

---

## Simulation-based optimisation for material dispatching in Vendor-Managed Inventory systems

---

Ganesh Subramaniam

American Solutions Inc.,  
100 Commerce Dr Suite # 103,  
Newark, DE 19713, USA

Abhijit Gosavi\*

Simulation-Optimization Laboratory,  
Department of Industrial Engineering,  
State University of New York at Buffalo,  
317 Bell Hall, Buffalo, NY 14260-2050, USA  
E-mail: agosavi@buffalo.edu  
\*Corresponding author

**Abstract:** This paper examines a problem related to replenishing inventories at retailers in distribution networks operated under the paradigm of Vendor Managed Inventory (VMI). In the real world, complexity in analysing such systems arises out of numerous factors, including several random variables and a multitude of costs in the system. Much of the literature makes simplifying assumptions about these factors to develop tractable mathematical models. In this paper, we develop a simulation-optimisation approach that can accommodate most of these factors. In our experiments, our approach outperforms a heuristic used in a local industry and a heuristic developed from the newsvendor model.

**Keywords:** retailer networks; supply chain; simulation-based optimisation; Vendor-Managed Inventory; VMI.

**Reference** to this paper should be made as follows: Subramaniam, G. and Gosavi, A. (2007) 'Simulation-based optimisation for material dispatching in Vendor-Managed Inventory systems', *Int. J. Simulation and Process Modelling*, Vol. 3, No. 4, pp.238–245.

**Biographical notes:** Ganesh Subramaniam received his MS from the University at Buffalo, State University of New York in September 2004. He is now employed as a Statistical Analyst with American Solutions, Newark, Delaware. His research interests include simulation and supply chain management.

Abhijit Gosavi is an Assistant Professor in the Department of Industrial Engineering at the University at Buffalo, State University of New York. His research areas include simulation-based optimisation machine learning preventive maintenance and revenue management.

---

### 1 Introduction

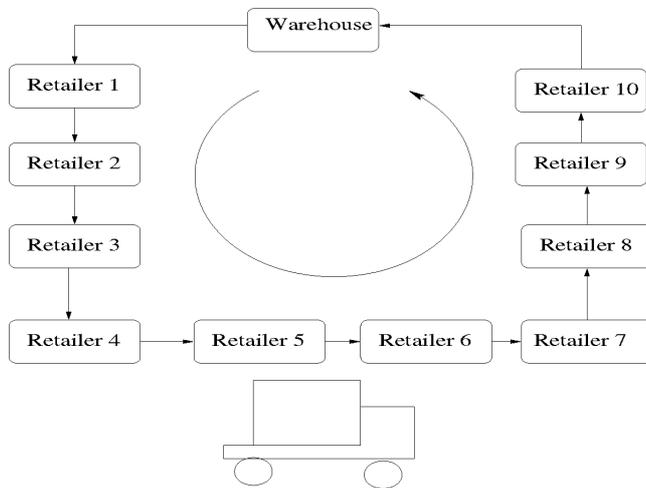
The science of supply chain management has provided several scientific procedures for reducing inventory and optimising service levels in distribution networks of manufactured goods. For many companies, the application of these techniques has resulted in achieving significant cost reductions. For example, Geon (see Hammer, 2001) obtained cost improvements by integrating its operations with those of its suppliers and customers for inventory management, it implemented advanced replenishment strategies like VMI. There are numerous examples in the industry of organisations that have utilised efficient inventory-management systems to reduce the total expenses incurred in transportation, storage, and providing service.

Examples of such companies are producers of food products and petroleum.

An important problem in many distribution networks operated under the VMI paradigm is to determine the optimal quantities of material to be dispatched to each retailer from the local warehouse. Usually, the goal is to optimise the trade-off between maintaining excessive inventory at the retailers, which increases the operating costs, and not maintaining enough inventory, which can cause stock-outs and lost sales. In addition, one can not ignore the issue of transportation costs which can be reduced by shipping large batches that unfortunately tend to increase inventory-holding costs. In this paper, we consider a retailer network of the type found in many real-world

systems. Generally, trucks are dispatched from the warehouse to the retailers carrying the material needed. We assume that the trucks follow a fixed route in which each retailer is visited once (see Figure 1). When the truck (s) leaves the warehouse it carries the material needed for replenishing each retailer.

