Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds.

DETERMINING THE OPTIMAL SAMPLING SET SIZE FOR RANDOM SEARCH

Chenbo Zhu

School of Management
Fudan University
Shanghai 200433, CHINA

Jie Xu
Chun-Hung Chen

Department of Systems Engineering and Operations Research
George Mason University
Fairfax, Virginia 22030, USA

Loo Hay Lee

Department of Industrial and Systems Engineering
National University of Singapore
SINGAPORE

Jianqiang Hu

School of Management
Fudan University
Shanghai 200433, CHINA

ABSTRACT

Random search is a core component of many well known simulation optimization algorithms such as nested partition and COMPASS. Given a fixed computation budget, a critical decision is how many solutions to sample from a search area, which directly determines the number of simulation replications for each solution assuming that each solution receives the same number of simulation replications. This is another instance of the exploration vs. exploitation tradeoff in simulation optimization. Modeling the performance profile of all solutions in the search area as a normal distribution, we propose a method to (approximately) optimally determine the size of the sampling set and the number of simulation replications and use numerical experiments to demonstrate its performance.

1 INTRODUCTION

Many existing stochastic simulation optimization algorithms belong to the class of convergent random search algorithms. Examples of globally convergent random search (GCRS) algorithms include the Model Reference Adaptive Search (MRAS) Algorithm of Hu, Fu, and Marcus (2007), and in particular its stochastic simulation counterpart SMRAS (Hu, Fu, and Marcus 2008), and the Nested Partition (NP) algorithms of Shi and Ólafsson (2000) and Pichitlamken and Nelson (2003). Examples of locally convergent random search (LCRS) algorithms include the COMPASS algorithm of Hong and Nelson (2006) and the AHA algorithm of Xu, Hong, and Nelson (2013).

A common step of GCRS and LCRS algorithms is to randomly sample a set of feasible solutions from a search area to be estimated via simulation. Most algorithms assign a fixed computing budget for

each iteration, typically measured in the number of total simulation replications assuming that simulating any feasible solution takes about the same amount of time. While procedures such as Optimal Computing Budget Allocation (OCBA) (Chen et al. 2000) may be used to adaptively allocate simulation replications to sampled solutions as did in the Industrial Strength COMPASS algorithm of Xu, Hong, and Nelson (2010), it is still common to assign the computing budget equally among sampled solutions. In this setting, a very important question is how many solutions should be sampled from the search area. Given a fixed simulation budget, this also determines the number of simulation replications each solution receives.

Obviously, sampling more solutions allow the algorithm to perform more exploration of the search area and increases the chance that a better solution will be sampled. However, the noise in simulation output means that the smaller the number of simulation replications allocated to each sampled solution, the smaller the probability that the best solution sampled is actually identified as the best. Therefore, given a fixed computation budget, there is a tradeoff between the size of the sampling set (exploration) and the number of simulation replications (exploitation) allocated to estimate the quality of each sampled solution. Achieving the right balance between exploration and exploitation in this regard requires optimally determining the random search sample size. Past research on GCRS and LCRS algorithms largely focused on the convergence property of the algorithms and did not consider how to determine the optimal sampling set size. This problem remains an open research question in simulation optimization research.

The main challenge in determining the optimal sampling set size is to identify an appropriate objective function. When the set of solutions to be estimated via simulation is given, Ranking & Selection algorithms such as the NSGS algorithm of Nelson et al. (2001) focus on determining the number of simulation replications allocated to each solution to guarantee the probability of correct selection (PCS), i.e., correctly selecting the actual best solution among the given set of solutions. OCBA also uses PCS as the objective function with a goal to optimally allocate simulation replications to maximize PCS.

