PhD Thesis 2001
Department of Operations Research
University of Aarhus

Computation of policies for inventory systems

Philip Melchiors

Preface

This thesis is the outcome of my 4 years of PhD studies at the Department of Operations Research at the University of Aarhus. The thesis is built on seven scientific papers which constitute the core of the thesis. The first part of the thesis is a general introduction to the field of inventory control and a survey of the seven papers. The survey provides the necessary background to understand the problems analysed, the main ideas behind the methods used, and the main conclusions of the thesis. The second part is the seven papers.

Acknowledgment

Several people have contributed to this thesis. First of all, I thank my supervisor Søren Glud Johansen for outstanding support and encouragement through all 4 years. Most of the ideas presented in the thesis were conceived during our long discussions. For this I am deeply indebted.

In the spring of 1998, I visited Professor Rommert Dekker at the Erasmus University Rotterdam, Holland. During the four valuable months I was introduced to the inventory rationing problem which ended up being an important component of this thesis. Later, in the fall of 1999 I visited Professor Paul Zipkin at Duke University, North Carolina, who showed me American hospitality at its best. I am grateful to both professors for the opportunity to visit them and for all our discussions.

The last four years would not have been the same without Jan Skriver - thanks for super friendship, countless coffee-breaks and for sharing the process of writing a thesis. Moreover, I thank Randi Moesgaard for silly jokes and for carefully reading the entire thesis, Marcel Kleijn for super friendship and Budapest 98, Kim Allan Andersen for "Nå, her knokles nok", Annemarie Melchiors for linguistic support, Oded Königsberg and Mark Ferguson for taking care of me at Duke, European post-docs and PhD students for good company at several conferences and workshops - especially Connie Gudum and the mini-Svens: Johan Marklund, Jonas Anderson and Peter Berling. Apart from my supervisor, Roger Hill and 'Vish' Viswanathan have offered constructive criticism on parts of the thesis, which they are greatly acknowledged for.

Finally, I thank my family, Wenche, Jacob, Maria and Nina, and, in particular, my wife Sys, for their love and support.

Philip Melchiors
Århus, July 2001.

Contents

1 Introduction

Supply Chain Management deals with management of information and material flows between manufacturers, distributors and customers. Efficient management of supply chains requires close cooperation between the agents in the supply chain, advanced information technology to provide instant and exact information of the flow of material, and finally, computerized decision support systems that, based on the information available, attempt to optimize the supply chain performance. Increased competition have led supply chain agents to engage in closer relationships with suppliers and vendors under banners like Customer Relations Management and Efficient Consumer Response. By sharing information of demand and supply, the efficiency of the entire supply chain can be improved to the benefit of all its agents. During the nineties there has been a tremendous development of information technology, and today, most larger companies have an Enterprise Resource Planning (ERP) system which coordinates and provides information on activities at all levels of the supply chain. The development of Electronic Data Interchange (EDI) moreover facilitates efficient seamless exchange of information between agents in the supply chain. The increased availability of information allows more advanced decision support systems. In fact, in order to fully utilize the benefit of the information available, the decision support systems need to be advanced. Advanced Planning and Scheduling systems are examples of decision support systems that can be built on top of ERP systems.

The decision support system is based on a representation of the real system, a model. The model represents the essential properties of the system, how demands occur, ordering opportunities, transportation times, cost of different actions, etc. Mathematical optimization is then applied to the model, in order to find good or even optimal model decisions. If the model captures the essential characteristics of the system, good model decisions will also be good system decisions.

How well a real system can be modelled is determined by several factors. First of all we need data information. If we do not have a forecast for the demand for a product, how can we determine good levels of safety-stock. The higher the level of complexity, the more computer power is needed to perform the numerical analysis. This used to be a significant restriction, but today we are much better off, although computation power still imposes certain limitations. The final limitation is the mathematical optimization. It is of no use to have a good model representation, if we are not able to optimize it. Methods for optimization of different supply chain models have been developed throughout the twentieth century. The first models were simple models, but based on these, techniques for solving more advanced models have been developed.

