Proceedings of the 2004 Winter Simulation Conference
R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

SIMULATION-BASED OPTIMIZATION FOR MATERIAL DISPATCHING IN A RETAILER NETWORK

Ganesh Subramaniam
Abhijit Gosavi

Department of Industrial Engineering
University at Buffalo,
The State University of New York
Buffalo, NY 14260, U.S.A.

ABSTRACT

This paper presents preliminary work done on simulation-based optimization of a stochastic *material-dispatching* system in a retailer network. The problem we consider is one of determining the optimal number of trucks and quantities to be dispatched in such a system. Theoretical solution models for versions of this problem can be found in the literature. Unlike most theoretical models, we can accommodate many real-life considerations, such as arbitrary distributions of the governing random variables, and all important cost elements, such as inventory-holding costs, stock-out costs, and transportation costs. We have used two techniques, namely, neuro-response surfaces and simulated annealing, for optimizing our system. We have also used a problem-specific heuristic, known as the mean demand heuristic, to provide us with a good starting point for simulated annealing and a benchmark for our other methods. Some computational results are also provided.

1 INTRODUCTION

Typically, a supply chain of consumer goods, such as gasoline, food products, and clothing items, consists of distribution centers, warehouses, and retailers. The distribution industry focuses on transporting goods from the manufacturer to the customers. A goal of this industry is to make the distribution process “lean", and thereby achieve cost benefits. The problem we have considered here is geared towards reducing the inventory in the distribution network and ensuring a satisfactory service level. Generally, a warehouse serves multiple retailers where customers arrive randomly to buy products. An optimization problem commonly-faced by managers is to determine (1) the number of trucks to be dispatched, and (2) the amount of goods to be dispatched. Associated with this problem are the costs of holding excess inventory (inventory-holding costs), not being able to meet customer demand (stock-out costs), and transporting goods from the manufacturer to the retailers (transportation costs). The cost function that we have developed in this paper accounts for all of these elements. When there are multiple retailers and each retailer has unique random characteristics, such as arrival rate of customers and size of the demand, one has a large-scale and complex stochastic optimization problem on which it is not easy to construct an exact theoretical model. In this paper, we study a complex problem with multiple retailers and a large number of governing random variables, and use a simulation-based approach for solving it.

Seminal work on this problem is from Clark and Scarf (1960). In their paper, they have assumed a holding and shortage cost but ignored the setup cost or the reorder cost. Jönsson and Silver (1987) have suggested the use of a “redistribution" strategy in which the inventories at the retailers are pooled and redistributed to standardize the inventory at each retailer. Their model is for systems in which demand variation is low and in which it can be shown that the stockouts can occur only in the last periods of an order cycle. McGavin, Schwarz, and Ward (1993) have suggested a so-called “between-replenishment," “risk-pooling" policy for this problem. They have a two-interval allocation policy in which the stock is withdrawn from the warehouse at two (unequal) intervals in the same order cycle. Their model ignores the inventory holding costs. Nahmias and Smith (1994) have developed a model for demands that have the negative-binomial distribution. Federgruen and Zipkin (1984) have modeled an extension of the previous work by Eppen and Schrage (1981) overcoming some of the limitations like normal distribution of the demand and identical holding and penalty costs across all the retailers. But they develop a myopic model, i.e., a model in which the system is optimized in the period during which the actual allocation occurs, ignoring the costs in all subsequent periods. Our model, on the other hand, minimizes the *long-run* average cost of operating the system. Kumar, Schwarz, and Ward (1995) have given static and dynamic models for a similar system but with a myopic allocation policy. Furthermore, their policies are valid only for low-variance demand at identical retailers. Our model is not restricted to this assumption.

We have modeled the problem of dispatching material between a single warehouse and N retailers using *simulation-based optimization* (see Andradottir 2002, Fu 2002, Gosavi 2003). The advantage of using simulation is that we have been able to consider all the costs involved in the real-world system. Our model, unlike some in the literature, does not approximate any feature of the cost function. We also have no restrictions on the distributions that can be used for the random variables governing the system.