In simulation optimization, the goal of random search is to maximize the quality of the solution selected by the algorithm as the best solution. The size of the sampling set has direct effect on the quality of the best solution in this set and thus PCS alone is not an appropriate objective function. Lee and Chew (2005) proposed to minimize the expected value of the sample best solution in a random search iteration for a minimization problem. However, the objective function was not analytically tractable and they used Monte Carlo simulation to demonstrate the effect of different sampling set sizes on the quality of the sample best solution. But they did not give an analytical method to determine the optimal sampling set size.

In this paper, for a given simulation budget, we present an analytical approach to (approximately) optimally determine the size of the sampling set and thus the number of simulation replications allocated to each solution assuming that each solution receives the same number of simulation replications. As to the best of our knowledge, this represents a first effort in simulation optimization research to optimally manage this inherent exploration and exploitation tradeoff in random search based simulation optimization algorithms. In the rest of the paper, we describe the problem in Section 2 and then present the optimization model and derive an (approximately) optimal solution to the optimization model in Section 3. We demonstrate the performance of our result via numerical experiments in Section 4 and present concluding remarks and future research directions in Section 5.

2 PROBLEM DESCRIPTION

We consider using random search on a general simulation optimization problem. Without loss of generality, we assume the objective is to minimize the expected value of a performance measure estimated via simulation. We use an equal simulation allocation rule, assigning the same number of simulation replications to each solution in the sampling set to be estimated via simulation. The computation budget is given *a priori* and fixed, measured in the total number of simulation replications and denoted as T. Such a setting is common in many GCRS and LCRS simulation optimization algorithms (Shi and Ólafsson 2000, Hong and Nelson 2006, Xu, Hong, and Nelson 2013). The size of the sampling set is denoted by k and the number of

replications for each solution is then given by $n = T/k$. For simplicity, we will treat k and n as continuous numbers in our analysis.

We model the sample mean of n IID simulation observations of a randomly sampled solution as

$$\tilde{J} = J + \omega. \tag{1}$$

In Equation (1), J denotes the true performance of a randomly sampled solution. We assume that the simulation noise ω has a common distribution across all sampled solutions. The sample average of the solution is $\tilde{J}$. We first randomly sample k solutions from the search space. Once all k sampled solutions have been simulated n times, we choose the solution with the smallest sample mean as the best solution found. However, due to the simulation noise, we might not be able to correctly select the best solution from the k sampled solutions as the best solution found. Therefore, a reasonable objective is to minimize the expected value of the true performance of the observed best solution with the best computation budget allocation decision (k,n). In what follows we introduce other notations that we will use throughout the paper:

$\tilde{J}_{[\tilde{1}]}$: the observed performance of the observed best solution.
$J_{[\tilde{1}]}$: the true performance of the observed best solution.
$z_{[1]}$: the first order statistic of a standard normal random variable.
$\phi(\cdot)$: the probability density function of the standard normal distribution.
$\Phi(\cdot)$: the cumulative distribution function of the standard normal distribution.

Then we have the following problem:

$$\begin{aligned} \min_{k,n} \quad & E[J_{[\tilde{1}]}] \\ s.t. \quad & kn = T. \end{aligned} \tag{2}$$

This problem is difficult to solve without any additional information on the distribution of J and ω. We assume that ω is normally distributed, which is a common assumption in literature. To facilitate analysis, we assume that J is also normally distributed. Such an assumption is reasonable when the performance of all solutions in the search area follows a S-curve, which is quite common in a large number of engineering and management problems (Ho, Zhao, and Jia 2007). We make these assumptions explicit below:

1. The true performance J follows a normal distribution with a mean of 0 and standard deviation σ_J.
2. The noise ω follows a normal distribution with a mean of 0 and standard deviation $\sigma_{\omega_0}/\sqrt{n}$.
3. J and ω are independent.

In the rest of the paper, unless explicitly written, we will not emphasize the dependence of σ_ω on n for notational simplicity.

