Optimization of (s, S) Inventory Systems with Random Lead Times and a Service Level Constraint

Sridhar Bashyam • Michael C. Fu
KPMG Peat Marwick LLP, 2300 Clarendon Boulevard, Arlington, Virginia 22201
The Robert H. Smith School of Business, University of Maryland, College Park, Maryland 20742

A major assumption in the analysis of (s, S) inventory systems with stochastic lead times is that orders are received in the same sequence as they are placed. Even under this assumption, much of the work to date has focused on the unconstrained optimization of the system, in which a penalty cost for unsatisfied demand is assigned. The literature on constrained optimization, wherein a service level requirement needs to be met, is more sparse. In this paper, we consider the *constrained* optimization problem, where orders are *allowed to cross in time*. We propose a feasible directions procedure that is simulation based, and present computational results for a large number of test cases. In the vast majority of cases, we come within 5% of estimated optimality.

((s, S) *Inventory Systems*; *Random Lead Times*; *Service Level Constraint*; *Constrained Simulation Optimization*; *Feasible Directions Search*; *Perturbation Analysis*)

1. Introduction

The case of random lead times for (s, S) inventory systems has been handled under the key assumption that orders arrive in the same sequence as they are placed (Ehrhardt 1984, Zipkin 1986, Zheng and Federgruen 1991). Specifically, if we define the following random variables:

$W(t)$ = inventory level (stock on hand minus backorders) at time t,

$I(t)$ = inventory position ($W(t)$ plus outstanding orders) at time t,

L = order lead time,

$D(t|L)$ = demand during the interval $(t, t + L]$,

then the analysis typically proceeds by first deriving the probability distribution of $I(t)$ via renewal theory, and then using the equation

$$W(t + L) = I(t) - D(t|L) \tag{1}$$

to determine the probability distribution of $W(t)$. Much of the work reported on (s, S) systems uses this information on $I(t)$ and $W(t)$ to find an (s, S) pair that minimizes (or approximately minimizes) a cost function that, in addition to setup and holding costs, assigns a penalty cost on backlogged demand. A review of a large number of optimal and approximately optimal algorithms to carry out this *unconstrained* optimization can be found in Porteus (1988). However, if orders are allowed to cross—which, in general, would be the case when there are multiple suppliers—(1) is no longer valid, and even the above unconstrained problem appears to become analytically intractable. To our knowledge, the literature does not report any work related to this more general problem.

In spite of the huge body of literature that has been developed on the cost minimization problem, it has been widely recognized that penalty costs, and in particular, the cost of losing customer goodwill, are usually difficult to assess. It is largely a result of this that service level measures are very popular in practice (Lee and Nahmias 1993). Under this approach, the objective is to determine an (s, S) pair that minimizes a cost function, defined only in terms of setup and holding costs, *subject*

0025-1909/98/4412/S243$05.00

to the constraint that the solution satisfies a prescribed level of customer service. Unfortunately, from an analytical viewpoint, this *constrained* optimization problem introduces the additional difficulty of having to deal with the probability distribution of backlogged demand. The purpose of our work is to incorporate two critical features in the determination of optimal settings of (s, S):

- service level constraint, and
- general random lead times that allow for order crossings.

A review of the literature on service level constraints reveals that the two most relevant works are Schneider and Ringuest (1990) and Tijms and Groenevelt (1984). Schneider and Ringuest consider a periodic review system operating under a γ-service level measure, where $(1 - \gamma)$ is the fraction of demand on backorder each period. They construct a Lagrangian function, and propose algorithms that give nearly exact solutions to the three first-order conditions $\partial\Lambda/\partial\theta = 0$, where Λ is the Lagrangian function, and $\theta = s, S$, and λ. However, they assume a fixed lead time.

On the other hand, Tijms and Groenevelt (1984) consider a general class of (s, S) models, covering both the periodic and continuous review cases, and allowing random lead times, though prohibiting order crossings. They compare their proposed approximation with ''optimal'' results obtained from a Lagrangian method, carried out via a line search on the Lagrange multiplier. However, they assume that $Q = S - s$ is predetermined (e.g., via the EOQ formula), and focus only on the determination of the reorder level s.

Another limitation of these approaches is their range of validity: both explicitly assume that $S - s$ be sufficiently large compared to the average period demand μ_D; in particular, that $S - s \geq 1.50\mu_D$. While Schneider and Ringuest suggest alternative schemes when this condition is violated, Tijms and Groenevelt specifically mention that their approximations are not warranted when this condition is violated. For example, if we arbitrarily fix the setup cost $K = 36$, and the per-period holding cost $h = 1$, and if Q is computed using the EOQ formula, then their algorithm is valid only when the average demand per period is less than or equal to 32. Moreover, when orders are allowed to cross, our simulation results indicate that the approximation methods might perform quite poorly.

In this paper, we consider a periodic review (s, S) inventory with random lead times that allow for orders to cross in time. We adopt the ''fill-rate'' service level measure used by Tijms and Groenevelt, defined as the fraction of demand that is met directly from stock on hand. Our objective is to determine values of s and $Q = S - s$ that minimize the average setup and holding cost per period, subject to the constraint that the fill-rate is above a prescribed level. Due to the analytic intractability caused by order crossings, we adopt a simulation-based approach to the problem. A review of the state of the art in optimization via simulation (Fu 1994b, see also Safizadeh 1990, and Jacobson and Schruben 1989) indicates that computationally efficient simulation-based gradient estimation methods have been developed for a variety of systems, which makes the use of gradient-based search methods a very viable tool to optimize analytically intractable models (another possible approach is response surface methodology). We use the gradient estimators of cost with respect to s and Q derived by Fu and Hu (1994) and also derive gradient estimators for the service level measure. Detailed expositions on gradient estimation for (s, S) inventory systems can also be found in Fu (1994a) and Bashyam and Fu (1994) (see also the monographs by Ho and Cao 1991, Glasserman 1991, and Fu and Hu 1997 for more general coverage of gradient estimation).

In terms of simulation optimization algorithms, a rich body of literature exists for the unconstrained version of the problem (see, for example, L'Ecuyer et al. 1994), but the literature on constrained optimization with ''noisy'' constraints is limited. Since our problem's service level constraint requires estimation of the service level measure by simulation, it falls into the latter category of problems. When the constraints are of known functional form, a number of solution techniques are available (Kushner and Clark 1978), but the only reported algorithm we know of that handles noisy constraints is the Lagrangian approach in Kushner and Clark (1978). There are several reasons that motivate us to consider alternative solution methods. The most significant among these reasons is the fact that the Lagrangian method guarantees a *feasible* solution only in the limit, which could be an unacceptable feature in practical application.

We propose an algorithm based on the feasible directions approach from nonlinear programming. The algorithm has several desirable properties:

- It is a direct search process that is very amenable to fine control.
- It converges relatively quickly in a stable manner.
- It gives the user several good feasible solutions as it evolves.
- It incorporates a simple but robust scheme for parameter settings.

In computational experiments involving various degrees of order crossings, our algorithm came within 5% of optimality in 95% of the cases, and within 2% of optimality in 68% of the cases. In contrast, the analytical methods produced optimality gaps in excess of 10% for over 75% of the cases. Thus, we believe that we offer a computationally viable algorithm that satisfactorily handles the case of order crossing in instances where analytical models can be expected to perform very poorly.

