

COORDINATED REPLENISHMENTS IN A MULTI-ITEM INVENTORY SYSTEM WITH COMPOUND POISSON DEMANDS*

A. FEDERGRUEN,[†] H. GROENEVELT[‡] AND H. C. TIJMS[‡]

In many practical applications of multi-item inventory systems significant economies of scale can be exploited when coordinating replenishment orders for groups of items. This paper considers a continuous review multi-item inventory system with compound Poisson demand processes; excess demands are backlogged and each replenishment requires a lead time. There is a major setup cost associated with any replenishment of the family of items, and a minor (item dependent) setup cost when including a particular item in this replenishment. Moreover there are holding and penalty costs. We present an algorithm which searches for a simple coordinated control rule minimizing the long-run average cost per unit time subject to a service level constraint per item on the fraction of demand satisfied directly from on-hand inventory. This algorithm is based on a heuristic decomposition procedure and a specialized policy-iteration method to solve the single-item subproblems generated by the decomposition procedure. The model applies to multi-location inventory systems with similar cost structures for coordinated deliveries.

(MULTI-ITEM INVENTORY SYSTEMS; COORDINATED REPLENISHMENTS; COMPOUND POISSON DEMAND PROCESSES; SERVICE LEVEL CONSTRAINT; POLICY-ITERATION ALGORITHM)

1. Introduction

In many practical multi-item inventory systems considerable savings may be achieved by the coordination of replenishment orders for groups of items, cf. Brown (1967) and Peterson and Silver (1979). We consider a continuous review multi-item inventory system where demands for the items are generated by independent compound Poisson processes; excess demands are backlogged and each replenishment requires a lead time.

There is a major setup cost associated with a replenishment of the family. In the procurement context this is the fixed cost of placing an order for the family of items, independent of its size or composition. In the production context, this corresponds with the changeover cost incurred when switching the facility from the production of some other family to production within the family of interest. For each individual item included in the replenishment, an item specific setup cost is added. In addition the cost structure consists of holding and penalty costs.

Our model applies equally well when there are several locations instead of several products: a central depot coordinates the replenishment process for a set of locations with exogenous, random and independent demand processes for a single commodity. Transportation costs can easily be incorporated in the above cost structure, as long as they are separable in the locations.

We wish to minimize the long-run average cost per unit time subject to a service level constraint per item on the fraction of demand satisfied directly from on-hand inventory. Service level constraints are widely used in practice, especially when penalty costs cannot be specified.

* Accepted by Warren H. Hausman, former Departmental Editor; received October 21, 1980. This paper has been with the authors 11 months for 2 revisions.

[†]Columbia University.

[‡]Vrije Universiteit, Amsterdam, The Netherlands.

The multi-item inventory problem can be modeled as a semi-Markov decision problem; however, in view of the dimensionality of the state space, standard solution methods are computationally intractable; moreover an optimal rule may fail to have a simple form (cf. Ignall 1969), and may thus be hard to implement.

As a consequence, we confine ourselves to the following class of simple coordinated control rules introduced in Balintfy (1964) and Silver (1974). Three parameters S_i , c_i and s_i are specified for each item i with $s_i \leq c_i < S_i$. An order is triggered by item i when its inventory level falls to or below the *reorder level* s_i ; any item j for which the inventory level is at or below its *can-order level*, c_j , is included in this order; and the inventory of each item k included in the order is replenished up to its order-up-to level S_k .

For the special case of unit-Poisson demand distributions and zero lead time, Silver (1974) provides an iterative method to compute a suboptimal rule in the above described class of (S, c, s) rules. This heuristic method decomposes the coordinated control problem into single-item problems for each item in the family. Each single-item problem has "normal" replenishment opportunities with *major* setup costs, occurring at the demand epochs for this item; in addition there are *special* replenishment opportunities, with *reduced* setup costs, at epochs generated by a Poisson process which is an approximation to the superposition of the ordering processes triggered by the other items. The single-item problems are solved by elementary heuristic search procedures in the parameter space of the (S, c, s) rules. The mean time between consecutive "special" replenishment opportunities is adapted iteratively.

Using the approach for unit Poisson demands and zero lead times as the key element, Silver (1974) also gives a heuristic solution method for the case of unit Poisson demands and positive lead times. Similarly, Thompstone and Silver (1975) present a heuristic solution method for the special case of compound Poisson demands and zero lead time by using a transformation of the compound Poisson distribution into an "equivalent" unit Poisson distribution.¹

This paper addresses the general case of compound Poisson demands and nonzero lead times and presents an efficient heuristic algorithm to search for an optimal (S, c, s) coordinated control rule. The approach in this paper uses the same decomposition principle as in Silver (1974). Apart from its more general applicability, it distinguishes itself from Silver (1974) and Thompstone and Silver (1975) by solving the single-item subproblems by a specialized policy-iteration algorithm.

For a review of the literature until 1978 on related models we refer to Chapter 13 in Peterson and Silver (1979). Here we merely mention Naddor (1975) treating a periodic review analogue of the above model. A few recent papers, Eppen and Schrage (1981) and Federgruen et al. (1984a, 1984b, 1984c), treat a periodic review variant of a related multi-item model. There replenishments are represented in the more general form of a two-stage process where the allocation of the replenishment batch among the various items in the family can be postponed until the end of the first stage. We note that our approach, as opposed to Eppen and Schrage (1981) and Federgruen et al. (1984a, 1984b, 1984c), allows for fixed costs per item included in the replenishment. §2 describes the specialized policy-iteration algorithm for the single-item subproblems. §3 exhibits how this algorithm is used in an iterative procedure to compute an optimal (S, c, s) coordinated control rule. Finally, in §4 numerical results are presented.

¹ After completion of this paper we became aware of the paper by E. SILVER, "Establishing Reorder Points in the (S, c, s) Coordinated Control System under Compound Poisson Demand," *Internat. J. Production Res.*, 9 (1981), 743–750. This paper generalises the results in Silver (1974) and Thompstone and Silver (1975) and deals with the case of compound Poisson demand and positive lead times.

