On the $(S-1, S)$ lost sales inventory model with priority demand classes

R. Dekker* R.M. Hill† M.J. Kleijn*

October 1997

Abstract

In this paper an inventory model with several demand classes, prioritised according to importance, is analysed. We consider a lot-for-lot or $(S-1, S)$ inventory model with lost sales. For each demand class there is a critical stock level at and below which demand from that class is not satisfied from stock on hand. In this way stock is retained to meet demand from higher priority demand classes. A set of such critical levels determines the stocking policy. For Poisson demand and a generally distributed lead time we derive expressions for the service levels for each demand class and the average total cost per unit time. Efficient solution methods for obtaining optimal policies, with and without service level constraints, are presented. Numerical experiments in which the solution methods are tested demonstrate that significant cost reductions can be achieved by distinguishing between demand classes.

Keywords: Inventory, demand classes, spare parts, lost sales, rationing.

1 Introduction

In this paper, we consider a single-location, single-item inventory problem, where demand may be categorised into classes of different importance. There are a number of practical contexts in which such a model is applicable. For example, demand from key customers may be given a higher priority than demand arising from less important customers. A second example occurs in a multi-echelon inventory system where the highest echelon may face demand both from customers and from lower echelon stocking points, where customer demand would normally be considered more important. A third example occurs in the context of a retail chain where outlets have their stock replenished from a central warehouse and it may be desirable to assign different priorities to different outlets. A fourth example where several demand classes

*Erasmus University Rotterdam, Econometric Institute, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands.

†University of Exeter, MSOR Department, Laver Building, North Park Road, Exeter EX4 4QE, England.

may be distinguished is an assemble-to-order system where a common component is shared by several end-products which have different values to the firm. A final example occurs in spare parts inventory control where an item is used in several types of equipment of varying degrees of criticality. If a part that is installed in some equipment fails, and hence causes a demand for the spare parts inventory system, then the cost of a stockout depends on the degree of criticality of the equipment in which the part was installed.

The most analysed rationing policy is a so-called critical-level policy, where part of the stock is reserved for high priority demand classes. This policy has been shown to be optimal for some specific models. In this paper a critical-level, lot-for-lot policy with n demand classes and Poisson demand is analysed. If physical stock is low then it is preserved to meet demand from higher priority demand classes while demand from lower priority demand classes is lost. Actually we only need to assume that demand not satisfied from stock on hand is lost from the normal replenishment system, for example it may be expedited or met in some other way. This assumption often applies in practical situations, especially when the replenishment lead time is relatively long. The key assumption is that an order not satisfied from stock on hand does not trigger a replenishment order. The cost per unit of lost sales is assumed to be different for each demand class, greater for high priority demand than for low priority demand.

An important contribution of this paper is the derivation of expressions for the service levels for each demand class and for the average total cost per unit time, which are easy to calculate and therefore are suitable for practical use. This is the topic of Section 3. Another valuable contribution is given in Section 4, where we present efficient solution methods for determining the optimal critical-level, lot-for-lot policy, with or without service level constraints. In addition, for the problem without a service level constraint a fast heuristic approach is presented. In Section 5 the quality of these algorithms is tested on a number of numerical examples and it turns out that our exact solution methods are efficient and the heuristic approach performs very well. In Section 2 an overview of the literature on inventory models with several demand classes is presented. We mention here that our work is closely related to a recent paper by Ha [8], who proved the optimality of the critical-level, lot-for-lot policy for exponentially distributed lead times. In our paper the results of Ha [8] are extended to generally distributed lead times, including a fixed lead time. It should be noted, however, that the optimality of our policy is no longer guaranteed for a general lead time distribution, because information on the remaining time until the next replenishment may influence the optimal critical levels and order-up-to level. However, in order to use this additional information the inventory policy would have to become much more complicated. Since we feel that in practical situations there is a need for policies which are easy to implement and to understand, we only consider the simple policy where the critical levels and the order-up-to level are constant. Finally, we like to stress that this paper has real practical value. The problem of dealing with priority demand classes has become more important but until now efficient tools to tackle the problem have been missing. The results in this paper are easy to use and thus are suitable for implementation in logistics software packages.

2 Related work

Veinott [15] was the first to consider the problem of several demand classes. He considered a periodic review inventory model, with n demand classes and zero lead time, and he introduced the concept of a critical-level policy. Topkis [14] proved the optimality of this policy both for the case of backordering and for the case of lost sales. He made the analysis easier by breaking down the period until the next ordering opportunity into a finite number of subintervals. In any given interval the optimal rationing policy was such that one should satisfy demand from a given class from existing stock as long as there is no unsatisfied demand from a higher class remaining and the stock level does not drop below a certain critical level for that class. These critical levels are generally decreasing with the remaining time until the next ordering opportunity. Independent of Topkis [14], Evans [5] and Kaplan [10] derived essentially the same results. Recently, Atkins and Katircioğlu [1] analysed a periodic review inventory system with several demand classes, backordering and a fixed lead time, where for each class a minimum service level was required. For this model they presented a heuristic rationing policy. Cohen, Kleindorfer & Lee [2, 3] also considered the problem of two demand classes, but they did not analyse a critical-level policy.

