DEPARTMENT OF OPERATIONS RESEARCH
UNIVERSITY OF AARHUS

Publication no. 99/2

Simple and optimal rationing policies for an inventory model with several demand classes

Philip Melchiors

ISSN 1398-8964

Department of Mathematical Sciences
Telephone: +45 8942 3188

Building 530, Ny Munkegade
DK-8000 Aarhus C, Denmark

Simple and optimal rationing policies for an inventory model with several demand classes

Philip Melchiors
University of Aarhus, Department of Operations Research,
Ny Munkegade, Building 530, 8000 Aarhus C, Denmark.
E–mail : philip@imf.au.dk

June 1999

Abstract

In this paper we analyze an (s, Q) inventory model with unit Poisson demand, several demand classes and lost sales. When dealing with different demand classes the usual approach is to control the inventory by critical levels at which stock is reserved for demand of high priority. We present two different rationing policies, a simple critical level policy where the critical levels are constant, and an optimal policy where the critical levels are allowed to depend on the time since the actual outstanding order (if any) was issued. As the simple policy is much easier to implement in practice we investigate the cost difference of using the simple policy instead of the optimal policy in a numerical study. We also compare the two rationing policies with the best non-rationing policy.

Keywords: Inventory, rationing, Markov processes, lost sales, several demand classes.

1 Introduction

In this paper we consider an inventory system with several demand classes. Usually it is assumed that all customers are equally important, but in practice this is rarely the case. As an example consider a spare part inventory company in the airline industry. Keeping an airplane grounded can be very expensive, and the cost of not being able to satisfy demand from an airline can therefore be very high. These costs are usually specified in a contractual agreement. Different airlines may, however, value the cost of a grounded airplane differently, and the company may reject demand from some airlines in order to be able to satisfy airlines with higher priority. Another example can be found in a two–echelon inventory model, where lateral shipments between retailers are allowed. Each retailer will then face two kinds of demand; normal demand and demand from other retailers. Usually the retailer will consider normal demand more important than demand from other retailers. Thirdly, we mention another two–echelon inventory model, where the warehouse faces demand from several retailers that might be of different importance, due to different stockout or expediting costs (See Axsäter, Kleijn & de Kok [1]).

The considered inventory system is controlled by a rationing policy specified by critical levels. For each demand class except the one with highest priority, demand is rejected when the actual inventory level is at or below the critical level assigned to the class. In this way it is possible to save stock for possible future high–priority demand. A *simple policy* has constant critical levels, whereas a *time remembering policy* allows the critical levels to depend on the time elapsed since the actual outstanding order (if any) was issued.

The first contributions in the area of rationing policies are periodic review models. Veinott [14] analyses a model with several demand classes and zero lead time, and introduces the concept of critical levels. Topkis [13] proves the optimality of a time remembering policy for the same model in both the backorder and the lost sales case. He divides each period into a finite number of subintervals, and allows the critical levels to depend on the time till the next review. Recently, Frank, Zheng & Duenyas [4] have considered a periodic review model with two demand classes, one stochastic and one deterministic. The deterministic demand has to be satisfied but the stochastic demand can be rejected. Demand is observed by the beginning of each period, after which a replenishment order can be placed. It is assumed that orders arrive instantaneously so that the replenishment can be used to satisfy the observed demand. The purpose of rationing is therefore not to save stock for high priority demand, but rather to postpone an order placement one period. They show that the optimal rationing policy either satisfies all the stochastic demand or results in a remaining inventory which is an integer multiple of the deterministic demand per period.

The literature on rationing policies in a continuous review setting deals with two types of inventory policies, base–stock policies and (s, Q) policies. Ha [5] and Dekker, Hill and Kleijn [3] both consider lot–for–lot inventory systems with several demand classes. Dekker, Hill and Kleijn [3] find good simple critical level policies for the case of generally distributed lead times. Since they do not consider time remembering policies they cannot guarantee optimality. Ha [5] shows optimality of the simple critical level policy for the same model with exponentially distributed lead times. Simple critical level policies for an (s, Q) inventory model are first analyzed by Nahmias & Demmy [10], who find fill rates for a model with two demand classes and Poisson demand. This is done by conditioning on the so–called 'hitting time', the time where the inventory level 'hits' the critical level. They do not consider optimization. Moon and Kang [9] generalize their results to a model with compound Poisson demand, and find optimal rationing levels in the case of deterministic demand and several demand classes. The paper most related to the present is Melchiors, Dekker and Kleijn [8], who present a method for finding an optimal simple policy for the (s, Q) inventory model with two demand classes and lost sales. They find the exact cost of a simple critical level policy and in a numerical study they compare the critical level policy with the best non–rationing policy. The only paper considering time remembering policies in a continuous review setting is Teunter & Klein Haneveld [11], who present simple methods for finding good time remembering policies for an inventory model with two demand classes and backordering. Using marginal analysis they recursively determine values of the remaining lead time for which it is optimal to reserve $1, 2, \ldots$ units of stock for high priority demand.

In this paper we analyze the (s, Q) inventory model with several demand classes, lost sales and constant lead times. Using Markov decision theory we find an optimal time remembering policy. Our decisions are allowed to depend on the inventory level and, if the inventory level is below the reorder level, the time since the order was placed. We show that this policy is a critical level policy and that the critical levels are decreasing in time. The Markov decision model discretizes the lead time, and approximates the demand during the lead time by a Bernoulli process. Johansen & Thorstenson [7] use a similar approach to find optimal emergency order policies. Since a time remembering policy can be difficult to implement in practice, we also show how to find a good simple critical level policy, and in a numerical study we compare the two policies with each other and with the best non-rationing policy. Note that the inventory system analyzed is identical to that analyzed in Melchiors, Dekker and Kleijn [8]. However, we consider time remembering policies and our model allows more than two demand classes.

The paper is organized as follows. In Section 2 we introduce the necessary notation and specify the average cost of a rationing policy. Section 3 focuses on finding the optimal time remembering policy, and in Section 4 we present a heuristic for finding good simple policies. In Section 5 we investigate the properties of the policies by means of some numerical examples, and finally, some concluding remarks and directions for further research are given in Section 6.