This thesis is entitled *"Computation of policies for inventory systems"*, and deals with a number of different inventory systems which can all be found at different stages in the supply chain. Supply chains come in different structures and shapes, but although recent trends like the Just-in-Time philosophy focus on eliminating them, inventories are important components of the supply chain. Inventories arise when demand and supply are not perfectly matched. Typically, there are economies of scale by ordering a batch of products, rather than ordering one unit for every demand. The supply may also be delayed by production or transportation times, and therefore we need to maintain an inventory to satisfy demands when they occur. In this thesis we show how different inventory systems can be modelled, and suggest and compute control policies for the systems. The methods developed can be used in decision support systems, as described above. Our focus is on the modelling and optimization of systems, rather than the actual implementation of the control policies, and the thesis is therefore theoretical rather than practical. The thesis is based on mathematical analysis, sometimes fairly complex and technical, sometimes requiring significant computer power. Although theoretical, all of our analyses is backed by numerical examples. In these we illustrate the performance of the suggested policies. One of the research objectives has been to keep the inventory policies suggested simple. Simple policies are easier to understand and to implement, and in many cases the cost difference between a simple policy and an optimal policy is small, this can be determined by numerical examples only. For some systems the optimal policy is unknown, and instead, heuristic policies have been developed. Again, numerical examples can be used to compare new approaches with existing ones.

We do not attempt to model a full-scale supply chain, but focus on a few sub-systems of the supply chain, where we find the control policies developed so far inadequate. The following systems are analysed:

- An inventory system with several demand classes.
- A make-to-order system with several demand classes.
- A multiple-item, one-supplier inventory system.
- A two-echelon inventory system.

Before we explore these systems in more detail, let us describe the properties of a simple inventory system.

We assume that the external source of *supply* for the inventory has infinite capacity and that orders can be placed at any review. Each replenishment arrives after a *lead time* that can represent production times, setup times, transportation times, or inspection times, or a sum of these. Lead times can be zero, constant, stochastic or variable. The *demand* can either be stochastic or deterministic. In many cases demand is treated as deterministic although it is stochastic, as an approximation. Sometimes deterministic models can capture the essence of an inventory problem, but in many cases, the fact that demand is stochastic is essential to the problem. This is the case for the inventory systems considered in this thesis. From an economic point of view, the level of the demand is a function of the price, but from a logistic point of view, prices and thereby the level of demand are typically considered fixed, and focus is on minimization of cost, rather than maximization of profit. Typically, customers are regarded as a homogeneous group. However, a significant part of this thesis is devoted to modelling and optimization of inventories with heterogeneous customers, with different requirements of service.

Now, let us consider the cost parameters. As mentioned earlier, inventories exist where supply and demand are not perfectly matched. Often there is a *fixed ordering cost*, which means that there are economies of scale by buying more than one unit at a time. This ordering cost includes cost of transportation, setup cost and handling cost. Normally, the actual purchase cost of the products ordered is not included in the analysis. If we use batch-ordering, some product are received before they are demanded. This affects our costs in two ways: First of all capital which could have been invested elsewhere is tied up as inventory. Secondly, there may be out-of-pocket cost of keeping products on storage. These costs are altogether called *holding costs*. The second driver for inventory is lead times in combination with stochastic demand. When demand cannot be predicted, we must maintain a safety stock to prevent stockouts. *Stockout costs* may represent cost of lost goodwill associated with not being able to satisfy a demand when it occurs. Such demands are either put on *backorder* and satisfied later, or are considered a *lost sale*. There may be a fixed stockout cost for every demand not satisfied immediately and, if a demand is backordered, a cost for every unit of time the demand is backordered. Sometimes a demand is expedited (i.e. satisfied in another way), if it cannot be satisfied directly from stock. In this case the fixed stockout cost can be the additional cost incurred by expediting the demand. If the demand is internal, the cost of goodwill may not be of concern. However, there may still be cost associated with stockouts. If the product is a critical spare part of a system, the stockout cost can represent the cost of not running the system for the backorder period. Also, some logistic providers make contracts with their customers, specifying a penalty fee to be paid in case of late deliveries.

The overall objective is to satisfy customer demands at the lowest expected long-run cost possible. Before deciding when and how much to order, we need to decide how often the inventory is reviewed. There are essentially two ways to monitor an inventory system. Either by *continuous review* or by *periodic review*. Whichever is chosen depends on the given situation. If the inventory is reviewed continuously, we know exactly when a demand has occurred and can react promptly by placing an order if necessary. Periodic-review policies have delayed reactions, but are on the other hand easier to implement and easier to coordinate with other activities. Practitioners tend to prefer periodic-review models, but many inventory models are continuous-review models, since they are often easier from an analytical point of view. Within this thesis we analyse policies for both continuous and periodic review.