The rest of the paper is organized as follows. In §2, we formally specify the model and define the problem of interest. In §3, we extend the analysis of Fu and Hu (1994) to derive PA gradient estimators of the service level measure with respect to s and Q. Readers interested mainly in the optimization algorithm may skip this section without any loss in continuity. In §4, we describe the proposed optimization algorithm based on a feasible directions approach, including the incorporation of some simple preprocessing ideas that provides the gradient search routine with a good starting point. Computational results on a large number of cases covering various demand and lead time distributions are reported in §5. Section 6 contains concluding remarks and directions for further research.

2. Problem Definition

We consider an infinite horizon periodic review inventory system with continuous-valued i.i.d. demands and full backlogging. The basic sequence of events in each period is as follows: Orders are received at the beginning of the period, the demand for the period is subtracted out, then an order review is carried out at the end of the period. The *inventory level* in period n is defined as the on hand stock minus backorders, and observed after demand subtraction, and the *inventory position* is the inventory level plus any outstanding orders. We define the following:

D_n = demand in period n, i.i.d. with p.d.f. $g(\cdot)$ and c.d.f. $G(\cdot)$,

I_n = inventory position in period n,

W_n = inventory level in period n,

V_n = inventory level in period n before demand subtraction $(=W_n + D_n)$,

Y_n = demand in period n not satisfied from on hand stock $(=(D_n - V_n^+)^+)$,

L_i = lead time of ith order placed,

h = per period holding cost,

K = per order setup cost,

u = per unit order cost,

where $x^+ = \max(x, 0)$.

Ordering decisions are made according to the well-known (s, S) policy: If $I_n < s$, an order for the amount $S - I_n$ is placed; otherwise, no action is taken. The lead times L_i for orders placed are assumed to be integer valued i.i.d. random variables. Under our convention, an order with lead time l placed in period n will arrive at the beginning of period $n + l + 1$. As indicated in the introduction, the performance of the system will be evaluated by a cost function and a service level measure, where the cost measure considers only setup and holding costs, and the service level measure tracks the extent of backlogging in the system. Defining the cost for a single period n by

$$C_n = hW_n^+ + I\{I_n < s\}(K + u(S - I_n)),$$

where $I\{\cdot\}$ is the indicator function, we define the expected cost per period:

$$\mathcal{C}_N = E[\bar{C}_N] = E\left[\frac{1}{N}\sum_{n=1}^{N} C_n\right], \tag{2}$$

where $\bar{C}_N$ is the *sample* performance measure. The service level measure is the fraction of the total demand that is *not satisfied* from on hand stock, the complement of the fill-rate measure considered in Tijms and Groenevelt (1984), given by

$$\mathcal{J}_N = E[J_N], \quad \text{where } J_N = \frac{\sum_{n=1}^{N} Y_n}{\sum_{n=1}^{N} D_n}. \tag{3}$$

Denoting $\mathcal{C}$ and $\mathcal{J}$ as the infinite horizon measures obtained by letting $N \to \infty$ in the corresponding measures,

and defining $\theta = (s, Q)$, the optimization problem is the following mathematical program (**P**):

$$\min_{\theta \in \mathcal{R} \times \mathcal{R}^+} \mathcal{C}(\theta)$$

$$\text{subject to } \mathcal{J}(\theta) \leq \beta.$$

The gradient estimators of $\mathcal{C}$ and $\mathcal{J}$ with respect to s and Q play a central role in the simulation optimization algorithm we propose in §4. In the next section, we derive these estimators using the technique of perturbation analysis. Since the estimators for $\partial \mathcal{C}/\partial s$ and $\partial \mathcal{C}/\partial Q$ have already been specified by Fu and Hu (1994), we focus here on extending their analysis to develop estimators for $\partial \mathcal{J}/\partial s$ and $\partial \mathcal{J}/\partial Q$.

3. Gradient Estimation

In this section, we use the gradient estimation technique of perturbation analysis to derive sample path estimators for $\partial \mathcal{J}_N/\partial s$ and $\partial \mathcal{J}_N/\partial Q$. These estimators then provide estimates for the infinite horizon gradients $\partial \mathcal{J}/\partial s$ and $\partial \mathcal{J}/\partial Q$ by taking N sufficiently large. Although the development here is largely self-contained, certain technical details have been omitted (cf., e.g., Bashyam and Fu 1994). In addition, implicit in the use of the finite horizon gradients for the infinite horizon problem is the assumption of strong consistency, requiring the interchange of two limits; although this interchange is not addressed here, the corresponding proofs for the average cost function can be found in Fu and Hu (1994). The interested reader is also referred to the monograph by Fu and Hu (1997) for a more detailed exposition on the perturbation analysis techniques used in this section.

In our derivations, we will refer to a **nominal path** as a sample path operating at the nominal values of the parameters (s, Q), whereas a **perturbed path** will refer to a sample path operating with one of the values perturbed, in particular $(s + \Delta s, Q)$ in §3.1 and $(s, Q + \Delta Q)$ in §3.2. To distinguish quantities in the respective paths, the perturbed parameter will be displayed explicitly, whereas the other parameter will be omitted, e.g., $I_n(s)$ vs. $I_n(s + \Delta s)$ in §3.1, and $I_n(Q)$ vs. $I_n(Q + \Delta Q)$ in §3.2. Finally, all estimators take the limit as the perturbation goes to zero ($\Delta s \to 0$ or $\Delta Q \to 0$), so that in the actual implementation there is no perturbation; it is merely an artifact of the presentation of the derivations.

3.1. Estimation with Respect to s

By definition, we have

$$\frac{\partial \mathcal{J}_N}{\partial s} = \frac{\partial E[J_N]}{\partial s} = \lim_{\Delta s \to 0} \frac{E[J_N(s + \Delta s) - J_N(s)]}{\Delta s}.$$

If both paths start with an initially ''full'' inventory (at the order-up-to level), and both are subjected to identical sequences of demand sizes and lead times, then keeping Q fixed means that the two paths are offset from each other by Δs, but are otherwise identical in shape. In the context of perturbation analysis, we say that the *event order sequences* are identical, from which the dominated convergence theorem can be used to justify that the interchange of limit and expectation operations in

$$\frac{\partial E[J_N]}{\partial s} = E\left[\frac{\partial J_N}{\partial s}\right] \tag{4}$$

is valid, so that the *infinitesimal perturbation analysis* (IPA) estimator $\partial J_N/\partial s$ is an *unbiased* estimator of $\partial \mathcal{J}_N/\partial s$. To derive an expression for the sample path estimator, we note that for all n, s, and Δs,

$$V_n(s + \Delta s) = V_n(s) + \Delta s, \quad \text{so } \frac{dV_n}{ds} = 1.$$

Since $Y_n = (D_n - V_n^+)^+$, we have by straightforward differentiation (NB: at the point $D_n - V_n^+ = 0$, the derivative technically does not exist, but this point occurs w.p. 0 since demands are continuous),

$$\frac{dY_n}{ds} = \begin{cases} \dfrac{d}{ds}(D_n - V_n^+) & \text{if } D_n - V_n^+ > 0, \\ 0 & \text{otherwise,} \end{cases}$$

and since D_n is independent of s, we have

$$\frac{d}{ds}(D_n - V_n^+) = -\frac{dV_n^+}{ds} = \begin{cases} -1 & \text{if } V_n > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Combining the two results, we get

$$\frac{\partial Y_n}{\partial s} = \begin{cases} -1 & \text{if } V_n > 0,\ V_n - D_n < 0, \\ 0 & \text{otherwise,} \end{cases}$$

so from (3), it follows that

$$\frac{\partial J_N}{\partial s} = \frac{\sum_{n=1}^N \partial Y_n/\partial s}{\sum_{n=1}^N D_n},$$

which results in the following IPA estimator for $\partial \mathcal{J}_N / \partial s$:

$$\left(\frac{\partial \mathcal{J}_N}{\partial s}\right)_{\text{IPA}} = -\frac{|\mathcal{N}_{\text{IPA}}|}{\Sigma_{n=1}^{N} D_n}, \tag{5}$$

where $\mathcal{N}_{\text{IPA}} = \{n \leq N: V_n > 0, V_n - D_n < 0\}$. The set $\mathcal{N}_{\text{IPA}}$ contains those periods where the fraction of demand satisfied from on hand stock is strictly positive and less than 1. As can be seen, the IPA estimator given by (5) can be implemented very efficiently.