3 MODEL DERIVATION AND ANALYTICAL SOLUTIONS

Based on the three assumptions in Section 2, the sample average of n IID simulations $\tilde{J}$ also follows a normal distribution:

$$\tilde{J} = J + \omega \sim N(0, \sigma_J^2 + \sigma_\omega^2) = N(0, \sigma_J^2 + \sigma_{\omega_0}^2/n),$$

and thus the conditional mean of the true performance given the sample mean value $\tilde{J} = v$ is given by (Tong 1989)

$$E\left[J|\tilde{J} = v\right] = \frac{\sigma_J^2}{\sigma_J^2 + \sigma_\omega^2} v.$$

Assume D is a set of alternative designs, notice that

$$\begin{aligned} f(J_i = u|\tilde{J}_i = v, \tilde{J}_j > v, j \in D \setminus i) &= \frac{f(\tilde{J}_i = v, \tilde{J}_j > v, j \in D \setminus i|J_i = u)f(J_i = u)}{f(\tilde{J}_i = v, \tilde{J}_j > v, j \in D \setminus i)} \\ &= \frac{f(\tilde{J}_i = v|J_i = u)f(J_i = u)f(\tilde{J}_j > v, j \in D \setminus i)}{f(\tilde{J}_i = v)f(\tilde{J}_j > v, j \in D \setminus i)} \\ &= f(J_i = u|\tilde{J}_i = v), \end{aligned}$$

which means

$$\begin{aligned} E\left[J_{[\tilde{1}]}|\tilde{J}_{[\tilde{1}]} = v\right] &= \int uf(J_i = u|\tilde{J}_i = v, \tilde{J}_j > v, j \in D \setminus i)du \\ &= \int uf(J_i = u|\tilde{J}_i = v)du \\ &= E\left[J|\tilde{J} = v\right]. \end{aligned}$$

Therefore, the expected true performance of the observed best can be expressed as follows:

$$E[J_{[\tilde{1}]}] = E_{\tilde{J}_{[\tilde{1}]}}\left\{E\left[J_{[\tilde{1}]}|\tilde{J}_{[\tilde{1}]} = v\right]\right\} = \frac{\sigma_J^2}{\sigma_J^2 + \sigma_\omega^2}E[\tilde{J}_{[\tilde{1}]}].$$

Because $E[\tilde{J}_{[\tilde{1}]}] = \sqrt{\sigma_J^2 + \sigma_\omega^2}E[z_{[1]}]$, the optimization formulation given in Equation (2) has the following form when both the true performance and simulation noise are normally distributed (Lee and Chew 2005):

$$\begin{aligned} \min_{k,n} \quad & E[J_{[\tilde{1}]}] = \frac{\sigma_J}{\sqrt{1 + \frac{\sigma_{\omega_0}^2}{n\sigma_J^2}}}E[z_{[1]}] \\ s.t. \quad & kn = T. \end{aligned} \tag{3}$$

Unfortunately, there does not exist a closed form for $E[z_{[1]}]$. Therefore, the problem stated in Equation (3) cannot be solved analytically. Lee and Chew (2005) used Monte Carlo simulation to estimate $E[z_{[1]}]$ and investigate the effect of using different sampling set sizes k on the true performance of the observed best solution. However, such a simulation-based procedure is computationally intensive and is not easy to implement in a simulation optimization algorithm.

Our goal is to obtain a closed-form (approximately) optimal solution that is easy to implement in a simulation optimization algorithm and yields insight into the exploration vs. exploitation tradeoff. To this end, we find a very accurate approximation for $E[z_{[1]}]$ and use the approximation formula to obtain a closed-form optimal solution.