The last few years have seen an explosion in the number of papers written on meta-heuristics and simulation-based optimization (Barton and Ivey 1996; Chen, Chen, and Yucesan 2000; Fu and Hu 1997; Glassserman 1991; Glynn 2002; Ho and Cao 1991; Ho, Sreenivas and Vakili 1992; Pflug 1996; Shi and Ólafsson 1998; Spall 1992; Yan and Mukai 1992). We have used two optimization techniques here, namely, ***simulated annealing*** and ***neuro-response surfaces***. We have also used an industrial heuristic, called the mean demand heuristic, which provides us with a benchmark for our methods and a starting point for simulated annealing.

A detailed description of the problem is given in Section 2. This is followed by the solution methodology in Section 3. Section 4 provides the computational results. Section 5 discusses the conclusions of this work and future work to be done.

2 PROBLEM DESCRIPTION

The distribution network, generally, has a hierarchical structure in which a warehouse serves a set of retailers. All warehouses are coordinated and replenished by a central distribution center. We assume that distribution network has already been designed. The network can be divided into three echelons (see Figure 1). Echelon 1 is the first level from the manufacturer to the regional distribution centers, Echelon 2 is the level from the regional distribution centers to the local distribution centers, and Echelon 3 is the level from the local distribution centers to the retailers. Our focus in this paper is on a problem in Echelon 3.

A "transshipment" point is a point in a supply chain where goods are transferred from one echelon to another. Goods are stored temporarily at the transshipment point. The warehouse acts as a transshipment point for the items to be distributed, and goods received by it are distributed among the retailers. The problem is to determine the optimal quantities to be delivered from the transshipment point to each of the retailers so as to minimize the average cost of

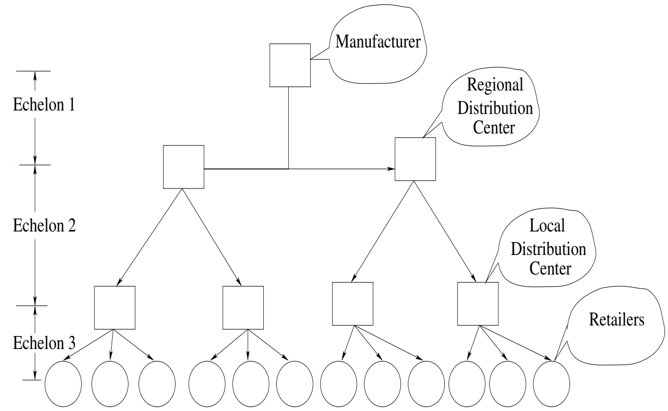

Figure 1: Distribution Network

operating the system. Our model takes the following costs into consideration. (1) Inventory-holding costs. (2) Stock-out costs, which include the cost of lost sales and the loss of goodwill. (3) Transportation costs, which include the operating cost of the truck and the cost of transporting the goods, which in turn depends on the quantities transported.

The system considered here is a complex real-world system. The random variables governing our system are: the inter-arrival time of each customer, the quantity demanded by each customer, the service time for each truck, and the travel time between the warehouse and retailers and the same between the retailers. It is difficult to develop an exact mathematical model that will account for all the random variables and costs that we have considered. This motivates a simulation-based model for performance evaluation and optimization. We have assumed that each retailer is distinct and has unique values for the system parameters. The rate of arrival of customers, the inventory-holding costs, and the stock-out costs are all different for each retailer. Our objective is to minimize the average cost per unit time of running the entire system. We next present a mathematical description of the problem.

