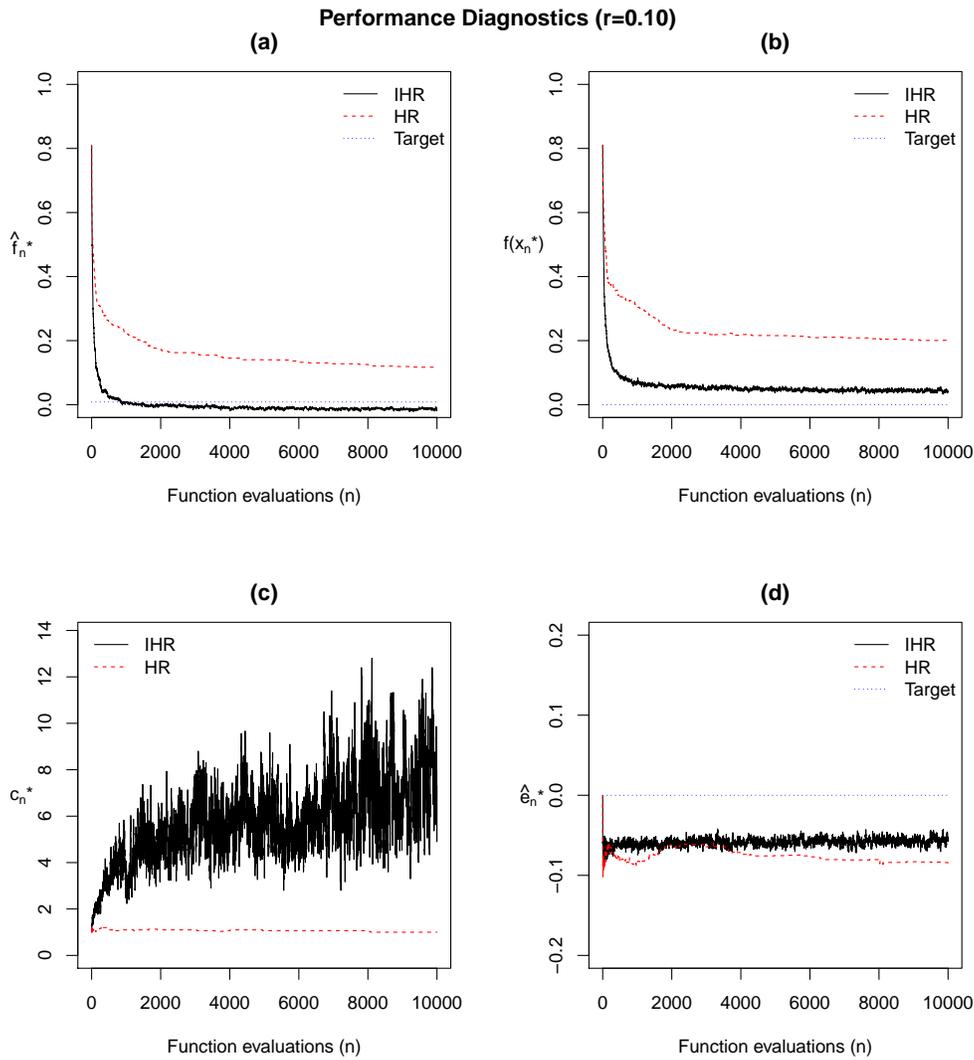


Figure 1: Performances diagnostics for IHR and HR when $r = 0.1$. Panels (a) and (b) exhibit the estimate of the objective function and the true objective function value at the best candidate (optimal solution estimate), respectively. Panels (c) and (d) show the contributions to the best candidates and the average noises of the best candidates as functions of objective function evaluations, respectively.

IHR and HR. That means the estimates of the optimal objective value are still noisy and the results are not very precise. Thus the fixed radius of the ball is an important parameter for future research.