3.2. Estimation with Respect to Q

When Q is varied by ΔQ, the nominal path operates at s and S, whereas the perturbed path operates at s and $S + \Delta Q$. Now consider a nominal sample path such that for some n, we have $I_n(Q) = s - \delta < s$, where $\delta \in (0, \Delta Q)$, i.e., an order is placed in period n in the nominal path. Then, in the perturbed path, we have $I_n(Q + \Delta Q) = s + \Delta Q - \delta > s$, i.e., no order is placed in period n in the perturbed path. This indicates *a change in the ordering sequence* between the two paths, which typically results in a significant change (of $O(1)$) in the performance measure. Under these conditions, an interchange of the kind specified in (4) is no longer valid, and the IPA estimator for this case is not unbiased by itself. As in Fu (1994), Bashyam and Fu (1994), and Fu and Hu (1994, 1997), we use *smoothed perturbation analysis* to derive an unbiased estimator. We begin by introducing some additional sample path notation:

$U_n = (I_n + D_n)$ = inventory position in period n before demand subtraction,

$Z_n = U_n - s$,

$e_n = \begin{cases} 1 & \text{if an order is placed in the } n\text{th period,} \\ 0 & \text{otherwise,} \end{cases}$

$\mathcal{O}_n = \{e_1(Q) = e_1(Q + \Delta Q), \ldots, e_{n-1}(Q) = e_{n-1}(Q + \Delta Q), e_n(Q) \neq e_n(Q + \Delta Q)\}$,

$\bar{\mathcal{O}} = \{e_1(Q) = e_1(Q + \Delta Q), e_2(Q) = e_2(Q + \Delta Q), \ldots, e_N(Q) = e_N(Q + \Delta Q)\}$,

$\eta(n)$ = number of orders placed up to and including period n,

$\mathcal{H}_n = \{D_1, D_2, \ldots, D_{n-1}, L_1, L_2, \ldots, L_{\eta(n)}\}$.

$\mathcal{O}_n$ defines the event that the first ordering sequence change (depending on s, Q, and ΔQ) between the nominal and perturbed paths occurs in period n, while $\bar{\mathcal{O}}$ represents no ordering sequence change (due to the perturbation) over the entire sample path. The set $\mathcal{H}_n$, which provides the history of the sample path up to the beginning of period n, will be key in the derivation of the conditional expectation estimator. Since the event set $\{\bar{\mathcal{O}}, \mathcal{O}_1, \ldots, \mathcal{O}_N\}$ provides a partitioning for the set of possible sample paths, we can write

$$\frac{\partial \mathcal{J}_N}{\partial Q} = \lim_{\Delta Q \to 0} \frac{E[\Delta J_N]}{\Delta Q} = \lim_{\Delta Q \to 0} \left(\frac{E[\Delta J_N | \bar{\mathcal{O}}]}{\Delta Q} P[\bar{\mathcal{O}}] \right) + \lim_{\Delta Q \to 0} \left(\sum_{n=1}^{N} E[\Delta J_N | \mathcal{O}_n] \frac{P[\mathcal{O}_n]}{\Delta Q} \right), \tag{6}$$

where $\Delta J_N \equiv J_N(Q + \Delta Q) - J_N(Q)$. The IPA estimator given by (5) is an unbiased estimator of the first term in the right-hand side of (6). Conditioning on $\mathcal{H}_n$, we rewrite the second term as

$$\lim_{\Delta Q \to 0} \left(\sum_{n=1}^{N} E[\Delta J_N | \mathcal{O}_N] \frac{P[\mathcal{O}_n]}{\Delta Q} \right) = \lim_{\Delta Q \to 0} \left(E\left[\sum_{n=1}^{N} E[\Delta J_N | \mathcal{O}_n, \mathcal{H}_n] \frac{P[\mathcal{O}_n | \mathcal{H}_n]}{\Delta Q} \right] \right).$$

Under appropriate conditions—in particular, if the period demand has finite mean with a Lipschitz continuous c.d.f., i.e., there exists a (Lipschitz) constant $K_l > 0$ such that $|G(x) - G(y)| \leq K_l |x - y| \ \forall x, y$—then the dominated convergence theorem can be used to justify the interchange (cf., e.g., Bashyam and Fu 1994 or Fu and Hu 1994)

$$\lim_{\Delta Q \to 0} \left(E\left[\sum_{n=1}^{N} E[\Delta J_N | \mathcal{O}_n, \mathcal{H}_n] \frac{P[\mathcal{O}_n | \mathcal{H}_n]}{\Delta Q} \right] \right) = E\left[\lim_{\Delta Q \to 0} \left(\sum_{n=1}^{N} E[\Delta J_N | \mathcal{O}_n, \mathcal{H}_n] \frac{P[\mathcal{O}_n | \mathcal{H}_n]}{\Delta Q} \right) \right],$$

so that

$$\sum_{n=1}^{N} \lim_{\Delta Q \to 0} E[\Delta J_N | \mathcal{O}_n, \mathcal{H}_n] \lim_{\Delta Q \to 0} \frac{P[\mathcal{O}_n | \mathcal{H}_n]}{\Delta Q} \tag{7}$$

is an unbiased estimator for the second term on the right-hand side of (6). Thus, our smoothed perturbation analysis (SPA) estimator for $\partial \mathcal{J}_N / \partial Q$ is given by

$$\left(\frac{\partial \mathcal{J}_N}{\partial Q}\right)_{\text{SPA}} = -\frac{|\mathcal{N}_{\text{IPA}}|}{\Sigma_{n=1}^{N} D_n} + \sum_{n=1}^{N} \lim_{\Delta Q \to 0} E[\Delta J_N | \mathcal{O}_n, \mathcal{H}_n] \lim_{\Delta Q \to 0} \frac{P[\mathcal{O}_n | \mathcal{H}_n]}{\Delta Q}.$$

(8)

Given $\mathcal{H}_n$ and ΔQ, an ordering change in period n can occur if and only if $Z_n < D_n \leq Z_n + \Delta Q$. The probability term in (7) is therefore

$$\lim_{\Delta Q \to 0} \frac{P[\mathcal{O}_n | \mathcal{H}_n]}{\Delta Q} = \lim_{\Delta Q \to 0} \frac{P[Z_n < D_n \leq Z_n + \Delta Q]}{\Delta Q} = g(Z_n),$$

where $g(\cdot)$ is the p.d.f. of period demand. The conditional expectation term in (8) represents the change in expected sample performance, due to an ordering sequence change in period n, under the limit $\Delta Q \to 0$. The bulk of the computational burden of the gradient estimation comes from the computation of this term. We begin by defining the following four sample paths (in what follows, we use the superscripts ''+'' and ''−'' to denote ''infinitesimally above'' and ''infinitesimally below,'' respectively):