Consider a probability space $(\Omega, \mathcal{F}, P)$, where P denotes the distribution of the profits generated by the system under consideration. From the simulator of the profits generated by the system, one can generate k samples: $(\omega^1, \omega^2, \ldots, \omega^k)$. Let $\vec{q} = (q_1, q_2, \ldots, q_n)$ denote the vector of delivery quantities, i.e., the *delivery vector*, where q_i denotes the quantity to be delivered to the ith retailer and n is the number of retailers. Let $f_i(\vec{q}, \omega^j)$ denote the average cost per unit time of running the system estimated from the jth sample (generated by the simulator) when the delivery vector is $\vec{q}$. Let C_t denote the truck operating cost per unit quantity per unit time. Let $L_i(\vec{q}, \omega^j)$ denote the total number of lost sales at the ith retailer in the jth sample when the delivery vector is $\vec{q}$, and ${C_l}^i$ denote the lost-sales or stock-out cost per unit quantity at the ith retailer. Let

$I_i(\vec{q}, t, \omega^j)$ denote the positive inventory at time t at the ith retailer in the jth sample, ${C_e}^i$ denote the inventory-holding cost per unit quantity at the ith retailer, and $T(\omega^j)$ denote the length of the trip in the jth sample. Then mathematically, the problem is to:

$$\text{Minimize } C_t \sum_{i=1}^{n} q_i + \sum_{i=1}^{n} \mathsf{E}[f_i(\vec{q}, \omega^j)],$$

$$\text{where } f_i(\vec{q}, \omega^j) = C_l L_i(\vec{q}, \omega^j) + \frac{C_e \int_0^{T(\omega^j)} I_i(\vec{q}, t, \omega^j)\, dt}{T(\omega^j)},$$

$$\text{and } \mathsf{E}[f_i(\vec{q}, \omega^j)] = \lim_{k \to \infty} \sum_{j=1}^{k} f_i(\vec{q}, \omega^j)$$

$$\text{such that } q_i \geq 0 \text{ for } i = 1, 2, \ldots, N.$$

3 SOLUTION METHODOLOGY

The two optimization techniques, namely, simulated annealing and the neuro-response surface method (NRSM), which we have used, are explained next.

3.1 Neuro-Response Surfaces

The response surface method (RSM) has been a popular method of optimization in simulation-based optimization due to its robustness and strong mathematical (statistical) backing. Traditional RSM uses regression for fitting the objective function. This requires the assumption of a metamodel. Unlike traditional RSM, NRSM does *not* assume a metamodel. In NRSM, function fitting is done using neural networks. Its power lies in its ability to fit *any* surface. The NRSM uses the well-known backpropogation algorithm. The steps involved in this algorithm are explained below.

Consider a neural network shown in Figure 2. The so-called "input" layer consists of a finite number of nodes; usually one node is associated with each decision variable. The number of nodes in the hidden layer is a function of the non-linearity of the function to be fitted. (Greater the non-linearity of the function larger is the required number of nodes in the hidden layer.) The neural network computes the so-called "weights" which represent its metamodel. Let $w(i,h)$ denote the weight from the ith input node to the hth hidden node and $x(h)$ the weight from the hth hidden node to the output node. The bias node is comparable to the constant term in regression-based function fitting. Let p denote the number of pieces of data used for training the neural net. The available data for the pth piece is $(\vec{u}_p, y_p)$ where $\vec{u}_p$ denotes a vector with I components.

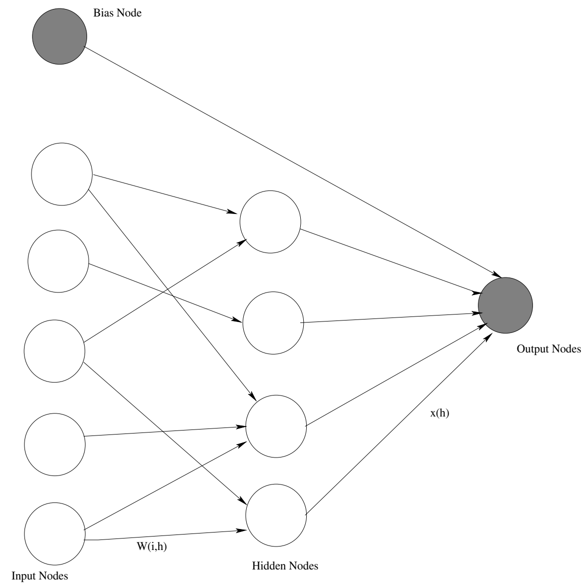

Figure 2: Neural Network

Step 1: The first step is to set all the weights to small random numbers. Set the value of the SSE (sum of squared errors) to a large number.

