Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

TOWARDS A FRAMEWORK FOR BLACK-BOX SIMULATION OPTIMIZATION

Sigurdur Ólafsson
Jumi Kim

Department of Industrial and Manufacturing Systems Engineering
Iowa State University
2019 Black Engineering
Ames, IA 50011, U.S.A.

ABSTRACT

Optimization using simulation has increased in popularity in recent years as more and more simulation packages now offer optimization features. At the same time, academic research in this area has grown, but more work is needed to bring results from the academic community to solve practical problems. This paper describes an effort in this direction. We present a framework that can be used to effectively solve large combinatorial-type simulation optimization problems in an automated or semi-automated manner.

1 INTRODUCTION

Simulation-based optimization has received considerable attention in recent years, but significant gaps appear to exist between theory and practice in the field. Although many advances have been made in algorithm development and analysis, they do not always seem to find their way readily into practice, where the use of relatively simple heuristic search methods prevails. Although such methods can often find quite good solutions, they are typically not designed specifically for simulation-based optimization and may therefore not account appropriately for the randomness (Boesel, Nelson, and Ishii 1999). With respect to methodological research, there have been many recent development, and comprehensive reviews of simulation-based optimization may be found in Jacobson and Schruben (1989), Andradóttir (1998), and Swisher et al. (2000).

One of the current difficulties in implementing many simulation optimization algorithms for practical problems appears to be that they can be quite complicated and require substantial knowledge on part of the user. Thus, as a step in the direction of bridging this gap, we present a framework for a black-box simulation-based optimization package that is intended for use by simulation practitioners. Accordingly, the required knowledge of optimization is kept at minimum, and the output is designed to mirror the output many simulation users are already familiar with when performing comparisons between a few alternative systems. On the other hand, the underlying algorithm remains globally convergent (almost surely), and the simulation noise is accounted for in every aspect of the framework. A prototype of a system designed according to this framework is currently in development. As the general problem of simulation-based optimization is extremely hard, the framework combines elements from a variety of methods that have been found to be effective in this context, including random search, adaptive sampling, and ranking-and-selection. To combine these elements, and simultaneously be able to make rigorous statements concerning the solution quality, the statistical selection based nested partitions (NP) algorithm of Ólafsson (1999) is employed.

The remainder of this paper is organized as follows. In Section 2 we review the NP algorithm used in our framework and point out the elements of this algorithm that must be automated or semi-automated for a black-box implementation. The key features of this algorithm are partitioning, random sampling, and local search, and automated implementations of these aspects are described in Section 3 - Section 5, respectively. Finally, Section 6 contains some concluding remarks.

2 THE NP ALGORITHM

In mathematical notation, we want to solve the problem

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

where Θ is a finite feasible region, and $f : \Theta \to \mathbf{R}$ is a performance function that is subject to noise. In other words, for any feasible point $\theta \in \Theta$, $f(\theta)$ cannot be evaluated analytically. Often $f(\theta)$ is an expectation of some random estimate of the performance of a complex stochastic system given a parameter θ, that is, $f(\theta) = E[L(\theta)]$. Here $L(\theta)$ is a random variable which depends on the parameter $\theta \in \Theta$. We assume that $L(\theta)$ is a discrete event simulation estimate

of the true performance, and refer to it as the sample performance.

We also make the assumption that the feasible region is n-dimensional, that is, $\Theta \subset \mathbf{R}^n$ such that

$$\Theta = \{\theta = (\theta_1, \theta_2, ..., \theta_n) : \theta_i \in \Theta_i\}, \tag{2}$$

where Θ_i is some finite set. Also we let $m(\theta_i) = |\Theta_i|$ be the number of values that the i-th dimensional variable can take.

In a recent paper, Shi and Ólafsson (2000) introduced an optimization method, the *nested partitions* (NP) method, for global optimization when the objective function is deterministic. In this context, the method has been found to be quite efficient for combinatorial optimization (Ólafsson and Shi 2000). Furthermore, as was first suggested by Shi and Ólafsson (1997), this method can also be applied to stochastic problems, where no analytical expression exists for the objective function and it must be evaluated using simulation. In Ólafsson (1999), this method is further improved by drawing on ideas from statistical sampling techniques that have proven useful in simulation in the past, namely ranking-and-selection methods. Thus, the resulting algorithm combines statistical sampling techniques traditionally used for comparing a few alternatives with a global optimization framework aimed at large-scale optimization problems.