The first contribution considering several demand classes in a continuous review inventory model was made by Nahmias & Demmy [11]. They analysed a (Q, R) inventory model with two demand classes, Poisson demand, backordering, a fixed lead time and a critical-level policy, under the assumption that there is never more than a single order outstanding. This assumption implies that whenever a replenishment order is triggered, the net inventory and the inventory position are identical. Their main contribution was the derivation of (approximate) expressions for the fill rates. Dekker, Kleijn & De Rooij [4] considered a lot-for-lot inventory model with the same characteristics, but without the assumption of at most one outstanding order. They discussed a case study on the inventory control of slow moving spare parts in a large petrochemical plant, where parts were installed in equipment of different criticality. Their most important result was the derivation of (approximate) expressions for the fill rates for both demand classes. Ha [9] discussed a similar model, but with exponentially distributed lead times, which allowed the problem to be formulated as a queueing model. He showed that in this setting a critical-level policy is optimal, with the critical level decreasing in the number of backorders of the low-priority class. A critical-level policy for two demand classes where the critical level depends on the remaining time until the next stock replenishment was discussed by Teunter & Klein Haneveld [13]. They showed that such a policy outperforms a simple policy where the critical levels are stationary.

The work most related to our paper is a recent contribution by Ha [8]. He also considers an inventory model with several demand classes, Poisson demand and lost sales. For exponentially distributed lead times he proved that a lot-for-lot ordering policy and a critical-level rationing policy is optimal. Moreover, for two demand classes he presented expressions for the expected inventory level and the stockout probabilities. To determine the optimal policy he used an exhaustive search, and he used (without proof) the assumption that the average cost is unimodal in the order-up-to level.

3 The model

3.1 Definitions and assumptions

Consider a critical-level, lot-for-lot policy with n demand classes and lost sales. The demand classes are ranked according to priority, with class 1 having the highest priority and class n having the lowest priority. The maximum stock level or 'order-up-to' level is S. Therefore, whenever a sale (a satisfied demand) of 1 unit occurs, a replenishment order for 1 unit is placed to raise the inventory position to S. The lead times for replenishment are assumed to be independent and identically distributed. Observe that a fixed lead time, which is more likely to be applicable in any practical context, is also included in this setting. The demand from each class is assumed to follow a Poisson process. The critical level for demand from class j is c_{j-1}, $j = 1, \ldots, n$. If the physical stock is above this level then demand from class j is satisfied from stock but if physical stock is at or below this level then demand from class j is lost.
The following terminology is used:

n	the number of demand classes
S	the maximum stock level or 'order-up-to' level; $S \in I\!N_+$
c_{j-1}	the critical level for demand class j, $j = 1, \ldots, n$; $0 =: c_0 \leq c_1 \leq \cdots \leq c_{n-1} \leq c_n := S$, $c_j \in I\!N_+$
λ_j	the (Poisson) demand rate for class j, $j = 1, \ldots, n$; $\lambda_j > 0$
Λ_i	the aggregate demand rate if the number of outstanding orders is i and hence physical stock is $S - i$, $i = 0, \ldots, S-1$; $\Lambda_i > 0$
L	the average lead time for replenishment orders; $L > 0$
h	the holding cost per unit per unit time; $h > 0$
π_j	the lost sales cost per unit of demand from class j, $j = 1, \ldots, n$; $\pi_1 > \cdots > \pi_n > 0$
K	the total cost associated with purchasing (producing) one unit; $K \geq 0$

The vector $(\boldsymbol{c}, S) := (c_1, \ldots, c_{n-1}, S)$ defines a policy. The characteristics of this policy can be illustrated by reference to Figure 1.
It is assumed that there are two demand classes and that the initial stock level equals the order-up-to level S ($S = 3$ is illustrated). Assume that at time τ_1 the stock level is c_1 ($c_1 = 1$

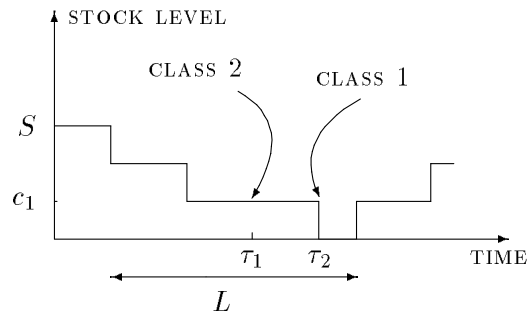

Figure 1: An illustration of the policy, with $n = 2$.

is illustrated) and a demand from class 2 occurs. In the traditional policy this demand would be satisfied from stock on hand and as a result the stock level would fall to zero. If at some subsequent time τ_2 a demand from class 1 occurs then there would be no stock available to satisfy this demand and hence a penalty cost of π_1 would be incurred. However, if the critical-level, lot-for-lot policy is applied, the demand from class 2 at time τ_1 would be lost and a cost of π_2 would be incurred and the class 1 demand occurring at τ_2 would now be satisfied from stock on hand. The cost benefit obtained by applying our strategy to this particular sequence of demands would be $\pi_1 - \pi_2$, to be offset, in this example, by an increase of $h(\tau_2 - \tau_1)$ in the stockholding cost (in addition to other possible repercussive costs).

3.2 Deriving the steady state probabilities for the system

Smith [12] analysed the $(S-1, S)$ lost sales inventory policy for a general independent lead time distribution. It is customary to define the state of the system i at any time as the number of replenishment orders outstanding, so that the physical stock is $S - i$. Given a fixed Poisson arrival rate of μ and a mean lead time of L, the steady state probability of being in state i is given by

$$p_i = \frac{(\mu L)^i}{i!} p_0, \qquad i = 0, 1, \ldots, S \tag{1}$$

with p_0 normalised in the usual way. For an exponential lead time distribution this result follows readily from an examination of the steady state transition diagram for the corresponding queueing system. That this result extends to a general lead time distribution (including a fixed lead time) is demonstrated in, for example, pages 244-248 of Gnedenko & Kovalenko [7]. If the Poisson arrival rate is state-dependent (μ_i in state i) then, for an exponential lead time distribution, an examination of the steady state diagram shows that (1) extends to

$$p_i = \frac{\prod_{k=0}^{i-1} (\mu_k L)}{i!} p_0, \qquad i = 0, 1, \ldots, S \tag{2}$$