Using the probability density function of $z_{[1]}$ (David and Nagaraja 2003), we have

$$E[z_{[1]}] = \int_{-\infty}^{\infty} xk[1 - \Phi(x)]^{k-1}\phi(x)dx. \tag{4}$$

Let $y = \Phi(x)$, then Equation (4) can be rewritten as

$$E[z_{[1]}] = k\int_0^1 \Phi^{-1}(y)(1-y)^{k-1}dy. \tag{5}$$

Let $G(k) = \int_0^1 \Phi^{-1}(y)(1-y)^{k-1}dy$. We plot $G(k)$ for $3 \le k \le 1000$, which covers all commonly used sampling set sizes in many GCRS and LCRS algorithms. We used the Matlab cftool and found that a power function

$(-a(k+b)^{-c}, a, c \in R^+, b \in R)$ provides the best fit of $G(k)$, with parameters $a = 1.594, b = 3.671, c = 0.8951$. The coefficient of determination (adjusted R^2) is 1.0000. Figure 1 plots both $G(k)$ and the approximation.

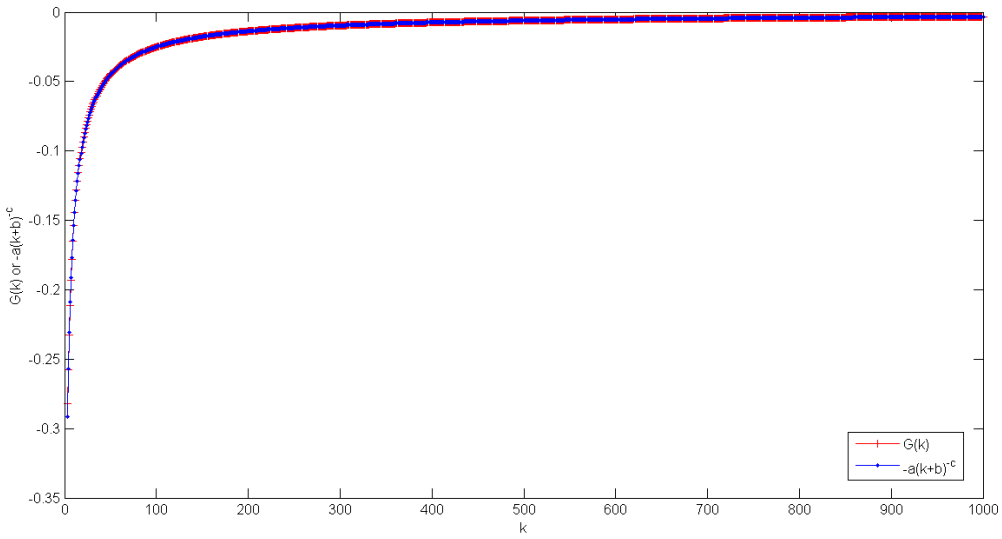

Figure 1: Comparisons of $G(k)$ and $-a(k+b)^{-c}$

Using the approximation function for $G(k)$ and eliminating the variable n from Equation (3), we reformulate the original optimization problem as

$$\min_k \quad E[J_{[\tilde{1}]}] = -a\sigma_J \frac{(k+b)^{-c}}{\sqrt{k^{-2} + \frac{\sigma^2_{\tilde{\omega}_0}}{T\sigma_f^2} k^{-1}}}, \tag{6}$$

where $a = 1.594, b = 3.671, c = 0.8951$.

Remark 1 Notice that a, b, and c are obtained for the case when $3 \leq k \leq 1000$. Their values will change slightly if the range of k changes. However, the values $a = 1.594, b = 3.671, c = 0.8951$ provides a quite robust approximation in a wide range of k, as can be seen from Table 1.

Table 1: Approximation function parameter robustness test

Range of k	$[3,50]$	$[3,100]$	$[3,500]$	$[3,1000]$	$[3,5000]$
Adjusted R^2	0.9993	0.9997	0.9998	1.0000	0.9999