Step 2: Calculate the output value at the hidden node as follows

$$V_p{}^*(h) = \sum_{i=1}^{I} w(i,h) u_p(i)$$

where, $V_p{}^*(h)$ denotes the output value at the hidden node h and $u_p(i)$ the input value to the i_{th} input node.

Step 3: Compute $V_p(h)$ as follows, using the *sigmoid* function.

$$V_p(h) = \frac{1}{1 + e^{-V_p{}^*(h)}}$$

Step 4: Compute each of the output terms O_p, for $p = 1, 2, \ldots, n$, where n is the number of data pieces, using

$$O_p = b + \sum_{h=1}^{H} x(h) V_p(h).$$

Step 5: Update $b, w(i,h), x(h)$ as follows

$$b \leftarrow b + \mu \sum_{p=1}^{n} (y_p - O_p).$$

$$w(i,h) \leftarrow w(i,h) + \mu \sum_{p=1}^{n} (y_p - O_p) x_h v_p(h)(1 - v_p(h)) u_p(i).$$

$$x(h) \leftarrow x(h) + \mu \sum_{p=1}^{n} (y_p - O_p) v_p(h).$$

Step 6: Calculate SSE_{new} using

$$SSE_{new} = \sum_{p=1}^{n} (y_p - O_p)^2$$

Reduce the value of μ. If $|SSE_{new} - SSE_{old}| < tolerance$, Stop. Otherwise, set $SSE_{old} = SSE_{new}$ and return to Step 2.

3.2 Simulated Annealing

Simulated annealing is a heuristic stochastic search technique that starts at an arbitrary solution and uses a so-called "neighborhood search strategy". It has an "exploratory" property which allows the algorithm to worsen the objective function value at times. It is claimed that this can help in finding the global optimum in a problem with multiple local optima. The algorithm accepts a worse solution with a probability, given in Step 4b, which is decayed with the number of iterations. This algorithm is written for minimizing the objective function value.

Step 1: Let the current solution vector (selected randomly) be denoted by $\vec{x}_{current}$. Set $\vec{x}_{best} \leftarrow \vec{x}_{current}$. Set phase number P to 0.

Step 2: Select a solution vector (at random) adjacent to the current solution vector and denote it $\vec{x}_{new}$

Step 3: If

$$f(\vec{x}_{new}) < f(\vec{x}_{best})$$

Set $\vec{x}_{best} \leftarrow \vec{x}_{new}$

Step 4: Calculate

$$\delta = f(\vec{x}_{new}) - f(\vec{x}_{current})$$

If $\delta \leq 0$, go to Step 4a otherwise go to Step 4b.

Step 4a: Set

$$\vec{x}_{current} \leftarrow \vec{x}_{new}.$$

Go to Step 5.

Step 4b: Generate a uniform random number, U, between 0 and 1. If

$$U \leq exp(-\frac{\delta}{T},)$$

then set

$$\vec{x}_{current} \leftarrow \vec{x}_{new}.$$

Otherwise do not change $\vec{x}_{current}$ and go to Step 5.

Step 5: One execution of Steps 2, 3 and 4 is an iteration of a "phase". Repeat the steps till the number of iterations associated with the current phase are completed.

Step 6: Increment the phase number P by 1. If $P < P_{\max}$, then reduce T and return to Step 2. Otherwise terminate the algorithm and declare $\vec{x}_{best}$ to be the best solution obtained.