with the product term, here and elsewhere in this paper, defined as 1 if the lower limit exceeds the upper limit by 1. That this result extends to a general lead time distribution (including a fixed lead time) is demonstrated in pages 250-252 of Gnedenko & Kovalenko [7]. In our critical-level, lot-for-lot policy the state-dependent arrival rate in state i is given by Λ_i, which is equal to $\lambda_1 + \cdots + \lambda_k$ if physical stock $S - i$ is in the range $[c_{k-1} + 1, c_k]$ or, equivalently, if $i \in [S - c_k, S - c_{k-1} - 1]$. Observe that this will be the case for one and only one value of k in the range $1, \ldots, n$ (remember that $c_0 := 0$ and $c_n := S$). Hence, by (2) it follows that the steady state probabilities for the system are given by

$$p_i = \left\{ \prod_{k=0}^{i-1} \Lambda_k \right\} \frac{L^i}{i!} p_0, \qquad i = 1, \ldots, S \tag{3}$$

3.3 Deriving the service levels and the average total cost per unit time

Let $\beta_j(\boldsymbol{c}, S)$ denote the service level (long-run fraction of demand satisfied from stock on hand) for demand class j, $j = 1, \ldots, n$, associated with a $(\boldsymbol{c}, S)$ policy. Then

$$\beta_j(\boldsymbol{c}, S) = 1 - \sum_{i=S-c_{j-1}}^{S} p_i \tag{4}$$

with p_i given by (3). It can easily be seen that $1 \geq \beta_1(\boldsymbol{c}, S) \geq \cdots \geq \beta_n(\boldsymbol{c}, S) \geq 0$.
In discussing the cost of the system in this and in subsequent sections 'cost' will be used as a shorthand for 'average total cost per unit time'. The cost associated with a $(\boldsymbol{c}, S)$ policy can be divided into the holding cost $C_h(\boldsymbol{c}, S)$, the penalty cost $C_p(\boldsymbol{c}, S)$ and the purchasing cost $C_o(\boldsymbol{c}, S)$, given by

$$\begin{aligned} C_h(\boldsymbol{c}, S) &= \sum_{i=0}^{S} h(S-i) p_i \\ C_p(\boldsymbol{c}, S) &= \sum_{j=1}^{n} \pi_j \lambda_j (1 - \beta_j(\boldsymbol{c}, S)) \\ C_o(\boldsymbol{c}, S) &= \sum_{j=1}^{n} K \lambda_j \beta_j(\boldsymbol{c}, S) \end{aligned} \tag{5}$$

The purchase cost $C_o(\boldsymbol{c}, S)$ can be incorporated into the penalty cost $C_p(\boldsymbol{c}, S)$ by defining new penalty costs $\pi'_j := \pi_j - K$, $j = 1, \ldots, n$, and adding a constant equal to $\sum_{j=1}^{n} \lambda_j K$. Observe that it may be assumed that π'_j is still positive, otherwise it would be optimal never to satisfy any demand from class j. Hence, in the remainder of the paper, the purchasing cost will not be taken into account ($K = 0$) and thus the total cost is assumed to be given by

$$C(\boldsymbol{c}, S) := C_h(\boldsymbol{c}, S) + C_p(\boldsymbol{c}, S) \tag{6}$$

In the next section we shall discuss the optimisation of the policy parameters, both with and without service level constraints.

4 Optimisation

In this section the optimisation of the policy will be analysed. First, we consider the problem of determining that critical-level, lot-for-lot policy which minimises the holding cost subject to the service levels attaining or exceeding prescribed minimum levels. Then we consider the problem of determining that policy which minimises cost.

4.1 Service level optimisation

In this subsection the following optimisation problem will be analysed.

$$\min\{C_h(\boldsymbol{c},S) : 0 \le c_1 \le \cdots \le c_{n-1} \le S, \beta_j(\boldsymbol{c},S) \ge \beta_j, j=1,\ldots,n\}$$

where $0 < \beta_n < \cdots < \beta_1 < 1$ gives the minimum required service levels for each demand class. We assume that only the holding cost needs to be minimised, which is a common assumption for service level optimisation models. Before deriving lower and upper bounds on the optimal order-up-to level we need the following results.

Lemma 4.1 *Consider an $(S-1,S)$ policy with a state-dependent Poisson arrival rate and a general lead time distribution with mean L. The system is in state i if there are i replenishment orders outstanding, and hence $S-i$ units in stock, and the arrival rate in state i is μ_i. Let $p_0,\ldots,p_S$ be the steady state probabilities for the system. If the arrival rate in state k $(0 \le k \le S-1)$ is reduced from μ_k to μ'_k $(\mu'_k < \mu_k)$ and the new steady state probabilities are $p'_0,\ldots,p'_S$, then it follows that*

$$p'_i > p_i$$

for $0 \le i \le k$ and

$$p'_i < p_i$$

for $k+1 \le i \le S$.

Proof: Define

$$\begin{aligned}
A_{1,i} &:= \left\{\prod_{j=0}^{i-1}\mu_j\right\}\frac{L^i}{i!} \qquad (0 \le i \le k)\\
A_{2,i} &:= \left\{\prod_{j=0,j\ne k}^{i-1}\mu_j\right\}\frac{L^i}{i!} \qquad (k+1 \le i \le S)\\
B &:= \sum_{m=0}^{k}\left\{\prod_{j=0}^{m-1}\mu_j\right\}\frac{L^m}{m!}\\
C &:= \sum_{m=k+1}^{S}\left\{\prod_{j=0,j\ne k}^{m-1}\mu_j\right\}\frac{L^m}{m!}
\end{aligned}$$