We then minimize $E[J_{[\tilde{1}]}]$ given in Equation (6). The first order derivative of $E[J_{[\tilde{1}]}]$ with respect to k is given as follows:

$$\frac{dE[J_{[\tilde{1}]}]}{dk} = \frac{a\sigma_J (k+b)^{-c}}{\sqrt{k^{-2} + Rk^{-1}}} \left[\frac{c}{k+b} - \frac{2+Rk}{2(k+Rk^2)} \right], \tag{7}$$

where $R = \sigma^2_{\tilde{\omega}_0} / (T\sigma_f^2)$. Solving the first order condition, we obtain two roots. One root is positive and the other is negative, which we do not consider because k has to be positive. So the only meaningful solution to the first order condition is:

$$k^* = \frac{1-c}{(2c-1)R} + \frac{b}{2(2c-1)} + \frac{\sqrt{b^2R^2 + 4b(3c-1)R + 4(1-c)^2}}{2(2c-1)R}. \tag{8}$$

Let

$$\delta = 2\sqrt{\frac{2(2-c)}{3c}}$$

and

$$\eta = \frac{b(2+\delta)}{2\delta(c-1)+\delta^2(2c-1)}.$$

We can demonstrate that, if $T \geq \eta\sigma_{\omega_0}^2/\sigma_J^2$ (to ensure that k^* resides in $[2,\delta/R]$), $E[J_{[\tilde{1}]}]$ reaches its minimum in $[2,\delta/R]$ when k equals to k^*. Therefore, we have the following proposition:

Proposition 1 Assume the true performance $J \sim N(0,\sigma_J)$ and the noise $\omega \sim N(0,\sigma_\omega)$, given the total computation budget T, if $T \geq \eta\sigma_{\omega_0}^2/\sigma_J^2$, then $E[J_{[\tilde{1}]}]$ is (approximately) minimized when k equals to k^* (so k^* is the optimal sampling set size).

Remark 2 Notice that for most common cases where $\sigma_{\omega_0}/\sigma_J$ is not too big, $\eta\sigma_{\omega_0}^2/\sigma_J^2$ is relatively small and approximately equals to $2.74\sigma_{\omega_0}^2/\sigma_J^2$ when $3 \leq k \leq 1000$. In these cases, it is easy to verify that k^* resides in $[2,\delta/R]$ when T is reasonably large. Furthermore, $T \geq \eta\sigma_{\omega_0}^2/\sigma_J^2$ is only a sufficient condition for the optimality of k^*, and for all the scenarios we test in our numerical experiment in Section 4, $E[J_{[\tilde{1}]}]$ always achieves its minimum when $k = k^*$, even though the condition is not satisfied for some of them.

4 NUMERICAL EXPERIMENTS

In this section, we provide results of numerical experiments to validate the performance of the analytical (approximately) optimal sampling set size we propose. During one iteration of a typical random search based simulation optimization algorithm such as COMPASS (Hong and Nelson 2006) and AHA (Xu, Hong, and Nelson 2013), one needs to randomly sample a set of feasible solutions from a given search area, simulate, and then record the observed best solution. Our method can be applied to help determine the number of solutions to sample and the number of replications assigned to them for these algorithms. We assume σ_J and σ_{ω_0} are known and consider eighteen scenarios with their parameter values given as follows:

1. Total computation budget (T). Set T to two values: $\{100, 500\}$.
2. Standard deviation of the true performance J (σ_J). Set σ_J to three values: $\{0.25, 1, 2\}$.
3. Standard deviation of the noise ω_0, i.e., the noise in one replication. Set σ_{ω_0} to three values: $\{0.6, 1.2, 4.8\}$.

Let k_c^* denote the the optimal sampling set size obtained using Equation (8), and k_s^* denote the optimal sampling set size obtained by running Monte Carlo simulations and enumeration. For each experiment, we sample k solutions from the distribution of the true performance, i.e., $N(0,\sigma_J^2)$, and for each solution, we generate a noise from $N(0,\sigma_{\omega_0}/\sqrt{n})$ to represent the noise of the sample mean of n IID replications. The noise is added to the true performance value to produce the observed performance value of the solution. These k solutions are then ranked by their observed performance values, and the true performance value of the observed best is recorded.