2. Algorithm for an Optimal (S, c, s) Rule in the Single-Item Model

Consider a single-item system in which demands occur at epochs generated by a Poisson process with rate λ ; demand sizes are independent nonnegative random variables with common discrete probability distribution $\{\phi(j), j \geq 0\}$. Excess demand is backlogged. The inventory level is continuously reviewed. We assume two types of ordering opportunities. "Normal" replenishment opportunities at setup cost K occur at the demand epochs, whereas "special" replenishment opportunities at reduced setup cost κ ($\leq K$) occur at epochs generated by a Poisson process with rate μ , assumed to be independent of the demand process. The replenishment lead time is a constant $L \geq 0$ (see Remark 2.1 below for the case of stochastic lead times). In addition to the ordering costs there are holding costs at a rate $h \cdot k$ when the inventory on hand equals k , and at a rate $p \cdot k$ for a backlog of k units where $h > 0$ and $p \geq 0$. Also a fixed penalty cost $\pi \geq 0$ is incurred for every requested unit that cannot be delivered immediately from current inventory. The holding and penalty costs are assumed to be linear only for ease of presentation. We wish to minimize the long-run average cost per unit time subject to a service level constraint on the fraction of demand satisfied directly from on-hand inventory.

The state of the system can be described by the inventory position ((the inventory on hand) + (outstanding orders) - (total backlog)). A reasonable control rule is the so-called (S, c, s) rule with $S > c \geq s$, under which the inventory position is ordered up to S when either: (i) at a demand epoch the inventory position drops to or below s ; or (ii) when at a special replenishment opportunity the inventory position is at or below c .

In this section we develop a specialized policy-iteration algorithm to compute the best rule within the class of (S, c, s) rules. Note that by taking $\mu = 0$ our algorithm also applies to the continuous review (s, S) inventory system studied in Archibald and Silver (1978) and Tijms and Groenevelt (1984), cf. Federgruen and Zipkin (1984d). We first ignore the service level constraint. Then, by a Lagrangian approach, our algorithm can easily be extended to handle the service level constraint.

To develop the algorithm we first note that the inventory control problem can be represented by a denumerable state semi-Markov decision model. Demand epochs and "special" replenishment opportunities represent the *decision epochs* and the state space is given by $X = \{(i, z) \mid i \text{ integer}; z = 0, 1\}$. Here state $(i, 0)$ [$(i, 1)$] corresponds to the situation where a demand [special replenishment opportunity] has just occurred leaving an inventory position of i units. At each decision epoch we specify the decision k as the inventory position just after a possible replenishment. The one-step expected transition times and the one-step transition probabilities $p_{xy}(k)$ are easily given. The time between consecutive decision epochs are independent and exponentially distributed with mean $(\lambda + \mu)^{-1}$. The next decision epoch is generated by a demand [special replenishment opportunity] with probability $\lambda(\lambda + \mu)^{-1}$ [$\mu(\lambda + \mu)^{-1}$]. Hence, noting that $p_{(i,0)y}(k) = p_{(i,1)y}(k)$ and using the shorthand notation $p_{iy}(k) = p_{(i,0)y}(k)$, we have

$$p_{i(k,1)}(k) = \mu(\mu + \lambda)^{-1}, \quad p_{i(j,0)}(k) = \lambda(\lambda + \mu)^{-1}\phi(k - j),$$

with the convention $\phi(j) = 0$ for $j < 0$. To give the one-step expected costs $\gamma(x, k)$, we first introduce $\{r(j), j \geq 0\}$ as the probability distribution of the total demand during the replenishment lead time L . The compound Poisson distribution $\{r(j)\}$ can be computed from the stable recursion scheme, cf. Adelson (1966):

$$r(0) = e^{-\lambda L(1-\phi(0))}, \quad r(j) = \frac{\lambda L}{j} \sum_{k=1}^j k\phi(k)r(j-k), \quad j \geq 1. \quad (2.1)$$

Next we observe that since excess demand is backlogged and the lead time of any order is a constant L , the inventory on hand at any time $t + L$ is distributed as the inventory position at time t minus the total demand during a period of length L .

Hence, letting $\{\tau_n\}$ be the sequence of decision epochs, the inventory position just after τ_n unambiguously determines the distribution of the inventory on hand at time $\tau_n + L$. We thus adopt the standard convention that when choosing decision k at epoch τ_n , one is charged with the immediate ordering cost (if any) as well as an amount $c(k)$ representing the expected holding and penalty costs incurred in $[\tau_n + L, \tau_{n+1} + L)$. Clearly, this shift in costs leaves the average cost of any policy unchanged. We now have for the one-step expected costs,

$$\gamma((i, 0), k) = K\delta(k - i) + c(k); \quad \gamma((i, 1), k) = \kappa\delta(k - i) + c(k),$$

where $\delta(j) = 0$ for $j \leq 0$ and $\delta(j) = 1$ for $j \geq 1$. By Federgruen and Schechner (1984) (cf. also Archibald 1976), $c(k)$ equals the expected holding and penalty costs incurred between time L and the first decision epoch following L under the condition that at time 0 the inventory position equals k and no replenishment orders are placed between time 0 and time L . Under this condition the inventory on hand minus backlogs at time L equals $k - j$ with probability $r(j)$ and remains constant after time L during an interval of expected length $(\lambda + \mu)^{-1}$. Further, fixed penalty costs can only be incurred at the end of this interval and only if the first decision epoch after time L is a demand epoch. Thus we obtain

$$\begin{aligned} c(k) = & (\lambda + \mu)^{-1} h \sum_{j=0}^k (k - j)r(j) + (\lambda + \mu)^{-1} p \sum_{j=k}^{\infty} (j - k)r(j) \\ & + \lambda(\lambda + \mu)^{-1} \pi \left\{ \sum_{j=0}^k r(j) \sum_{t=k-j}^{\infty} (t - k + j)\phi(t) + ED \sum_{j=k+1}^{\infty} r(j) \right\}, \end{aligned}$$

$k \geq 1,$

where $ED = \sum_j j\phi(j)$. Also $c(k) = -pk(\lambda + \mu)^{-1} + \pi\lambda(\lambda + \mu)^{-1}ED$ for $k \leq 0$. It now follows readily that $c(k)$, $k \geq 1$ can recursively be computed from

$$c(k) = c(k - 1) + (\lambda + \mu)^{-1} \left\{ (h + p) \sum_{j=0}^{k-1} r(j) - p - \pi\lambda \sum_{j=0}^{k-1} r(j) \sum_{t=k-j}^{\infty} \phi(t) \right\},$$

$k \geq 1.$

This completes the specification of the basic elements of the semi-Markov decision model for the inventory problem.

REMARK 2.1. In case the replenishment lead times are stochastic and in case the probability that orders cross in time is negligible for relevant control rules, we can apply the same model provided $r(j)$ is replaced by $r(j) = \sum_l r(j; l)P[L = l]$. Here $r(j; l)$, $j \geq 0$ can recursively be computed from (2.1) with $L = l$. The resulting formula for $c(k)$ is exact up to neglecting the probability of orders crossing in time.