If $0 \leq i \leq k$ then

$$p_i = \frac{A_{1,i}}{B + C\mu_k}$$

and

$$p'_i = \frac{A_{1,i}}{B + C\mu'_k}$$

and the first part of the result follows directly. To show the second part we observe that for $k+1 \leq i \leq S$

$$p_i = \frac{A_{2,i}\mu_k}{B + C\mu_k}$$

and

$$p'_i = \frac{A_{2,i}\mu'_k}{B + C\mu'_k}$$

Hence, it follows that

$$\begin{aligned} p_i - p'_i &= \frac{A_{2,i}\mu_k(B + C\mu'_k) - A_{2,i}\mu'_k(B + C\mu_k)}{(B + C\mu_k)(B + C\mu'_k)} \\ &= \frac{A_{2,i}B(\mu_k - \mu'_k)}{(B + C\mu_k)(B + C\mu'_k)} \\ &> 0 \end{aligned}$$

and thus the desired result is proved. □

Theorem 4.1 *For all policies* $(\boldsymbol{c}, S)$ *it follows that*

$$\beta_j(\boldsymbol{c}, S+1) > \beta_j(\boldsymbol{c}, S)$$

for $j = 1, \ldots, n$, *and*

$$C_h(\boldsymbol{c}, S+1) > C_h(\boldsymbol{c}, S)$$

Moreover,

$$C_h(\boldsymbol{c} + \boldsymbol{\delta}, S) > C_h(\boldsymbol{c}, S)$$

with $\boldsymbol{\delta} \in I\!N_+^{n-1}$ *such that* $0 \leq c_1 + \delta_1 \leq \ldots \leq c_{n-1} + \delta_{n-1} \leq S$.

Proof: Consider a lot-for-lot policy with state-dependent arrival rates μ_i. Assume that initially we have a maximum stock of $S+1$, but μ_0 is set equal to some arbitrarily large number, so effectively we have a maximum stock of S. Then by reducing μ_0 to a finite μ'_0 we obtain a genuine maximum stock level of $S+1$. The first two results can now easily be verified from Lemma 4.1 and relations (4) and (5). To show the last part we observe that increasing the critical levels from $\boldsymbol{c}$ to $\boldsymbol{c} + \boldsymbol{\delta}$ implies that in some of the states the arrival rate is reduced. From Lemma 4.1 and (5) one can verify that reducing μ_k to μ'_k ($\mu'_k < \mu_k$) leads to an increase in the holding cost, and so the last part of the theorem follows by successively applying Lemma 4.1. □

From the above result we obtain that if $\beta_1(\mathbf{0},\Delta) \geq \beta_1$ (and thus $\beta_j(\mathbf{0},\Delta) \geq \beta_j$ for all j) it is never optimal to consider values of $S > \Delta$ and so Δ is an upper bound on the optimal order-up-to level. From the last part of the theorem it follows that the optimal policy with a maximum stock level of Δ is $(\mathbf{0},\Delta)$. The next result enables us to find a lower bound on the optimal order-up-to level.

Theorem 4.2 *If policy $(\mathbf{0},S)$ does not meet the service level requirements for demand class n, then no other policy having a maximum stock level of S meets that service level requirement.*

Proof: This proof uses a queueing theory argument based on the implicit assumption that lead times are exponentially distributed. Since the steady state probabilities for the system are the same for a generally distributed lead time as they are for an exponentially distributed lead time, the arguments carry across to a general lead time distribution. The initial policy $(\mathbf{0},S)$ corresponds to a queue with S servers, no queueing, a fixed arrival rate of Λ_0 in states $0,\ldots,S-1$ and an exponential service rate of j/L in state j. Consider an alternative policy $(\boldsymbol{c},S)$ with $0 < c_{n-1} < S$. In states $0,\ldots,S-c_{n-1}-1$ the arrival rate is still Λ_0. In states $S-c_{n-1},\ldots,S-1$ the arrival rate is less than Λ_0 but greater than zero. The total probability of being in any of the states $S-c_{n-1},\ldots,S-1$ is greater than the probability of being in state $S-c_{n-1}$ on its own if access to any of the higher states $(S-c_{n-1}+1,\ldots,S)$ were not possible, since access to these higher states increases the mean time spent in states $S-c_{n-1},\ldots,S$ before a return is made to state $S-c_{n-1}-1$. If no access to a higher state were possible then the probability of being in state $S-c_{n-1}$ is the probability of being out of stock in a lot-for-lot system with order-up-to level $S-c_{n-1}$ and arrival rate Λ_0. This probability is greater than the probability of being out of stock in a lot-for-lot system with order-up-to level $S > S-c_{n-1}$ and arrival rate Λ_0, which is the initial policy (this follows from the fact that the service level of a lot-for-lot policy increases with the order-up-to level). Therefore, the total probability of being in any of the states $S-c_{n-1},\ldots,S$ for the alternative policy is greater than the probability of being in state S for the initial policy and the result follows. □

From this result we obtain that if $\beta_n(\mathbf{0},\Delta) < \beta_n$ for some $\Delta \geq 0$ then it is never optimal to consider $S \leq \Delta$ and thus $\Delta+1$ is a lower bound on the optimal S. We are now able to construct an algorithm to obtain an optimal solution for the above optimisation problem.
The algorithm starts by determining the smallest value of Δ_{max} for which $\beta_1(\mathbf{0},\Delta_{max}) \geq \beta_1$. Since it follows from Theorem 4.1 that $\beta_1(\mathbf{0},x)$ is strictly increasing in x this value is easily determined. Similarly, it is easy to compute the lower bound Δ_{min}. With these bounds we can use enumeration over all policies $(\boldsymbol{c},S)$ satisfying $\Delta_{min}+1 \leq S \leq \Delta_{max}-1$ and $\beta_j(\boldsymbol{c},S) \geq \beta_j$, $j=1,\ldots,n$, to find the best policy. It can be shown that the number of possible policies $(\boldsymbol{c},S)$ with $\Delta_{min}+1 \leq S \leq \Delta_{max}-1$ is given by