Inventory policies specify when and how much to order. When taking this decision, it is important to focus on the *inventory position* rather than the on-hand inventory level. The inventory position equals the on-hand inventory level minus backorders plus outstanding replenishments. If the fixed cost of ordering is negligible or zero, we use a *base-stock policy*. This policy specifies a base-stock level, S, which will be the safety stock during the lead time. If the inventory position is below S, an order is placed, raising the inventory position to S. If the cost of ordering is positive, a base-stock policy may lead to very high ordering cost and instead, a *batch-order policy* should be used. There are two slightly different types of batch-order policies: (s, Q) policies and (s, S) policies. For both, an order is placed whenever the inventory position is below the reorder point s. Under an (s, S) policy an order is placed to bring the inventory position up to the target-stock S. An (s, Q) policy, on the other hand, just places an order of size Q. The advantage of the (s, S) is that our inventory position after an order placement will always be the same, while the order size may differ from time to time. The (s, Q) policy has the opposite properties. If demand is unit-sized and the inventory is reviewed continuously, the (s, S) and the (s, Q) polices are identical.

By combining simple inventory systems we can represent more complex supply chains. Consider for example, a supply chain where products are shipped from a manufacturer to several regional distribution centers before they are distributed to local retailers. Inventories are held at all installations in the supply chain. Inventory decisions taken at upstream installations obviously influence the downstream installation, and therefore joint optimization is important as argued earlier.

This thesis is based on seven scientific papers. In the following sections which constitute the first part of the thesis, we provide a survey of our results for the four systems mentioned above.

Section 2 deals with the management of inventory systems with more than one demand class. When stocks are low, the on-hand inventory can be rationed in order to provide better service for high-priority customers. We analyse both simple and optimal rationing policies and perform numerical tests that illustrate the strength of the policies found. The section is based on Melchiors, Dekker and Kleijn [37], Melchiors [35] and Melchiors [36].

In Section 3 we employ the principle of rationing to a make-to-order system. In a make-to-order system products are tailored to the actual demand, and it is impossible to keep inventory (and strictly speaking this system is therefore not an inventory system). The asset to be managed is instead the production capacity. Due to the complexity of the problem, computation power actually prevents the computation of optimal policies for large systems. We therefore focus on simple policies which are compared with the optimal policy on smaller systems, where they are found to have a fine performance. The section is based on Melchiors [34].

The joint replenishment of several items with a common supplier is considered in Section 4, based on the Melchiors [33] and Johansen & Melchiors [29]. The so-called can-order policy is analysed in the context of a continuous-review model and in the context of a periodic-review model. It seems to be generally believed that the can-order policy is dominated by the periodic replenishment policies. However, we demonstrate that, measured in costs, the can-order policy can be the best policy available today.

In Section 5 we discuss a two-echelon model where demand not satisfied immediately is lost, based on Andersson & Melchiors [2]. The prevalent assumption in multi-echelon inventory literature is that demand not satisfied immediately is backlogged rather than lost, and in this section we demonstrate how to relax this assumption.

Finally we present our conclusions in Section 6.

In the second part of the thesis we present the seven papers in their full length.

2 Inventory rationing

2.1 Introduction

Traditional inventory literature deals with the problem of how to replenish an inventory facing deterministic or stochastic demand. It is usually assumed that there is a cost of holding inventory, a fixed ordering cost (perhaps zero) and either a service level constraint, or a specification of a stockout cost for unsatisfied demand. In this section we will take a closer look at the demand process and how it influences our inventory decisions. The prevalent assumption in the inventory literature is that demand can be deterministic or stochastic following some known or unknown distribution, but that the demand is homogeneous. This means that, from a cost or service-level perspective, it is of no influence which demands are satisfied and which are not, in case of stockouts. A demand is a demand. This assumption is in many cases a realistic one. However, companies are today creating closer relationships with their suppliers, who in turn need to provide these key-customers the service they require. Simultaneously, the supplier faces demand from regular customers who may not be willing to pay for an increased level of service. In many cases a company can therefore divide their customers into demand classes of different priority. Service can have several meanings depending on the actual situation. Good service may be short lead times, access to EDI, flexible ordering opportunities, etc. By service we understand, within this thesis, the capability to satisfy a demand when it occurs, and nothing more.