Fix now a rule R of the (S, c, s) type. First we discuss the computation of the average cost and the relative values for rule R . The relative values will be needed to identify potential improvements of R in our policy-iteration algorithm. Both the average cost and the relative values can be related to the expected costs incurred during a single *regeneration cycle*, which is defined as the time interval between two consecutive replenishment orders. For the system which starts at epoch 0 in state $(i, 0)$, with $i > s$, and is controlled by rule R , define:

$t_R(i)$ = the expected time until the next epoch at which a replenishment order is placed,

$h_R(i)$ = the expected holding and penalty costs incurred until the next epoch at which a replenishment order is placed (excluding any costs incurred at this replenishment epoch), and

$q_R(i)$ = the probability that the next replenishment order is triggered by a demand.

Also, we introduce for ease of notation,

$$k_R(i) = h_R(i) + Kq_R(i) + \kappa(1 - q_R(i)), \quad i > s. \quad (2.2)$$

Thus $k_R(i)$ denotes the total expected costs of going from state $(i, 0)$ to the regeneration state (= state just after a replenishment order) when using rule R , where among the costs incurred at the replenishment epoch only the ordering costs are included. Note that $t_R(i)$ and $k_R(i)$ for $i > c$ also give the expected time and the expected costs of going from state $(i, 1)$ to the regeneration state under rule R . Now, by the theory of regenerative processes (cf. Ross 1970), the long-run average cost per unit time under rule R equals

$$g_R = k_R(S)/t_R(S). \quad (2.3)$$

Also, by the theory of regenerative processes and the observation that $q_R(S)$ can be interpreted as the expected number of orders triggered by a demand in a single regeneration cycle, we have

$$q_R(S)/t_R(S) = \text{the long-run average number of replenishments} \\ \times \text{per unit time triggered by a demand.} \quad (2.4)$$

This result will be needed in the algorithm for the multi-item problem. Finally, the relative values for rule R , $v_R(x)$ (which denotes the relative costs—relative to a constant cost rate g_R —of going from state x to the regeneration state) are defined by:

$$v_R(i, 0) = \begin{cases} k_R(i) - g_R t_R(i), & i > s, \\ K, & i \leq s, \end{cases} \quad (2.5)$$

$$v_R(i, 1) = \begin{cases} k_R(i) - g_R t_R(i), & i > c, \\ \kappa, & i \leq c. \end{cases} \quad (2.6)$$

Thus the difference $v_R(x) - v_R(y)$ is equal to the long-run decrease in the total expected costs caused by starting in state x rather than in state y if the system is operated under rule R , cf. also Howard (1960). It remains to be shown how the functions t_R , h_R and q_R can be computed. To determine $t_R(i)$, condition on the state of the system at the first decision epoch after time 0. This decision epoch is a special replenishment opportunity with probability $\mu(\mu + \lambda)^{-1}$ and a demand epoch with probability $\lambda(\lambda + \mu)^{-1}$. In the former case an order is placed only when $i \leq c$, while in the latter case inventory is replenished only when the demand is at least $i - s$. Hence we obtain

$$t_R(i) = (\lambda + \mu)^{-1} + \mu(\lambda + \mu)^{-1} t_R(i) \delta(i - c) + \lambda(\lambda + \mu)^{-1} \sum_{j=0}^{i-s-1} t_R(i-j) \phi(j), \\ i > s. \quad (2.7)$$

Similarly, using the convention that the expected holding and penalty costs incurred in $[\tau_n + L, \tau_{n+1} + L)$ are assigned as immediate costs to the n th decision epoch τ_n , we find

$$h_R(i) = c(i) + \mu(\lambda + \mu)^{-1} h_R(i) \delta(i - c) + \lambda(\lambda + \mu)^{-1} \sum_{j=0}^{i-s-1} h_R(i-j) \phi(j), \\ i > s. \quad (2.8)$$

Also,

$$q_R(i) = \mu(\lambda + \mu)^{-1} q_R(i) \delta(i - c) + \lambda(\lambda + \mu)^{-1} \left\{ \sum_{j=i-s}^{\infty} \phi(j) + \sum_{j=0}^{i-s-1} q_R(i-j) \phi(j) \right\},$$

$$i > s. \quad (2.9)$$

The relations (2.7)–(2.9) enable the recursive computation of the quantities $t_R(i)$, $h_R(i)$ and $q_R(i)$ for $i = s + 1, s + 2, \dots$.

We need the following lemma, the proof of which is given in the appendix.

LEMMA 1. *The numbers $\{g_R, v_R(x), x \in X\}$ satisfy the linear system of value determination equations*

$$v(x) = \gamma(x, R_x) - g(\lambda + \mu)^{-1} + \sum_{y \in X} p_{xy}(R_x) v(y), \quad x \in X, \quad (2.10)$$

where R_x denotes the decision prescribed by rule R in state x .

We now turn to the problem of designing a policy-improvement step that results in a new rule \bar{R} having the desired (S, c, s) structure. Define the policy-improvement test quantity $I_R(x, k)$ by

$$I_R(x, k) = \gamma(x, k) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{xy}(k) v_R(y). \quad (2.11)$$

The quantity $I_R(x, k)$ may be interpreted as the relative costs incurred until the next replenishment order when choosing action k in the initial state x and using rule R thereafter. Note that, by (2.10), $I_R(x, R_x) = v_R(x)$ for any x . Let $\bar{R} = (\bar{S}, \bar{c}, \bar{s})$ be any rule such that

$$I_R(x, \bar{R}_x) \leq v_R(x) \quad \text{for all } x \in X. \quad (2.12)$$

Then (cf. Howard 1960, Tijms 1980)

$$g_{\bar{R}} \leq g_R, \quad (2.13)$$

where strict inequality holds in (2.13) if strict inequality holds in (2.12) for some state x which is positive recurrent under rule \bar{R} . See also the proof of Lemma 3 in the appendix. To construct a rule \bar{R} satisfying (2.12), we first specify $I_R(x, k)$ for the combinations (x, k) required in our algorithm. Using (2.10) and $\gamma((i, 0), k) = K\delta(k - i) + c(k)$ for $k \geq i$,

$$\begin{aligned} I_R((i, 0), k) &= K + c(k) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{iy}(k) v_R(y) \\ &= K + c(k) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{ky}(k) v_R(y) \\ &= K + v_R(k, 0) \quad \text{all } i \text{ and } k > \max(s, i). \end{aligned} \quad (2.14)$$