$$\binom{n+\Delta_{max}-1}{n} - \binom{n+\Delta_{min}}{n}$$

Step 0: Set $\Delta_{max} := \text{argmin}\{x \geq 0 : \beta_1(\mathbf{0}, x) \geq \beta_1\}$.
Set $\Delta_{min} := \text{argmax}\{x \geq 0 : \beta_n(\mathbf{0}, x) < \beta_n\}$.
Step 1: If $\Delta_{max} = \Delta_{min} + 1$ then `stop` and take $(\mathbf{0}, \Delta_{max})$ as an optimal solution; otherwise go to Step 2.
Step 2: Solve the optimisation problem
$\min\{C_h(\boldsymbol{c}, S) : 0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S, \Delta_{min} + 1 \leq S \leq \Delta_{max} - 1,$
$\beta_j(\boldsymbol{c}, S) \geq \beta_j, j = 1, \ldots, n\}$ and let $(\boldsymbol{c}^*, S^*)$ denote an optimal solution.
Step 3: Take $(\boldsymbol{c}^*, S^*)$ as an optimal solution if $C_h(\boldsymbol{c}^*, S^*) < C_h(\mathbf{0}, \Delta_{max})$, otherwise take $(\mathbf{0}, \Delta_{max})$ as the optimal solution.

Algorithm 4.1: Determination of optimal parameters for service level optimisation.

(see e.g. Feller [6], pages 38-39). In most real-life applications the values of n and Δ_{max} will be relatively small, and thus the algorithm will be fast enough. For example, for $n = 4$, $\Delta_{min} = 3$ and $\Delta_{max} = 20$, the number of possible policies is 8820.

4.2 Cost optimisation

In this subsection the following optimisation problem is considered.

$$\min\{C(\boldsymbol{c}, S) : 0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S\}$$

with $C(\boldsymbol{c}, S)$ given by (6). First, we derive an algorithm to determine the true optimal cost, and then we present a heuristic approach which gives a solution very close to (or the same as) the optimal one but requiring much less computer time. To obtain lower and upper bounds on the optimal order-up-to level we introduce an equivalent optimisation problem but with all unit lost sales costs equal to π_n, the cost of the least important class. This will have an optimal solution with all critical levels equal to zero since now we are not differentiating between demand. Hence, the optimal policy will be an $(S-1, S)$ policy, with arrival rate $\Lambda_0 = \lambda_1 + \cdots + \lambda_n$ and penalty cost π_n. We refer to this policy as the *traditional* policy and denote its cost by $C^T(S)$. For a *given* value Δ the cost of the traditional policy with $S = \Delta$ must be less than the cost of the optimal critical-level, lot-for-lot policy with $\pi_1 > \cdots > \pi_n$ and $S = \Delta$, i.e.

$$C^T(\Delta) \leq \min\{C(\boldsymbol{c}, S) : 0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S, S = \Delta\} \tag{7}$$

We define S' as the optimal order-up-to level given that all critical levels equal zero and by $(\boldsymbol{c}^*, S^*)$ we denote the optimal critical-level, lot-for-lot policy. We may now determine a lower bound and an upper bound on S^*.

Theorem 4.3 *A lower bound on the optimal order-up-to level S^* is given by*

$$\min\{x \geq 0 : C^T(x) \leq C(\mathbf{0}, S')\}$$

An upper bound on S^* is given by $x \geq 0$, where x satisfies $C^T(x+1) > C^T(x)$ and

$$C^T(x+1) \geq \min\{C(\boldsymbol{c}, S) : 0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S, S \leq x\}$$

Proof: Consider an arbitrary $x \geq 0$ such that $C^T(x) > C(\mathbf{0}, S')$. Then by (7) and the trivial inequality $C(\mathbf{0}, S') \geq C(\boldsymbol{c}^*, S^*)$ we obtain

$$\min\{C(\boldsymbol{c}, S) : 0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S, S = x\} > C(\boldsymbol{c}^*, S^*)$$

implying that $S^* \neq x$. Hence, it follows that S^* satisfies $C^T(S^*) \leq C(\mathbf{0}, S')$ and the first result is proved. To prove the second part of the theorem we observe from Smith [12] that $C^T(x+1) > C^T(x)$ implies $C^T(y) > C^T(x)$ for all $y > x$. If additionally we have

$$C^T(x+1) \geq \min\{C(\boldsymbol{c}, S) : 0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S, S \leq x\}$$

then the desired result follows by (7). □

In Figure 2 these bounds are illustrated. Note that *real* applies to the minimum cost of the critical-level, lot-for-lot policy for a given order-up-to level, *traditional* refers to the above mentioned traditional policy and *lot-for-lot* applies to the optimal policy given that all critical levels equal zero. A solution method to determine the optimal policy parameters, using the lower and upper bound, is proposed in Algorithm 4.2.

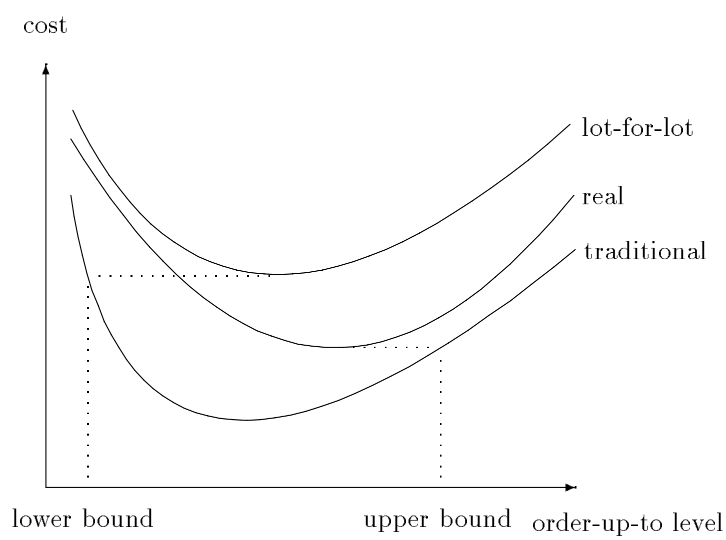

Figure 2: Illustration of bounds for cost optimisation.