1. *Nominal path* (NP): generated by the sequence $\{D_1, \ldots, D_N, L_1, \ldots, L_{\eta(N)}\}$;
2. nth *degenerated nominal path* (DNP_n): generated by the sequence $\mathcal{H}_n \cup \{D_n = Z_n^+, D_{n+1}, \ldots, D_N, L_1', L_2', \ldots\}$, and referenced by the superscript '$n1$';
3. nth *perturbed path* (PP_n): generated by the sequence $\mathcal{H}_n \cup \{D_n = Z_n^-, D_{n+1}, \ldots, D_N, L_1', L_2', \ldots\}$, and referenced by the superscript '$n2$';
4. nth *modified perturbed path* (MPP_n): generated by the sequence $\mathcal{H}_n \cup \{D_n = Z_n^-, D', D_{n+1}, \ldots, D_{N-1}, L_1', L_2', \ldots\}$, and referenced by the superscript '$n3$';

where $\{L_i'\}$ represents a sequence of i.i.d. order lead times distinct from (albeit equal in distribution to) the original sequence $\{L_i\}$. Thus, the four sample paths are coupled, with the last three paths differing from the nominal path by the value of D_n and by the subsequent order lead times. In addition, subsequent demands in MPP_n are offset by the inclusion of an inserted generic demand D' for period $n + 1$. The expectation term is now given by

$$\lim_{\Delta Q \to 0} E[\Delta J_N | \mathcal{O}_n, \mathcal{H}_n] = E[J_N^{n2} - J_N^{n1}] = E[J_N^{n3} - J_N^{n1}], \quad (9)$$

where the first equality holds by definition, and the second one is justified by the fact that PP_n and MPP_n are equal in probability law. We now develop an estimator for $E[J_N^{n3} - J_N^{n1}]$ for large N. From (9), and from the strong law of large numbers, we get

$$E[J_N^{n3} - J_N^{n1}] = E\left[\frac{\Sigma_{i=1}^N Y_i^{n3}}{D' + \Sigma_{i=1}^{N-1} D_i} - \frac{\Sigma_{i=1}^N Y_i^{n1}}{\Sigma_{i=1}^N D_i}\right]$$

$$\approx \frac{E[\Sigma_{i=1}^N (Y_i^{n3} - Y_i^{n1})]}{NE[D]}. \quad (10)$$

Henceforth, we will use ''$\approx$'' to denote the large N approximation. For path DNP_n, let $n_1, n_2, \ldots, n_l$ be the periods in which outstanding orders in period n (including the one just placed) are due to arrive, and define $a_n = \max(n_1, n_2, \ldots, n_l)$. By construction, the portion of MPP_n in the interval $[a_n + 1, N]$ is identical to that of DNP_n over $[a_n, N - 1]$. To show this, we begin by noting that the inventory positions in DNP_n and MPP_n at the beginning of periods $n + 1$ and $n + 2$, respectively, are equal to S. Let us first consider the simpler case where no additional orders are placed in DNP_n during the periods $n + 1$ to $a_n - 1$. Accordingly, no orders will be placed in MPP_n during the periods $n + 2$ to a_n, since it experiences the same demand stream (in particular, $D_{n+1}, D_{n+2}, \ldots, D_{a_n-1}$) as DNP_n in this interval. By the definition of a_n it follows that DNP_n would have received all its outstanding orders by the beginning of period a_n. Similarly, MPP_n will have received its outstanding orders no later than the beginning of period $a_n + 1$ (in fact, if the lead time for the order placed by DNP_n in period n was less than a_n, MPP_n would also receive all its outstanding orders at the beginning of period a_n). Thus,

$$V_{a_n}^{(\text{DNP}_n)} = V_{a_n+1}^{(\text{MPP}_n)} = S - \sum_{i=n+1}^{a_n-1} D_i.$$

The result holds even if order placements and subsequent receipts occur during the interval $[n + 1, a_n - 1]$ for DNP_n, since they will occur identically in MPP_n during the interval $[n + 2, a_n]$.

Since both paths are identical over $[1, n]$, we get the following simplification for (10):

$$E[J_N^{n3} - J_N^{n1}] \approx \frac{E[Y_{a_n}^{n3} + \Sigma_{i=n+1}^{a_n-1} (Y_i^{n3} - Y_i^{n1})]}{NE[D]} - \frac{E[Y_N^{n1}]}{NE[D]}.$$

Finally, from the ergodicity arguments provided in Fu and Hu (1994), we get

$$\frac{E[Y_N^{n1}]}{NE[D]} \approx \frac{E[Y_N]}{NE[D]} \approx \frac{\mathcal{J}_N}{N},$$

so the final form of the conditional expectation term is

$$\lim_{\Delta Q \to 0} E[\Delta J_N | \theta_n, \mathcal{H}_n] \approx \frac{E[Y_{a_n}^{n3} + \Sigma_{i=n+1}^{a_n-1} (Y_i^{n3} - Y_i^{n1})]}{NE[D]} - \frac{J_N}{N}.$$

Thus, the final estimator is given by

$$\left(\frac{\partial J_N}{\partial Q}\right)_{\text{SPA}} = -\frac{|\mathcal{N}_{\text{IPA}}|}{\Sigma_{n=1}^{N} D_n} - \frac{J_N}{N} \sum_{n=1}^{N} g(Z_n) + \frac{1}{NE(D)} \sum_{n=1}^{N} g(Z_n) \left[Y_{a_n}^{n3} + \sum_{i=n+1}^{a_n-1} (Y_i^{n3} - Y_i^{n1}) \right], \quad (11)$$

In terms of implementation, estimation of the bracketed term in (11) does require generating additional sample path quantities related to DNP_i and MPP_i. However, if the lead times are bounded from above by a constant M, we need not handle more than M pairs of DNP_i and MPP_i at any stage in the simulation. This indicates that the additional computational effort required is linear in the upper bound of lead times. Finally, we point out that when lead times are noncrossing (which includes the deterministic case), drastic simplifications exist for estimating the conditional expectation term, and in fact, all the required quantities can be estimated from NP itself (see Fu and Hu 1994).

A similar analysis for the cost function gives the following estimators (Fu and Hu 1994):

$$\left(\frac{\partial C_N}{\partial s}\right)_{\text{IPA}} = \frac{h|\mathcal{N}^+|}{N}, \quad \mathcal{N}^+ = \{n: W_n > 0\}, \quad (12)$$

$$\left(\frac{\partial C_N}{\partial Q}\right)_{\text{SPA}} = \frac{h|\mathcal{N}^+|}{N} + \frac{1}{N} \sum_{n=1}^{N} g(Z_n) \left[C_{a_n}^{n3} + \sum_{i=n+1}^{a_n-1} (C_i^{n3} - C_i^{n1}) \right] + \frac{uE(D) - \bar{C}_N}{N} \sum_{n=1}^{N} g(Z_n). \quad (13)$$

4. The Proposed Simulation Optimization Algorithm

We present an adaptation of the classical feasible directions method for nonlinear programs with inequality constraints (Zoutendijk 1960). Our algorithm is, to our knowledge, the first application of the feasible directions idea to carry out constrained optimization via simulation. We have not here attempted to characterize the convergence properties of the algorithm; instead we subject the algorithm to extensive heuristic testing, with the results presented in §5. Unlike most of the recent applications of stochastic approximation methods for optimization via simulation (cf., Fu 1994b) where short simulation runs are used to generate large numbers of iterations, our algorithm emulates its nonlinear programming roots by using long simulation runs and far fewer iterations, and by placing emphasis on preprocessing methods to generate good starting points.