Similarly, using $v_R(j, 0) = K$ and $v_R(j, 1) = \kappa$ for $j \leq s$, we find

$$\begin{aligned} I_R((i, 0), i) &= I_R((i, 1), i) \\ &= c(i) - g_R(\lambda + \mu)^{-1} + \mu(\lambda + \mu)^{-1} \kappa + \lambda(\lambda + \mu)^{-1} K, \quad i \leq s. \end{aligned} \quad (2.15)$$

Using (2.10) we find by analogous arguments

$$\begin{aligned} I_R((i, 1), k) &= \kappa + c(k) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{iy}(k) v_R(y) \\ &= \kappa + v_R(k, 0) \quad \text{all } i \text{ and } k > \max(s, i). \end{aligned} \quad (2.16)$$

$$I_R((i, 1), i) = c(i) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{iy}(i) v_R(y) = v_R(i, 0), \quad s < i \leq c. \quad (2.17)$$

We can now formulate a policy-improvement procedure for our policy-iteration algorithm. Using the computed average cost g_R and the relative values for the current rule R , this procedure yields a new rule $\bar{R} = (\bar{S}, \bar{c}, \bar{s})$ which satisfies (2.12). In the algorithm below we assume that bounds U and L can be given for the relevant values of S and s such that $L \leq s \leq c < S \leq U$; i.e., we only consider the control rules which never order up to a level exceeding U and which always place an order when the inventory level falls at or below L (our empirical results suggest that L and U could be chosen as the lower and upper bounds in Archibald and Silver (1978) for the (s, S) policy that would be optimal if $\mu = 0$).

Policy-Improvement Step

(a) Choose an integer \bar{S} with $c < \bar{S} \leq U$ such that (cf. (2.14) and (2.16)),

$$v_R(\bar{S}, 0) = \min_{c < i \leq U} v_R(i, 0),$$

where we take $\bar{S} = S$ when S attains the above minimum (otherwise it suffices to choose \bar{S} as any integer with $c < \bar{S} \leq U$ such that $v_R(\bar{S}, 0) < v_R(S, 0) = 0$).

(b) Determine the largest integer t with $s < t \leq c$ such that (cf. (2.14)),

$$K + v_R(\bar{S}, 0) < v_R(i, 0) \quad \text{for all } s < i \leq t.$$

If such an integer t exists, define $\bar{s} = t$ and go to part (c) of the policy-improvement step. Otherwise, determine t as the smallest integer with $L < t \leq s$ such that (cf. (2.5) and (2.15)),

$$c(i) + (\lambda + \mu)^{-1} [-g_R + \mu\kappa + \lambda K] < K \quad \text{for all } t \leq i \leq s.$$

If such an integer t exists, define $\bar{s} = t - 1$, otherwise let $\bar{s} = s$.

(c) Determine t as the largest integer with $c < t < \bar{S}$ such that

$$\kappa + v_R(\bar{S}, 0) < v_R(i, 0) \quad \text{for all } c < i \leq t$$

(cf. (2.16) and note by (2.5)–(2.6) that $v_R(i, 1) = v_R(i, 0)$ for $i > c$). If such an integer exists define $\bar{c} = t$. Otherwise determine t as the smallest integer with $\bar{s} < t \leq c$ such that (cf. (2.17) and (2.5))

$$v_R(i, 0) < \kappa \quad \text{for all } \max(t, s + 1) \leq i \leq c \quad \text{and}$$

$$c(i) + (\lambda + \mu)^{-1} [-g_R + \mu\kappa + \lambda K] < \kappa \quad \text{for all } t \leq i < \max(t, s + 1),$$

and define $\bar{c} = t - 1$ if such an integer exists and let $\bar{c} = c$ otherwise.

LEMMA 2. Let R be a given (S, c, s) rule and let \bar{R} be an $(\bar{S}, \bar{c}, \bar{s})$ rule obtained from applying the policy-improvement step to rule R . Then,

$$I_R(x, \bar{R}_x) \leq v_R(x) \quad \text{for all } x \in X \quad \text{and} \quad g_{\bar{R}} \leq g_R.$$

PROOF. We first verify (2.12) for $x = (i, 0)$, i integer.

Case I. $\bar{s} < s$. For $i \leq \bar{s}$, using (2.5) and (2.14),

$$I_R(x, \bar{R}_x) = K + v_R(\bar{S}, 0) \leq K + v_R(S, 0) = K = v_R(x). \quad (2.18)$$

For $\bar{s} < i \leq s$, $I_R(x, \bar{R}_x) = c(i) + (\lambda + \mu)^{-1}[-g_R + \lambda K + \mu\kappa] < K = v_R(x)$ (the first equality follows from (2.15) and the last one from (2.5)). Finally for $i > s$, $I_R(x, \bar{R}_x) = I_R(x, R_x) = v_R(x)$, cf. (2.10)–(2.11).

Case II. $\bar{s} \geq s$. For $i \leq s$, (2.12) follows from (2.18). For $s < i \leq \bar{s}$, $I_R(x, \bar{R}_x) = K + v_R(\bar{S}, 0) \leq v_R(x)$, cf. (2.14). For $i > \bar{s}$, $I_R(x, \bar{R}_x) = I_R(x, R_x) = v_R(x)$, cf. (2.10)–(2.11).

Verification of (2.12) for $x = (i, 1)$, i integer, requires a similar distinction between the cases $\bar{c} > c$ and $\bar{c} \leq c$. Finally, by (2.13), $g_{\bar{R}} \leq g_R$. Q.E.D.

We now state the algorithm for the problem without the service level constraint.

Algorithm (No Service Level Constraint)

Step 1. Let $R = (S, c, s)$ be the current rule. Using (2.2) and (2.7)–(2.9), compute recursively the numbers $k_R(i), t_R(i)$ for $i = s + 1, \dots, S$. Next, by (2.3) and (2.5), compute the average cost g_R and the relative values $v_R(i, 0)$ for $s < i \leq S$.

Step 2. Perform the above described policy-improvement step where any required $v_R(i, 0)$ for $S < i \leq U$ is obtained by an application of the recursive schemes (2.7)–(2.9). This results in a new rule $\bar{R} = (\bar{S}, \bar{c}, \bar{s})$.

Step 3. If $\bar{R} = R$, stop; otherwise go to Step 1 with R replaced by \bar{R} .

Although the algorithm is bound to terminate in a finite number of iterations (cf. Lemma 3 in the appendix) we were unable to prove that it always converges to a policy which is optimal among all stationary policies. However, in all examples tested convergence to an overall optimal policy was numerically verified through the average cost optimality equation, cf. also Appendix 2 in Federgruen et al. (1983). The number of iterations required by this algorithm is remarkably small (typically less than ten) and each iteration involves only simple computations.