Step 0: Set $C^* = \infty$ and $\Delta := \min\{x \geq 0 : C^T(x) \leq C(\mathbf{0}, S')\}$.
Step 1: Solve the optimisation problem given by
$\min\{C(\boldsymbol{c}, S) : 0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S, S = \Delta\}$
Let $(\boldsymbol{c}_\Delta, \Delta)$ denote an optimal solution,
and let C_Δ denote its corresponding objective value.
If $C_\Delta < C^*$ then set $C^* := C_\Delta$ and $(\boldsymbol{c}^*, S^*) := (\boldsymbol{c}_\Delta, \Delta)$.
Step 2: If $C^T(\Delta+1) > C^T(\Delta)$ and $C^T(\Delta+1) \geq C^*$ then `stop`,
and take $(\boldsymbol{c}^*, S^*)$ as an optimal solution;
Otherwise, set $\Delta := \Delta + 1$ and go back to Step 1.

Algorithm 4.2: Determination of optimal parameters for cost optimisation.

Observe that one may determine $C(\mathbf{0}, S')$ using the results of Smith [12] for the $(S-1, S)$ policy with demand rate Λ_0 and lost sales cost $(\lambda_1\pi_1 + \cdots + \lambda_n\pi_n)/\Lambda_0$. Basically, the algorithm checks the minimal cost for all order-up-to levels, starting with the lower bound, until a stopping criterion is satisfied. An important part of the method is the determination of C_Δ, the minimum cost when the order-up-to level equals Δ. To solve this problem, we again suggest complete enumeration over all possible policies satisfying $0 \leq c_1 \leq \cdots \leq c_{n-1} \leq S = \Delta$.

We now consider a heuristic approach to solve the cost optimisation problem. The idea is that one starts by determining the optimal simple lot-for-lot policy and then increases the critical levels, starting with c_{n-1}, until no cost improvement is obtained. We will illustrate the heuristic by means of an example for $n = 4$. Suppose that the optimal simple policy is $S' = 10$. We increase c_3 by 1 unit and optimise the order-up-to level given that $\boldsymbol{c} = (0, 0, 1)$. Let the new optimal order-up-to level be 9. If there is a cost improvement, i.e. if $C(0, 0, 1, 9) < C(0, 0, 0, 10)$, we increase c_2 by 1 unit and re-optimise the order-up-to level given that $\boldsymbol{c} = (0, 1, 1)$. Assume that again the order-up-to level is 9 and that there is a cost improvement. Then we consider $\boldsymbol{c} = (1, 1, 1)$ and we assume that the new order-up-to level is 8 but there is no cost improvement, so we are left with $\boldsymbol{c} = (0, 1, 1)$. Now we start the "second round" and raise again c_3 by 1 unit, and so on. Suppose that we end up with $\boldsymbol{c} = (1, 2, 2)$ and $S = 8$. If increasing c_3 again with 1 unit does not lead to a cost improvement, then we stop and take $(1, 2, 2, 8)$ as the optimal policy. The main motivation for this approach is the fact that increasing the critical level of the least important demand class will have the smallest negative effect and the largest positive effect, because the service level for the least important demand class will decrease, whereas the other service levels will increase. The heuristic approach is formalised in Algorithm 4.3. By $\boldsymbol{e}_j$, $j = 1, \ldots, n-1$, we denote an $(n-1)$-dimensional vector containing all zeros except in the jth position, where it equals one. With respect to the optimisation problem $\min\{C(\boldsymbol{c}^{(k+1)}, x) : x \geq 0\}$, which has to be solved many times in Step 1, we conjecture that it suffices to consider $x = S^{(k)}$ or $x = S^{(k)} - 1$. In

Step 0: Set $\boldsymbol{c}^{(0)} := \boldsymbol{0}$, $S^{(0)} = S' := \arg\min\{C(\boldsymbol{0}, x) : x \geq 0\}$,
$C^{(0)} := C(\boldsymbol{0}, S^{(0)})$, $k := 0$ and $j := n-1$.
Step 1: Let $\boldsymbol{c}^{(k+1)} = \boldsymbol{c}^{(k)} + \boldsymbol{e}_j$, $S^{(k+1)} = \arg\min\{C(\boldsymbol{c}^{(k+1)}, x) : x \geq 0\}$
and $C^{(k+1)} := C(\boldsymbol{c}^{(k+1)}, S^{(k+1)})$.
Step 2: If $C^{(k+1)} < C^{(k)}$ and $j > 1$ then
set $j := j-1$, $k := k+1$ and goto Step 1,
else, if $C^{(k+1)} < C^{(k)}$ and $j = 1$ then
set $j := n-1$, $k := k+1$ and goto Step 1,
else, if $C^{(k+1)} \geq C^{(k)}$ and $j < n-1$ then
set $j := n-1$ and goto Step 1,
else, if $C^{(k+1)} \geq C^{(k)}$ and $j = n-1$ then
`stop`, and take $(\boldsymbol{c}^{(k)}, S^{(k)})$ as a solution.