The basic idea of the method is to systematically partition the feasible region into subsets and focus the computational effort in those subsets that are considered promising. The main components of the method are:

- **Partitioning**: at each iteration the feasible region is partitioned into subsets that cover the feasible region but concentrate the search in what is believed to be the most promising region.
- **Random sampling**: to evaluate each of the subsets, a random sample of solutions are obtained from each subset and used to estimate the performance of the region as a whole.
- **Local improvement**: to improve the estimate of each of the subset, the sample points can be used as starting points for a local search procedure that is constrained within the region.

This method can be understood as an optimization framework that combines adaptive global sampling with local heuristic search. It uses a flexible partitioning method to divide the design space into regions that can be analyzed individually and then aggregates the results from each region to determine how to continue the search, that is, to concentrate the computational effort. Thus, the NP method adaptively samples from the entire feasible region and concentrates the sampling effort by systematic partitioning of the design space.

The key aspects of determining how to implement the method is developing a partitioning method, deciding how much sample effort to use in each region, and how much local search effort to use in each iteration. These factors are of course interconnected. A high quality partition will lessen the need for sampling and local search, and in general increased effort along one of these dimensions decreases the need for the other two. Implementing the NP method can therefore be quite problem dependent, and in particular, partitioning schemes that have been devised in the past have drawn heavily on specific structure related to the application itself. This, however, requires substantial effort on part of the practitioner using the method, and in this paper we present a new framework for automating these decision, namely an intelligent partitioning method, guided random sampling, and guided local search.

In each iteration of the NP method it maintains what is called the most promising region, that is, a sub-region that is considered the most likely to contain the best solution. This most promising region is partitioned into a given number of sub-regions, these sub-regions and the surrounding region are sampled using random sampling, and the sampling information used to determine which region should be the most promising region in the next iteration.

As opposed to purely heuristic optimization methods, the NP method guarantees that the optimum solution is eventually found (Shi and Ólafsson 2000). Furthermore, Ólafsson (1999) uses standard ranking and selection procedures to develop an algorithm that allows us to specify a probability, say 90% or 95%, and terminate the algorithm when the probability that the correct solution has been selected exceeds this value. Here the correct solution is defined as a solution that has a performance within certain distance, that is an indifference zone, of the optimal performance. The key to this result is to guarantee in each iteration of the algorithm that the correct move is made with a minimum probability P^*, which can be calculated numerically from the following equation:

$$\Psi = \frac{(P^*)^n}{(1-P^*)^n + (P^*)^n}, \tag{3}$$

where Ψ is the probability of terminating correctly, that is, within the indifference zone. This type of termination of the algorithm when solution with an indifference zone has been found with a certain probability should be appealing to many practitioners, as this is widely known in the comparison of two or more systems.

3 INTELLIGENT PARTITIONING

The selected partition imposes a structure on the feasible region. When it is done in such a way that good solution as clustered together in the same subsets then those subsets are selected by the algorithm with relatively little effort. On the other end of the spectrum, if the optimal solution is surrounded by solutions of poor quality it is unlikely that the algorithm will move quickly towards those subsets. This can be made more rigorous by looking at how the minimum probability P^* of moving in the right direction can be guaranteed. We will see in Section 4 below that the amount of computational effort is directly proportional to the variance of the solution in each region. This implies we should attempt to cluster together similar solutions, that is, the diversity of the solutions, with respect to the simulation estimates of the objective function values, should be as small as possible.

Recall that a solution to the optimization problem can be written as $\theta = (\theta_1, \theta_2, ..., \theta_n)$. Our generic partitioning scheme will be to fix one of these n values at a time. Thus, for example, we can partition Θ into $m(\theta_i)$ subsets defined by

$$\sigma_j = \left\{\theta \in \Theta : \theta_i = \theta_{ij}\right\},$$

where $\theta_{ij} \in \Theta_i$ for $j = 1, 2, ..., m(\theta_i)$. The only decision to be made is the order in which these variables should be selected, that is, which variable should be fixed first, and so forth. This order defines the structure imposed on the feasible region by the search, and if it is done intelligently, that is in a way that distinguishes good parts of the feasible region from poor ones, the search is likely to proceed easily. Our aim is an automated procedure for intelligently selecting this order.