We begin with a broad overview of the procedure. Specific details of the actual implementation and numerical results will be presented in the following sections. We first introduce some vector/matrix notation, where all vectors being defined are of the column variety. Let $\nabla C(\theta)$ and $\nabla J(\theta)$ be the gradient vectors evaluated at θ, and let $\{A_n\}$ be a family of diagonal matrices with diagonal elements $a_n(s)$ and $a_n(Q)$ that represent the step-size sequences for the iterative updates. For the sake of convenience (slightly abusing notation), we shall refer to the PA gradient estimators, evaluated at the parameter setting θ, as $\partial C(\theta)/\partial s$, $\partial C(\theta)/\partial Q$, $\partial J(\theta)/\partial s$, and $\partial J(\theta)/\partial Q$.

Denote a vector x normalized to unit length by

$$\langle x \rangle = \frac{x}{|x|}.$$

The algorithm we consider generates the sequence $\{\theta_n\}$ as

$$\theta_{n+1} = \theta_n + A_n D(\theta_n), \quad (14)$$

where the normalized direction vector $D(\theta_n)$ is given by

$$D(\theta) = \begin{cases} -\langle \nabla C(\theta) \rangle & \text{if } J(\theta) < \beta_l, \\ -\langle \nabla J(\theta) \rangle & \text{if } J(\theta) > \beta_u, \\ \langle D_f(\theta) \rangle & \text{otherwise,} \end{cases} \quad (15)$$

with $\beta_l \le \beta \le \beta_u$ and the *feasible directions* vector D_f to be specified. The basic motivation of this algorithm is to generate a subsequence $\{\theta_{n_i}\}$ of *feasible and improving solutions*. To achieve this, we force the $\{J(\theta_n)\}$ process to spend most of the time within a suitably constructed interval $I = [\beta_l, \beta_u]$. The construction of I ensures that

every θ such that $J(\theta) \in I$ is a feasible solution in a sense that shall be subsequently made clear. Within this interval, the direction of movement given by D_f is such that it points strictly toward the interior of the feasible region, and represents a reduction in cost. Whenever $J(\theta) > \beta_u$, the direction $-\langle \nabla J(\theta) \rangle$ forces the process back into the feasible region, whereas $J(\theta) < \beta_l$ indicates that the process is well within the feasible region—and the algorithm in this case proceeds in an unconstrained fashion.

Given the observed values of $\nabla C(\theta_n)$ and $\nabla J(\theta_n)$ associated with θ_n, a number of options are available to determine a $D_f(\theta_n)$ with the desired properties. One standard approach (cf., e.g., Luenberger 1973) is to use the following linear program to get the direction vector $D_f = (d_s, d_Q)$:

$$\text{Max } \sigma_0 \tag{16}$$

$$\text{subject to } \frac{\partial C}{\partial s} d_s + \frac{\partial C}{\partial Q} d_Q \leq -k_1 \sigma_0,$$

$$\frac{\partial J}{\partial s} d_s + \frac{\partial J}{\partial Q} d_Q \leq -k_2 \sigma_0,$$

$$-1 \leq d_s \leq 1, -1 \leq d_Q \leq 1,$$

where the parameters k_1 and k_2 are strictly positive real numbers. Noting that $d_s = 0$, $d_Q = 0$ with $\sigma_0 = 0$ is a feasible solution to the above LP, optimization will yield a solution, if one exists, for which $\sigma_0 > 0$. The constraints imply that such a solution will represent a direction of strict decrease in both $C(\theta)$ and $J(\theta)$, as desired. This formulation takes into account the trade-off between the cost decrease offered by a particular feasible direction, and the maximum amount of displacement (to maintain feasibility) that it permits. The degree of this trade-off can be manipulated via the constants k_1, and k_2.

Given a target service level β and a ''tolerance'' limit ϵ, our aim is to select the parameters β_l and β_u in the following manner:

$$\beta_u = \inf_{\theta} \sup\{x: P(J(\theta) \leq x | \mathcal{J}(\theta) \geq \beta) \leq \epsilon\}, \tag{17}$$

$$\beta_l = \inf_{\theta} \sup\{x: P(J(\theta) \leq x | \mathcal{J}(\theta) \geq \beta_u) \leq \epsilon\}. \tag{18}$$

The value of β_u is important since the algorithm forces the sequence $J(\theta_n)$ to ''oscillate'' about β_u. The expression given in (17) aims at setting the ''tightest'' possible value to β_u such that for any θ, if a particular realization ω results in $J(\theta, \omega) < \beta_u$, then $P(J(\theta) \geq \beta) \leq \epsilon$. As β_u becomes loose (i.e., decreases), the algorithm, in effect, overcompensates; and the resulting solutions, though feasible, become further removed from optimality. However, evaluating (17) a priori is usually not possible, or else simulation would probably not be needed to solve the problem. In the following sections, while describing our implementation of the algorithm, we address this issue in further detail.

We now describe our proposed simulation optimization algorithm incorporating the feasible directions method, and report the results of extensive empirical testing in the next section. The implementation we describe below can be further refined in a number of ways, which we shall point out as we go along. The algorithm consists of three stages:

Stage 1: analytical approximation;

Stage 2: line search on s, with Q kept constant;

Stage 3: update via feasible directions based on gradient estimates.

Stages 1 and 2 are carried out once each, and are considered preprocessing steps to Stage 3, which is iterative. The gradient estimation uses the PA gradient estimators given by (5), (11), (12), and (13). We now describe the other components of the algorithm.

4.1. Preprocessing: Stages 1 and 2

We have noted earlier that, for rapid convergence towards the optimizer, it is quite crucial to provide the algorithm with a good starting point. Our algorithm does this in two stages. In Stage 1, we compute Q using the EOQ formula, and then implement the two moment approximation proposed by Tijms and Groenevelt (1984) to determine s. In particular, for each test case, we select among the *simplified normal*, *normal*, and *gamma* approximations, using the guidelines suggested by the authors. While this provides a reasonably good starting point in cases where the order crossing is quite low, for more extreme cases, the approximation performs poorly. Thus, in Stage 2, we use the Stage 1 solution and implement a simple preprocessing idea that provides us with a starting point that has proved to be very good in the vast majority of problem instances we considered. The idea is based on the assumption that

the optimal value of Q is fairly insensitive to the lead time process. Accordingly, we begin at point $(\hat{s}_0, Q_0)$ given by Stage 1, and carry out a line search (via simulation) on s, terminating the process at a point (s_0, Q_0) for which $J(s_0, Q_0) \in [\beta, \beta_u]$. This particular stopping criterion was used so that the solutions given by the line search and the constrained optimization to follow would have roughly equal service levels (up to the given precision), making the costs comparable. The feasible directions algorithm is then invoked with (s_0, Q_0) as the initial point.