**Figure 1** A schematic of the route followed by a truck



*Contributions of this paper:* We develop a simulation-optimisation methodology to solve the problem of determining the optimal quantities to be delivered to each retailer. We show that the methodology presented here can accommodate a large number of features of real-world systems and can outperform two heuristics. One of the heuristics outperformed is used in a local industry. The other heuristic is derived from the newsvendor model. An interesting finding is that our newsvendor heuristic produces reasonably good solutions – although, of course, it is outperformed by simulation-based optimisation. This implies that managers not interested in pursuing an elaborate simulation-optimisation approach can resort to the simpler newsvendor model for solution purposes. In addition, we also prove that under the general assumptions made here, the cost function to be minimised is non-convex.

A number of factors contribute to randomness in the system. Some examples of such factors are: randomness in customer arrival and size of customer demand and the randomness in the transportation. As mentioned previously, there are three sources of costs in these systems: transportation costs, inventory-holding costs, and stock-out costs. Modelling such a system mathematically is often difficult and challenging. Although many mathematical models have been developed in the literature, they usually disregard some complexity – either a cost or a governing random variable – to keep the model tractable. As a result, we focus on developing a simulation-based model that accommodates many features of a real-world system. The simulation model is combined with optimisation techniques to generate implementable solutions.

The problem considered in this paper is of tremendous relevance to managers of warehouses who have to dispatch

trucks with the right amounts of material. VMI-based systems face this problem on a daily or a weekly basis. In fact, this study was motivated by a problem faced in a local industry. Solutions using the methods proposed in this paper can be obtained easily with computer programs, which can be run on personal computers, and used directly in the decision-making process. (Computer codes are available from the second author upon request). Since the simulation model is very general, the manager can easily change the system parameters for retailers, travel times, and the demand rates as and when needed.

Seminal work on the static inventory-allocation problem is from Clark and Scarf (1960) and Eppen and Schrage (1981), which forms the foundation of the underlying science in this field. Much of the existing literature is devoted to the development of mathematical models, and as such, it ignores either one or more of the three costs (mentioned above) involved or else makes simplifying assumptions about governing random variables. For instance, McGavin et al. (1993) ignore inventory-holding costs, Federgruen and Zipkin (1984) develop a so called ‘myopic’ model which optimises in the current time period but disregards costs in future time periods, and Nahmias and Smith (1994) develop models for the negative-binomial distribution. Some other works that look at dynamic allocation and especially routing – an aspect that we do not consider – are Kumar et al. (1995), Minkoff (1993), Berman and Larson (2001) and Kleywegt et al. (2002). Many of these use stochastic dynamic programming.

The rest of the paper is organised as follows: Section 2 provides a problem statement. Section 3 describes the solution methodology adopted and the computational results obtained. Conclusions drawn from this work are presented in Section 4.

## 2 A problem description

The distribution network, generally, has a hierarchical structure in which a warehouse serves a set of retailers. The problem considered in this paper is to determine the optimal quantities to be delivered from the warehouse (also called the *transshipment point*) to each of the retailers, so as to minimise the expected long-run cost of operating the system. Our model takes the following costs into consideration.

- inventory-holding costs
- stock-out costs, which include the cost of lost sales and the loss of goodwill
- transportation costs, which include the operating cost of the truck and the cost of transporting goods, which, in turn, depends on the quantities transported.

The random variables governing our system are: the inter-arrival time of customers at each retailer, the quantity demanded by the customers, the service time for each truck, and the travel time between the warehouse and the retailers

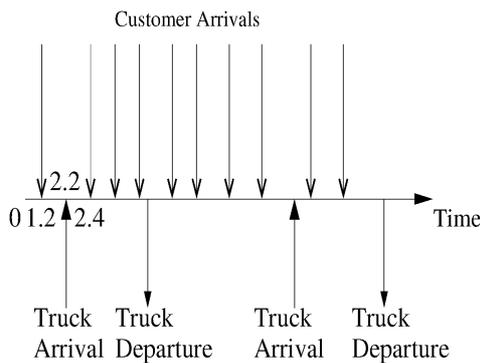
and the same between the retailers. We have assumed that each retailer is distinct and has unique values for system parameters. The rate of arrival of customers, the inventory-holding costs and the stock-out costs are different for each retailer. Our objective is to minimise the average cost per unit time of operating the entire system. We now present details of the simulation model developed.