In these experiments, one of the key properties of an efficient algorithm is the capability of an algorithm to effectively reduce the noise from function evaluations when there is only single observation of the objective value per sampled design. In Panel (d) of Figure 1, the average noises are decreasing over iterations. That means noise cancellation is actually in effect. Andradóttir and Prudius (2010) provide a framework that can prove the convergence to zero of the average noises in this conditional independence case of HR. However, the noise reduction is in fact observed in the case of IHR, where the function evaluations play a key role in sampling designs. Therefore, given sampled points x_1, x_2, \dots , unlike the case of HR, the noises from IHR are not independent. From the best of our knowledge, there is no exact

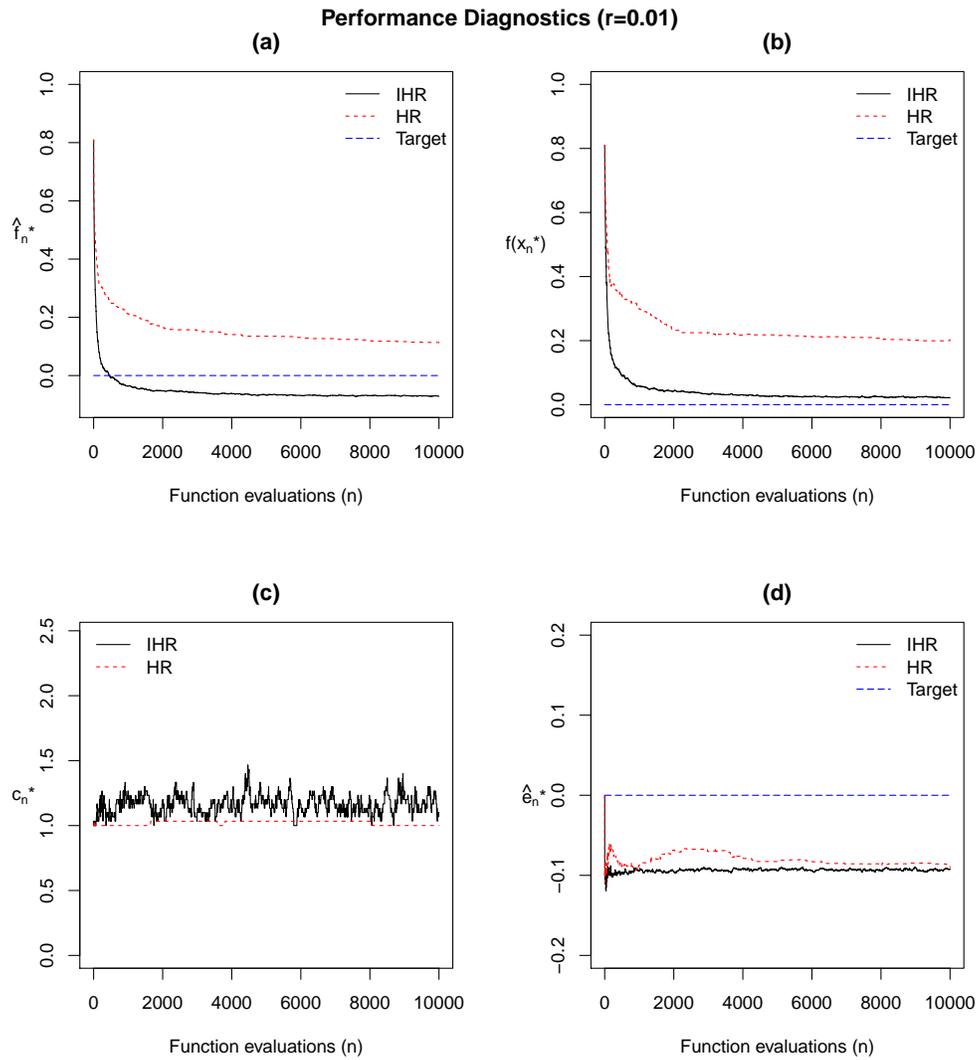


Figure 2: Performances diagnostics for IHR and HR when $r = 0.01$. Panels (a) and (b) exhibit the estimate of the objective function and the true objective function value at the best candidate (optimal solution estimate), respectively. Panels (c) and (d) show the contributions to the best candidates and the average noises of the best candidates as functions of objective function evaluations, respectively.

framework that this adaptive single-observation-per-design setting of IHR can fit into. Such a framework is required to fully understand the behavior of IHR shown in Figure 1 and 2.