2 The model

We now introduce the assumptions and notation used throughout the paper. We consider an inventory model with n demand classes. Class j has unit Poisson demand with rate λ_j. All demand not satisfied immediately is assumed to be lost (or expedited). The classes are distinguished by their stock out cost π_j, and we rank the classes such that $0 < \pi_n < \pi_{n-1} < \cdots < \pi_1$. Let $\Lambda_i = \sum_{j=1}^{i} \lambda_j$ be the demand rate from customers of the classes 1 to i. The ranking ensures that Λ_i is the demand rate from customers with a stockout cost of at least π_i. Let $\Pi(i)$ be the expected stockout cost incurred per unit time by not satisfying demand from customers of the classes $i+1$ to n, i.e. $\Pi(i) = \sum_{j=i+1}^{n} \lambda_j \pi_j$ for $0 \leq i < n$ and $\Pi(n) = 0$. For each replenishment order there is a fixed ordering cost K, and a constant lead time L. The unit holding cost per time unit is $h > 0$.

We analyze the rationing policy in the context of an (s, Q) policy where $Q > s$. This condition and the lost sales assumption ensure that at most one order is outstanding at any time. This means that in contrast to Nahmias & Demmy [10] (where the assumption of only one outstanding order is an approximation, due to the backorder environment) our results are exact so far. Assuming that s and Q are fixed, we shall formulate a semi–Markov decision model with finite state space $S_0 \cup S_1$. Let $\mathbb{N}$ denote the set of non–negative integers, and suppose that the constant lead time consists of N subintervals each of length L/N. The set of states when no order is outstanding is

$$S_0 = \{i \in \mathbb{N} |\ s < i \leq s + Q\}$$

and the set of states when one order is outstanding is

$$S_1 = \{(i, t) \in \mathbb{N} \times \mathbb{N} |\ 0 \leq i \leq s, \quad 0 \leq t \leq N\}.$$

Here i denotes the inventory level and t denotes the number of subintervals elapsed since the outstanding order was issued. There are two kinds of decision epochs: just after a demand has been satisfied when no order is outstanding and the beginning of each subinterval when one order is outstanding. In each decision epoch we choose an action. An action prescribes the set of classes we are willing to satisfy until a new decision is made. Let the action $a \in \{0, 1, 2, \ldots, n\}$ prescribe that we satisfy demand from classes 1 to a and that we reject demand from classes $a + 1$ to n. Let $\mathcal{A}$ be the set of actions that can be represented in this way. We will later show that the optimal action in each state belongs to $\mathcal{A}$. The rate of demand that we are willing to satisfy when choosing action a is Λ_a. Since we do not allow backlogging we set $a = 0$ in states where the inventory level is zero.

The number N of subintervals is chosen such that the probability of more than one demand in each subinterval is negligible. We can then approximate the real demand process during the lead time (which is Poisson) by a Bernoulli process (see e.g. Çinlar [2]). A Bernoulli process is a sequence of independent trials with outcome either one or zero. Each of the subintervals can be viewed as such a trial where the outcome is one if a demand that we are willing to satisfy occurs, and zero otherwise. The success probability in each subinterval, i.e. the probability of outcome one, depends on the chosen action and is $p_1(a) = L\Lambda_a/N$. Also let $p_0(a) = 1 - p_1(a)$ denote the probability of outcome zero. The approximation considerably simplifies the further calculations and we have verified by simulation that it has almost no influence on the obtained results as long as the subintervals are small enough.

The system evolves as follows: When there is no order outstanding we jump from state $i \in S_0$ to state $i-1 \in S_0$ if $i > s+1$, since all demand has unit size. When a demand is satisfied in state $s+1 \in S_0$, an order is placed and we jump to state $(s,0) \in S_1$. During the lead time in states $(i,t) \in S_1$ with $i > 0$ we can jump to two different states. With probability $p_0(a)$ we jump to state $(i,t+1)$ and with probability $p_1(a)$ we jump to state $(i-1,t+1)$. In states $(0,t) \in S_1$ we jump to state $(0,t+1)$ since we do not allow backlogging. When the replenishment arrives in state $(i,N) \in S_1$ we jump to state $i+Q \in S_0$.

The expected time between two decision epochs when choosing action a, and no order is outstanding, is

$$\tau_i(a) = 1/\Lambda_a \text{ for } i \in S_0.$$

During the lead time the expected time between two decision epochs is

$$\tau_{i,t} = L/N \text{ for } (i,t) \in S_1,$$

independently of the chosen action. Now let us consider the expected one–step cost. The expected one–step cost incurred in state i, when no order is outstanding and the action a is chosen, is

$$C_i(a) = \tau_i(a)[hi + \Pi(a)] \text{ for } i \in S_0.$$

During the lead time the one–step cost incurred in state (i,t) when choosing action a is

$$C_{i,t}(a) = \tau_{i,t}[hi + \Pi(a)] \text{ for } (i,t) \in S_1.$$

Note that we make a small error by assigning holding costs based on the stock in the beginning of each subinterval, but when N is large this error is negligible. Finally, we have to add the order cost K in each order cycle. The timing of the allocation of the order cost does not influence the analysis, so for convenience we will add it when the state $Q \in S_0$ occurs.

We will consider a policy described by the following parameters:

- s — Reorder point at which an order is placed
- Q — Order quantity, $Q > s$
- $k(i)$ — When no order is outstanding and the inventory level is i, satisfy demand from classes 1 to $k(i)$
- $l(i,t)$ — When one order is outstanding, the inventory level is i and the time since the replenishment order was placed is between tL/N and $(t+1)L/N$, satisfy demand from classes 1 to $l(i,t)$.

Observe that the assumption of at most one demand per subinterval has only to do with the analysis. If the policy is implemented in practice, it is able to deal with more than one demand per subinterval, and the assumption of at most one demand per subinterval is therefore not restrictive. The considered policy is not necessarily a critical level policy. To be a critical level policy it must satisfy

$$l(i+1,t) \geq l(i,t) \text{ for } i = 1,2,\ldots,s-1 \text{ and } t = 0,1,\ldots,N-1 \tag{1}$$

and

$$k(i+1) \geq k(i) \text{ for } i > s. \tag{2}$$

This means that there, for each class $j \geq 2$ and for all t, exists a unique critical level $c_j(t)$ =max$\{i|l(i,t) < j\}(= 0$ if $l(1,t) \geq j)$. This is the highest level of inventory where we will not serve class j. Similarly let $c_j(-)$ be the highest inventory level above s at which we will not satisfy demand class j. If $k(s+1) \geq j$ we will always satisfy demand from class j when there is no order outstanding and $c_j(-)$ is not defined. Policies with a critical level above the reorder point was introduced by Melchiors, Dekker and Kleijn [8], who also characterize when this type of policies is optimal. Observe that, if $l(i,t)$ is a constant function of t for all i, then the policy is a simple critical level policy.