We next discuss the computation of an (S, c, s) rule which minimizes the average holding and ordering costs (i.e., assuming $p = 0$) subject to the requirement that the fraction of demand satisfied directly from on-hand inventory is at least α , with $0 < \alpha < 1$ given. Service level constraints are widely used in practice, especially when penalty costs cannot be specified. Note that for the earlier cost structure with a fixed penalty cost π , the average cost g_R of rule R contains a term π times the average demand that goes short per unit time. Dividing this term by π times the average demand λED per unit time, we find for rule R the fraction of demand that is backlogged. Hence the service level of any (S, c, s) rule can easily be computed. Note that in the recursive computation of the cost functions $c(k)$ and $h_R(k)$ the different cost terms can be dealt with separately. To find an (S, c, s) rule which minimizes the average holding and ordering costs subject to the service level constraint, we apply the above algorithm repeatedly for different values for the fixed penalty cost $\pi \geq 0$, as in ordinary Lagrangian methods. We continue until we have found the *smallest* value of π for which the associated (S, c, s) rule resulting from the algorithm still satisfies the service level constraint. In our experience, recovering an average cost optimal (S, c, s) rule when making (small) changes in the value of π requires very few iterations, provided that the algorithm is restarted with the rule that was found to be optimal under the previous value of π . We now state the algorithm.

Algorithm (Service Level Constraint)

Initialization. Choose a positive number π .

Step 1. For the current value of the fixed penalty cost π and the given holding and ordering costs, compute by the preceding algorithm the (S_π, c_π, s_π) rule which mini-

mizes the average cost per unit time (initialize this algorithm with the (S, c, s) rule that was found to be optimal for the previous value of π). Let $\alpha(\pi)$ be the fraction of demand satisfied directly from on-hand inventory under the (S_π, c_π, s_π) rule.

Step 2. If $\alpha(\pi) \geq \alpha$ select a smaller $\pi \geq 0$ (e.g., using bisection); if $\alpha(\pi) < \alpha$ select a larger π value. Go to Step 1.

The algorithm is terminated when the new (S_π, c_π, s_π) rule is the same as the previous one and the value for π has sufficiently converged. This algorithm will be used as the key element in the algorithm for the multi-item inventory problem to be discussed in the next section.

3. The Algorithm for the Multi-Item Inventory Problem with Joint Replenishments

This section deals with the coordinated inventory control problem for a group of n items where a saving in the replenishment cost can be achieved by combining several items in a single order. More precisely, the setup cost of a replenishment involving j items is $K + \kappa(j - 1)$ for $j \geq 1$ where $0 \leq \kappa \leq K$. We note that the analysis below also applies when the major setup cost K is item-dependent. The demands are assumed to be generated by independent compound Poisson processes, i.e. for each item i demands occur at epochs generated by a Poisson process with rate λ_i and the demand sizes are independent random variables with common discrete probability distribution $\{\phi_i(j), j \geq 0\}$. Excess demand is backlogged. For item i the replenishment lead time is assumed to be a constant L_i (however stochastic lead times can also be handled, cf. Remark 2.1 in §2). Also, for item i we incur a holding cost at rate $h_i \cdot k$ when the inventory on hand equals k , and at rate $p_i \cdot k$ when there is a backlog of k units.

In this section we discuss a computational procedure which searches for a multi-item (S, c, s) rule which minimizes the long-run average cost per unit time subject to a service level constraint per item on the fraction of demand satisfied directly from on-hand inventory.

The multi-item inventory problem can be modeled as a semi-Markov decision problem with an n -dimensional state space. However, in view of the dimensionality of the state space, standard solution methods are computationally intractable. Moreover, they may result in optimal policies that fail to have a simple form (cf. Ignall 1969) and are therefore hard to implement. For the case in which no service level constraint is imposed, one could apply a heuristic method which is based on the following decomposition approach (cf. Norman 1972 and Wijngaard 1979): Determine first for each item i independently the (S_i, s_i) rule which would be optimal if joint ordering were not possible, as well as the corresponding relative values $v_i(\cdot)$. This problem with independent control can be solved by the algorithm in the previous section provided one sets $\mu = 0$, $c = s$ in part (a), $c = \bar{S} - 1$ in part (b), and deletes part (c) of the policy-improvement step, cf. also Federgruen and Zipkin (1984d).

Next consider the original multi-item inventory problem with joint ordering. Perform for the corresponding semi-Markov decision problem a single standard policy-improvement step (cf. Norman 1972 and Wijngaard 1979) using $v_1(i_1) + \dots + v_n(i_n)$ as an approximation to the relative value of state (i_1, \dots, i_n) , and implement the resulting policy. This heuristic approach will typically result in a policy that fails to exhibit a simple (say (S, c, s)) structure. Moreover, the approximation to the relative value function fails to reflect the economies of scale in the replenishment process.

As a consequence we focus on the decomposition approach in Silver (1974). This decomposition approach is based on the fact that under general conditions superpositions of n point processes converge to a Poisson process as $n \rightarrow \infty$ and as the individual processes get more and more sparse (cf. Cinlar 1972; see also Albin 1982 for a discussion of the quality of this approximation). The approach thus consists of a decomposition of the coordinated control problem into n independent single-item

problems of the type studied in the previous section. Here the Poisson process generating replenishment opportunities at reduced setup cost represents an approximation of the replenishment processes triggered by the other items. The rate μ_i of this Poisson process in the single-item model for item i is determined in an iterative procedure. We now describe the algorithm which searches for a coordinated (S, c, s) rule minimizing the average holding and ordering costs subject to the requirement that for any item i the fraction of demand satisfied directly from on-hand inventory is at least α_i with $0 < \alpha_i < 1$ given.

Coordinated Control Algorithm

Initialization. Choose positive numbers β_2, \dots, β_n . Let $i := 1$ and $\mu_i := \sum_{j \neq i} \beta_j$.

Step 1. Compute in the single-item control problem for item i the best rule $R_i = (S_i, c_i, s_i)$ by the corresponding algorithm of §2 with μ_i as the rate of the Poisson process describing the replenishment opportunities at the reduced setup cost. Adjust β_i by letting $\beta_i = q_{R_i}(S_i)/t_{R_i}(S_i)$, cf. (2.4).

Step 2. $i := i + 1$ (or 1 if $i = n$), $\mu_i := \sum_{j \neq i} \beta_j$ and go to Step 1.

