Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

SIMULTANEOUS SIMULATION EXPERIMENTS AND NESTED PARTITION FOR DISCRETE RESOURCE ALLOCATION IN SUPPLY CHAIN MANAGEMENT

Leyuan Shi
Dept. of Industrial Engineering
University of Wisconsin-Madison
Madison, WI 53706, U.S.A.

Chun-Hung Chen
Dept. of Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104, U.S.A.

Enver Yücesan
INSEAD
Technology Management Area
Fontainebleau, FRANCE

ABSTRACT

Discrete resource allocation is a common problem in supply chain management. However, stochastic discrete resource allocation problems are difficult to solve. In this paper, we propose a new algorithm for solving such difficult problems. The algorithm integrates the nested partitions method with an optimal computing budget allocation method. The resulting hybrid algorithm retains the global perspective of the nested partitions method and the efficient simultaneous simulation experiments of the optimal computing budget allocation. Numerical results demonstrate that the hybrid algorithm can be effectively used for a large-scale discrete resource allocation problem.

1 INTRODUCTION

Many resource allocation problems in supply chain management such as facility planning, job scheduling, buffer allocation, pollution control, and portfolio management can be modeled as stochastic discrete optimization problems. Owing to the complexity inherent in these systems, the search of optimal solutions can be a difficult task. Two key difficulties for solving the problem are: (1) the combinatorial explosion of alternatives normally leads to NP-hard optimization problems; (2) the lack of analytical expressions relating performance functions to solutions usually results in noise estimates of the performances. Recent methods proposed for this problem include: simulating annealing (Gelfand and Mitter 1989), the stochastic ruler method (Yan and Mukai 1993), the stochastic comparison method (Gong et al. 1992), ordinal optimization (Ho et al. 1992, Dai 1996, Cassandras et al. 1998), the stochastic branch-and-bound method (Norkin et al. 1996), the method of Andradottir (1995), the nested partitions method (Shi and Olafsson 1999), and the simulated entropy method (Rubinstein 1999).

In this paper, we develop a hybrid algorithm that integrates with the *nested partitions* (NP) method, and an efficient technique for simultaneous simulation experiments. The NP method is a randomized optimization method that has recently been developed for global optimization (Shi and Olafsson 1999). This method has been found to be promising for difficult combinatorial deterministic optimization problems (Shi et al. 1999). The NP method may be described as an adaptive sampling method that uses partitioning to concentrate the sampling effort in those subsets of feasible region that are considered the most promising. It combines global search through global sampling of the feasible region, and local search that is used to guide where the search should be concentrated.

In each iteration, the NP method needs to identify the most promising region by conducting a set of simultaneous simulation experiments. However, simulation can be both expensive and time consuming. In our hybrid approach, we apply our efficient technique to control the simultaneous simulation experiments. As a result, the simulation efficiency is significantly improved and the overall computation time for searching the optimal design is drastically reduced. Intuitively, to have a set of simultaneous simulation experiments, a larger portion of the computing budget should be allocated to those designs that are critical in the process of identifying good designs. In other words, a larger number of simulations must be conducted with those critical designs in order to reduce estimator variance. On the other hand, limited computational effort should be expanded on non-critical designs that have little effect on identifying the good designs even if they have large variances. In doing so, less computational effort is spent on simulating non-critical designs and more computational effort is spent on simulating critical designs; hence, the overall simulation efficiency is improved. Ideally, we want to optimally choose the number of simulation samples for all designs to maximize simulation efficiency with a given computing

budget. This is the basic idea of *optimal computing budget allocation* (OCBA) (Chen et al. 1996, 1999).

We apply the hybrid algorithm for a stochastic resource allocation problem, where no analytical expression exists for the objective function, and it is estimated through simulation. Numerical results show that our proposed algorithm can be effectively used for solving large-scale stochastic discrete optimization problems.

The paper is organized as follows: In section 2 we formulate the resource allocation problem as a stochastic discrete optimization problem. In section 3 we present the hybrid algorithm. The performance of the algorithm is illustrated with one numerical example in Section 4. Section 5 concludes the paper.