To develop a generic intelligent partitioning scheme, we focus on the idea of diversity, which is well known from information theory and its applications to areas such as machine learning (Mitchell 1987). For this purpose we need a way in which solutions can be classified as being the same from the point of view of our simulation optimization. A natural way to think about this is to say that two solutions are the same if there is in no statistically significant difference in their estimated performance. Thus, a subset of solutions where there are many solutions with statistically different performance is considered diverse, and vice versa. Using this definition, diverse subsets are clearly undesirable as it makes it difficult to determine which subset should be selected.

To use traditional diversity measures we need to classify each solution into one of several fixed categories. We do this in such a way that solutions in each category are statistically equivalent, and for any two categories there are at least one solution in each such that those two have statistically different estimated performance. The boundaries between the categories are determined from the simulation error. Now we cannot, of course, do this for every possible solution, but we can do it for a representative sample of solution, that is, we can use the following algorithm as a preprocessing step before the main iteration of the NP algorithm starts.

3.1 Intelligent Partition

Step 1. Start by using random sampling to obtain a set of M_0 sample solutions.

Step 2. Simulate the performance $L(\theta)$ of each one of these sample solutions, and record the average standard error $\bar{s}^2$.

Step 3. Construct $g(\bar{s}^2)$ intervals or categories for the sample solutions such that each solution in the same interval has statistically equivalent performance, but for two different intervals there are at least two solutions that make them distinguishable.

Step 4. Let $i = 1$.

Step 5. Fix $\theta_i = \theta_{ij},\ j = 1, 2, ..., m(\theta_i)$.

Step 6. Drawing on the idea of entropy (Mitchell 1997), calculate the following quantity for each dimension θ_i of the feasible region:

$$E(\theta_i) = \sum_{j=1}^{m(\theta_i)} \frac{M(\theta_{ij})}{M_0} \cdot \sum_{l=1}^{g(\bar{s}^2)} -p_{lj} \log_2 p_{lj}, \quad (4)$$

where p_{lj} is the proportion of samples with $\theta_i = \theta_{ij}$ and fall in category l, and

$$\begin{aligned} M(\theta_{ij}) &= \text{Number of samples for which} \\ &\quad\ \theta_i = \theta_{ij} \\ &= \sum_l p_{lj}. \end{aligned}$$

Step 7. If $i = n$, continue to Step 8; otherwise let $i = i + 1$ and go back to step 5.

Step 8. A high entropy value (4) indicates high diversity, so it is desirable to partition by fixing the lowest entropy dimensions first. Thus, we order the dimensions according to their entropy values

$$E(\theta_{[1]}) \le E(\theta_{[2]}) \le ... \le E(\theta_{[n]}), \quad (5)$$

and let this order determine the *intelligent partition*.

This method of intelligent partitioning is quite generic. It can be applied to any problem that has been formulated according to equation (2) above, which is often easy to satisfy in practice. Furthermore, it automates the process of

determining what is a good partition. The only undefined constants in the process is M_0, the number of sample solution used to build the partitioning. This can essentially be taken as a measure of how much computational time should be used to assure the quality of the partition. The higher this value the more reliable the partition. Thus, the user can simply indicate how much time is available and this will be mapped directly into a M_0 value.

From the discussion above it is clear that the NP method samples from the entire feasible region in an adaptive fashion, and concentrates the sampling effort by systematically partitioning the feasible region. Thus, in each iteration it selects a most promising region, that is, the subset that is considered the most likely to contain the global optimum. This selection can be considered a success if the region selected contains the true global optimum, and it would clearly be of practical interest if a minimum probability of success could be guaranteed in each iteration. Intelligent partitioning should intuitively increase the probability of correct selection, but it does no assure it in a rigorous manner. This, however, can be accomplished using the guided random sampling described in the next section.