4.2. The Direction Vector

The direction vector D_f given in (15) can be found by solving the LP given by (16). However, we found that the following modified version, which is more restrictive but easier to implement (it has a very simple graphical representation), worked well in our numerical experimentation:

$$\text{Min } \frac{\partial C}{\partial s} d_s + \frac{\partial C}{\partial Q} d_Q \tag{19}$$

$$\text{subject to } \left[\frac{\partial J}{\partial s} d_s + \frac{\partial J}{\partial Q} d_Q\right] - \left[\frac{\partial C}{\partial s} d_s + \frac{\partial C}{\partial Q} d_Q\right] \le 0,$$

$$-1 \le d_s \le 1, -1 \le d_Q \le 1.$$

Basically, the constants k_1 and k_2 have been set to unity in (16), and the first inequality constraint converted to an equality constraint. Based on our experiments with other possible formulations of the linear program, it appears that these constants could, in general, be manipulated advantageously. For instance, it seems intuitively attractive to favor cost decrease (using a relatively higher k_1) in the initial iterations, and emphasize more on feasibility (using a relatively higher k_2) in the final stages.

4.3. Parameter Settings

We discuss the setting of the parameters β_l, β_u, $\{a_n(s)\}$, and $\{a_n(Q)\}$. The setting of β_l and β_u was based on a ''3σ'' approach as an approximation to the specifications imposed in (17) and (18). Based on some preliminary simulations, and in view of the lead time distributions and precision level to be used, we defined the interval I to be $[\beta - 0.0025, \beta + 0.0025]$ and then used the observed J of the resulting solutions to generate a comparable ''true'' optimal solution (via a brute-force search described in §5). Our objective here was to ensure that the heuristic solutions had service levels that were no more than $\beta + 0.01$. Alternatively, the solution (s_0, Q_0) given by the line search could be evaluated to get an estimate of the variance of $J(s_0, Q_0)$, which could then be used to set an appropriate I.

For the setting of the sequences $\{a_n(s)\}^\alpha$ and $\{a_n(Q)\}^\alpha$ associated with each problem instance α, we chose a fixed step-size sequence denoted by

$$a_n^\alpha(s) = a_s(\alpha),\ a_n^\alpha(Q) = a_Q(\alpha), \text{ for all } n.$$

The constants $a_s(\alpha)$ and $a_Q(\alpha)$ are calibrated to a benchmark problem α_0 by

$$a_s(\alpha) = K_\alpha K_s(\alpha) a_s(\alpha_0), \tag{20}$$

$$a_Q(\alpha) = K_\alpha K_Q(\alpha) a_Q(\alpha_0), \tag{21}$$

where the scaling coefficients K_α, $K_s(\alpha)$, and $K_Q(\alpha)$ are given by

$$K_\alpha = \frac{\text{Var}(L(\alpha))}{\text{Var}(L(\alpha_0))},\ K_s(\alpha) = \frac{s_0(\alpha)}{s_0(\alpha_0)},$$

$$K_Q(\alpha) = \frac{Q_0(\alpha)}{Q_0(\alpha_0)}, \tag{22}$$

with $L(\cdot)$ being the lead time random variable. The coefficients $K_s(\alpha)$, and $K_Q(\alpha)$ reflect ''magnitude,'' whereas K_α reflects the ''distance'' from optimality. The latter is based on the notion that as the variance of lead time increases, so does the probability of order crossing, which, in turn, might suggest a relative increase in the gap $|\theta^* - \theta_0|$.

The constants $a_s(\alpha_0)$ and $a_Q(\alpha_0)$ in the calibration problem α_0 are selected such that the recursions

$$s_{n+1} = s_n + a_s d_s(\theta_n),$$

$$Q_{n+1} = Q_n + a_Q d_Q(\theta_n),$$

result in a solution that is close to optimal in terms of both the cost and the service level, where optimality is determined by brute-force exhaustive search on α_0. Note that this search would not be required for all subsequent problem instances α. We recommend choosing a problem with the most stringent service level constraint and a relatively large probability of order crossing. In practice, a more sophisticated calibration

process could conceivably be used to fine tune the algorithm. However, even this rather simple strategy was quite successful in numerical experiments, which seems to be indicative of the general robustness of the feasible directions approach.

4.4. Summary of Algorithm

Before proceeding to the numerical experiments, we first summarize the implementation of the algorithm more concisely here for the reader's convenience.

Calibration (One Time over Entire Problem Space)
Run one calibration problem α_0 to find the constants $a_s(\alpha_0)$ and $a_Q(\alpha_0)$. Alternatively, just use the ones derived in the next section:

$$a_s(\alpha_0) = 2.25, \quad a_Q(\alpha_0) = 0.15,$$

where $\text{Var}(L(\alpha_0)) = 6$, $s_0(\alpha_0) = 1435$, and $Q_0(\alpha_0) = 85$.

Algorithm

- Stage 1: Analytical Approximation

  Set $Q_0 = \sqrt{2KE[D]/h}$.
  Set $\hat{s}_0$ according to the procedures specified in Tijms and Groenevelt (1984). Specifically, define $\mu_1 = (1 + E[L])E[D]$, $\sigma_1^2 = (1 + E[L])V[D] + V[L](E[D])^2$, $\mu_2 = E[L]E[D]$, $\sigma_2^2 = E[L]V[D] + V[L](E[D])^2$, $\nu_1 = \sigma_1/\mu_1$ and $\nu_2 = \sigma_2/\mu_2$. Then,
  (i) for the case of $\nu_1 \leq 0.5$, use the *simplified normal approximation* provided $\beta \leq 0.10$ and $\nu_2 - \nu_1 \geq 0.01$; otherwise use the *normal approximation* procedure; and
  (ii) for the case of $\nu_1 > 0.5$, use the *gamma approximation* procedure.
- Stage 2: Line Search on s

  Determine a s_0 such that $J(s_0, Q_0) \in [\beta, \beta_u]$ as follows:
  (i) Initialize: $s = \hat{s}_0$, $s_L = 0$, $s_U = 0$, $\Delta = 0.10\hat{s}_0$, $n = 0$.
  (ii) Estimate $J(s, Q_0)$ (no gradient estimation required) and set $n \leftarrow n + 1$.
  If $J(s, Q_0) \in [\beta, \beta_u]$ or $n = 25$, set $s_0 = s$ and STOP;
  else if $J(s, Q_0) < \beta$, go to (iii); else go to (iv).
  (iii) Set $s_U = s$.
  If $s_L > 0$, set $s = (s_L + s_U)/2$; else set $s \leftarrow s - \Delta$.
  Return to (ii).
  (iv) Set $s_L = s$.
  If $s_U > 0$, set $s = (s_L + s_U)/2$; else set $s \leftarrow s + \Delta$.
  Return to (ii).
- Stage 3: Feasible Directions Search

  Use (s_0, Q_0) as the initial starting point ($n = 0$), and calculate the step sizes a_s and a_Q via Equations (20), (21), and (22).
  (i) Run simulation at $\theta_n = (s_n, Q_n)$ to get $J(\theta_n)$ and gradient estimates via (5), (11), (12), and (13).
  (ii) Form $\langle \nabla J(\theta_n) \rangle$ and $\langle \nabla C(\theta_n) \rangle$.
  (iii) Form $\langle D_f \rangle$ by solving the LP (19).
  (iv) Update $\theta_n \leftarrow \theta_{n+1}$ via Equations (14) and (15).
  (v) Repeat until stopping condition met with $n \leftarrow n + 1$.