The inventory rationing problem arises when customers for a single product are divided into classes of different importance. Assume for example that customers are divided into high-priority and low-priority customers. In order to provide a high service level for high-priority customers, the inventory manager must maintain a high safety stock to protect the inventory from stockouts. By doing so, low-priority customers will receive the same, unnecessarily high, service level. Alternatively, the inventory manager can ration his inventory. This can for example be done by using a critical level policy. The critical level policy, introduced by Nahmias & Demmy [39], specifies a critical level for each demand class. A demand is only satisfied if the inventory level is above the critical level for the demand class. In this way it is possible to reserve stock for possible future high-priority demand. In this section we will analyse and optimize the critical level policy for an (s, Q) inventory model with two demand classes.

This critical level policy is not the best rationing policy. For example, if it is known that a replenishment is about to arrive, there is no need to reject a low-priority demand even if the inventory level is below the critical level. We therefore analyse an inventory rationing model and find the optimal policy, where actions are taken based on information about both inventory level and elapsed lead time. We define a *simple policy* to be a rationing policy with constant critical levels, and a *time-remembering policy* to be a rationing policy that allows the critical levels to depend on the time elapsed since the actual outstanding order (if any) was issued.

The advantage of the optimal policy may also be its disadvantage. Since the critical levels change over time it is only easily implemented in highly computerized implementations. In other implementations a much more simple policy is needed. Finally, we therefore analyse a *restricted time-remembering policy* that shares the simplicity of the simple policy and has a performance close to that of the optimal rationing policy.

While the rationing policy can facilitate a reduction in safety stock, and still meet the required service levels for all demand classes, there are some potential disadvantages. While high-priority customers are only rejected if the inventory is empty, customers of lower priority may experience to have their demand rejected when there is still stock on hand, and observe other customers, arriving later, having their demand filled. Firstly, the situations where it is possible to divide customers into classes are mostly characterized by a buying process where sales are not made over the shelf in a store, but rather by phone, EDI, or via mail correspondence, which means that customers do not meet. Secondly, many customers are aware of which class they belong to and accept the corresponding service level.

We proceed as follows. First we give two examples where customers are divided into classes of priority, whereupon we discuss the background of the rationing policies and the related literature in Section 2.2. In Section 2.3 we discuss the underlying assumptions of the models, and in Section 2.4 we analyse a critical level policy for a case with two demand classes and constant lead times. The optimal time-remembering policy is analysed in

Section 2.5, assuming several demand classes and stochastic lead times. Finally, in Section 2.6, we analyse the restricted time-remembering policy, followed by our conclusions in Section 2.7. The content is based on Melchiors, Dekker & Kleijn [37], Melchiors [35] and Melchiors [36], which are referred to for details of the analysis and the numerical results.

First, let us take a look at two practical examples where demand can be distinguished based on priority.

Arla Foods

Arla Foods (www.arlafoods.com) is Europe's largest dairy group, manufacturing dairy products like milk, cheese and yoghurt. They have divided their English customers (which are the supermarket chains, not the consumers) into four classes of priority: A, B, C and D. Class A consists of the major supermarkets chains that are the key customers of Arla Foods. Class B consists of smaller supermarkets, while customer groups C and D, typically, are kiosks and gas stations.

Arla Foods uses EDI to obtain information about demand from A-customers. While this lay the basis for future forecasts, it can only indicate near-future demand. Short-term fluctuations in demand are primarily caused by changes in consumer behavior due to competitor promotion campaigns which cannot be predicted by Arla Foods. Demand not satisfied immediately is either lost or backlogged: Customers that order just enough to cover the demand for one or two days will typically not be interested in receiving the replenishment two days later, since they cannot backlog unsatisfied consumer demand. However, customers that only order every second week will require to have their demand backed up as soon as possible.