We will now specify the long–run average cost per unit time (henceforth referred to as *cost* for simplicity) of using the considered policy. Note that the inventory process is regenerative with regeneration points when the state $Q \in S_0$ occurs, and define a cycle as the time between two consecutive regeneration points. We then have from the renewal–reward theorem (see e.g. Tijms [12]) that the cost of the policy is the expected cost of one cycle divided by the expected length of one cycle.

We compute the expected cost and length of a cycle by a backwards recursive procedure starting in the regeneration point. Let $Z(i)$ be the expected cost incurred until we reach the next regeneration point starting in state $i \in S_0$. Let $Y(i)$ be the expected time until we reach the next regeneration point starting in state $i \in S_0$. Note that $Z(i)$ and $Y(i)$ can be found by the recursive formulae

$$Z(i) = C_i(k(i)) + Z(i-1) \text{ for } i \in S_0 \tag{3}$$

and

$$Y(i) = \tau_i(k(i)) + Y(i-1) \text{ for } i \in S_0. \tag{4}$$

The recursion is initialized with $Z(Q) = K$ and $Y(Q) = 0$. Since the inventory level cannot be higher than $s+Q$, we compute $Z(i)$ and $Y(i)$ for $i = Q, Q+1, \ldots, s+Q$. We can now jump to the situation just before the order arrives. Let $z(i,t)$ be the expected cost incurred until we reach the regeneration point starting in state (i,t). Also let $y(i,t)$ be the expected time until we reach the next regeneration point starting in state (i,t).

Initialize with $z(i,N) = Z(i+Q)$ and $y(i,N) = Y(i+Q)$ for $0 \le i \le s$. Now

$$\begin{aligned} z(i,t) &= C_{i,t}(l(i,t)) + p_0(l(i,t))z(i,t+1) \\ &\quad + p_1(l(i,t))z(i-1,t+1) \text{ for } 0 < i \le s \text{ and } 0 \le t < N \\ z(0,t) &= \Pi(0) + z(0,t+1) \end{aligned}$$

and

$$\begin{aligned} y(i,t) &= \tau_{i,t} + p_0(l(i,t))y(i,t+1) \\ &\quad + p_1(l(i,t))y(i-1,t+1) \text{ for } 0 < i \le s \text{ and } 0 \le t < N \\ y(0,t) &= \tau_{i,t} + y(0,t+1) \end{aligned}$$

can be found by recursion for $t = N-1, N-2, \ldots, 0$ and $i = 0, 1, \ldots, s$. Finally, let $Z(s) = z(s,0)$ and $Y(s) = y(s,0)$ and compute $Z(i)$ and $Y(i)$ by (3) and (4) for $i = s+1, s+2, \ldots, Q$. The cost of the policy is

$$g = \frac{Z(Q)}{Y(Q)}.$$

3 The optimal policy

The optimization procedure in this section is based on the semi-Markov decision theory (see e.g. Tijms [12]). We will find the optimal policy within the class of policies discussed in Section 2. We assume that the order-size Q is fixed, and use a tailor–made policy iteration algorithm, described in the Appendix, to find optimal values of $k(i)$, $l(i,t)$ and s. The algorithm is designed such that the policy found satisfies the average cost optimality equations for the semi–Markov decision model, which means that the policy is optimal.

In the following theorem we characterize the structure of the optimal policy. The three statements are all based on the average cost optimality equations. The theorem is proved in the appendix.

Theorem. *The optimal rationing policy is described as follows:*

a *The optimal action in each state belongs to $\mathcal{A}$.*

b *The optimal policy is a critical level policy.*

c *The critical levels of the optimal policy are decreasing in the time t, i.e. the policy found by the policy iteration algorithm satisfies*

$$l(i,t+1) \ge l(i,t) \text{ for } i = 1,2,\ldots s, \text{ and } t = 0,1,\ldots,N-1.$$

The theorem considerably simplifies the search for the optimal policy. The policy iteration algorithm does not consider optimization of the order size Q. However, all our numerical tests have indicated that the minimum cost is quasi–convex in Q, and Q can therefore be found by neighbourhood search starting e.g. with the Economic Order Quantity computed by considering the deterministic version of the problem where the demand classes are aggregated to one. Our procedure for finding the optimal policy computes the optimal value of Q in this way. For each value of Q, the optimal values of $k(i)$, $l(i,t)$ and s are found by the policy iteration algorithm. Let R_{opt} denote the optimal policy. The procedure has been implemented in Pascal, and is very efficient.

4 The simple policy

The optimal policy can be difficult to implement in practice. We shall therefore in this section describe how to find good simple policies with constant critical levels that do not depend on the time t. Define $\boldsymbol{c} = (c_2, c_3, \ldots, c_n)$ where c_j denotes the critical level of demand class j. We denote the simple policy by $(\boldsymbol{c}, s, Q)$. This policy can obviously be evaluated by the method presented in Section 2, by letting

$$k(i) = \max\{j | c_j < i\} \text{ for } i \in S_0.$$

Similarly, let

$$l(i,t) = \max\{j | c_j < i\} \text{ for } (i,t) \in S_1.$$

Let $g(\boldsymbol{c}, s, Q)$ denote the cost of the simple policy $(\boldsymbol{c}, s, Q)$. We will not try to find an optimal simple policy but instead focus on a heuristic that performs very well.

We use neighbourhood search to determine the optimal value of Q. For each Q, we search for the optimal value of s by enumeration from $Q-1$ to 0. For given values of s and Q we find a good $\boldsymbol{c}$–vector by an algorithm similar to the one suggested by Dekker, Hill & Kleijn [3]. Let $\boldsymbol{e}_j$ be the vector consisting of zeroes at all entries except at the j'th entry where it equals one, and let $\boldsymbol{c}^k$ be the $\boldsymbol{c}$–vector considered in iteration k. If $s = Q-1$ then start with $\boldsymbol{c}^1 = (0, 0, \ldots, 0)$ otherwise let $\boldsymbol{c}^1$ be equal to the best $\boldsymbol{c}$-vector found for $(s+1, Q)$. Let $j = n$. Let $\boldsymbol{c}^2 = \boldsymbol{c}^1 + \boldsymbol{e}_j$. If $g(\boldsymbol{c}^2, s, Q) < g(\boldsymbol{c}^1, s, Q)$ let $j := j-1$ and continue like this until $g(\boldsymbol{c}^{k+1}, s, Q) > g(\boldsymbol{c}^k, s, Q)$ or $j = 2$. Now let $j = n$ and start over improving the so far best obtained vector, and continue until no further improvements can be made.