The temperature (T) of the process represents the exploratory property (Step 4 of the algorithm) and is reduced gradually. Hence determining this temperature properly is an important factor in the success of the algorithm. It has been established that the algorithm does provide the optimal solution in an asymptotic sense if the temperature is decreased properly (Lundy and Mees 1986). In our experiments, the starting point for the algorithm was provided by a problem-specific heuristic, which we discuss next.

3.3 The Mean Demand Heuristic

The Mean Demand Heuristic (MDH) is a problem-specific heuristic that provides us with a good starting point for simulated annealing and also provides a lower bound on the search region for the neuro-response surface method. The heuristic can be explained as follows. Let T denote the average cycle time, i.e., the time required to go around the route once and return to the retailer. Let d_i denote the average demand per customer at the ith retailer. Then, the optimal quantity Q_i for the ith retailer, according to this heuristic, is given by $Q_i = T d_i \lambda_i$, where λ_i denotes the mean rate of arrival of customers at the ith retailer.

4 COMPUTATIONAL RESULTS

We compare the performance of SA and NRSM to that of MDH. We ran the three methods on a system with two retailers (See Fig 3). For NRSM we sampled the solution space of the objective function at various points using a full factorial experiment to do the sampling. To train the neural network properly we needed to sample the solution space at no less than 5 points for each retailer. Thus the number of sampled points of the simulator for the 2-retailer network became 2^5. The parameters used for each system are provided in Table 1, and the computational results are shown in Table 2. We used a value of 1 for δ in Step 4b of the SA algorithm because larger values of δ caused excessive exploration. NRSM takes an exceedingly long time on a 10-retailer network. Hence we are conducting experiments with SA and gradient-based techniques on larger problems.

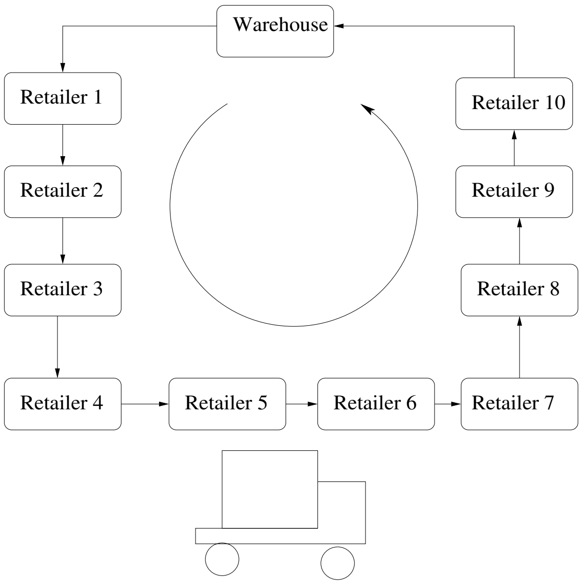

Figure 3: Route Followed By Truck

Table 1: Parameters Used For Each System

	$C_e{}^i$		$C_l{}^i$		λ_i	
System	$i=1$	$i=2$	$i=1$	$i=2$	$i=1$	$i=2$
1	0.01	0.09	1.5	1.4	90	110
2	0.01	0.009	3	3.1	90	110

Table 2: Comparison Of Costs From The 3 Methods

System	Mean-Demand Heuristic (MDH)	Improvement of NRS over MDH	Improvement of SA over MDH
1	22311	51.42 %	62.55 %
2	17329	63.53 %	70.37 %

5 CONCLUSIONS & FUTURE WORK

We considered a material-dispatching problem found in many real-world retailer networks. We developed a model that accommodates almost all real-life considerations, and also showed how it can be used for optimization. Thus far, our tests have been on medium-sized problems but we are in the process of conducting tests on very-large scale problems. (We have already generated the simulator for such tests; see Fig 4.) We have also used SA successfully on medium-sized problems. It is expected that further research will lead to valuable insights on solving the full blown, large-scale version of this problem.