Algorithm 4.3: Heuristic approach to solve cost optimisation problem.

words, the optimal order-up-to level will not change or decrease at most by 1 if one of the critical levels is raised by 1 unit. For large values of $S^{(k)}$ this conjecture may be used to speed up the algorithm. Finally, observe that the heuristic approach always leads to a policy which is at least as good as the simple lot-for-lot policy where all critical levels equal zero.

This concludes our discussion on the determination of the optimal critical-level, lot-for-lot policy. In the next section the results of some numerical examples are presented.

5 Computational results

To illustrate the benefits of using a critical-level, lot-for-lot policy instead of a standard lot-for-lot policy, the results of a number of 20 numerical examples are reported. In all examples we have taken $n = 4$, $h = 1$ and $L = 0.5$. We consider 10 cases with service level optimisation and 10 cases with cost optimisation. The data were arbitrarily chosen and are presented in Tables 1 and 2.

We have determined both the optimal critical-level, lot-for-lot policy and the optimal simple lot-for-lot policy, which is denoted by S'. The results are presented in Tables 3 and 4. The first column refers to Tables 1 and 2. The next 5 columns represent the optimal critical-level, lot-for-lot policy with associated cost. In columns 7 and 8 the optimal simple policy with associated cost is presented and the last column denotes the cost reduction obtained by using the critical-level, lot-for-lot policy instead of the simple lot-for-lot policy.

One can see that the total cost can be reduced significantly by applying the critical-level, lot-for-lot policy. For the examples with a service level constraint the cost reductions are quite large. An interesting feature can be observed from Tables 3 and 4. On the one hand,

case	λ_1	λ_2	λ_3	λ_4	β_1	β_2	β_3	β_4
1	0.5	0.5	0.5	0.5	0.99	0.95	0.75	0.50
2	5	0.5	0.5	0.5	0.99	0.95	0.75	0.50
3	0.5	5	0.5	0.5	0.99	0.95	0.75	0.50
4	0.5	0.5	5	0.5	0.99	0.95	0.75	0.50
5	0.5	0.5	0.5	5	0.99	0.95	0.75	0.50
6	0.5	0.5	0.5	0.5	0.99	0.95	0.90	0.75
7	5	0.5	0.5	0.5	0.99	0.95	0.90	0.75
8	0.5	5	0.5	0.5	0.99	0.95	0.90	0.75
9	0.5	0.5	5	0.5	0.99	0.95	0.90	0.75
10	0.5	0.5	0.5	5	0.99	0.95	0.90	0.75

Table 1: Data for service level optimisation.

case	λ_1	λ_2	λ_3	λ_4	π_1	π_2	π_3	π_4
11	0.5	0.5	0.5	0.5	10000	1000	100	10
12	5	0.5	0.5	0.5	10000	1000	100	10
13	0.5	5	0.5	0.5	10000	1000	100	10
14	0.5	0.5	5	0.5	10000	1000	100	10
15	0.5	0.5	0.5	5	10000	1000	100	10
16	0.5	0.5	0.5	0.5	500	100	50	10
17	5	0.5	0.5	0.5	500	100	50	10
18	0.5	5	0.5	0.5	500	100	50	10
19	0.5	0.5	5	0.5	500	100	50	10
20	0.5	0.5	0.5	5	500	100	50	10

Table 2: Data for cost optimisation.

for high values of λ_1 (cases 2, 7, 12 and 17), the optimal critical-level, lot-for-lot policy is significantly different from the optimal standard lot-for-lot policy, i.e. all critical levels are positive. However, the cost reductions are relatively small. On the other hand, for high values of λ_4 (cases 5, 10, 15 and 20), the optimal policy is characterised by low values for the critical levels, but the cost reduction is relatively large, in one case more than 50%.

The reason that the cost reductions for the cases with a service level constraint are large is the fact that the constraint for the highest priority class completely determines the order-up-to level in the standard lot-for-lot policy. We have also analysed the total cost if, for all classes, separate stocks were maintained. However, in all cases the cost of this policy was higher than the cost of the standard policy.

In Table 5 we present the lower and upper bounds for the 10 cases with service level constraints and the 10 cases without service level constraints. The second and the fourth column refer to the bounds $\Delta_{min} + 1$ and $\Delta_{max} - 1$ (see Algorithm 4.1) and the *optimal* columns denote

case	c_1	c_2	c_3	S	$C_h(\mathbf{c},S)$	S'	$C_h(\mathbf{0},S')$	reduction
1	0	1	1	4	3.04	5	4.00	24.00%
2	1	1	1	8	4.80	9	5.76	16.67%
3	0	1	2	8	4.81	9	5.76	16.49%
4	0	1	1	7	3.95	9	5.76	31.42%
5	0	0	2	5	2.81	9	5.76	51.22%
6	0	1	1	4	3.04	5	4.00	24.00%
7	1	1	1	8	4.80	9	5.76	16.67%
8	0	1	2	8	4.81	9	5.76	16.49%
9	0	1	1	7	3.95	9	5.76	31.42%
10	0	0	1	7	3.94	9	5.76	31.60%

Table 3: Results for service level optimisation.

case	c_1	c_2	c_3	S	$C(\mathbf{c},S)$	S'	$C(\mathbf{0},S')$	reduction
11	0	1	2	7	6.19	7	6.41	3.43%
12	1	3	5	13	10.62	14	11.08	4.15%
13	0	2	4	12	9.61	12	9.88	2.73%
14	0	1	3	11	8.77	12	9.43	7.00%
15	0	1	2	10	7.77	12	9.38	17.16%
16	0	0	1	5	4.84	5	5.02	3.59%
17	1	1	3	11	8.63	11	8.82	2.15%
18	0	0	2	10	7.77	10	7.85	1.02%
19	0	0	1	10	7.50	10	7.53	0.40%
20	0	0	1	9	6.76	10	7.28	7.14%

Table 4: Results for cost optimisation.

the optimal order-up-to levels. The sixth column represents the lower bound presented in Theorem 4.3 an the last column denotes the largest value of the order-up-to level for which the stopping criterion in Algorithm 4.2 was not satisfied.