As in Melchiors, Dekker & Kleijn [8], we have observed that it is possible to end up in local minimas when searching for s for a given value of Q. Therefore we use enumeration. The backwards enumeration over s is chosen because we have observed that the best $\boldsymbol{c}$–vector increases as s decreases, which mean that we can use the best $\boldsymbol{c}$–vector for $(s+1, Q)$

as a start vector when searching for the best $\boldsymbol{c}$–vector for (s, Q). The order size Q is found by neighbourhood search starting with the Economic Order Quantity. As observed for the optimal policy found in Section 3, the cost of a simple policy has been quasi–convex with respect to Q in all our numerical tests.

The policies described in this and the previous section are applicable to inventories with n equal to the number m of different demand classes, but in practice it might be too confusing if m is large. Typically one then would try to aggregate the m customers into a small number, n, of demand classes. The problem of finding n optimal partitions of m demand classes is very complicated, but it should be possible to find sound partitions by aggregating similar demand classes according to their stockout costs.

5 Numerical Results

In this section we illustrate the properties of the policies introduced in the previous sections. We will first investigate one of the base cases described in Melchiors, Dekker and Kleijn [8], and then compare the simple and the optimal policy with each other and with the best non-rationing policy on a larger set of data.

Our results obviously depend on the choice of N. Using high values of N when computing the policies will lead to a more precise representation of the Poisson process, and the policies found will be better than those found with lower values of N. For the examples in this section, the cost of a policy R is found by evaluating the policy with $N = 10000$. The cost of a policy is denoted $\gamma(R)$. In a numerical study we have found that the difference in cost between evaluating using $N = 10000$ and $N = 100000$ are within 0.004%. We do not, however, use $N = 10000$ when we find the optimal and the simple policy. We have experienced that using $N = 500$ gives solutions where the costs are within 0.002 % of that of the policy found using $N = 10000$. The policies in this section are all found using $N = 500$.

Example 1

In this example, we consider an inventory system with two demand classes and the following characteristics: $L = 1$, $h = 1$, $K = 100$, $\lambda_1 = 1$, $\lambda_2 = 10$, $\pi_1 = 1000$, and $\pi_2 = 10$. The optimal policy has $s_{opt} = 13$ and $Q_{opt} = 48$, and the best simple policy is $(\boldsymbol{c}, s_s, Q_s) = (2, 14, 48)$. In Figure 1, the critical level $c_2(t)$ of the optimal policy is depicted together with the critical level of the best simple policy. The critical level of the optimal policy is decreasing in time, illustrating part b of the theorem. The figure illustrates the advantage of the optimal policy. In the beginning of the lead time we will reject demand class 2 at a higher level, and by the end of the lead time we will not reject demand class 2 at all. That is, the optimal policy dominates the simple in two situations: when the

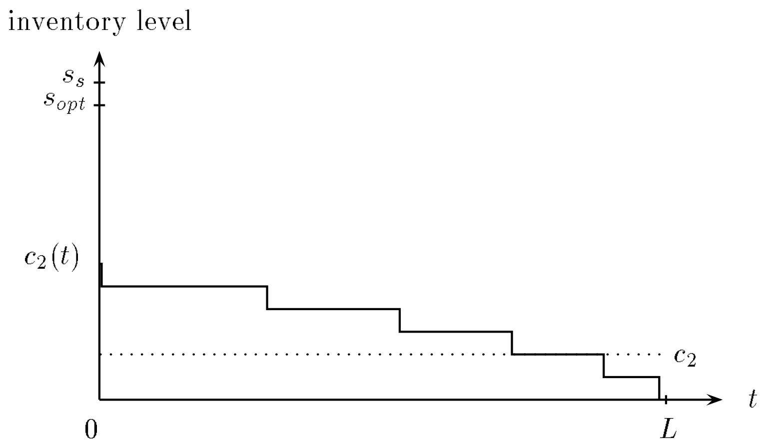

Figure 1: The critical levels $c_2(t)$ of the optimal policy and c_2 of the simple policy for Example 1 (n=2).

demand in the beginning of the lead time is high, and when demand from class 2 appears by the end of the lead time. In most cases the inventory level will not reach the critical level and the only difference between the simple and the optimal policy will in these cases be the reorder level. The cost of the two policies are $\gamma(R_{opt}) = 51.84$ and $\gamma(R_s) = 52.49$, respectively, a difference of 1.25%.

In order to illustrate the rationing policy with several demand classes, we will change the example slightly, by diving demand class 2 up into 3 different demand classes, with $\lambda_2 = 1$, $\lambda_3 = 2$, $\lambda_4 = 7$ and $\pi_2 = 40$, $\pi_3 = 12.5$, $\pi_4 = 5$. The optimal policy has $s_{opt} = 11$ and $Q_{opt} = 48$ with $\gamma(R_{opt}) = 50.72$, and the simple policy is $(\boldsymbol{c}, s_s, Q_s) = (1, 2, 3, 13, 48)$ with $\gamma(R_s) = 51.79$, a difference of 2.1%. The critical levels are shown in Figure 2. The structure is basicly the same as in the original example. The way the two examples are constructed, allows us to compare the cost of the original example with the cost of the modified example to see what difference it makes when three very similar demand classes (2,3 and 4 in the modified example) are joined into one (class 2 in the original example), as mentioned in the discussion by the end of Section 4. For the optimal policies the increase in cost incurred by aggregating the three demand classes is 2.2 %, and for the simple policy the increase in cost is 1.3 % compared with the cost of the unaggregated problems.