2 RESOURCE ALLOCATION PROBLEMS

There are many resource allocation problems in the design of discrete event systems. In this paper we consider the following resource allocation optimization problem:

$$\min_{\theta \in \Theta} J(\theta) \tag{2.1}$$

where Θ is a finite discrete set and J: $\Theta \rightarrow \mathbf{R}$ is a performance function that is subject to noise. Often $J(\theta)$ is an expectation of some random estimate of the performance,

$$J(\theta) = \mathrm{E}[L(\theta, \xi)] \tag{2.2}$$

where ξ is a random vector that represents uncertain factors in the systems. The "stochastic" aspect has to do with the problem of performing numerical expectation since the functional $L(\theta,\xi)$ is available only in the form of a complex calculation via simulation. The standard approach is to estimate $\mathrm{E}[L(\theta, \xi)]$ by simulation sampling, i.e.,

$$\mathrm{E}[L(\theta, \xi)] \approx \hat{J}(\theta) \equiv \frac{1}{t}\sum_{i=1}^{t} L(\theta, \xi_i) \tag{2.3}$$

Unfortunately, t can not be too small for a reasonable estimation of $\mathrm{E}[L(\theta, \xi)]$. And the total number of simulation samples can be extremely large since in the resource allocation problems, the number of $(\theta_1, \theta_2, \ldots, \theta_N)$ combinations is usually very large as we will show the following example.

2.1 Buffer Allocation in Supply Chain Management

We consider a 10-node network shown in Figure 1. There are 10 servers and 10 buffers, which is an example of a supply chain, although such a network could be the model for many different real-world systems, such as a manufacturing system, a communication or a traffic network. There are two classes of customers with different arrival distributions, but the same service requirements. We consider both exponential and non-exponential distributions (uniform) in the network. Both classes arrive at any of Nodes 0~3, and leave the network after having gone through three different stages of service. The routing is not probabilistic, but class dependent as shown in Figure 1. Finite buffer sizes at all nodes are assumed which is exactly what makes our optimization problem interesting. More specific, we are interested in distributing optimally

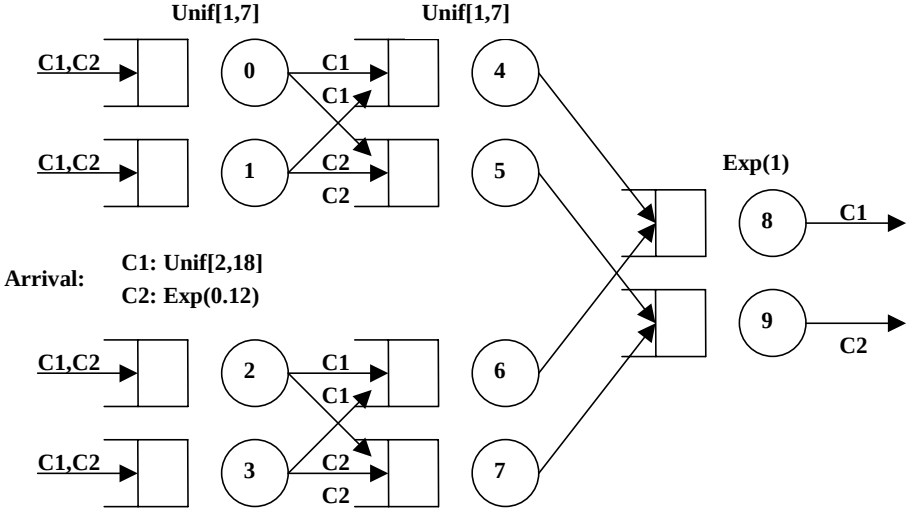

Figure 1: A 10-node Network in the Resource Allocation Problem

buffer spaces to different nodes given a limited budget for them. A buffer is said to be full if there are as many customers as its size in it, not including the customer being served in the server. We consider the problem of allocating 12 buffer units, among the 10 different nodes numbered from 0 to 9. We denote the buffer size of node i by B_i. Specifically,

$$B_0 + B_1 + B_2 + \ldots + B_9 = 12. \qquad (2.4)$$

Note that there are 293,930 different combinations of $[B_0, B_1, B_2, ..., B_9]$ which satisfy the constrain in (2.4). Unfortunately, due to the dynamic nature of the system, there is no closed-form analytical formula to evaluate the performance function. For each combination, the performance measure estimation involves a very long simulation (for steady state simulation) or a huge number of independent replications (for transient simulation). The total simulation cost is prohibitively large even if the simulation cost for a single design alternative is not expensive. In Section 4, we will illustrate the benefits of using the proposed algorithm to this buffer allocation problem.