2.1 A simulation model

The simulator is written using the standard procedure for discrete-event systems (see Chapter 2 of Law and Kelton, 2000). Our programs were written in C language, but could be just as easily written in commercial simulation packages. Figure 2 shows the event clock for the simulator. Let  $q_i$  denote the quantity to be delivered to the  $i$ th retailer and  $n$  be the number of retailers. Then  $\vec{q} = (q_1, q_2, \dots, q_n)$  will represent the vector of delivery quantities. We will now define the following quantities:

- $L_i(\vec{q}, t)$ : the total number of lost sales by time  $t$  in the simulation (time starts at 0) at the  $i$ th retailer, and
- $I_i(\vec{q}, t)$ : the positive inventory at time  $t$  at the  $i$ th retailer
- $C_{tr}$ : the operating cost per unit time of a truck carrying a unit quantity of material (transportation cost)
- $C_l^i$ : the stock-out (lost-sales) cost per unit quantity of sales lost at the  $i$ th retailer, and
- $C_s^i$ : the inventory-holding cost per unit quantity at the  $i$ th retailer.

Figure 2 The event clock of the simulator showing the different types of events



Then, mathematically, the problem is to determine the solution vector  $\vec{q}$  in order to minimise the average cost per unit time i.e., to minimise

$$G(\vec{q}) = C_{tr} \sum_{i=1}^n q_i + \sum_{i=1}^n \lim_{t \rightarrow \infty} \frac{g_i(\vec{q}, t)}{t}, \tag{1}$$

where

$$g_i(\vec{q}, t) = C_l^i L_i(\vec{q}, t) + C_s^i \int_0^t I_i(\vec{q}, \tau) d\tau, \tag{2}$$

such that  $q_i \geq 0$  for  $i = 1, 2, \dots, n$ . The first term in equation (1) denotes the transportation cost per unit time, and the second term denotes the expected cost of stock-outs and holding inventory on a unit time basis. The first term on the right-hand side of equation (2) denotes the costs due to stock-outs and the second term denotes the costs of holding inventory. In equation (2)  $L_i(\vec{q}, t)$  is evaluated with a separate counter for the  $i$ th retailer that is incremented whenever a lost sale occurs at the  $i$ th retailer, while the second quantity is estimated as follows:

$$\int_0^t I_i(\vec{q}, \tau) d\tau = \sum_{j=0}^{\infty} j \Delta_j(i, t)$$

where  $\Delta_j(i, t)$  denotes the total duration of time interval, starting from time 0 until the simulation clock strikes  $t$ , during which the  $i$ th retailer has  $j$  units of positive inventory.  $\Delta_j(i, t)$  can be easily valued in the simulation program.

In the next sub-section, we provide a brief description of the main optimisation technique used in this paper.

2.2 Simultaneous Perturbation (SP)

Gradient-descent methods are defined as those methods in which values of partial derivatives of the objective function are used in the optimisation (search) process. Derivatives can be calculated numerically with a finite difference technique when the closed form is unavailable. The age-old gradient-descent rule (Cauchy, 1847), which is also called the steepest descent rule, can be expressed as follows:

$$\vec{x} \leftarrow \vec{x} - \mu \nabla f(x),$$

$$\nabla f(x) = \left\{ \frac{\partial f(\vec{x})}{\partial x(1)}, \frac{\partial f(\vec{x})}{\partial x(2)}, \dots, \frac{\partial f(\vec{x})}{\partial x(n)} \right\}.$$

In the above equation,  $\partial f(\vec{x}) / \partial x(i)$  denotes a partial derivative of  $f(\cdot)$  with respect to  $x(i)$  where  $f(\cdot) : \mathfrak{R}_+^n \rightarrow \mathfrak{R}$ , and  $\mu$  denotes the step size.

In the finite differences method, the gradient is calculated numerically. This is useful when we do not have the closed form of the objective function. In model-free simulation-based optimisation (Carson and Maria, 1997; Andradóttir, 2002; Fu and Hu, 1997; Gosavi, 2003), we do not have access to the closed form, but since the performance measure of the system may always be evaluated with the simulator, the finite differences approach becomes especially useful. To calculate the gradient numerically, we can use the central differences formula which is given by

$$\frac{\partial f(w)}{\partial x} = \frac{f(w+h) - f(w-h)}{2h}.$$

The difficulty with the above is that the simulator has to be run twice for each decision variable in every iteration of the search algorithm – once to calculate  $f(w+h)$  and once to calculate  $f(w-h)$ . Thus the simulator has to be run  $2n$  times if  $n$  denotes the number of decision variables. Then, as  $n$

increases, the number of runs of the simulator also increases and consequently, the computational burden increases considerably.