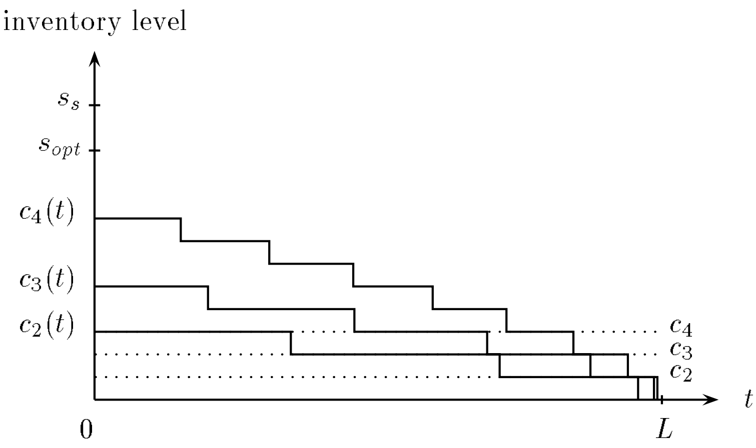

Figure 2: The critical levels of the optimal and the simple policies for Example 1 (n=4).

Cost comparisons

Let R_{non} be the best non-rationing policy. To find R_{non}, we aggregate all demand classes into one and let

$$\lambda_1^{non} = \sum_i \lambda_i \text{ and } \pi_1^{non} = \sum_i \pi_i \lambda_i / \Lambda_n.$$

We can then find the best simple policy for the one demand class problem. Let $\gamma(R_{non})$ be the cost of the best non-rationing policy. Define the (relative) cost reduction of using the simple rationing policy instead of the non-rationing policy as

$$CR_s = \frac{\gamma(R_{non}) - \gamma(R_s)}{\gamma(R_{non})}$$

and the (relative) cost reduction of using the optimal rationing policy instead of the non-rationing policy as

$$CR_{opt} = \frac{\gamma(R_{non}) - \gamma(R_{opt})}{\gamma(R_{non})}.$$

Finally, define the (relative) cost difference of using the optimal rationing policy instead of the simple rationing policy as

$$CD = \frac{\gamma(R_s) - \gamma(R_{opt})}{\gamma(R_{opt})}.$$

We have computed the best non-rationing policy, the best simple policy and the optimal policy for 27 examples with 4 demand classes. In all examples we have chosen $h = 1$ as the monetary unit, and $L = 1$ as the time unit. In the examples we have varied the total demand rate Λ_4, and for each value of Λ_4 investigated three different allocations of the total demand. Let λ_j/Λ_4 denote class j's share of total demand, and let $\boldsymbol{\lambda}/\Lambda_4$ be a demand allocation vector. Finally we have considered three different values of the stockout cost vector, $\boldsymbol{\pi}$. We have not investigated changes in the order cost K. The results are reported in Table 1. CR_s, CR_{opt} and CD all increase as the difference between the stockout costs increase, with the highest values for $\boldsymbol{\pi} = (10000, 1000, 100, 10)$ as expected. Similarly, CR_s, CR_{opt} and CD all increase as we allocate more demand to the classes with lowest priority. The benefit of the rationing policy is that it rejects low priority demand during the lead time, thereby reducing the need for safety stock. As we allocate more demand to the low-priority classes, the cost reductions CD_s and CR_{opt} increase. A possible explanation for the influence on CD is that the situations where the optimal policy dominates the simple policy depend on demand from low priority demand. In order not to satisfy too much low priority demand during the first part of the lead time, there has to be low priority demand. And similarly, in order to satisfy low priority demand when there is sufficient stock by the end of the lead time, we need low priority demand. We note, however, that the observation (at least with respect to CD) does not hold rigorously. We have made small numerical tests where the demand allocation is changed marginally, where the observed effect on CD is quite unpredictable.

Now, let us investigate the effect of increasing the total demand rate Λ_4. There seems to be no systematic effect on CR_s. However, both CR_{opt} and CD increase (except in the case $\boldsymbol{\pi} = \boldsymbol{\pi}(1)$, $\boldsymbol{\lambda} = \boldsymbol{\lambda}(3)$). Since demand is Poisson, increasing the demand rate will increase the variance of the demand as well and we will thereby more often end up in the situations where the optimal policy dominates the simple one. This explains why it is only the optimal policy (and hence CR_{opt} and CD) that benefits from the increase in total demand rate. We note that in 26 of the 27 examples the reorder point of the optimal policy is lower than that of the non-rationing policy. The reorder point of the simple policy is lower than that of the non-rationing policy in 21 examples.

The average values of CR_s, CR_{opt} and C are 2.02%, 3.39% and 1.43%, respectively. Whether or not it is worthwhile to use the optimal policy instead of the simple, depends on the given problem. For the investigated examples it appears that by using the simple policy the cost reduction compared with the non-rationing policy is about half the size of the cost reduction obtained by using the optimal policy.

6 Conclusions

In this paper we have shown how to find simple and optimal rationing policies for an (s, Q) inventory model with lost sales and several demand classes. The optimal policy is a critical level policy with critical levels that are decreasing in the time t elapsed since the outstanding order (if any) was issued. The simple policy is easy to implement and in many cases the cost difference of using the simple policy instead of the optimal is very small. However, in cases with high demand rate, and in particular if the stockout cost of the most important class is high, the difference between the simple and the optimal policy can be significant (in the investigated examples we recorded a difference of 2.71 %).

We have only considered Poisson demand. The extension to compound Poisson is an interesting challenge for future research. This will increase the variability of the demand and we therefore expect that the performance of the optimal policy will be considerably better than that of the simple policy. The development of a model with non–deterministic lead times is also very relevant in the context of inventory rationing.