We repeat this experiment 100000 times to estimate the expected true performance value of the observed best solution. For each pair of (k,n), we repeat the same procedure and k_s^* is determined by finding the minimum expected true performance of the observed best. Results obtained from Equation (8) and from simulation experiments are presented in Table 2. In the table, $E[J_{[\tilde{1}]}]_c^*$ and $E[J_{[\tilde{1}]}]_s^*$ are the estimated expected value of the true performance of solutions found using k_c^* and simulations, respectively. It is clear from the table that our closed-form formula obtains results very close to those of simulation experiments in various scenarios, which shows that our method is quite effective and is able to find approximately optimal solutions in a wide range of scenarios.

Table 2: Comparisons between analytical solutions and simulation results

Sce.	T	σ_J	σ_{ω_0}	k_c^*	$E[J_{[\tilde{1}]}]_c^*$	k_s^*	$E[J_{[\tilde{1}]}]_s^*$	Error
1	100	0.25	0.6	18	-0.3186	19	-0.3187	0.03%
2	100	0.25	1.2	10	-0.2109	9	-0.2121	0.57%
3	100	0.25	4.8	5	-0.0656	4	-0.0659	0.46%
4	100	1	0.6	100	-2.1492	100	-2.1492	0.00%
5	100	1	1.2	39	-1.7236	39	-1.7236	0.00%
6	100	1	4.8	10	-0.8473	8	-0.8494	0.25%
7	100	2	0.6	100	-4.8035	100	-4.8035	0.00%
8	100	2	1.2	100	-4.2983	100	-4.2983	0.00%
9	100	2	4.8	18	-2.5489	19	-2.5493	0.02%
10	500	0.25	0.6	45	-0.4491	45	-0.4491	0.00%
11	500	0.25	1.2	20	-0.3369	20	-0.3369	0.00%
12	500	0.25	4.8	7	-0.1369	7	-0.1369	0.00%
13	500	1	0.6	405	-2.6148	406	-2.6179	0.12%
14	500	1	1.2	123	-2.2211	123	-2.2211	0.00%
15	500	1	4.8	20	-1.3528	20	-1.3528	0.00%
16	500	2	0.6	500	-5.8207	500	-5.8207	0.00%
17	500	2	1.2	405	-5.2297	406	-5.2358	0.12%
18	500	2	4.8	45	-3.5844	48	-3.5909	0.18%

We also investigate the impact of different values of k on the performance $E[J_{[\tilde{1}]}]$. Given the total computation budget T, we set k to $T/5$, $\lfloor\sqrt{T}\rfloor$, and T. We then compare the true performance of the observed best solutions in these cases to what we have using k_c^* in Table 3. We also report the percentages of the improvement in performance in the table. It is clear that choosing the right sampling set size leads to significant increase in the performance of the random search algorithm.

Notice that in this paper, σ_J and σ_{ω_0} are assumed to be known when we calculate the optimal sampling set size. In practice, σ_J and σ_{ω_0} are not known *a priori*. But we can employ a sequential procedure to estimate σ_J and σ_{ω_0} similar to the sequential OCBA procedure; see Chapter 3 in Chen and Lee (2010).

5 CONCLUSION

The objective of this study is to determine the optimal sampling set size for random search when we have a fixed computation budget and each sampled solution receives the same number of simulation replications. Modeling the performance profile of all solutions in the search area as a normal distribution, we derive a formula to (approximately) optimally determine the size of the sampling set. Numerical experiments show that our formula is very accurate and either give the optimal sampling set sizes or sampling set sizes very close to the optimal ones.

In future study, we will apply our result to determine the optimal sampling set size for GCRS and LCRS simulation optimization algorithms such as NP (Shi and Ólafsson 2000) and AHA (Xu, Hong, and Nelson 2013). The immediate extension is to design a sequential procedure that allows the estimation of σ_J and σ_{ω_0} as the search progresses.