A way of working around the above-mentioned difficulty is via the Simultaneous Perturbation (SP) method (Spall, 1992). This is a recent but remarkable development. SP requires only two function evaluations per iteration of the algorithm regardless of the number of decision variables. Hence, in the context of simulation-based optimisation, the method is of special interest. The algorithm is a stochastic search algorithm. What we present below is a slightly modified version of the original algorithm; the modification is in Steps 6 and 7.

*Step 1:* Set  $m = 1$ . Initialise  $\bar{x}^m$ , the solution vector in the  $m$ th iteration, to a feasible solution obtained from the problem specific heuristic. The algorithm will be terminated when the step size  $\mu$  becomes smaller than a pre-determined value  $\mu_{\min}$ . Define a sequence, whose  $m$ th term is  $c^m = 1/m^\zeta$ . We will fix  $\zeta$  to a value in the open interval  $(0, 1)$ . Define  $A$  and  $B$  such that they satisfy:  $0 < A < 1$ ,  $0 < B < 1$  and  $B < A$ .

*Step 2:* Generate a random number  $H(i)$ , for  $i \in 1, 2, \dots, n$ , from a Bernoulli distribution, whose two permissible and equally likely values are 1 and  $-1$ . Compute the values of  $h(i)$  using the following formula.

$$h(i) = H(i)c^m.$$

*Step 3:* Compute  $f(\bar{x}^m + \vec{h})$  and  $f(\bar{x}^m - \vec{h})$  as follows, where

$$f(\bar{x}^m + \vec{h}) = f(x^m(1) + h(1), x^m(2) + h(2), \dots, x^m(n) + h(n)),$$

$$f(\bar{x}^m - \vec{h}) = f(x^m(1) - h(1), x^m(2) - h(2), \dots, x^m(n) - h(n)).$$

*Step 4:* For variables  $i = 1, 2, \dots, n$  we obtain the partial derivatives using

$$\frac{\partial f(\bar{x}^m)}{\partial x^m(i)} \approx \frac{f(\bar{x}^m + \vec{h}) - f(\bar{x}^m - \vec{h})}{2h(i)}.$$

*Step 5:* Compute  $\vec{y}$  using the following rule:

$$\vec{y} \leftarrow x^m(i) - \mu \frac{\partial f(\bar{x}^m)}{\partial x^m(i)} \quad \text{for } i = 1, 2, \dots, n.$$

*Step 6:* If  $f(\vec{y}) < f(\bar{x}^m)$  then set  $\bar{x}^{m+1} = \vec{y}$  and  $\mu \leftarrow \mu B$ . Otherwise, go to Step 7.

*Step 7:* Increase  $m$  by 1, set  $\bar{x}^{m+1} = \bar{x}^m$ , and update  $\mu$  using:  $\mu \leftarrow \mu A$ .

*Step 8:* If  $\mu \leq \mu_{\min}$ , stop. Otherwise, return to Step 2.

SP takes one to the vicinity of a good solution very rapidly. However, one can fine tune the performance of SP by using the well-known meta-heuristic, Simulated Annealing (SA) (Kirkpatrick et al., 1983), after using SP. In other

words, the solution generated by SP can be used as a starting solution for SA. In the next subsection, we discuss the SA algorithm.

### 2.3 Simulated Annealing (SA)

SA starts at a user-provided solution and searches in the neighbouring region for a better solution. The power of SA depends on the neighbourhood search strategy used. It has a so-called *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 system having a lot of local optima. The algorithm accepts the worse solution with a probability which decays with the number of iterations of the algorithm. The algorithm is terminated when the probability of moving to a worse neighbour approaches 0. At each iteration, the best solution obtained so far is stored. As a result, the best of all the solutions tested by the algorithm is returned as the near-optimal solution. An important parameter, called temperature is used to control the exploratory property. It has been established that the algorithm does provide the optimal solution if the temperature is decreased properly (Lundy and Mees, 1986). We will skip the steps in the algorithm referring the reader to any standard text on meta-heuristics. However, we do define our neighbour-generating strategy next.