REFERENCES

Andradóttir, S, 2002. Simulation Optimization: Integrated Research and Practice, *Journal on Computing*, Volume 14, Number 3, pp. 216-219.

Barton, R. R. and J. S. Ivey, 1996. Nelder-Mead Simplex Modifications for Simulation-Optimization, *Management Science*, Volume 42, pp. 954-973

Chen, H.C, C. H. Chen, and E. Yucesan, 2000. Computing Efforts Allocation for Ordinal Optimization and Discrete Event Simulation, *IEEE Transactions on Automatic Control*, Volume 45, Number 5, pp. 960-964.

Clark, A. J and H. Scarf, 1960. Optimal Policies for A Multi-Echelon Inventory Problem, *Management Science*, Volume 6, Number 4, pp. 475-490.

Eppen, G. and L. Schrage, 1981. Centralized Ordering Policies in A Multi-Warehouse System With Lead Times and Random Demand, in *Multi-Level Production/Inventory Control Systems: Theory And Practice,* North Holland, Amsterdam, pp. 51-69.

Federgruen, A and P. Zipkin, 1984. Approximations Of Dynamic, Multi-Location Production and Inventory Problems, *Management Science*, Volume 30, Number 1, pp. 69-84.

Fu, M. C, 2002. Optimization for Simulation: Theory Vs Practice, *Journal on Computing*, Volume 14, Number 3, pp. 192-215.

Fu, M. C and J. Q. Hu, 1997. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*, Kluwer Academic, Massachusetts.

Glassserman, P, 1991. *Gradient Estimation Via Perturbation Analysis*, Kluwer Academic, Massachusetts.

Glynn, P. W, 2002. Additional Perspectives On Simulation for Optimization, *INFORMS Journal of Computing*, Volume 14, Number 3, pp. 220-222.

Gosavi, A, 2003. *Simulation-Based Optimization Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academic Publishers, Boston, Massachusetts, USA.

Ho, Y. C. and Cao. X. R, 1991. *Perturbation Analysis of Discrete-Event Dynamic Systems*, Kluwer Academic, Massachusetts.

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

Jönsson, H and E. A. Silver, 1987. Analysis of A Two-Echelon Inventory Control System With Complete Redistribution, *Management Science*, Volume 33, Number