5. Numerical Experiments

The algorithm was implemented on the following test cases:

- *Demand Distributions*: Exponential, 2-Erlang, Normal, Uniform
- *Mean Demand Sizes*: 20, 50, 100, 150, 200, 250
- *Lead Time Distributions*:
  LT1—$P(L = 1) = 0.25$, $P(L = 2) = 0.50$, $P(L = 3) = 0.25$
  LT2—Discrete Uniform [0, 5]
  LT3—Discrete Uniform [0, 10]
  LT4—Poisson with $\lambda = 10$
- *Service Levels*: 0.10, 0.05, 0.01

The above cases represent a wide variety of demand distributions in terms of shape and location. The lead time distributions have been listed according to increasing probabilities of order crossings. In fact, LT1 appears in Tijms and Groenevelt (1984) and was selected specifically to test the behavior of the algorithm in the close vicinity of the optimal solution. The inventory system was simulated using a discrete-event stochastic simulation model programmed in C. For each test case, we ran five replications (using different starting seeds) to check that similar final solutions would be obtained by the algorithm. Thus, treating each sample path as a separate problem instance, we report results for a total of 1440 instances.

We took the final solution to a problem instance to be the *best observed feasible solution* over one run of a given number of iterations, say T, of the algorithm:

Table 1 Algorithm Output for the Calibration Problem

	Stage 1 Solution		Stage 2 Solution		Final Solution		Optimal Solution	
β	C	$\mathcal{J}$	C	$\mathcal{J}$	C	$\mathcal{J}$	C	$\mathcal{J}$
0.01	1465.05	0.0004	1120.84	0.0117	1093.26	0.0143	1071.59	0.0148
0.05	1100.16	0.0133	847.35	0.0533	838.26	0.0566	823.29	0.059
0.10	933.87	0.0337	717.16	0.1025	708.20	0.1076	702.76	0.1099

$$\theta_{\min} = \arg_{\theta_n} \min\{C(\theta_n): J(\theta_n) \leq \beta_u, n = 1, \ldots, T\}. \quad (23)$$

We set the unit cost $u = 2$, the holding cost $h = 1$, and the setup cost $K = 36$ in all our test cases (including calibration). Results for other values of K and h, while not reported here, exhibited similar behavior to the experiments described here. In general, T is a stopping time that is a function of the stopping condition chosen for the algorithm. Here, we just choose a fixed number of iterations, $T = 50$, in order to study the improvement of the algorithm. Improved performance in actual implementation can be achieved by a judicious choice of this condition. Each iteration consists of a simulation run of 20,000 periods, as this was sufficient to achieve steady-state conditions.

The true cost ($\mathcal{C}$) and service level ($\mathcal{J}$) of the Stage 1, Stage 2, and final solutions were estimated by performance evaluation of the inventory system at each solution point, using 30,000 periods and averaging over 10 replications. The additional periods and replications were chosen for further precision in the estimates, since they were used to determine ''true'' costs and solutions. The ''true'' optimal solutions were estimated by brute-force simulation over the (s, S) plane, initially using a 5 $\times$ 5 grid, and then searching over a 1 $\times$ 1 grid in the neighborhood of the coarse solution. Each θ on the grid was evaluated via ten 20,000 period replications, and the optimal solution θ^* was estimated as

$$\theta^* = \arg_\theta \min\{\mathcal{C}(\theta): \mathcal{J}(\theta) \leq \beta^*, \theta \in \Theta\}, \quad (24)$$

where Θ represents the search space, and β^* is an appropriately chosen ''feasibility limit.'' In particular, we set β^* so that $\mathcal{J}(\theta^*) \approx \mathcal{J}(\theta_{\min})$, which then allows a fair comparison of the associated solution costs.

To measure the quality of our final solutions, we considered the gap in the cost function $\mathcal{C}(\theta^*) - \mathcal{C}(\theta)$ rather than in the parameter value $\theta^* - \theta$. Henceforth, the word ''precision'' will denote the standard error of the concerned estimator. In terms of simulation, this will depend on the number of replications and the length of each run.

We calibrated the algorithm using exponentially distributed demands, Poisson lead times with means of 100 and 6, respectively, and $\beta = 0.01$. This choice reflects our recommendation that the calibration problem be carried out on a case reflecting high lead time variability relative to the likely scenarios to be considered. The feasibility limit (β^*) to generate the ''true'' optimal solution was set to be 0.015. Based on this calibration exercise, we fixed $a_s(\alpha_0) = 2.25$ and $a_Q(\alpha_0) = 0.15$. Also, the values of $s_0(\alpha_0)$ and $Q_0(\alpha_0)$ were determined via this exercise to be 1435 and 85, respectively. We next carried out optimizations of the same problem instance using service levels of 0.10 and 0.05. In these cases, the step sizes were determined using (20), (21), and (22), and the optimal solutions were generated using feasibility limits of 0.11 and 0.06, respectively. The results of this calibration are shown in Table 1.

The optimality gaps defined by $(\mathcal{C} - \mathcal{C}^*)/\mathcal{C}^*$, the CPU times (in seconds on a Sun Sparc10 workstation), and the fraction of total orders that arrive before previously placed orders (denoted as Cross Ratio, CR), are shown in Table 2.

The above results are quite representative of the large number of test cases that we considered. As can be seen, the final solution comes very close to optimality, and furthermore, the gap decreases with increasing β. The results also indicate that the Stage 2 solution given by the line search does well and is computationally very efficient. The order noncrossing assumption is grossly violated here, and consequently the Stage 1 solution (given by the analytic model of Tijms and Groenevelt) performs poorly. We also experimented with larger step

Table 2 Solution Quality for the Calibration Problem

	Stage 1 Solution			Stage 2 Solution			Final Solution		
β	Gap (%)	CPU	CR (%)	Gap (%)	CPU	CR (%)	Gap (%)	CPU	CR (%)
0.01	36.72	≈0	33.80	4.60	5.25	33.81	2.02	229.87	34.60
0.05	33.63	≈0	33.80	2.92	5.25	33.81	1.81	229.93	34.80
0.10	24.75	≈0	33.80	2.05	6.70	33.81	0.08	229.48	34.70

Table 3 Summary of Solution Quality for 1440 Problem Instances

	Final Solution		Stage 2 Solution		Stage 1 Solution	
Optimality Gap (%)	Freq. (%)	Cum. Freq. (%)	Freq. (%)	Cum. Freq. (%)	Freq. (%)	Cum. Freq. (%)
0–1	27.78	27.78	16.80	16.80	2.43	2.43
1–2	40.21	67.99	19.58	36.38	1.74	4.17
2–3	17.57	85.56	15.42	51.80	1.74	5.91
3–5	10.14	95.70	26.67	78.47	4.52	10.43
5–7	2.85	98.55	12.43	90.90	10.76	21.19
7–10	1.45	100.00	7.50	98.40	3.13	24.32
10–15	0	100.00	1.60	100.00	8.68	33.00
>15	0	100.00	0	100.00	60.41	93.41
Indeterminate	0	100.00	0	100.00	6.59	100.00

Table 4 Observed Service Levels of Final Solutions for 1440 Test Cases

	LT1 and LT2		LT3 and LT4	
Target Service	Observed Service	Number of Cases	Observed Service	Number of Cases
$\beta = 0.10$	≤0.105	236	≤0.11	230
	0.105–0.109	4	0.1100–0.1105	8
			0.1105–0.1108	2
	$\beta^* = 0.105$		$\beta^* = 0.11$	
$\beta = 0.05$	≤0.055	239	≤0.06	239
	0.055–0.057	1	0.0600–0.0605	1
	$\beta^* = 0.055$		$\beta^* = 0.06$	
$\beta = 0.01$	≤0.0125	212	≤0.0150	220
	0.0125–0.0130	24	0.0150–0.0155	17
	0.0130–0.0135	4	0.0155–0.0160	3
	$\beta^* = 0.0125$		$\beta^* = 0.015$	

sizes, and we observed a trade-off between the optimality gap and the true service level of the final solution. Specifically, as the step size is increased, the optimality gap in general reduces, but the probability of the true service level exceeding feasibility limits is also increased. For instance, in the calibration case described above, as we increased a_s and a_Q from 2.25 and 0.15 to 4.50 and 0.75, respectively, the solution for $\beta = 0.01$ improved significantly, but the corresponding solution for $\beta = 0.10$ had a true service level higher than 0.11.