If  $\bar{x}_{\text{current}}$  denotes the current solution in the SA algorithm, a neighbour (i.e., the new solution) is selected as follows. Let  $\pi$  denote the thickness of the neighbourhood. When a neighbour is to be selected, generate a random number  $u_i$  for  $i = 1, 2, \dots, n$  between 0 and 1 from the uniform distribution  $(0, 1)$ . If  $u_i \leq 0.5$  then for  $i = 1, 2, \dots, n$ , set:

$$x_{\text{neighbour}}(i) = x_{\text{current}}(i) + \pi,$$

and otherwise set:

$$x_{\text{neighbour}}(i) = \max(0, x_{\text{current}}(i) - \pi).$$

In our experiments,  $\pi$  was set to 10.

### 2.4 Problem-specific heuristics

We now describe some problem-specific heuristics that can be used on this problem. These heuristics provide good starting solutions and serve as benchmarks for SA and SP

*The mean demand heuristic.* The Mean Demand Heuristic (MDH) is used in a local dairy product industry in New York. It utilises the mean demand at each retailer. The computations involved can be explained as follows. Let  $T$  denote the average cycle time, i.e., the time required for one round trip and  $d_i$  denote the average demand per customer at the  $i$ th retailer. Then, the optimal quantity  $q_i$  for the  $i$ th retailer is given by  $q_i = Td_i\lambda_i$ , where  $\lambda_i$  denotes the mean rate of arrival of customers at the  $i$ th retailer.

*The newsvendor heuristic.* The popular newsvendor model (see e.g., Nahmias, 2001) can be used to determine the optimal delivery quantity when the commodity is perishable and the demand is normally distributed. By using a normal approximation to the Poisson distribution, we can also apply it to the Poisson distributed demand. The model works as follows. The optimal quantity for the  $i$ th retailer satisfies the following condition:

$$F(q_i) = \frac{C_i^i}{C_i^i + C_s^i},$$

where  $F(\cdot)$  denotes the cumulative distribution function of the demand. If the demand is normally distributed, then the optimal quantity is given by:

$$q_i = \mu_i + Z\sigma_i,$$

where  $Z$  denotes the  $z$ -value in the normal table associated with  $F(\cdot)$  obtained from the above, and  $\mu_i$  is the mean and  $\sigma_i$  the standard deviation of the demand at the  $i$ th retailer. Here  $\mu_i = Td_i\lambda_i$ . For the normal approximation to the Poisson distribution,  $\sigma_i = \sqrt{\mu_i}$  for each  $i$ .

### 3 Computational results

For testing the efficacy of the simulation-optimisation approach, we conducted experiments with a large number of systems. We compared the performance with the two heuristics discussed above. A popular method for simulation-based optimisation uses response surfaces (Law and Kelton, 2000). For a full-blown system with ten or more retailers, the response surfaces would require too many samples. In particular, for a 10-retailer network, one would require at the very least  $2^{10}$  points. Hence we used response surfaces on smaller versions of the problem with retailers.

#### 3.1 A 2-retailer network

The central idea underlying the Response Surface Method (RSM) is to obtain an approximate form of the objective function by simulating the system at a number of points in the solution space and fitting a function to the set of sampled points. Conventional RSM uses regression for fitting the objective function. For regression, it is necessary to assume a metamodel. The need to assume the metamodel makes the application of RSM limited because it is often difficult to recognise an obvious closed form of the function. In such cases, we may need to assume a number of metamodels sequentially and test each of them to obtain a good fit. This can be time-consuming. An alternative is to use neuro-response surfaces (Gosavi, 2003) in which function fitting is done using artificial neural networks (via a standard algorithm like back-propagation), which do not require the meta-model *a priori*.

For the Neuro-Response Surface Method (NRSM) the solution space was sampled at 32 points per retailer.

Thus, the total number of data points used for training the neural net became  $32 \times 32 = 1024$ . We trained the neural net for 3500 iterations. After training the neural net, we generated the cost of the running the system (for a given set of parameters) using the neural net. We next tested if the fit obtained with the neural network was good, using the coefficient of determination (Montgomery et al., 2001).