4 GUIDED RANDOM SAMPLING

Determining the amount of random sampling is a key issue in NP and as its determination is difficult it is important to automate this process. Also note that when applying the NP method to a simulation optimization problem there are two sources of randomness that complicate the selection of the correct subset. First, there is a sampling error due to a relatively small sample being used to estimate the performance of an often large set. Secondly, the performance of each sample points is estimated using simulation and is hence noisy. It is important to observe that the former of these elements implies that the variation within a subset differs greatly from one region to the next. As an extreme case consider a singleton region that is being compared to the entire surrounding region. That is, a region containing only one solution being compared to a region containing all of the other solutions. Clearly the first source of randomness has been completely eliminated in the singleton region, whereas it probably accounts for almost all of the randomness in the surrounding region. This implies that to make better use of the sampling effort the number of sample points from each region should be variable and dependent on the variation within the region. The two-stage sampling procedure of Ólafsson (1999) addresses this by incorporating ranking-and-selection into the NP framework, and at the same time automates the process of determining the sampling effort. In particular, Ólafsson (1999) uses the classical Rinott's procedure for the statistical selection (Rinott, 1978). Here we provide an alternative based on the procedure of Matejcik and Nelson (1995), that our simulation experiments have found to be more efficient.

To state this approach rigorously, we let $\mathcal{D}_{ij}(k)$ be the i-th set of points selected from the region $\sigma_j(k)$ using a uniform random sampling procedure, $i \geq 1,\ j = 1, 2, ..., M+1$ in the k-th iteration. We let $N = |\mathcal{D}_{ij}(k)|$ denote the number of sample points, which is assumed to be constant. We let $\theta \in \mathcal{D}_{ij}(k)$ denote a point in this set and let $L(\theta)$ be a simulation estimate of the performance of this point. Then in the k-th iteration,

$$X_{ij}(k) = \min_{\theta \in \mathcal{D}_{ij}(k)} L(\theta),$$

is an estimate of the performance of the region σ_j, which we can now also refer to as the i-th system performance for the j-th system, $i \geq 1,\ j = 1, 2, ..., M+1$. The two-stage ranking-and-selection procedure first obtains n_0 such system estimates, and then uses that information to determine the total number N_j of system estimates needed from the the j-th system, that is, subregion $\sigma_j(k)$. This number is selected to be sufficiently large so that the correct subregion is selected with probability at least P^*, subject to an indifference zone of $\varepsilon > 0$.

More precisely, the procedure, which also takes advantage of the intelligent partitioning scheme, can be defined as follows:

4.1 Algorithm NP with Nelson-Matejcik Statistical Selection

Step 0. Apply algorithm **Intelligent Partition** to obtain a ranking:

$$E(\theta_{[1]}) \leq E(\theta_{[2]}) \leq ... \leq E(\theta_{[n]}), \quad (6)$$

Let $k = 0$ and $\sigma(0) = \Theta$.

Step 1. Given the current most promising region $\sigma(k)$, partition $\sigma(k)$ into

$$M = m(\theta_{[d(\sigma(k))]}) \quad (7)$$

subregions $\sigma_1(k), ..., \sigma_M(k)$ by fixing $\theta_{[d(\sigma(k))]}$, that is,

$$\sigma_j(k) = \left\{\theta \in \sigma(k) : \theta_{[d(\sigma(k))]} = \theta_{ij}\right\} \quad (8)$$

and aggregate the surrounding region

$$\sigma_{M+1}(k) = \Theta \setminus \sigma(k) \quad (9)$$

into one region (which can be empty).

Step 2. Let $i = 1$.

Step 3. Use uniform sampling to obtain a set $\mathcal{D}_{ij}(k)$ of N sample points from region $j = 1, 2, ..., M+1$.

Step 4. Use discrete event simulation of the system to obtain a sample performance $L(\theta)$ for every $\theta \in \mathcal{D}_{ij}(k)$ and estimate the performance of the region as

$$X_{ij}(k) = \min_{\theta \in \mathcal{D}_{ij}(k)} L(\theta), \qquad (10)$$

$j = 1, 2, ..., M+1$.

Step 5. If $i = n_0$ continue to Step 6. Otherwise let $i = i + 1$ and go back to Step 3.

Step 6. Estimate the first stage variance

$$S^2(k) = K_0 \sum_{j=1}^{N} \sum_{i=1}^{n_0} \left(X_{ij}(k) - \bar{X}_{i\cdot}(k) - \bar{X}_{\cdot j}(k) + \bar{X}_{\cdot\cdot}(k) \right)^2, \qquad (11)$$

where

$$K_0 = \frac{2}{(N-1)(n_0-1)},$$

$$\bar{X}_{i\cdot}(k) = \frac{1}{N} \sum_{j=1}^{N} X_{ij}(k),$$

$$\bar{X}_{\cdot j}(k) = \frac{1}{n_0} \sum_{i=1}^{n_0} X_{ij}(k),$$

$$\bar{X}_{\cdot\cdot}(k) = \frac{1}{N \cdot n_0} \sum_{j=1}^{N} \sum_{i=1}^{n_0} X_{ij}(k).$$

Step 7. Compute the total sample size

$$N(k) = \max \left\{ n_0, \left\lceil \frac{g^2 S^2(k)}{\epsilon^2} \right\rceil \right\}, \qquad (12)$$

where ϵ is the indifference zone and g is a constant that is determined by n_0 and the minimum probability P^* of correct selection.