3 A HYBRID ALGORITHM

In this section, we will present our hybrid algorithm for solving optimization problems discussed in the previous section. Our approach integrates Nested Partitions method and optimal computing budget allocation (OCBA). Optimal Computing Budget Allocation (OCBA) enhances the efficiency of simultaneous simulation experiments by intelligently determining the best allocation of simulation trials or samples necessary to maximize the probability of identifying the optimal ordinal solution. The integration with a Nested Partitions method further extends the applicability to an optimization with an extremely huge design space.

3.1 Nested Partitions Method

The Nested Partitions (NP) method has recently been proposed to solve global optimization problems. The method can be briefly described as follows. In each iteration we assume that we have a region that is considered the most promising. We partition this most promising region into M subregions and aggregate the entire surrounding region into one region. At each iteration, we therefore look at M +1 disjoint subsets that cover the feasible region. Each of these M +1 regions is sampled using some random sampling scheme and the estimated performance function values at randomly selected points are used to estimate the promising index for each region. This index determines which region becomes the most promising region in the next iteration. If one of the subregions is found to be best this region becomes the most promising region. If the surrounding region is found to be best the method backtracks to a larger region. To choose this larger region we use a fixed backtracking rule. The new most promising region is then partitioned and sampled in a similar fashion. The methodology described above may be divided into four main steps that constitute the NP method. Each of these steps can be implemented in a generic fashion, but can also be combined with other optimization methods and adapted to take advantage of any special structure of a given problem.

1. **Partitioning**. The first step is to partition the current most promising region into several subregions and aggregate the surrounding region into one region. The partitioning strategy imposes a structure on the feasible region and is therefore very important for the speed of convergence of the algorithm. If the partitioning is such that most of the good solutions tend to be clustered together in the same subregions, it is likely that the algorithm quickly concentrates the search in these subsets of the feasible region. It should be noted that since the feasible region is finite the partitioning can be done by grouping arbitrary points together in each subregion. Therefore, a good partitioning strategy always exists, although it may not be easy to identify.
2. **Random Sampling**. The next step of the algorithm is to randomly sample from each of the subregions and from the aggregated surrounding region. This can be done in almost any fashion. The only condition is that each solution in a given sampling region should be selected with a positive probability. Clearly uniform sampling can always be used. However, it may often be worthwhile to incorporate special structures into the sampling procedure. The aim of such a sampling method should be to select good solutions with a higher probability than poor solutions.
3. **Calculation of Promising Index**. Once each region has been sampled the next step is to use the sample points to calculate the promising index of each region. However, the total number of designs that must be evaluated using simulation in each iteration is equal to the total number of samples in all regions. The total simulation time in this step

could be very long. So Step 3 is the most time consuming step in the NP algorithm. Therefore, the improvement of computation efficiency at Step 3 is crucial to the efficiency of the hybrid algorithm. OCBA will be applied to improve simulation efficiency as shown in the following subsection.

4. **Backtracking.** If one of the subregions has the best promising index, the algorithm moves to this region and considers it to be the most promising region in the next iteration. If the surrounding region has the best promising index the algorithm backtracks to a larger region.

3.2 The OCBA Technique

In the Step 3 of the NP algorithm, we have to conduct a set of simultaneous simulation experiments, which is the most time-consuming step in the whole algorithm. The OCBA technique is applied to improve the efficiency of this bottleneck.

More specifically, suppose we select a design (or a solution) θ_a using the following criterion in this set of simultaneous simulation experiments:

$$\theta_a \equiv \arg\min_{\theta} \hat{J}(\theta) \left(\equiv \frac{1}{t}\sum_{i=1}^{t} L(\theta,\xi_i)\right). \qquad (3.1)$$

Define the *probability of correct selection*, $P(\mathrm{CS}) \equiv P\{$ The current top-raking design θ_b is actually the best design $\}$. Let t_θ be the number of simulation samples of design θ. If simulation is performed on a sequential computer and the difference of computation costs of simulating different designs is negligible, the total computation cost can be approximated by $\sum_{\theta\in\Theta} t_\theta$. The goal is to choose t_θ for all θ such that the total computation cost is minimized, subject to the restriction that the confidence level defined by $P\{\mathrm{CS}\}$ is greater than some satisfactory level.

$$\min_{t_\theta} \sum_{\theta\in\Theta} t_\theta$$
$$\text{s.t. } P\{\mathrm{CS}\} \geq P^*.$$

where P^* is a user-defined confidence level requirement, which corresponds to the stopping criterion in each iteration of the Nested Partition Method.