The systems studied are described in Table 1. The values of the coefficient of determination are provided in Table 2. All values are close to 1, indicating good fits. First, SP was employed on every problem. SP rapidly took us to very good solutions. Thereafter SA was used to improve upon the solution provided by SP by using the solution of SP as the starting solution. In all the systems, the SP-SA combination outperformed the best heuristic. See Table 3 for a comparison of the performance of NRSM and the SP-SA combination. Of the two heuristics, the newsvendor heuristic obtains the better solution. NRSM performs much better than both of the heuristics, but fares worse than the SP-SA combination.

**Table 1** The values of parameters used for each system

System	$C_s^1$	$C_s^2$	$C_i^1$	$C_i^2$	$\lambda_1$	$\lambda_2$
1	0.01	0.009	3.0	3.1	110	90
2	0.01	0.009	1.5	1.4	55	75
3	0.01	0.009	2.0	1.9	110	90
4	0.01	0.009	2.0	1.9	55	75

**Table 2** Coefficient of determination

System	Coefficient of determination
1	0.991
2	0.995
3	0.996
4	0.965

**Table 3** Cost comparison for optimising methods (small cases)

System	Mean-demand heuristic	Newsvendor heuristic	Imp of NRSM over newsvendor heuristic (%)	Imp of SP-SA over newsvendor heuristic (%)
1	17329	3375	63.53	70.37
2	7646	7153	62.14	84.65
3	7646	7008	76.64	84.77
4	23922	20311	86.67	90.34

#### 3.2 A 10-retailer network

In this section, we describe our computational results for a more realistic 10-retailer network. Some modifications had to be made to both SA and SP. We test 18 systems, which are described via Tables 4–6.

**Table 4** Level definitions

System parameters	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5
$C_s-1$	0.01	0.009	0.0085	0.012	0.015
$C_s-2$	0.025	0.022	0.026	0.023	0.024
$C_s-3$	0.05	0.049	0.048	0.056	0.054
$C_l-1$	1.5	1.4	1.45	1.52	1.56
$C_l-2$	2	1.9	1.95	2.13	2.2
$C_l-3$	3	3.1	2.95	3.2	3.16
$\lambda-1$	50	60	70	75	80
$\lambda-2$	65	75	50	62	92

**Table 5** Level definitions (continued from previous table)

System parameters	Retailer 6	Retailer 7	Retailer 8	Retailer 9	Retailer 10
$C_s-1$	0.0086	0.013	0.015	0.01	0.01
$C_s-2$	0.028	0.027	0.022	0.025	0.026
$C_s-3$	0.054	0.055	0.052	0.05	0.046
$C_l-1$	1.46	1.53	1.35	1.6	1.3
$C_l-2$	2.05	1.9	1.85	2.3	2
$C_l-3$	2.89	2.85	3.05	3	2.99
$\lambda-1$	85	90	100	110	120
$\lambda-2$	42	108	129	101	111

**Table 6** System parameters (large cases)

System	$C_s$ -level	$C_l$ -level	$\lambda$ -level
1	1	1	2
2	1	3	2
3	1	1	1
4	1	2	2
5	1	2	1
6	1	3	1
7	2	3	1
8	2	2	2
9	2	2	1
10	2	3	2
11	2	1	2
12	2	1	1
13	3	2	1
14	3	2	2
15	3	3	2
16	3	1	2
17	3	1	1
18	3	3	1

We were able to improve the performance of SP by using a 'reactive' step size. In such a scheme, when the algorithm strikes a worse solution, it is returned to the original solution, and the step-size-reduction rate for that step is increased. The rate is restored to its original value only after

an improved solution is found. Table 7 shows the behaviour of regular S and our reactive SP on System.

**Table 7** Comparison of cost from the two SP methods for system 1

	Regular SP	Reactive SP
Optimised cost	6422	5185

The results obtained for the 10-retailer network are provided in Table 8. Like in the 2-retailer case, an SP-SA combination was used. In all the cases but 5, 13 and 18 SA was able to improve marginally upon the solution of SP. The SP-SA combination outperformed the problem-specific heuristics in all cases. The SP-SA combination took no more than ten minutes for the 10-retailer case on a UNIX Sunblade machine. The percentage improvement of the algorithm over the heuristic is denoted by Imp %.