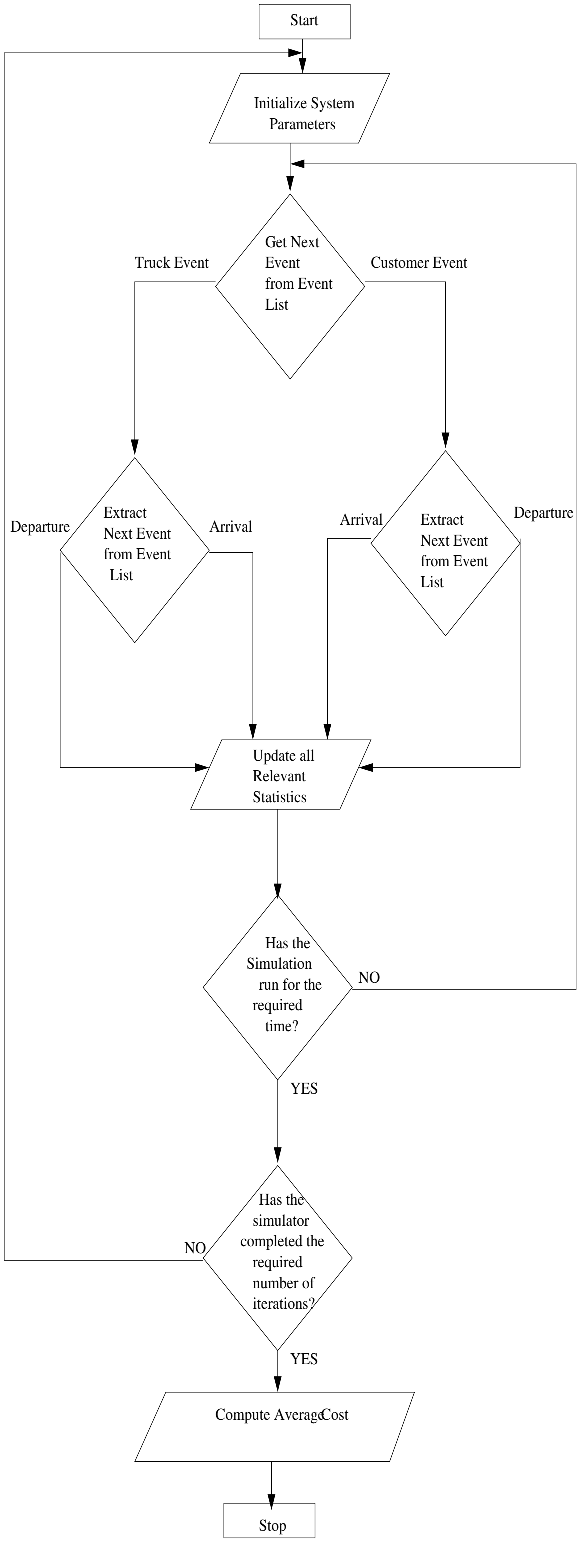

Figure 4: Flowchart Depicting The Working Of Simulator

2, pp. 215- 227.

Kumar, A, L. B. Schwarz, and J. E. Ward, 1995. Risk-Pooling Along a Fixed Delivery Route Using a Dynamic Inventory-Allocation Policy, *Management Science*, Volume 41, Number 2, pp. 344-362.

Lundy, M and A. Mees, 1986. Convergence of The Annealing Algorithm. *Mathematcial Programming*, Volume 34, pp. 111-124.

McGavin, E. J, L. B. Schwarz, J. E. Ward, 1993. Two Inventory Allocation Policies In A One Warehouse N-Identical Retailer Distribution System, *Management Science*, Volume 39, Number 9, pp. 1092-1107.

Nahmias, S and S. A. Smith, 1994. Optimizing Inventory Levels In A Two-Echelon Retailer System With Partial Lost Sales, *Management Science*, Volume 40, Number 5, pp.582-596.

Pflug, G. C, 1996. *Optimization of Stochastic Models: The Interface Between Simulation And Optimization*, Kluwer Academic, Massachusetts.

Shi, L. and S. Ólafsson, 1998. Nested Partitions Method for Stochastic Optimization, *Operations Research*, Volume 48, pp. 390-407.

Spall, J. C, 1992. Multivariate Stochastic Approximation Using A Simultaneous Perturbation Gradient Approximation, *IEEE Transactions on Automatic Control* Volume 37, pp. 332-341.

Yan, D. and H.Mukai, 1992. Stochastic Discrete Optimization. *SIAM Journal on Control and Optimization*, Volume 30, pp. 594-612.

AUTHOR BIOGRAPHIES

GANESH SUBRAMANIAM is a graduate student in the department of industrial engineering at the University at Buffalo, State University of New York. He is working as a research assistant in the simulation-optimization laboratory at the State University of New York. His research interest is in applying simulation-based optimization to variety of production planning and supply chain management problems. He is a member of IIE and Omega Rho. His e-mail address is `<ganesh@ganeshs.net>`, and his web page is `<www.ganeshs.net>`.

ABHIJIT GOSAVI is an assistant professor in the department of industrial engineering at the University at Buffalo, State University of New York. His research interests are in simulation-based optimization, reinforcement learning, and Markov decision modeling. He has published in journals such as *Management Science*, *IIE Transactions*, *Communications in Statistics*, etc. He is a member of IIE and INFORMS. His e-mail address is `<agosavi@buffalo.edu>`, and his web page is `< www.eng.buffalo.edu/ agosavi >`.