Chen et al. (1999) approximate $P\{\mathrm{CS}\}$ using the Chernoff bounds (Ross 1994) and a Bayesian model (Chen 1996) and offer an asymptotically solution, which is summarized in the following theorem.

Theorem 1. Given total number of simulation budget T to be allocated to a finite number of competing designs, the $P\{\mathrm{CS}\}$ can be asymptotically maximized when

$$\text{(a) } \frac{t_a}{t_b} \to \frac{s_a}{s_b}\left[\sum_{\substack{i=1\\ i\neq a}}^{k}\left(\frac{\delta_{a,b}^2}{\delta_{a,i}^2}\right)\right]^{1/2}$$

$$\text{(b) } \frac{t_i}{t_b} \to \left(\frac{\sigma_i/\delta_{a,i}}{\sigma_b/\delta_{a,b}}\right)^2 \text{ for } \theta \in \Theta \text{ and } \theta\, i \neq a \neq b,$$

where a is the design having the largest sample mean, b is the design having the second largest sample mean, and

$$\delta_{i,j} = \frac{1}{t_i}\sum_{u=1}^{t_i} L(i,\xi_u) - \frac{1}{t_j}\sum_{u=1}^{t_j} L(j,\xi_u), \text{ for any } i, j \in \Theta. \quad \#$$

4 NUMERICAL RESULTS

In this section, we apply the hybrid algorithm to the buffer allocation problem discussed in section 2. Before we report the numerical result of the hybrid algorithm, we first demonstrate in section 4.1 how OCBA technique can be applied to a simplified version of the buffer allocation problem. In this simplified version, where the total of designs (or solutions) is 210. We show that OCBA can achieve a speedup factor as high as 23. This means that the total computation time is reduced by 96% with the use of OCBA. In section 4.2, we apply the hybrid algorithm to deal with the original buffer allocation problem that has a much larger design space. We show that a better solution can be obtained with a reasonable simulation cost.

4.1 A Reduced Problem

Consider the 10-node network presented in section 2 in which the objective is to select a design with minimum expected time to process the first 100 customers from a same initial state that the system is empty. Multiple simulation runs are needed to seatmate $E[L(\bullet\ \bullet\ \bullet)]$ for each θ. As discussed in section 2, even for an allocation of 12 buffer units to 10 nodes, there are 293,930 different combinations. While the simulation time for each combination is not very long, the total simulation time for 293,930 designs are not affordable. By observation, we can see the network is symmetric. To reduce the number of designs for consideration to a much smaller size, we set three constraints for symmetry reasons:

$$B_0 = B_1 = B_2 = B_3 \qquad (4.1.a)$$
$$B_4 = B_6 \qquad (4.1.b)$$
$$B_5 = B_7 \qquad (4.1.c)$$

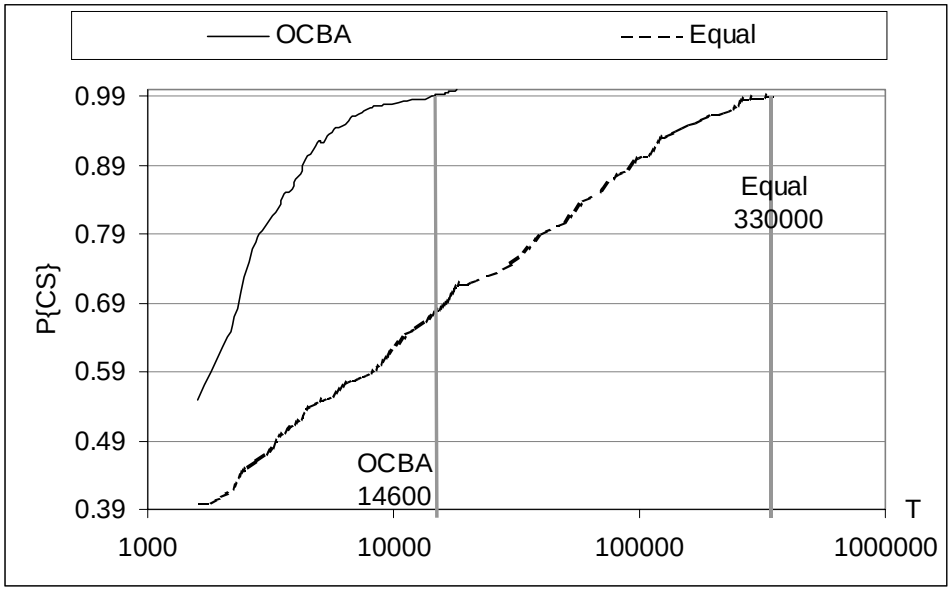

Figure 2: $P\{CS\}$ vs. the Computation Budget T. Note the x-axis is in log scale. The computation cost for obtaining $P\{CS\}$=99% with OCBA is 14,600. On the other hand, the cost is 330,000 if OCBA is not used (equal simulation)