The numerical results for the 1440 problem instances are summarized in Table 3. The true service levels for these solutions, and the corresponding feasibility limits used to generate optimal solutions, are summarized in Table 4. For the Stage 1 solutions, the class ''indeterminate'' indicates service levels that, though not grossly infeasible, significantly exceeded the feasibility limits and hence were incomparable with the generated optimal solutions.

The results indicate the following:

- The Stage 1 solutions have optimality gaps of over 15% for 60% of the cases; a further breakdown (not indicated in the results reported here) reveals that solutions for the low order-crossing LT1 cases (comprising 25% of the total instances) are within 10% of optimality, and as the extent of order crossing increases, so does the optimality gap.
- The Stage 2 solutions performed very well for the problem instances considered, with optimality gaps of under 5% in almost 80% of the cases.
- The optimization step significantly improves on the Stage 2 solution, with optimality gaps of under 5% in 95% of the cases.
- There is some degradation of solution quality as β becomes smaller.

The results also clearly indicate that the vast majority of our solutions have service levels of no more than $\beta + 0.01$. The few cases that exceed this limit do so by a negligible amount. Further reductions (i.e., improvements) in service levels of the solutions (i.e., lower values of β) can be achieved by suitably decreasing the value of β_u. The results shown in Table 4 also justify the feasibility limits used for generating the optimal solutions.

6. Conclusions

In this paper, we have used gradient-based simulation optimization methods to address the problem of optimizing periodic review (s, S) systems that allow order crossings and are subject to a service level constraint. Simulation experiments show that currently available analytic models perform poorly under these conditions. We propose a simulation-based algorithm using the feasible directions approach from nonlinear programming. The implementation includes various preprocessing steps that generate good starting solutions. The algorithm is generally applicable to any problem having a single ''noisy'' constraint, with only the gradient estimation problem being system dependent.

The numerical results indicated the following:

1. Currently available analytic approximations would not be useful for managing (s, S) inventory systems that are subject to even moderately high probabilities (such as that associated with LT2) of order crossings.

2. The proposed algorithm performed very well, using a single calibration over the entire set of test cases. The calibration provides benchmark settings on which parameter settings for a problem instance are easily scaled via Equations (20), (21), and (22). This robustness is attractive for practical implementation.

3. The Stage 2 preprocessing step provided impressive improvement at a relatively cheap cost in terms of computation. Although the optimization step usually reduces the gap of the Stage 2 solution by at least 50%, it is computationally much more intensive; the time for one optimization run ranges from one minute for LT1 to six minutes for LT4. A practitioner might consider terminating the procedure at Stage 2 for certain classes of products, e.g., those deemed less critical through Pareto analysis.

The algorithm performed very well over a large number of test cases, and, in fact, the preprocessing step, by itself, yielded reasonable solutions within very short computing times. These promising computational results motivate the need for further research on the algorithm's theoretical properties. In addition, the need for a good stopping rule in practical implementation is an important topic for investigation. Lastly, extending the proposed algorithm to problems with multiple noisy constraints is also a fruitful topic for further investigation.[1]

[1] M. C. Fu was supported in part by the National Science Foundation under Grant No. NSF EEC-9402384 and by a 1997 Summer Research Grant from the Maryland Business School. The authors sincerely thank the two anonymous referees for their constructive comments and suggestions that have led to a much improved paper.

References

Bashyam, S., M. C. Fu. 1994. Application of perturbation analysis to a class of periodic review (s, S) inventory systems. *Naval Res. Logistics* **41** 47–80.

Ehrhardt, R. 1984. (s, S) policies for a dynamic inventory model with stochastic lead times. *Oper. Res.* **32** 1 121–132.

Fu, M. C. 1994a. Sample path derivatives for (s, S) inventory systems. *Oper. Res.*, **42** 351–364.

——. 1994b. Optimization via simulation: A review. *Ann. Oper. Res.* **53** 199–248.

——. J. Q. Hu. 1994. (s, S) inventory systems with random lead times: Harris recurrence and its implications in sensitivity analysis. *Probability Engng. Inform. Sci.* **8** 355–376.

——, ——. 1997. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. Kluwer Academic Publishers, Boston, MA.

Glasserman, P. 1991. *Gradient Estimation Via Perturbation Analysis*. Kluwer Academic Publishers, Boston, MA.

Ho, Y. C., X. R. Cao. 1991. *Discrete Event Dynamic Systems and Perturbation Analysis*. Kluwer Academic Publishers, Boston, MA.

Jacobson, S. H., L. W. Schruben. 1989. A review of techniques for simulation optimization. *Oper. Res. Letters* **8** 1–9.

Kushner, H. J., D. C. Clark. 1978. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York.

L'Ecuyer, P., N. Giroux, P. W. Glynn. 1994. Stochastic optimization by simulation: numerical experiments with a simple queue in steady-state. *Management Sci.* **40** 1245–1261.

Lee, H. L., S. Nahmias. 1993. Single-product, single-location models. S. C. Graves, A. H. G. Rinnooy Kan, P. H. Zipkin Eds. *Handbooks in Operations Research and Management Science, Vol. 4: Logistics of Production and Inventory*, North-Holland, NY.

Luenberger, D. G. 1973. *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA.

Porteus, E. L. 1988. Stochastic inventory theory. D. Heyman, M. Sobel Eds. *Handbooks in Operations Research and Management Science*, *Vol. II*: *Stochastic Models*, North-Holland, NY.

Safizadeh, M. H. 1990. Optimization in simulation: current issues and the future outlook. *Naval Res. Logistics* **37** 807–825.

Schneider, H., J. L. Ringuest. 1990. Power approximation for computing (*s*, *S*) policies using service level. *Management Sci.* **36** 7 822–834.

Tijms, H. C., H. Groenevelt. 1984. Simple approximations for the reorder point in periodic and continuous review (*s*, *S*) inventory systems with service level constraints. *European J. Oper. Res.* **17** 175–190.

Zheng, Y., A. Federgruen. 1991. Finding optimal (*s*, *S*) policies is about as simple as evaluating a single policy. *Oper. Res.* **39** 654–665.

Zipkin, P. 1986. Stochastic lead times in continuous-time inventory models. *Naval Res. Logistics Quarterly* **33** 763–774.

Zoutendijk, G. 1960. *Methods of Feasible Directions*. Elsevier Publishing Company.

Accepted by Hau L. Lee; received May 16, 1994. *This paper has been with the authors* 28 *months for* 2 *revisions.*