The algorithm is terminated when for each item i the new values of S_i , c_i and s_i are the same as the previous ones and the value of μ_i has sufficiently converged.

4. Numerical Results

In this section we discuss some numerical results for the multi-item inventory system with coordinated replenishments. We require for each item i that the fraction of demand satisfied directly from on-hand inventory is at least α_i . Also, the cost structure consists only of holding and replenishment costs (i.e., $p_i = 0$). We minimize the average cost per unit time subject to the service level constraint. In the examples below we assume that for each item i the demand size has a negative binomial distribution on the positive integers. For item i we denote by ED_i and $cv(D_i) = \sigma(D_i)/ED_i$ the average demand and the coefficient of variation of the demand. We consider $n = 4$ items with the following numerical data:

item i	λ_i	ED_i	$cv^2(D_i)$	L_i	h_i	p_i
1	10	5	0.5	1	1	0
2	5	5	0.5	1	1	0
3	10	5	0.5	1	2	0
4	5	5	1	1	1	0

We assume for each item i the same service level $\alpha_i = \alpha$ where α has one of three values: 0.90, 0.95 and 0.99. For the setup costs K and κ we consider the three cases $(K, \kappa) = (33, 3)$, $(30, 5)$ and $(15, 5)$ with $\kappa/(K - \kappa) = 0.1$, 0.2 and 0.5 respectively. In Table 4.1 we give for the multi-item problem the best (S_i^*, c_i^*, s_i^*) control rule as computed by the coordinated control algorithm described in §3. We compare this coordinated control rule with the best independent control rule where each item is controlled independently by an (S_i^*, s_i^*) rule assuming the setup cost of a replenishment equals K . For these control rules we report for each item i the associated values for the service level α_i^* and the average cost C_i^* . Since the coordinated control algorithm involves approximations for the superpositions of the ordering processes triggered by the items, the coordinated control model has been validated by computer simulation. For each case one long simulation of 3600 time units was run. The actual values obtained by simulation are reported between brackets. A Pascal computer program for the coordinated control algorithm took on the average 5 seconds of CPU-time on a Cyber 170-750.

TABLE 4.1
Numerical Results

<i>i</i>	Coordinated Control			Independent Control			
	(S_i^*, c_i^*, s_i^*)	α_i^* (sim)	C_i^* (sim)	(S_i^*, s_i^*)	α_i^*	C_i^*	
1	(112, 93, 40)	0.900(0.871)	50.0(48.1)	(126, 54)	0.901	65.0	$K = 33$ $\kappa = 3$ $\alpha = 0.90$
2	(68, 54, 13)	0.903(0.889)	35.0(33.4)	(79, 29)	0.902	47.2	
3	(105, 91, 50)	0.902(0.873)	90.3(88.7)	(113, 58)	0.901	105.3	
4	(72, 57, 15)	0.901(0.876)	39.0(37.5)	(84, 32)	0.903	50.9	
1	(120, 102, 55)	0.952(0.945)	60.5(59.0)	(135, 65)	0.951	75.0	$K = 33$ $\kappa = 3$ $\alpha = 0.95$
2	(74, 60, 26)	0.952(0.949)	42.5(41.2)	(86, 37)	0.951	54.6	
3	(113, 99, 62)	0.951(0.932)	108.9(107.2)	(121, 69)	0.951	124.7	
4	(80, 65, 28)	0.951(0.949)	48.0(46.5)	(92, 42)	0.952	60.0	
1	(137, 119, 78)	0.991(0.988)	80.0(79.3)	(152, 86)	0.990	94.9	$K = 33$ $\kappa = 3$ $\alpha = 0.99$
2	(87, 73, 44)	0.990(0.987)	57.1(56.2)	(99, 53)	0.990	69.8	
3	(130, 116, 83)	0.990(0.986)	146.3(146.3)	(138, 89)	0.990	162.7	
4	(97, 83, 51)	0.990(0.988)	66.9(66.2)	(110, 62)	0.991	79.3	
1	(113, 89, 42)	0.902(0.877)	51.4(49.0)	(125, 55)	0.904	63.5	$K = 30$ $\kappa = 5$ $\alpha = 0.90$
2	(69, 51, 15)	0.904(0.886)	36.2(34.8)	(78, 30)	0.906	46.3	
3	(105, 87, 51)	0.900(0.876)	90.6(88.9)	(111, 59)	0.902	103.1	
4	(73, 54, 17)	0.901(0.895)	40.1(38.4)	(83, 33)	0.906	50.0	
1	(121, 97, 56)	0.951(0.940)	61.3(60.5)	(133, 66)	0.952	73.3	$K = 30$ $\kappa = 5$ $\alpha = 0.95$
2	(75, 57, 27)	0.952(0.949)	43.5(42.4)	(85, 38)	0.953	53.7	
3	(113, 95, 63)	0.950(0.938)	109.3(108.2)	(120, 69)	0.950	121.5	
4	(81, 62, 30)	0.951(0.945)	49.1(47.9)	(91, 42)	0.951	58.3	
1	(138, 115, 79)	0.990(0.988)	81.1(80.7)	(150, 86)	0.990	92.4	$K = 30$ $\kappa = 5$ $\alpha = 0.99$
2	(88, 71, 45)	0.990(0.988)	58.4(57.6)	(98, 53)	0.990	68.1	
3	(130, 113, 84)	0.990(0.988)	147.1(146.9)	(137, 89)	0.990	159.4	
4	(98, 80, 52)	0.990(0.990)	67.8(66.9)	(108, 62)	0.990	77.4	
1	(107, 80, 51)	0.903(0.890)	46.9(45.9)	(111, 59)	0.902	51.5	$K = 15$ $\kappa = 5$ $\alpha = 0.90$
2	(65, 46, 24)	0.904(0.910)	33.9(33.6)	(69, 33)	0.906	38.1	
3	(100, 80, 57)	0.901(0.891)	83.0(81.5)	(103, 63)	0.906	89.0	
4	(69, 49, 26)	0.901(0.908)	37.7(37.4)	(73, 36)	0.902	41.7	
1	(115, 89, 62)	0.950(0.945)	55.9(55.3)	(120, 69)	0.950	60.8	$K = 15$ $\kappa = 5$ $\alpha = 0.95$
2	(71, 53, 33)	0.952(0.955)	40.9(40.5)	(76, 41)	0.955	45.5	
3	(108, 89, 68)	0.951(0.950)	101.1(100.6)	(111, 73)	0.953	106.9	
4	(77, 58, 37)	0.951(0.950)	46.6(46.0)	(82, 45)	0.950	50.3	
1	(132, 107, 84)	0.990(0.990)	75.2(75.0)	(137, 89)	0.990	79.7	$K = 15$ $\kappa = 5$ $\alpha = 0.99$
2	(85, 67, 50)	0.991(0.992)	55.8(55.4)	(89, 56)	0.991	59.9	
3	(125, 107, 88)	0.990(0.991)	137.9(137.8)	(127, 92)	0.990	142.7	
4	(95, 76, 57)	0.990(0.990)	65.0(64.5)	(99, 65)	0.990	69.2	