In all cases, biases are observed both in the objective function value estimates and in the optimal solution estimates. Under some additional experiments (not exhibited here), we find that those biases are caused mainly by the size of ball covering each candidate design. To see this, imagine an extreme case when IHR is applied to a problem with zero noise but with non-zero ball radius. Even when the true optimal solution (the origin) is sampled as a candidate, its corresponding estimated objective function is still not the true optimal value but the average of the objective functions of design points within the balls, which will be greater than the true value, creating some bias of the optimal value estimate. To eliminate the biases, one might be tempted to shrink the ball as the algorithm progresses. However, as Figure 2 demonstrates, shrinking the ball too quickly may result in too few points to reduce the noise to zero. Therefore, there

exists a trade-off between the estimation bias and noise reduction mechanisms. An adaptive algorithm with single observation per design suggests a new paradigm for simultaneous exploration and refinement of noisy objective functions.

REFERENCES

- Andradóttir, S., and A. A. Prudius. 2010. “Adaptive random search for continuous simulation optimization”. *Naval Research Logistics* 57:583–604.
- Baumert, S., and R. L. Smith. 2002. “Pure random search for noisy objective functions”. Technical Report Technical report 01-03, University of Michigan.
- Kiatsupaibul, S., R. L. Smith, and Z. B. Zabinsky. 2011. “An analysis of a variation of hit-and-run for uniform sampling from general regions”. *ACM Transactions on Modeling and Computer Simulation* 21 (3): Article 16.
- Kiefer, J., and J. Wolfowitz. 1952. “Stochastic estimation of the maximum of a regression function”. *Annals of Mathematical Statistics* 23:462–466.
- Kushner, H., and G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer.
- Pasupathy, R., and S. Ghosh. 2013. “Simulation optimization: a concise overview and implementation guide”. In *Tutorial in Operations Research*, edited by H. Topaloglu, Number 7, 122–150. INFORMS.
- Robbins, H., and S. Monro. 1951. “A stochastic approximation method”. *Annals of Mathematical Statistics* 22:400–407.
- Yakowitz, S., P. L’Ecuyer, and F. Vázquez-Abad. 2000. “Global stochastic optimization with low-dispersion point sets”. *Operations Research* 48:939–950.
- Zabinsky, Z. B., and R. L. Smith. 2013. “Hit-and-Run Methods”. In *Encyclopedia of Operations Research & Management Science 3rd Edition*, edited by S. I. Gass and M. C. Fu. Springer Science+Business Media.
- Zabinsky, Z. B., R. L. Smith, J. F. McDonald, H. E. Romeijn, and D. E. Kaufman. 1993. “Improving hit-and-run for global optimization”. *Journal of Global Optimization* 3 (2): 171–192.

AUTHOR BIOGRAPHIES

SEKSAN KIATSUPAIBUL is an Associate Professor of the Department of Statistics at Chulalongkorn University, Bangkok, Thailand. His research interests include Markov chain Monte Carlo methods in global optimization, simulation optimization, and constrained statistical models. His email address is seksan@cbs.chula.ac.th.

ROBERT L. SMITH is the Altarum/ERIM Russell D. O’Neal Professor Emeritus of Engineering at the University of Michigan, Ann Arbor, Michigan. His research interests lie in the fields of infinite horizon and global optimization. He is an INFORMS Fellow and has served on the editorial boards of *Operations Research* and *Management Science*. He was Program Director for Operations Research at the National Science Foundation and Director of the Dynamic Systems Optimization Laboratory at the University of Michigan. His e-mail address is rlsmith@umich.edu.

ZELDA B. ZABINSKY is a Professor in the Department of Industrial and Systems Engineering at the University of Washington, with adjunct appointments in the departments of Electrical Engineering, Mechanical Engineering, and Civil and Environmental Engineering. She is an IIE Fellow. Professor Zabinsky’s research interests are in global optimization under uncertainty for complex systems. Her book, *Stochastic Adaptive Search in Global Optimization*, describes research on theory and practice of global optimization algorithms. Her email address is zelda@u.washington.edu.