With the above three constraints, the number of designs considered here is reduced to 210. Since the network is symmetric, we originally anticipated that the optimal design should satisfy the above three constraints. This turns out to be wrong after we apply the hybrid algorithm, as we will show later in next subsection. Now we first focus on the reduced 210 designs and apply OCBA to this simplified problem. Different computing budgets are allocated. 10,000 independent experiments are performed to estimate $P\{CS\}$. In all the numerical illustrations, we estimate $P\{CS\}$ by counting the number of times we successfully find the true best design in those 10,000 independent experiments. $P\{CS\}$ is then obtained by dividing this number by 10,000, representing the correct selection frequency. Figure 2 shows the test results using OCBA and equal allocation of simulation budget (without OCBA).

From Figure 2, we observe that a higher computing budget can obtain a higher $P\{CS\}$. Using the OCBA scheme, however, significantly reduces the computation cost for a desired level of $P\{CS\}$. The speedup factor is as high as 23. This means that our OCBA can further reduce the required simulation time for a crude NP by 96%. This is a tremendous saving already.

In order to have a better idea about the optimal design, we conduct a simulation experiment with $P^* = 99.999\%$. The best design we obtained is $[B_0, B_1, B_2, ..., B_9] = [1, 1, 1, 1 , 2, 1, 2, 1, 1, 1]$. We will show that the proposed hybrid algorithm can obtain a better design with a reasonable simulation cost in the next subsection.

4.2 The Original Resource Allocation Problem

In this subsection we apply our hybrid algorithm to the original 10-node network. Note that the problem considered here has 293,930 different designs, which is dramatically bigger than the 210 designs considered in the reduced problem.

In each iteration, we randomly sample 45 designs from the promising region, and 105 designs from the surrounding regions, making the total 150 design for consideration in an iteration. The stopping criterion is that the confidence level of identifying the best in the 150 design is no less than 90%, i.e., $P\{CS\}>90\%$.

In order to improve the quality of our sampling designs, we adopt the very simple heuristic presented in section 3.4.2 for our sampling scheme. Our algorithm converges to a design $[B_0, B_1, B_2, ..., B_9] = [2, 1, 1, 1, 2, 1, 2, 1, 0, 1]$. It turns out this design is better than the design we found in Section 4.1. Obviously, this design does not satisfy the symmetric constraints in (4.1). The total number of simulation runs to converge to this design is only 3 times bigger than the needed cost for the reduced problem in Section 4.1. Given that the design space is much bigger (293,930/210≈1400 bigger), the timesaving is tremendous.

5 CONCLUSIONS

In this paper we introduced a hybrid algorithm for stochastic discrete resource allocation optimization. The hybrid algorithm combines a recently developed

optimization framework, the *Nested Partitions* methods with the paradigm of an efficient ranking and selection technique called *optimal computing budget allocation* (OCBA). We applied the proposed algorithm to a stochastic buffer allocation problem. Our numerical results show that we are able to quickly obtain a near optimal solution by evaluated a very small fraction of the solution space.

ACKNOWLEDGEMENTS

This work has been supported in part by NSF under grants DMI-9713647 and DMI-9732173, by Sandia National Laboratories under contract BD-0618, and by the University of Pennsylvania Research Foundation.

REFERENCES

Andradottir, S. 1995. “A Method for Discrete Stochastic Optimization,” *Management Science*, 41:1946-1961.

Cassandras, C, L. Dai, and C. G. Panayiotou. 1998. "Ordinal Optimization for a Class of Deterministic and Stochastic Discrete Resource Allocation Problems," *IEEE Trans. on AC*., 43:881-900.

Chen, C. H., H. C. Chen, and L. Dai. 1996. "A Gradient Approach of Smartly Allocating Computing Budget for Discrete Event Simulation," *Proceedings of the 1996 Winter Simulation Conference*, 398-405.