Appendix

The tailor–made policy iteration algorithm

Suppose that the order size Q is fixed. For a policy with cost g, the relative values are defined as

$$\begin{aligned} w(i) &= Z(i) - gY(i) \text{ for } i \in S_0 \\ v(i,t) &= z(i,t) - gy(i,t) \text{ for } (i,t) \in S_1 \end{aligned}$$

The relative value of each state can be interpreted as the difference in expected long–run total cost of starting in this state rather than in the regeneration state $Q \in S_0$. The semi–Markov version of Theorem 3.2.1 in Tijms [12] tells that an optimal policy, i.e. one that minimizes the cost of running the system, can be found by solving the following equations with respect to v, w and g.

$$w(i) = \min_a \left\{ C_i(a) - g\tau_i(a) + \sum_{j \in S_0} P_{(i),(j)}(a) w(j) \right\} \text{ for } i \in S_0 \tag{5}$$

$$v(i,t) = \min_a \left\{ C_{i,t}(a) - g\tau_{i,t} + \sum_{(j,r) \in S_1} P_{(i,t),(j,r)}(a) v(j,r) \right\} \text{ for } (i,t) \in S_1 \tag{6}$$

Here $P_{(\cdot),(\cdot)}$ are the transition probabilities. When a solution to these equations is found, the optimal policy is specified by the actions minimizing the right hand side of the equations. The cost of this policy is g. We solve the equations by a policy iteration algorithm. Initially, g is computed as the cost of some easily evaluated policy with cost $g < \Pi(0)$. In each iteration g is given and we solve the equations with respect to v and w. Let g' be the cost of the new policy specified by the actions minimizing (5) and (6). If $g' = g$, we have solved (5) and (6) and thereby found the optimal policy. If $g' \geq \Pi(0)$, it is optimal to reject all demand and hold no inventory. Otherwise we set $g := g'$ and perform another iteration based on the new value of g.

We will now describe how to solve the equations in more detail. Let g be the cost of the previously found policy. Due to the structure of the Markov chain, we can write (5) as

$$w(i) = \min_a \{j(i,a) = C_i(a) - g/\Lambda_a + w(i-1)\ \} \text{ for } i > Q.$$

Initialize the recursion scheme by letting $w(Q) = K$ and compute the values of the states $i = Q+1, Q+2, \ldots, 2Q-1$ recursively (Recall that the reorder point can be at most $Q-1$). It is easy to show that $j(i,a)$ convex in a and that its minimum is found as the highest value of a that satisfies

$$hi + \Pi(a) + \Lambda_a \pi_a \geq g. \tag{7}$$

From part b of the theorem we get that $k(i) \geq k(i-1)$, and we can therefore use the following algorithm to find $k(i)$. If $k(i-1) = n$ or if $a = k(i-1)+1$ does not satisfy (7), then set $k(i) = k(i-1)$. Otherwise increase a by one until (7) is not satisfied and set $k(i) = a-1$.

Now consider the situation just before the order arrives. Initialize with $v(i,N) = w(i+Q)$ for $i = 0,1,\ldots,Q-1$. For $i = 0$ and $t = N-1, N-2, \ldots, 0$, the values are easily found since we can only choose $a = 0$,

$$v(0,t) = \Pi(0) - g\frac{L}{N} + v(0,t+1).$$

For $i = 1,2,\ldots,Q-1$ and $t = N-1, N-2, \ldots, 0$, the values are given by

$$v(i,t) = \min_a \left\{ f_t(i,a) = C_{i,t}(a) - g\frac{L}{N} + p_0(a)v(i,t+1) + p_1(a)v(i-1,t+1) \right\}.$$

It is easy to show that $f_t(i,a)$ is convex in a. Moreover, the action a that minimizes $f_t(i,a)$ is the highest value of a that satisfies

$$v(i-1,t+1) - v(i,t+1) \leq \pi_a. \tag{8}$$

To find this a, we use part b and c of the theorem. Let $\tilde{a} = \max\{l(i-1,t), l(i,t+1)\}$. If $\tilde{a} = n$ or if $a = \tilde{a}+1$ does not satisfy (8), then $l(i,t) = \tilde{a}$. Otherwise increase a by one until (8) is not satisfied and set $l(i,t) = a-1$.

All we need now is to compute the values $w(i)$ for $i \leq Q$. At this point we have to choose the reorder point s. The average cost optimality equation with respect to ordering is

$$w(i) = \min \left\{ v(i,0), \min_a \left\{ C_i(a) - g(R)\tau_i(a) + \sum_{j \in S_0} P_{(i),(j)}(a) w(j) \right\} \right\} \text{ for } i \in S_0.$$

Since we have to place an order when $i = 0$, set $w(0) = v(0,0)$.
Now if

$$v(i,0) < \min_a \{j(i,a)\} \tag{9}$$

we will place an order in state $i \in S_0$ and set $w(i) = v(i,0)$. Otherwise we set

$$w(i) = \min_a \{j(i,a)\}$$

This minimization is identical to that for values $w(i)$ with $i > Q$. Compute in this way the values $w(i)$ for $i = 1, 2, \ldots, Q$. We have not been able to prove that if (9) is not satisfied for i, then it will not be satisfied for $i+1$ either. Therefore, to ensure global optimality we investigate all $i < Q$. The reorder point s is found as the highest $i \in S_0$ that satisfies (9).

We have now described how to find the decisions that lead to the minimum value of all $w(i)$ and $v(i,t)$. For these decisions, compute $Y(i)$ and $y(i,t)$ as described in Section 2. We can then find the cost of the new improved policy $g' = g + w(Q)/Y(Q)$. If $w(Q)/Y(Q) = 0$ we have found a solution $(g, \{w(i)\}_{i \in S_0}, \{v(i,t)\}_{(i,t) \in S_1})$ to the average optimal cost equations and the algorithm terminates with the optimal policy specified by the reorder point s and $\{l(i,t)\}_{(i,t) \in S_1}$ and$\{k(i)\}_{i \in S_0}$. Otherwise we repeat the iteration with g equal to g'.
The algorithm converges in a finite and small number (typically 4-6) of iterations.

Proof of part a) of the theorem

Let $B \notin \mathcal{A}$ be a set of classes. For $n < 3$ the proof is trivial. We therefore assume $n \geq 3$. By definition there must exist $a, b, c \in \mathbb{R}$ with $a < b < c$ such that $a, c \in B$ and $b \notin B$. We will prove that the action B is dominated by either $B_a = B \setminus \{c\}$ or $B_{abc} = B \cup \{b\}$. For a set of classes A,

$$f_t(i,A) = \frac{L}{N}(hi - g + \sum_{j \in A} \lambda_j \pi_j + (\sum_{j \in A} \lambda_j) v(i-1, t+1) + (N/L - \sum_{j \in A} \lambda_j) v(i, t+1)).$$

Recall that the optimal action in state (i,t) is the set of classes that minimizes $f_t(i,A)$. Thus for B to dominate B_a and B_{abc} in state $(i,t) \in S_1$

$$f_t(i, B_{abc}) - f_t(i,B) = \frac{\lambda_b L}{N} [v(i-1,t+1) - v(i,t+1) - \pi_b]$$

and

$$f_t(i, B_a) - f_t(i,B) = \frac{\lambda_c L}{N} [v(i,t+1) - v(i-1,t+1) + \pi_c]$$

must both be positive. This cannot happen since the classes are ordered such that $\pi_b > \pi_c$, and the action B is therefore either dominated by B_a or B_{abc}. By a similar argument we can prove that B is dominated by B_a or B_{abc} in states $i \in S_0$ as well. Now repeat the procedure on the dominating action, until the action belongs to $\mathcal{A}$.