MDH, although prevalent in a local industry, is not likely to perform well because of its inability to take all the system variability into account. On the other hand, the robust performance of the newsvendor model is an encouraging finding; after all, the newsvendor model is well-understood, and can be easily adapted into existing systems. Thus, managers who prefer not to use 'black-box' methods could use the newsvendor model instead.

**Table 8** Cost-performance comparison (equation (1)) of heuristics and SP-SA

System	Mean-demand heuristic	Newsvendor heuristic	Imp of SP-SA over best heuristic (%)
1	5687	5605	21.86
2	8538	8307	38.23
3	5462	5582	21.22
4	6715	6581	28.76
5	6719	6582	18.73
6	8514	8281	42.05
7	10196	10032	24.43
8	8305	8227	13.31
9	8302	8323	13.01
10	10127	9964	24.51
11	7277	7244	5.55
12	7348	7314	5.32
13	11650	11654	0.44
14	11570	11573	3.58
15	13392	13324	13.57
16	10541	10579	8.19
17	10597	10635	8.15
18	13445	13378	3.17

### 3.3 Non-convexity of the objective function

We wanted to determine if the cost function is convex. An important reason for this is that convex functions are easier to optimise. Hence, we performed experiments

with a two-retailer system in which the quantity delivered to Retailer 2 was varied and that delivered to Retailer 1 was fixed at a constant value. Table 9 provides cost values at some values of the quantity delivered to Retailer 2 ( $Q$ ); the table clearly shows the presence of two local optima. It is thus clear that even for the two-retailer problem, the function is not convex.

**Table 9** Non-convexity of the objective function

	$Q$	Cost
1	4000	2442.878
2	8400	1996.237
3	8650	2391.665
4	14050	1962.567
5	14700	2908.750

### 3.4 A sensitivity analysis

A full factorial experiment was designed to determine which factors affect the optimal objective function obtained from our simulation-optimisation approach. The three factors that we studied are:

- the inventory-holding costs at three levels
- the stock-out costs at three levels
- the rate of arrival of customers at two levels.

The values of the factors are enumerated in Table 6. The results of the ANOVA are presented in Table 10. A number of two-way interactions and one three-way interaction were also considered. The inventory-holding cost and the stock-out cost were found to be significant. This implies that managers interested in using the simulation-optimisation approach should exercise great care in estimating these elements of the cost.

**Table 10** An analysis of variance

Source	Degrees of freedom	Sum of squares	Mean squares	F	P-value
$C_s$	2	310850715	155425357	184.16	0.000
$C_l$	2	27972248	13986124	16.57	0.000
$\lambda$	1	242004	242004	0.29	0.596
$C_s \times C_l$	4	2388655	597164	0.71	0.592
$C_s \times \lambda$	2	457207	228603.5	0.27	0.764
$C_l \times \lambda$	2	147208	73604	0.09	0.917
$C_s \times C_l \times \lambda$	4	926565	231641	0.27	0.893
Error	36	30383177	843977		
Total	53	373367780			

## 4 Conclusions

Determining the optimal quantities to be dispatched to each retailer from the warehouse is a long-standing problem in the industry. In the past, customers placed orders and

warehouses met them as and when needed. This led to a peak of demands on certain days of the week. With the advent of vendor managed inventory systems, the warehouse manager sends material at regular time intervals on the basis of average demand at each retailer. The problem is complicated by randomness arising from customer behaviour and transportation delays. There are three major costs associated with managing this system: the transportation cost, the inventory-holding cost and the cost due to lost sales. Accommodating all the governing random variables and costs in one mathematical model is a challenging task, and hence, much of the literature makes simplifying assumptions to develop tractable mathematical models. In this paper, we have developed a simulation model that accounts for most features of a real-world system, and have used optimisation techniques that can be combined efficiently with simulation to generate solutions. On the basis of our experimentation, we find that a combination of SP and SA is able to outperform a heuristic used in the real world, namely, MDH. An interesting finding is that the newsvendor model can be adapted to develop an efficient heuristic for solving this problem. Although it was outperformed by simulation-optimisation methods, its encouraging performance indicates that managers who do not intend to simulate the entire network can resort to the newsvendor model for obtaining reliable solutions.

The solutions developed by the simulation-optimisation approach can be directly incorporated into the decision-making technology of warehouse managers. Simulation is a widely used tool in the industry, and hence the solution methodology presented here should find ready acceptance in the industry. Furthermore, the simulation-optimisation approach can be run on any personal computer with great ease.