Our numerical investigations show that the coordinated control algorithm provides good approximations for the quantities of interest, including the service level. *Ceteris paribus* the differences $S_i^* - c_i^*$, for each item i , seem practically independent of both the service level constraint and the lead time; dependence on the parameters of the compound Poisson demand process seems restricted to $\lambda_i ED_i$, the average demand per unit time. In comparison with the heuristics in Silver (1974) and Thompstone and Silver (1975), our algorithm gives in general slightly better results; more importantly it can also handle nonzero lead times for the case of compound Poisson demands. Finally, we note that our numerical results show that considerable cost savings may be achieved by using suboptimal coordinated control instead of the best independent control.

Appendix

PROOF OF LEMMA 1. It readily follows from the definition of the functions k_R , t_R and $\gamma((i, 0), i) = c(i)$ that for any $i > s$,

$$k_R(i) = \gamma((i, 0), i) + \sum_{j>s} p_{i(j,0)}(i)k_R(j) + K \sum_{j\leq s} p_{i(j,0)}(i) \\ + p_{i(i,1)}(i)\{\delta(i-c)k_R(i) + (1-\delta(i-c))\kappa\},$$

$$t_R(i) = (\lambda + \mu)^{-1} + \sum_{j>s} p_{i(j,0)}(i)t_R(j) + p_{i(i,1)}(i)\delta(i-c)t_R(i).$$

Subtracting the second equality g_R times from the first one and using (2.5)–(2.6) we find for any $i > s$,

$$k_R(i) - g_R t_R(i) = \gamma((i, 0), i) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{iy}(i)v_R(y).$$

In view of $i = R_{(i,0)}$ for $i > s$, and $i = R_{(i,1)}$ and $\gamma((i, 1), i) = \gamma((i, 0), i)$ for $i > c$ this verifies that $v_R(i, 0)$, $i > s$ and $v_R(i, 1)$, $i > c$ satisfy (2.10). Using the fact that $v_R(S, 0) = k_R(S) - g_R t_R(S) = 0$ (cf. (2.3) and (2.5)), we next verify that (2.10) is satisfied for $x = (i, 0)$, $i \leq s$ by:

$$v_R(i, 0) = K = K + v_R(S, 0) \\ = K + \gamma((S, 0), S) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{Sy}(S)v_R(y) \\ = \gamma((i, 0), R_{(i,0)}) - g_R(\lambda + \mu)^{-1} + \sum_{y \in X} p_{iy}(R_{(i,0)})v_R(y).$$

The verification for the final case $(i, 1)$, $i \leq c$ is analogous.

LEMMA 3. *The policy-iteration algorithm for the one-item model in §2 converges in a finite number of steps.*

PROOF. We first show that under any rule R of the (S, c, s) class the Markov chain describing the state at the decision epochs has a single ergodic set which is positive recurrent. Let $j^* = \min\{j | \phi(j) > 0\}$ and note that state $(S - j^*, 0)$ can be reached from any other state in X . Clearly the mean recurrence time from state $(S - j^*, 0)$ to itself is finite, thus verifying the assertion. Next, let $\bar{R} \neq R$ be a successor policy of R in the policy-iteration algorithm. Let $\bar{\pi}(\cdot)$ denote the unique steady state distribution of the Markov chain under rule \bar{R} . Multiply (2.12) with $\bar{\pi}(\cdot)$ to conclude that $g_{\bar{R}} \leq g_R$. We show that either

$$(i) \quad g_{\bar{R}} < g_R \quad \text{or}$$

$$(ii) \quad g_{\bar{R}} = g_R \quad \text{and} \quad v_{\bar{R}}(x) \leq v_R(x),$$

$$x \in X \quad \text{with strict inequality for some } x \in X. \quad (A1)$$

Since the algorithm only considers a finite number of policies (note $L \leq s < c < S \leq U$) and since (A1) excludes cycling, this proves the lemma.

First consider the case $\bar{S} \neq S$. Note that $\{(i, 0) | i \leq \bar{s}\} \cup \{(i, 1) | i \leq \bar{c}\}$ contains states x that are positive recurrent under rule \bar{R} and for which $I_R(x, \bar{R}_x) < v_R(x)$ (cf. part (a) of the policy-improvement step to verify $v_R(\bar{S}, 0) < v_R(S, 0)$ and use (2.14) and (2.16)). Multiply (2.12) with $\bar{\pi}(\cdot)$ using the above strict inequality to conclude that $g_{\bar{R}} < g_R$.

Next, let $\bar{S} = S$. If (A1)(i) fails to hold, $g_{\bar{R}} = g_R$. For $i \leq \bar{s}$, we have by (2.14), (2.3)

and (2.5) that $v_R(i, 0) \geq I_R((i, 0), \bar{R}_{(i,0)}) = K + v_R(S, 0) = K = v_{\bar{R}}(i, 0)$. For $i \leq \bar{c}$, one finds $\kappa = v_{\bar{R}}(i, 1) \leq v_R(i, 1)$ in a similar way. Finally let $X^N = \{(i, 0) | i > \bar{s}\} \cup \{(i, 1) | i > \bar{c}\}$ be the set of states in which no replenishment is ordered under \bar{R} . For given starting state $\xi_0 = x \in X^N$, denote by $\{\xi_k\}$ the sequence of states adopted by the Markov chain under rule \bar{R} and let $\nu \geq 0$ be the number of transitions before the first visit to the set $X \setminus X^N$. By repeated substitution in the policy-improvement test quantity $I_R(\cdot)$ and using $v_R(i, 0) \geq K$ for $i \leq \bar{s}$ and $v_R(i, 1) \geq \kappa$ for $i \leq \bar{c}$, we obtain for $x \in X^N$

$$v_R(x) \geq I_R(x, \bar{R}_x) \\ \geq E \left\{ \sum_{k=0}^{\nu} \left[\gamma(\xi_k, \bar{R}_{\xi_k}) - g_R(\lambda + \mu)^{-1} \right] \right\} + q_{\bar{R}}(x)K + (1 - q_{\bar{R}}(x))\kappa. \quad (A2)$$