Proof of part b) of the theorem

First we will prove that the optimal policy satisfies (1). By (8) this is the case if for all t

$$v(i+1,t) - v(i,t) \geq v(i,t) - v(i-1,t) \text{ for } i = 1,2,\ldots,Q-2. \tag{10}$$

Note that this is the condition for convexity in i. We will prove (10) by induction on t. Recall that $k(i)$ =arg $\min_a j(i,a)$. When $t = N$ we have

$$\begin{aligned}
&v(i+1,N) - 2v(i,N) + v(i-1,N) \\
&\quad = \; w(i+Q+1) - 2w(i+Q) + w(i+Q-1) \\
&\quad = \; j(i+Q+1; k(i+Q+1) - w(i+Q) - j(i+Q; k(i+Q) + w(i+Q-1) \\
&\quad \geq \; j(i+Q+1; k(i+Q+1) - w(i+Q) - j(i+Q; k(i+Q+1) + w(i+Q-1) \\
&\quad = \; c(i+Q+1; k(i+Q+1)) - g/\Lambda_{k(i+Q+1)} - (c(i+Q; k(i+Q+1)) + g/\Lambda_{k(i+Q+1)}) \\
&\quad \geq \; 0
\end{aligned}$$

Now suppose inductively that (10) is true for $t = N, N-1, \ldots, r+1$.
Recall that $l(i,t)$ =arg $\min_a\{f_t(i,a)\}$. Now

$$\begin{aligned}
&v(i+1,r) - 2v(i,r) + v(i-1,r) \\
&\quad = \; f_r(i+1, l(i+1,r)) - f_r(i, l(i,r)) - f_r(i, l(i,r)) + f_r(i-1, l(i-1,r)) \\
&\quad \geq \; f_r(i+1, l(i+1,r)) - f_r(i, l(i+1,r)) - f_r(i, l(i-1,r)) + f_r(i-1, l(i-1,r)).
\end{aligned}$$

Since the holding and the penalty costs cancel out together with gL/N, this equals

$$\begin{aligned}
&\quad p_0(l(i+1,r))[v(i+1,r+1) - v(i,r+1)] + (1 - p_0(l(i+1,r)))[v(i,r+1) - v(i-1,r+1)] \\
&\quad +(1 - p_1(l(i-1,r)))[v(i-1,r+1) - v(i,r+1)] \\
&\quad +p_1(l(i-1,r))[v(i-2,r+1) - v(i-1,r+1)] \\
&= \; p_0(l(i+1,r))[v(i+1,r+1) - v(i,r+1) - v(i,r+1) + v(i-1,r+1)] \\
&\quad +p_1(l(i-1,r))[v(i,r+1) - v(i-1,r+1) - v(i-1,r+1) + v(i-2,r+1)] \\
&\quad +v(i,r+1) - v(i-1,r+1) + v(i-1,r+1) - v(i,r+1) \\
&\geq \; 0.
\end{aligned}$$

The last inequality follows from the induction hypothesis, completing the induction.
To conclude that the optimal policy is a critical level policy we only need to prove (2), which follows directly from (7).

Proof of part c) of the theorem

We will now prove that $l(i,t+1) \geq l(i,t)$ for all t by induction on t. By (8) this is the equivalent to

$$v(i+1,t+1) - v(i,t+1) \geq v(i+1,t) - v(i,t) \text{ for } i = 1,2,\ldots,Q-2 \tag{11}$$

for all t. We note that (11) is equivalent to the definition of a two-dimensional supermodalar function (see e.g. Heyman & Sobel [6]). First we need to prove

$$v(i+1,N) - v(i,N) - v(i+1,N-1) + v(i,N-1) \geq 0.$$

It is easy to show that $l(i,N-1) = k(i+Q)$.
Now insert

$$w(i+1) - w(i) = \frac{h(i+1) + \Pi(k(i+1)) - g}{\Lambda_{k(i+1)}}$$

and $p_1(a) = \Lambda_a L/N$ and we find

$$\begin{aligned}
&v(i+1,N) - v(i,N) - v(i+1,N-1) + v(i,N-1) \\
&= w(i+1+Q) - w(i+Q) - f_{N-1}(i+1, l(i+1,N-1)) + f_{N-1}(i, l(i,N-1)) \\
&= w(i+1+Q) - w(i+Q) - \frac{L}{N}[h(i+1) + \Pi(k(i+1+Q)) - g] \\
&\quad -p_1(k(i+1+Q))[w(i+Q) - w(i+1+Q)] - w(i+1+Q) \\
&\quad +\frac{L}{N}[h(i) + \Pi(k(i+Q)) - g] + p_1(k(i+Q))[w(i-1+Q) - w(i+Q)] - w(i+Q) \\
&= -\frac{L}{N}[h(i+1) + \Pi(k(i+1+Q)) - g - \Lambda_{k(i+1+Q)}[w(i+1+Q) - w(i+Q)]] \\
&\quad +\frac{L}{N}[h(i) + \Pi(k(i+Q)) - g - \Lambda_{k(i+Q)}[w(i+Q) - w(i-1+Q)]] \\
&= -\frac{L}{N}[h(i+1) + \Pi(k(i+1+Q)) - g - h(i+1+Q) - \Pi(k(i+1+Q)) + g] \\
&\quad +\frac{L}{N}[h(i) + \Pi(k(i+Q)) - g - h(i+Q) - \Pi(k(i+Q)) + g] \\
&= 0.
\end{aligned}$$

Suppose inductively that (11) is true for $t = N, N-1, \ldots, r+1$. Now

$$\begin{aligned}
&v(i+1,r) - v(i,r) - v(i+1,r-1) + v(i,r-1) \\
&= f_r(i+1, l(i+1,r)) - f_r(i, l(i,r)) - f_{r-1}(i+1, l(i+1,r-1)) + f_{r-1}(i, l(i,r-1)) \\
&\geq f_r(i+1, l(i+1,r)) - f_{r-1}(i+1, l(i+1,r)) - f_r(i, l(i,r-1)) + f_{r-1}(i, l(i,r-1)) \\
&= p_0(l(i+1,r))[v(i+1,r+1) - v(i,r+1) - v(i+1,r) + v(i,r)] + v(i,r+1) - v(i,r) \\
&\quad +p_1(l(i,r-1))[-v(i-1,r+1) + v(i,r+1) + v(i-1,r) - v(i,r)] - v(i,r+1) - v(i,r) \\
&\geq 0
\end{aligned}$$

by (11) and the induction is complete.