Several extensions to this research can be envisaged. Firstly, a possible alternative is to develop an approximate Markov chain model for this system, and use it for optimisation. Secondly, one can consider the entire network and incorporate the effects of transportation from the regional warehouse to the local warehouses into the model.

## Acknowledgements

This research was partially supported by a research grant to the second author from the National Science Foundation. The authors are grateful to the anonymous reviewers of this paper and to the special-issue editor, Dr. Sanjay Jain, who made many suggestions for improving this paper.

## References

Andradóttir, S. (2002) ‘Simulation optimization: integrated research and practice’, *INFORMS Journal on Computing*, Vol. 14, No. 3, pp.216–219.

Berman, O. and Larson, R. (2001) ‘Deliveries in a inventory/routing problem using stochastic dynamic programming’, *Transportation Science*, Vol. 35, No. 2, pp.192–213.

- Carson, Y. and Maria, A. (1997) 'Simulation optimization: methods and applications', *Proceedings of the 1997 Winter Simulation Conference*, Atlanta, GA, pp.118–126.
- Cauchy, A. (1847) 'Méthode Générale pour la Résolution des Systèmes d'Équations Simultanées', *Comp. Rend. Acad. Sci.*, Paris, pp.536–538.
- Clark, A. and Scarf, H. (1960) 'Optimal policies for a multi-echelon inventory problem', *Management Science*, Vol. 6, No. 4, pp.475–490.
- Eppen, G. and Schrage, L. (1981) 'Centralized ordering policies in a multi-warehouse system with lead times and random demand', *Multi-Level Production/Inventory Control Systems: Theory and Practice*, Amsterdam, North Holland, p.169.
- Federgruen, A. and Zipkin, P. (1984) 'Approximations of dynamic, multi-location production and inventory problems', *Management Science*, Vol. 30, No. 1, pp.69–84.
- Fu, M. and Hu, J. (1997) *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*, Kluwer Academic Publishers, Boston, MA.
- Gosavi, A. (2003) *Simulation-based Optimization: Parametric Optimization Techniques and Reinforce Ment Learning*, Kluwer Academic Publishers, Massachusetts, USA.
- Hammer, M. (2001) 'The superefficient company', *Harvard Business Review*, September–October Issue, pp.82–91.
- Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) 'Optimization by simulated annealing', *Science*, Vol. 220, pp.671–680.
- Kleywegt, A.J., Nori, V.S. and Savelsbergh, M.W.P. (2002) 'The stochastic inventory routing problem with direct deliveries', *Transportation Science*, Vol. 36, No. 1, pp.94–118.
- Kumar, A., Schwarz, L.B. and Ward, J.E. (1995) 'Risk-pooling along a fixed delivery route using a dynamic inventory-allocation policy', *Management Science*, Vol. 41, No. 2, pp.344–362.
- Law, A. and Kelton, W. (2000) *Simulation Modeling and Analysis*, 3rd ed., McGraw-Hill, New York, NY.
- Lundy, M. and Mees, A. (1986) 'Convergence of the annealing algorithm', *Mathematical Programming*, Vol. 34, pp.111–124.
- McGavin, E.J., Schwarz, L.B. and Ward, J.E. (1993) 'Two inventory allocation policies in a one warehouse N-identical retailer distribution system', *Management Science*, Vol. 39, No. 9, pp.1092–1107.
- Minkoff, A.S. (1993) 'A Markov decision model and decomposition heuristic for dynamic vehicle dispatching', *Operations Research*, Vol. 41, No. 1, pp.77–90.
- Montgomery, D., Ruer, G. and Hubele, N. (2001) *Engineering Statistics*, 2nd ed., John Wiley and Sons, New York, NY, USA.
- Nahmias, S. (2001) *Production and Operations Analysis*, McGraw-Hill, Irwin, USA.
- Nahmias, S. and Smith, S.A. (1994) 'Optimizing inventory levels in a two-echelon retailer system with partial lost sales', *Management Science*, Vol. 40, No. 5, pp.82–96.
- Spall, J.C. (1992) 'Multivariate stochastic approximation using a simultaneous perturbation gradient approximation', *IEEE Transactions on Automatic Control*, Vol. 37, pp.332–341.