Step 8. Obtain $N(k) - n_0$ more simulation estimates of the system performance as in Step 2 - Step 5 above, that is $(N(k) - n_0) \cdot N$ more sample points.

Step 9. Let the over all sample mean be the promising index for each region,

$$\hat{I}\left(\sigma_j(k)\right) = \bar{X}_j(k) = \frac{\sum_{i=1}^{N(k)} X_{ij}(k)}{N(k)}, \qquad (13)$$

$j = 1, 2, ..., M+1$.

Step 10. Select the index of the region with the best promising index.

$$\hat{j}_k \in \arg \min_{j=1,...,M+1} \hat{I}(\sigma_j). \qquad (14)$$

If more than one region is equally promising, the tie can be broken arbitrarily. If this index corresponds to a region that is a subregion of $\sigma(k)$, then let this be the most promising region in the next iteration. Otherwise, if the index corresponds to the surrounding region, backtrack to a larger region containing the current most promising region. That is, let

$$\sigma(k+1) = \begin{cases} \sigma_{\hat{i}_k}(k), & \text{if } \hat{i}_k < M+1, \\ s\left(\sigma(k)\right), & \text{otherwise.} \end{cases} \qquad (15)$$

Step 11. If $d(\sigma(k)) = n$, that is, all the dimensions have been fixed then stop; otherwise go back to Step 1.

An identical argument as the one in Ólafsson (1999) can be used to show that when this algorithm is applied the probability of terminating correctly is bounded by the following inequality:

$$\Psi = P\left[\hat{\sigma} = \sigma^*_{opt}\right] \geq \frac{(P^*)^n}{(1-P^*)^n + (P^*)^n},$$

where it is assumed that $P^* > \frac{1}{2}$, the indifference zone is selected sufficiently small, and that the sampling is from a normal population. Note that since we can prescribe the value of P^* then we can calculate this probability a priori for a given n, which is simply part of the simulation model. Thus, the only quantities the user must specify for the guided random sampling is some probability Ψ and an indifference zone ϵ that are such that we desire to have a solution with ϵ of the optimal performance with probability at least Ψ. As this is very similar to the traditional indifference zone approach to comparing a few alternatives using simulation, these type of specifications are familiar to many simulation users already.

5 GUIDED LOCAL SEARCH

The NP method can use local search to improve the estimates of the best solution in each region. The search uses the sample points as starting points and continues until some predetermine criteria is satisfied. This will in general lower the variance of the first stage samples, and thus reduce the

total number of random sample (and computational effort) needed in the second stage. However, determining how long the random search should be run is a challenging task, especially since the presence of randomness complicates analysis of the output of common random search methods. For example, in the deterministic context one might run a simple local search algorithm until it terminates at a local optimum, but for simulation-based optimization identifying such a local optimum is problematic in its own right.

In general, any local search method can be incorporated into the NP method, taking each of the points in the sample point set $\mathcal{D}$ as a starting point. Past applications have had good success with simple but highly specialized heuristics. However, our objective here is a black-box framework that can be applied without extensive knowledge of what specialized optimization algorithms might perform well for a particular application. Furthermore, as the NP algorithm already starts with a set of points, using a genetic algorithm (GA) is a natural companion and we will do so in our framework. In this context, the initial set of random samples is the initial population of the genetic algorithm, and the final population of the genetic algorithm is the set of points used by the NP method to calculate the performance measures for each region. Thus, in the algorithm above (Section 4), a GA step would be inserted in between Step 3, which obtains the initial population, and Step 4, which calculates the final performance measures.