References

[1] S Axsäter, M.J. Kleijn, and A.G. De Kok. Stock rationing in a continuous review two echelon inventory model. Technical Report 9827/A, Econometric Institute, Erasmus University Rotterdam, The Netherlands, 1998.

[2] E. Çinlar. *Introduction to Stochastic Processes.* Prentice Hall, Englewood Cliffs, NJ, 1975.

[3] R. Dekker, R.M. Hill, and M.J. Kleijn. On the $(S-1, S)$ lost sales inventory model with priority demand classes. Technical Report 9743/A, Econometric Institute, Erasmus University Rotterdam, The Netherlands, 1997.

[4] K.C. Frank, R.Q. Zhang, and I. Duenyas. Optimal policies for inventory systems with with priority demand classes. Technical Report 99-01, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan, 1999.

[5] 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.

[6] D.P. Heyman and M.J. Sobel. *Stochastic models in operations research, Vol. II.* McGraw Hill inc., New York, 1984.

[7] S.G. Johansen and A. Thorstenson. An inventory model with Poisson demands and emergency orders. *Int. J. Production Economics*, 56–57:275–289, 1998.

[8] P. Melchiors, R. Dekker, and M.J. Kleijn. Inventory rationing in a (s, Q) inventory model with two demand classes and lost sales. *Department of Operations Research, University of Aarhus*, 1998.

[9] I. Moon and S. Kang. Rationing policies for some inventory systems. *Journal of the Operational Research Society*, 49:509–518, 1998.

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

[11] 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, 1997.

[12] H.C. Tijms. *Stochastic Models: An Algorithmic Approach.* Wiley, New York, 1994.

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

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

Problem Parameters			R_{non}	R_s	R_{opt}			
	$\boldsymbol{\pi}$	$\boldsymbol{\lambda}/\Lambda_4$	(s,Q)	$(\boldsymbol{c},s,Q)$	(s,Q)	CR_s	CR_{opt}	CD
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	8,33	(0,0,1,3),8,32	7,33	0.36%	0.75%	0.39%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	7,33	(0,0,0,2),7,33	7,33	0.65%	1.32%	0.68%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	7,33	(0,0,0,1),6,33	6,33	0.73%	1.57%	0.85%
	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	11,32	(0,0,2,5),10,33	10,33	0.94%	1.74%	0.81%
$\Lambda_4 = 5$	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	10,33	(0,0,1,5),9,33	9,33	1.72%	2.94%	1.26%
	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	10,32	(0,0,1,3),8,33	7,33	3.06%	4.38%	1.38%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	13,32	(0,1,3,7),12,32	12,32	1.92%	2.79%	0.89%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	12,33	(0,1,2,5),11,32	10,33	3.28%	4.90%	1.70%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	12,32	(0,0,2,4),9,33	8,33	5.14%	6.98%	1.98%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	15,46	(0,0,1,3),15,46	14,46	0.36%	1.18%	0.82%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	14,46	(0,0,1,2),14,47	13,46	0.57%	1.55%	0.99%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	13,46	(0,0,0,2),12,47	12,46	0.63%	1.72%	1.11%
	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	18,47	(0,0,2,6),18,46	17,46	1.17%	2.19%	1.05%
$\Lambda_4 = 10$	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	18,46	(0,0,1,5),17,46	16,46	1.77%	3.31%	1.59%
	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	17,46	(0,0,1,4),15,46	14,46	2.97%	4.72%	1.83%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	21,46	(0,1,4,8),20,46	19,46	2.00%	3.51%	1.57%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	20,46	(0,1,3,7),18,46	17,46	3.55%	5.54%	2.11%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	20,46	(0,0,2,5),16,46	15,46	5.23%	7.49%	2.44%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	21,57	(0,0,2,4),21,57	20,57	0.50%	1.33%	0.84%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	21,56	(0,0,1,3),20,56	19,57	0.76%	1.77%	1.03%
	$\boldsymbol{\pi}(1)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	19,57	(0,0,0,2),19,56	18,57	0.80%	1.87%	1.09%
	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	26,56	(0,0,2,7),25,56	24,56	1.10%	2.35%	1.28%
$\Lambda_4 = 15$	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	25,56	(0,0,2,6),23,56	22,56	1.85%	3.66%	1.88%
	$\boldsymbol{\pi}(2)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	24,56	(0,0,1,4),21,57	20,56	2.95%	4.97%	2.12%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(1)$	29,56	(0,2,5,10),27,56	25,57	2.05%	3.66%	1.67%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(2)$	28,56	(0,1,3,8),25,56	23,56	3.47%	5.80%	2.48%
	$\boldsymbol{\pi}(3)$	$\boldsymbol{\lambda}/\Lambda_4(3)$	27,56	(0,1,2,5),23,56	20,57	5.11%	7.62%	2.71%

Table 1: R_{non}, R_s, R_opt, CR_s, CR_{opt} and CD for 27 examples. For all examples $K = 100$, $h = 1$ and $L = 1$. Stockout costs are $\boldsymbol{\pi}(1) = (100, 50, 25, 10)$, $\boldsymbol{\pi}(2) = (1000, 500, 100, 10)$, $\boldsymbol{\pi}(3) = (10000, 1000, 100, 10)$. Demand allocations are $\boldsymbol{\lambda}/\Lambda_4(1) = (1/2, 1/4, 1/8, 1/8)$, $\boldsymbol{\lambda}/\Lambda_4(2) = (1/4, 1/4, 1/4, 1/4)$, $\boldsymbol{\lambda}/\Lambda_4(1) = (1/8, 1/8, 1/4, 1/2)$.