Using the relations

$$E \left\{ \sum_{k=0}^{\nu} \gamma(\xi_k, \bar{R}_{\xi_k}) \right\} + q_{\bar{R}}(x)K + (1 - q_{\bar{R}}(x))\kappa = k_{\bar{R}}(x), \\ (E\nu + 1)(\lambda + \mu)^{-1} = t_{\bar{R}}(x) \quad \text{and} \quad g_R = g_{\bar{R}},$$

we have by (A2) that $v_R(x) \geq I_R(x, \bar{R}_x) \geq v_{\bar{R}}(x)$ for all $x \in X^N$. Hence we have verified that $v_R(x) \geq I_R(x, \bar{R}_x) \geq v_{\bar{R}}(x)$ for all $x \in X$. Since $\bar{R} \neq R$, we have $v_R(x) > I_R(x, \bar{R}_x)$ for some $x \in X$, thus proving (A1)(ii) and hence the lemma.²

²We are indebted to the referees for helpful suggestions on the presentation of the paper. Also, we thank Luuk Seelen for his help with the numerical work.

References

- ADELSON, R. M., "Compound Poisson Distributions," *Oper. Res. Quart.*, 17 (1966), 73-75.
- ALBIN, S., "On Poisson Approximations for Superposition Arrival Processes in Queues," *Management Sci.*, 28 (1982), 126-137.
- ARCHIBALD, B. C., "Continuous Review (s, S) Policies for Discrete Compound Poisson Demand Processes," unpublished Ph.D. thesis, University of Waterloo, Waterloo, Ontario, 1976.
- AND E. A. SILVER, "(s, S) Policies under Continuous Review and Discrete Compound Poisson Demands," *Management Sci.*, 24 (1978), 899-909.
- BALINTFY, J. L., "On a Basic Class of Multi-Item Inventory Problems," *Management Sci.*, 10 (1964), 287-297.
- BROWN, R. G., *Decision Rules for Inventory Management*, Holt, Rinehart & Winston, New York, 1967.
- CINLAR, E., "Superposition of Point Processes," in P. A. Lewis (Ed.), *Stochastic Point Processes, Statistical Analysis, Theory and Applications*, Wiley, New York, 1972.
- EPPEN, G. AND L. SCHRAGE, "Centralized Ordering Policies in a Multi-Warehouse System with Lead Times and Random Demand," in L. Schwarz (Ed.), *Multi-Level Production/Inventory Control Systems: Theory and Practice*, North-Holland, Amsterdam, 1981.
- FEDERGRUEN, A. AND Z. SCHECHNER, "Cost Formulae for Continuous Review Inventory Models with Fixed Delivery Lags," *Oper. Res.*, (1984) (to appear).
- , P. J. SCHWEITZER AND H. C. TIJMS, "Denumerable Undiscounted Semi-Markov Decision Processes with Unbounded Rewards," *Math. Oper. Res.*, 8 (1983), 298-314.
- AND P. ZIPKIN, "Approximations of Dynamic Multi-Location Production and Inventory Problems," *Management Sci.*, (1984a) (to appear).
- AND —, "Computational Issues in an Infinite-Horizon, Multi-Echelon Inventory Model," *Oper. Res.*, (1984b) (to appear).
- AND —, "Allocation Policies and Cost Approximations for Multilocation Inventory Systems," *Naval Res. Logist. Quart.*, (1984c) (to appear).
- AND —, "An Efficient Algorithm for Computing Optimal (s, S) Policies," *Oper. Res.*, (1984d), (to appear).
- HOWARD, R. A., *Dynamic Programming and Markov Processes*, Wiley, New York, 1960.
- IGNALL, E., "Optimal Continuous Review Policies for Two Product Inventory Systems with Joint Set-up Costs," *Management Sci.*, 15 (1969), 277-279.
- NADDOR, E., "Optimal and Heuristic Decisions in Single and Multi-Item Inventory Systems," *Management Sci.*, 21 (1975), 1234-1249.

- NORMAN, J. M., *Heuristic Procedures in Dynamic Programming*, Manchester University Press, Manchester, 1972.
- PETERSON, R. AND E. A. SILVER, *Decision Systems for Inventory Management and Production Planning*, Wiley, New York, 1979.
- ROSS, S. M., *Applied Probability Models with Optimization Applications*, Holden-Day, Inc., San Francisco, 1970.
- SILVER, E. A., "A Control System for Coordinated Inventory Replenishment," *Internat. J. Production Res.*, 12 (1974), 647-670.
- THOMPSTONE, R. M. AND E. A. SILVER, "A Coordinated Inventory Control System for Compound Poisson Demand and Zero Lead Time," *Internat. J. Production Res.*, 13 (1975), 581-602.
- TIJMS, H. C., "An Algorithm for Average Cost Denumerable State Semi-Markov Decision Problems with Applications to Controlled Production and Queueing Systems," in R. Hartley et al. (Eds.), *Recent Developments in Markov Decision Processes*, Academic Press, New York, 1980.
- AND H. GROENEVELT, "Approximations for (s, S) Inventory Systems with Stochastic Lead Times and a Service Level Constraint," *European J. Oper. Res.* (to appear).
- WIJNGAARD, J., "Decomposition for Dynamic Programming in Production and Inventory Control," *Engineering and Process Econom.*, 4 (1979), 385-388.

CALL FOR PAPERS

Mathematical Programming Study on:

Probabilistic Analysis of the Simplex and Related Methods

Papers are solicited for a *Mathematical Programming Study* on the topic mentioned above. We are looking for papers that deal with the probabilistic analysis of the simplex method, its variants and homotopy methods like Lemke's algorithm and complementary pivoting methods of fixed point theory. Also, papers dealing with random polytopes and their statistical properties, as well as a probabilistic or statistical analysis of the pivoting process are also welcome.

Mathematical Programming Studies are published by the Mathematical Programming Society, and there are typically two or three studies a year. Each study is devoted to a specific topic of current interest, and some of the recent titles are Network Models and Applications, Engineering Optimization, and Mathematical Programming in Use. All papers submitted for such studies are refereed, and for us to put out this study in a timely manner, we would appreciate your contributions by June 30, 1984.

Please submit your contributions to one of us listed below:

PROFESSOR R. SAIGAL
Department of Industrial Engineering
and Management Sciences
Northwestern University
Evanston, Illinois 60201

PROFESSOR ILAN ADLER
Department of Industrial Engineering
and Operations Research
University of California
Berkeley, California 94720