We will also study how to extend our results into the case of non-equal simulation replication allocation. Procedures such as OCBA (Chen et al. 2000) have been used in the simulation optimization solver ISC (Xu, Hong, and Nelson 2010) to increase the PCS using a fixed computation budget. However, the sampling set size was predetermined. How to jointly determine both the sampling set size and the adaptive allocation of simulation replications will be a major research task to be undertaken.

Table 3: Comparisons of different sampling set sizes

Sce.	T	σ_J	σ_{ω_0}	$E[J_{[\tilde{1}]}]$						
				$k=k_c^*$	$k=T/5$	diff.	$k=\lfloor\sqrt{T}\rfloor$	diff.	$k=T$	diff.
1	100	0.25	0.6	-0.3186	-0.3178	0.25%	-0.3053	4.17%	-0.2402	24.61%
2	100	0.25	1.2	-0.2109	-0.1971	6.54%	-0.2109	0.00%	-0.1276	39.50%
3	100	0.25	4.8	-0.0656	-0.0545	16.82%	-0.0618	5.78%	-0.0338	48.50%
4	100	1	0.6	-2.1492	-1.8035	16.08%	-1.5088	29.79%	-2.1492	0.00%
5	100	1	1.2	-1.7236	-1.6417	4.75%	-1.4350	16.75%	-1.6039	6.95%
6	100	1	4.8	-0.8473	-0.7875	7.06%	-0.8473	0.00%	-0.5103	39.78%
7	100	2	0.6	-4.8035	-3.7021	22.93%	-3.0582	36.33%	-4.8035	0.00%
8	100	2	1.2	-4.2983	-3.6062	16.10%	-3.0166	29.82%	-4.2983	0.00%
9	100	2	4.8	-2.5489	-2.5427	0.24%	-2.4427	4.16%	-1.9217	24.61%
10	500	0.25	0.6	-0.4491	-0.4279	4.70%	-0.4259	5.15%	-0.2925	34.86%
11	500	0.25	1.2	-0.3369	-0.2638	21.69%	-0.3364	0.15%	-0.1547	54.07%
12	500	0.25	4.8	-0.1369	-0.0736	46.27%	-0.1152	15.82%	-0.0380	72.24%
13	500	1	0.6	-2.6148	-2.4223	7.36%	-1.8935	27.59%	-2.6047	0.39%
14	500	1	1.2	-2.2211	-2.2081	0.59%	-1.8511	16.66%	-1.9432	12.51%
15	500	1	4.8	-1.3528	-1.0554	21.99%	-1.3486	0.31%	-0.6189	54.25%
16	500	2	0.6	-5.8207	-4.9704	14.61%	-3.8091	34.56%	-5.8207	0.00%
17	500	2	1.2	-5.2297	-4.8446	7.36%	-3.7870	27.59%	-5.2094	0.39%
18	500	2	4.8	-3.5844	-3.4235	4.49%	-3.4075	4.94%	-2.3400	34.72%

ACKNOWLEDGMENTS

This work is supported in part by the Award for Excellent Doctoral Student granted by ministry of Education of China 2010, by the National Science Foundation under Grant No. CMMI- 1233376, by the National Natural Science Foundation of China under Grants 71071040, 71028001, 70832002, 71061160506, by the Shanghai Science and Technology PuJiang Funds under Grant 09PJ1401500, and by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institution of Higher Learning.

REFERENCES

Chen, C., and L. Lee. 2010. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*. Singapore: World Scientific.

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

David, H. A., and H. N. Nagaraja. 2003. *Order Statistics*. Wiley.

Ho, Y.-C., Q.-C. Zhao, and Q.-S. Jia. 2007. *Ordinal Optimization*. New York: Springer.

Hong, L. J., and B. L. Nelson. 2006. "Discrete optimization via simulation using COMPASS". *Operations Research* 54:115–129.

Hu, J., M. Fu, and S. Marcus. 2007. "A model reference adaptive search method for global optimization". *Operations Research* 55:549–568.