A generic GA implementation is quite straightforward for the problem as it is formulated in Section 2, namely satisfying equation (2) above. We let $\theta = (\theta_1, \theta_2, ...\theta_n)$ be the string that the cross-over and mutation operations modify, and feasibility is always assured as we assume that any combination of values is acceptable. Cross-over and mutation are not affected by the randomness or noise in the performance measure, however, the selection of solutions into the next population, that is, the survival, does depend on the estimated objective function values. It may thus be appropriate to control the simulation run lengths accordingly as discussed in Boesel, Nelson, and Ishii (1999). Finally, note that there clearly is an interaction between the amount of local search, that is number of GA generations, and the amount of random sampling. As the GA is allowed to run longer, the variance (11) should be expected to decrease, implying that less computational effort is dictated by equation (12) for the second stage sampling. Thus, although there will in general exist some optimal balance that is problem dependent, insufficient GA run lengths will be automatically compensated for by increased effort in the second stage, and vice versa.

6 CONCLUSIONS

This paper proposes a simulation-based optimization framework that can be adapted to a wide variety of practical problems, and is automated with capabilities of incorporating special knowledge of the user when it exists. The main components of the framework are:

- Intelligent partitioning;
- Guided random sampling; and
- Guided local search.

A prototype optimizer based on this framework is currently being developed. Using this framework, a practitioner can effectively utilize the NP method for simulation-based optimization and draw on the power of local search through a genetic algorithm, while still maintaining a global convergence property and the capability of making rigorous statements about the quality of the solution obtained.

This framework is simple to implement effectively as the user only needs to formulate the problem and set a few constants, such as n_0 and N for the intelligent partitioning scheme. However, exact selection of such constants is not critical to the success of the method, and work is underway to determine how default values can be set in a robust manner.

REFERENCES

Andradóttir, S. 1998. "A Review of Simulation Optimization Techniques," in D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan (eds.), *Proceedings of the 1998 Winter Simulation Conference*, 151-158.

Boesel, J., B.L. Nelson, and N. Ishii. 1999. "A Framework for Simulation-Optimization Software," Preprint, Northwestern University.

Jacobson, S.H. and L.W. Schruben. 1989. "A Review of Techniques for Simulation Optimization," *Operations Research Letters*, 8, 1-9.

Mitchell, T.M. 1987. *Machine Learning,* McGraw-Hill, Boston, MA.

Matejcik, F.J. and B.L. Nelson. 1995. "Two-Stage Multiple Comparisons with the Best for Computer Simulation," *Operations Research*, 43, 633-640.

Ólafsson, S. 1999. "Iterative Ranking-and-Selection for Large-Scale Optimization," in P.A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans (eds.), Proceedings of the 1999 Winter Simulation Conference, 479-485.

Ólafsson, S. and L. Shi. 2000. "A Method for Scheduling in Parallel Manufacturing Systems with Flexible Resources," *IIE Transactions on Scheduling and Logistics*, 32: 135-146.

Rinott, Y. 1978. "On Two-Stage Selection Procedures and Related Probability-in-Equalities," *Communications in Statistics*, A7, 799-811.

Shi, L. and S. Ólafsson. 1997. "An Integrated Framework for Deterministic and Stochastic Optimization," in S.

Andradóttir, K.J. Healy, D.H. Withers, and B.L. Nelson (eds.), *Proceedings of the 1997 Winter Simulation Conference,* 358-365.

Shi, L. and S. Ólafsson. 2000. "Nested Partitions Method for Global Optimization," *Operations Research*, 48, 390-407.

Swisher, J.R., P.D. Hyden, S.H. Jacobson, and L.W. Schruben. 2000. "A Survey of Simulation Optimization Techniques and Procedures," in J.A. Jones, R.R. Barton, K. Kang, and P.A. Fishwick (eds.), *Proceedings of the 2000 Winter Simulation Conference,* 119-128.

AUTHOR BIOGRAPHIES

SIGURDUR ÓLAFSSON is an assistant professor in the Department of Industrial and Manufacturing Systems Engineering at Iowa State University. He received a B.S. in Mathematics from the University of Iceland in 1995, and an M.S. and a Ph.D. in Industrial Engineering from the University of Wisconsin - Madison in 1996 and 1998, respectively. His research interests include applied probability, stochastic optimization, and simulation. He is a member of IIE and INFORMS.

JUMI KIM is a Ph.D. candidate in the Department of Industrial and Manufacturing Systems Engineering at Iowa State University. She received a B.S. in Industrial Engineering from the Kangnung National University of Korea, and a M.S. in Industrial Engineering from the Seoul National University of Korea. Her research interests are in simulation optimization.