It can be seen that the stopping criterion for the cost optimisation algorithm is very good. Only in case 14 was the optimal order-up-to level lower than the largest evaluated value. For the cases with service level optimisation we noticed that the upper bound was very tight. The lower bound was significantly smaller than the optimal value, but also significantly larger than zero.

Finally, we analysed the quality of the heuristic approach for the cost optimisation problem, presented in Algorithm 4.3. We randomly generated 5000 test problems for $n=4$ by taking $h \in [1,10]$, $\pi_1 \in [1000, 10000]$, $\pi_2 \in [500, 2000]$, $\pi_3 \in [100, 1000]$, $\pi_4 \in [10, 200]$, $\lambda_j \in [0.1, 1.6]$, $j = 1, \ldots, 4$, and $L \in [0.1, 1.6]$. We sorted π_j, $j = 1, \ldots, 4$, in descending order. In the heuristic approach we used the conjecture mentioned below Algorithm 4.3. The average computation time of the heuristic was 0.01 seconds on a 486DX/66 Personal Computer,

case	lower	optimal	upper	case	lower	optimal	upper
1	1	4	4	11	1	7	7
2	3	8	8	12	6	13	13
3	3	8	8	13	6	12	12
4	3	7	8	14	6	11	12
5	3	5	8	15	7	10	10
6	2	4	4	16	1	5	5
7	4	8	8	17	6	11	11
8	4	8	8	18	6	10	10
9	4	7	8	19	6	10	10
10	4	7	8	20	6	9	9

Table 5: Lower and upper bounds for 20 cases.

which is much less than the 1.05 seconds average computation time of the exact method. The maximum running time of the heuristic was 0.03 seconds vs. 12.11 seconds for the exact method. In 13 cases the heuristic did not find an optimal solution, with a maximum relative error of only 0.5%. Hence, we may conclude that, although it can not guarantee an optimal solution, the heuristic approach performs very well and saves a lot of computation time.

6 Conclusions

There are many examples in practice where demand for a product can be classified into categories of different priorities. If this is the case, an inventory manager would wish to implement an inventory policy which takes advantage of this knowledge. In this paper we have analysed a lot-for-lot or $(S-1,S)$ inventory model with n demand classes, where demand not satisfied from stock on hand is lost. We considered a critical-level policy to handle the different demand classes. This policy reserves some of the stock for higher priority classes and thus allows different service levels for different classes to be obtained. We have derived easy and exact expressions for the average total cost per unit time and for the service levels, for generally distributed replenishment lead times. Exact solution procedures were presented to obtain an optimal policy parameters, for models with or without service level constraints. For the problem without service level constraints we developed a good heuristic approach. The computational results show that the cost reduction obtained by applying the critical-level policy can be up to 50%. The running time of the exact solution method was on average about 1 second. The quality of the heuristic approach turned out to be very good. In almost all cases an optimal solution was found and the average computation time was only 0.01 seconds. Since the critical-level policy is easy to understand and to implement by practitioners and an optimal policy can be determined in little time using relatively easy expressions, we feel that this policy has real practical value.

Acknowledgement: The authors wish to thank Sven Axsäter for his suggestions leading to the heuristic approach presented in Section 4.2.

References

[1] D. Atkins and K.K. Katircioğlu. Managing inventory for multiple customers requiring different levels of service. Working Paper 94–MSC–015, University of British Columbia, Vancouver, B.C., 1995.

[2] M.A. Cohen, P.R. Kleindorfer, and H.L. Lee. Service constrained (s, S) inventory systems with priority demand classes and lost sales. *Management Science*, 34:482–499, 1988.

[3] M.A. Cohen, P.R. Kleindorfer, and H.L. Lee. Near–optimal service constrained stocking policies for spare parts. *Operations Research*, 37:104–117, 1989.

[4] R. Dekker, M.J. Kleijn, and P.J. De Rooij. A spare parts stocking system based on equipment criticality. *to appear in the International Journal of Production Economics*, 1996.

[5] R.V. Evans. Sales and restocking policies in a single item inventory system. *Management Science*, 14:463–472, 1968.

[6] W. Feller. *An Introduction to Probability Theory and Its Applications (3rd ed.), vol. 1.* Wiley, New York, 1968.

[7] B.V. Gnedenko and I.N. Kovalenko. *Introduction to Queueing Theory.* Israel Program for Scientific Translations, Jerusalem, 1968.

[8] A.Y. Ha. Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Management Science*, 43:1093–1103, 1997.

[9] A.Y. Ha. Stock-rationing policy for a make-to-stock production system with two priority classes and backordering. *Naval Research Logistics*, 44:457–472, 1997.

[10] A. Kaplan. Stock rationing. *Management Science*, 15:260–267, 1969.

[11] S. Nahmias and S. Demmy. Operating characteristics of an inventory system with rationing. *Management Science*, 27:1236–1245, 1981.

[12] S.A. Smith. Optimal inventories for an $(S-1, S)$ system with no backorders. *Management Science*, 23:522–528, 1977.

[13] R.H. Teunter and W.K. Klein Haneveld. Reserving spare parts for critical demand. Research Report, Graduate School/Research Institute System, Organisations and Management (SOM), University of Groningen, 1996.

[14] D.M. Topkis. Optimal ordering and rationing policies in a nonstationary dynamic inventory model with n demand classes. *Management Science*, 15:160–176, 1968.

[15] A.F. Veinott Jr. Optimal policy in a dynamic, single product, nonstationary inventory model with several demand classes. *Operations Research*, 13:761–778, 1965.