The dairy products manufactured by Arla Foods are perishable in the sense that the sales manager has approximately one week to sell the products after they arrive at the warehouse. Lead times are fairly long, which means that it is not possible to issue emergency orders in order to cope with stockout situations. There are, therefore, only two ways Arla Foods can protect itself from stockouts: By ordering enough to practically prevent stockouts, or by rationing the stock on hand. Due to the perishability of the dairy products, all unsold products are lost by the end of the sales horizon, which makes overstocking expensive. Rationing is employed on a common-sense basis, but there is currently no formal rationing policy in use.[1]

[1] Personal communication with Per Kristensen, Supply Manager, Arla Foods, July 2000

Multipart

Multipart (www.multipart.com) encompasses product support, logistics, marketing and after-demand services for their clients. On the logistics side they provide complete inventory management service including capture of demand data, forecasting, inventory control, placement and management of replenishment orders, and expediting. They have contracts for providing spare part support to the following customers:

- Hyundai - motor cars in the UK.
- Leyland/DAF trucks, vans, buses.
- Other commercial vehicles and tractor spares.
- Ministry of Defence - Challenger II tanks.

The number of spare parts serviced totals around 300,000, a lot of these being slow movers. The spare parts are generally not repairable. Multipart writes contracts with their customers, committing Multipart to provide a given service level for a selection of spare parts. These service levels depend on the customer's need. If an expensive truck is off the road and urgently needs a spare part before it can be road bound, this will require a higher service level than if the truck owner could easily wait a few days. This is reflected in the price of the contract. Multipart considers to employ a rationing system that would allow them to maintain service levels and reduce holding costs. The inventory holding cost constitutes a significant part of the total costs at Multipart, so even half a percentage point reduction in inventory cost would have significant impact on total profit. [2]

2.2 Background

Despite their practical applicability the models of inventory rationing have not yet found their way to textbooks and state-of-the-art overviews of inventory modelling. In recent books by Graves et al. [19], Silver, Pyke and Peterson [49], Tayur et al. [51], Zipkin [63] and Axsäter [6] inventory rationing is not mentioned. There have, however, been several contributions in the literature. The first contributions were periodic-review models. Veinott [56] analyses a model with several demand classes and zero lead time, and introduces the concept of critical levels. Topkis [54] 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 to the next review. The optimal policy is found by dynamic programming. More or less simultaneously Kaplan [30] and Evans [14] derive essentially the same results for the case

[2] Personal communication with Dr Richard Marett, Research Controller, Multipart Distribution Ltd., January 2001

with two demand classes. Cohen, Kleindorfer & Lee [12] consider a periodic-review (s, S) policy where demands from two demand classes are collected during a period. By the end of the period, the inventory is used to satisfy high-priority demand, after which remaining inventory is used to satisfy low-priority demand. They do not consider a critical level policy. Zhang and Sobel [61] consider 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 backlogged. Demand is observed at the beginning of each period, whereupon 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. Backlogging is only allowed if the inventory is empty, which means that it is not a rationing model. Frank, Zhang and Duenyas [17] consider a modification of the model, where the stochastic demand can be rejected even if there is stock on hand, in which case the demand is lost. However, since replenishments are instantaneous, the purpose of rationing is not to save stock for future high-priority demand, but rather to postpone an order placement for a 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 [20] analyses a lot–for–lot lost sales model with n demand classes and Poisson demand. He assumes exponentially distributed lead times and models the system as a single–product $M/M/1/S$ queue (Tijms [53]) with state–dependent service times. This enables him to prove optimality of the lot–for–lot critical level policy. The model is extended by Ha [21] to cover Erlang distributed production times. Dekker, Hill & Kleijn [13] analyse the same model with a general lead time distribution. They model the system as an $M/M/S/S$ queue (Tijms [53]) and develop efficient methods to determine the best policy. Since they do not consider time-remembering policies, optimality cannot be guaranteed. Inspired by this paper Axsäter, Kleijn & De Kok [7] analyse a two-echelon system with one warehouse and N retailers, with base-stock policies applied at all installations. Here, the warehouse rations its stock in order to be able to provide a high service level to the retailers with high costs of stockouts (or expediting).