Chen, C. H. 1996. “A Lower Bound for the Correct Subset-Selection Probability and Its Application to Discrete Event System Simulations.” *IEEE Transactions on Automatic Control*, 41:1227-1231.

Chen, C. H., V. Kumar, and Y. C. Luo. 1998. "Motion Planning of Walking Robots Using Ordinal Optimization," *IEEE Robotics and Automation Magazine*, 22-32.

Chen, C. H., S. D. Wu, and L. Dai. 1999. "Ordinal Comparison of Heuristic Algorithms Using Stochastic Optimization," *IEEE Transactions on Robotics and Automation*, 15: 44-56.

Chen, H. C., C. H. Chen, and E. Yücesan. 1999. "Computing Efforts Allocation for Ordinal Optimization and Discrete Event Simulation," To appear in *IEEE Transactions on Automatic Control*.

Dai, L. 1996. "Convergence Properties of Ordinal Comparison in the Simulation of Discrete Event Dynamic Systems," *Journal of Optimization Theory and Applications*, 91: 363-388.

Gelfand, S. B., and S. K. Mitter. 1989. "Simulated Annealing with Noisy or Imprecise Energy Measurements," Journal of Optimiation: Theory and Application, 62:49-62.

Gong, W. B., Y. C. Ho, and W. Zhai. 1995. "Stochastic Comparison Algorithm for Discrete Optimization with Estimations," *Discrete Event Dynamic Systems: Theory and Applications*.

Ho, Y. C., R. S. Sreenivas, and P. Vakili. 1992. "Ordinal Optimization of DEDS," *Journal of Discrete Event Dynamic Systems*, 2:61-88.

Inoue, K. and S. Chick. 1998. "Comparison of Bayesian and Frequentist Assessments of Uncertainty for Selecting the Best System," *Proceedings of the 1998 Winter Simulation Conference*, 727-734.

Norkin, W.I., Y.M. Ermoliev, and A. Ruszczynski. 1996. “On Optimal Allocation of Indivisables Under Uncertainty,” Operations Research, 46:381-395.

Patsis, N. T., C. H. Chen, and M. E. Larson. 1997. "SIMD Parallel Discrete Event Dynamic System Simulation," *IEEE Transactions on Control Systems Technology*, 5:30-41.

Rubinstein, R.Y. 1999. “The Simulated Entropy Method for Combinatorial And Continuous Optimization” Manuscript.

Shi, L. and S. Olafsson. 1999. “Nested Partitions Method for Global Optimization.” To appear in *Operations Research*.

Shi, L., S. Olafsson, and N. Sun. 1999. “New Parallel Randomized Algorithms for the Traveling Salesman Problem,” *Computers and Operations Research*, 26:371-394.

Ross, S. 1994. *A First Course in Probability*, Prentice Hall Inc.

Yan, D. and H. Mukai. 1993. "Optimization Algorithm with Probabilistic Estimation," *Journal of Optimization Theory and Applications*, 79: 345-371.

AUTHOR BIOGRAPHIES

LEYUAN SHI is an Assistant Professor in the Department of Industrial Engineering at the University of Wisconsin-Madison. She holds a B.S. degree in Mathematics from Nanjing Normal University, China (1982), an M.S. degree in Applied Mathematics from Tsinghua University, China (1985), and an M.S. and a Ph.D. degrees in Applied Mathematics from Harvard University (1990 ,1992). Her research interests include modeling, analysis, and optimization of discrete event systems, discrete-event simulation, and sensitivity analysis.

CHUN-HUNG CHEN is an Assistant Professor of Systems Engineering at the University of Pennsylvania, Philadelphia, PA. He received his Ph.D. degree in Simulation and Decision from Harvard University in 1994. His research interests cover a wide range of areas in Monte Carlo simulation, web-based simulation, optimal control, stochastic decision processes, ordinal optimization, and their applications to manufacturing systems. Dr. Chen won the 1994 Harvard University Eliahu I. Jury Award for the

best thesis in the field of control. He is also one of the recipients of the 1992 MasPar Parallel Computer Challenge Award.

ENVER YÜCESAN is a Professor of Operations Research at INSEAD in Fontainebleau, FRANCE. He holds a BSIE degree from Pudue University, and an MS and a Ph.D. both in OR, from Cornell University. The work described in this paper has been initiated while he was visiting the Department of Systems Engineering at the University of Pennsylvania. His research interests include web-based simulation, systems design and optimization, and supply chain management.