Hu, J., M. Fu, and S. Marcus. 2008. "A model reference adapative search method for stochastic global optimization". *Communications in Information and Systems* 8:245–276.

Lee, L. H., and E. P. Chew. 2005. "Design Sampling and Replication Assignment under Fixed Computing Budget". *Journal of Systems Science and Systems Engineering* 14:289–307.

Nelson, B. L., J. Swann, D. Goldsman, and W. Song. 2001. "Simple procedures for selecting the best simulated system when the number of alternatives is large". *Operations Research* 49:950–963.

Pichitlamken, J., and B. L. Nelson. 2003. "A combined procedure for optimization via simulation". *ACM Transactions on Modeling and Computer Simulation* 13:155–179.

Shi, L., and S. Ólafsson. 2000. "Nested partitions method for stochastic optimization". *Methodology and Computing in Applied Probability* 2:271–291.

Tong, Y.-L. 1989. *The Multivariate Normal Distribution*. New York: Springer.

Xu, J., L. J. Hong, and B. L. Nelson. 2010. "Industrial Strength COMPASS: A Comprehensive Algorithm and Software for Optimization via Simulation". *ACM Transactions on Modeling and Computer Simulation* 20:3:1–3:29.

Xu, J., L. J. Hong, and B. L. Nelson. 2013. "An Adaptive Hyperbox Algorithm for High-Dimensional Discrete Optimization via Simulation Problems". *INFORMS Journal on Computing* 25:133–146.

AUTHOR BIOGRAPHIES

CHENBO ZHU is a PhD candidate in the Department of Management Science, School of Management, Fudan University. He received his B.S. degree in management science from Fudan University. His research interests include service science, revenue management, and simulation optimization. His email address is 091025002@fudan.edu.cn.

JIE XU is an assistant professor in the Department of Systems Engineering and Operations Research at George Mason University. His research interests include Monte Carlo simulation, stochastic optimization, computational intelligence, and applications in risk management and aviation. He is a member of INFORMS, IIE, and IEEE. His email address is jxu13@gmu.edu.

CHUN-HUNG CHEN is a Professor of Systems Engineering and Operations Research at George Mason University. He received his Ph.D. from Harvard University in 1994. His research interests are mainly in development of very efficient methodology for simulation and optimization and its applications. Dr. Chen has served as Co-Editor of the Proceedings of the 2002 Winter Simulation Conference and Program Co-Chair for 2007 Informs Simulation Society Workshop. He is currently an associate editor of *IEEE Transactions on Automation Science and Engineering*, departmental editor of *IIE Transactions*, and editorial board member of *International Journal of Simulation and Process Modeling*. His email address is cchen9@gmu.edu.

LOO HAY LEE is an Associate Professor and Deputy Head (Graduate Studies and Research) in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B.S. (Electrical Engineering) degree from the National Taiwan University in 1992 and his Ph.D. degree in 1997 from Harvard University. He is currently a senior member of IEEE, a member of INFORMS and ORSS. His research interests include simulation-based optimization, maritime logistics and supply chain systems. His email address is iseleelh@nus.edu.sg.

JIANQIANG HU is a Professor with the Department of Management Science, School of Management, Fudan University. Before joining Fudan University, he was an Associate Professor with the Department of Mechanical Engineering and the Division of Systems Engineering at Boston University. He received his B.S. degree in applied mathematics from Fudan University, China, and M.S. and Ph.D. degrees in applied mathematics from Harvard University. He is a managing editor of OR Transactions. His research interests include discrete-event stochastic systems, simulation, queuing network theory, stochastic control theory, with applications towards supply chain management, risk management in financial markets and derivatives, communication networks, and flexible manufacturing and production systems. He is a co-author of the book, Conditional Monte Carlo: Gradient Estimation and Optimization Applications (Kluwer Academic Publishers, 1997), which won the 1998 Outstanding Simulation Publication Award from INFORMS College on Simulation. His email address is hujq@fudan.edu.cn.