Simple critical level policies for a continuous review (s, Q) inventory model are first analysed by Nahmias & Demmy [39] 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 [38] generalize their results and find optimal rationing levels for a model with compound Poisson demand, deterministic demand and several demand classes. The only paper considering time-remembering policies in a continuous review setting is that of Teunter and Klein Haneveld [52], presenting 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.
The theme of different demand classes is subject to research in other settings as well. Kleijn [31] considers a model where customers' demand is distinguished based on demand sizes. A so-called break-quantity rule is applied to determine how to separate small demands which are satisfied from stock on hand and large demands which are satisfied in an alternative manner, e.g. directly from the supplier.
Closely related is also perishable asset revenue management (PARM, see Weatherford and Bodily [59]), also known as yield management, which is mainly applied in the hotel industry and in particular in the airline industry. The essence of PARM is captured by American Airlines [1], who in 1987 stated their objective as "selling the right seat to the right customer at the right time". Inventory rationing is not classified as perishable asset revenue management, since we assume that products can be kept on stock for as long as we like without perishing, in contrast to one of the fundamental characteristics of PARM. In Section 3 we discuss, in more detail, the relation between PARM and the rationing of a make-to-order system.

2.3 Notation and assumptions

In this section we focus on the computation of rationing policies in the context of an (s, Q) policy. Since this is one of the most used policies in practice, it is of high relevance to develop good rationing policies in this context. We will first state the assumptions made, and then discuss some of them in more detail:

1. There are n demand classes. Class j has Poisson demand with arrival rate λ_j.
2. All demand not satisfied immediately is assumed to be lost (or expedited).
3. The classes are distinguished only by their stockout cost π_j. We rank the classes such that $0 < \pi_n < \pi_{n-1} < \cdots < \pi_1$.
4. For each replenishment order there is a fixed ordering cost, $K > 0$, and a replenishment lead time, which may be constant or stochastic. The unit holding cost per unit time is $h > 0$.
5. Our objective is to minimize the long-run average cost of the system.

Assumption 1: In principle, each customer may be treated as one demand class. However, many customers will have similar profiles, and it seems reasonable, as an approximation, to join them in order to reduce the number of demand classes. In practice we believe that the number of demand classes should be between 1 and 5.

Assumption 2: The assumption of lost sales as opposed to backordering is made primarily in order to facilitate the analysis of the model. By restricting ourselves to a lost sales environment, we do not have to keep track of a backorder list for each demand class, which allows a smooth model formulation.
Assumption 3: If there were two demand classes with demand rates λ_i and λ_j with identical stockcost $\pi = \pi_i = \pi_j$, we could join these into one class with demand rate $\lambda_i + \lambda_j$ and stockout π, since demand is Poisson. Although our analysis does not prohibit negative stockout costs, we assume that the stockout costs are positive. Since a demand not satisfied immediately is lost and not backordered, there is no time dependent penalty cost.
The use of an (s, Q) ordering policy is justified by the fixed positive ordering cost K. The remaining assumptions are discussed in Section 1.

2.4 Inventory rationing with two demand classes

First let us consider the case where $n = 2$ and all lead times are constant. This section is based on Melchiors, Dekker & Kleijn [37], which we refer to for a more detailed analysis and extensive numerical results. First of all, this is a simple model from a computational point of view and the advantage of the rationing policy compared to a non–rationing policy is still considerable. Our analysis departs in the paper by Nahmias & Demmy [39]. By the use of 'hitting' times we derive an expression for the long-run average cost of using a simple critical level policy, which is minimized by a simple procedure.

2.4.1 The model

We analyse the rationing policy in the context of an (s, Q) policy with reorder point s and order size Q, where $Q > s$. This condition and the lost sales assumption ensure that at most one order is outstanding at any time. In principle, the critical level c is unbounded, but for the model to be tractable we require that $c < Q$. The critical level policy is denoted (s, c, Q). The inventory process can be split into independent and identically distributed renewal cycles. Using the renewal–reward theorem (see e.g. Tijms [53]) we know that the average cost per time unit equals the expected cost incurred during a cycle divided by the expected length of a cycle. In case the inventory policy satisfies the condition $c < s$, we let H be a random variable denoting the hitting time of the critical level, i.e. the time from placing a replenishment order (or the time when the inventory level 'hits' the reorder level s) until the time where the inventory level 'hits' the critical level c. Since the total demand from both classes follows a Poisson distribution with parameter $\lambda := \lambda_1 + \lambda_2$, it readily follows that H is Erlang distributed with parameters $s - c$ and λ. Furthermore, we define R as the random variable denoting the inventory level just before a replenishment order arrives. Figure 1 illustrates the inventory process over two cycles for a (c, s, Q) policy with $c < s$.

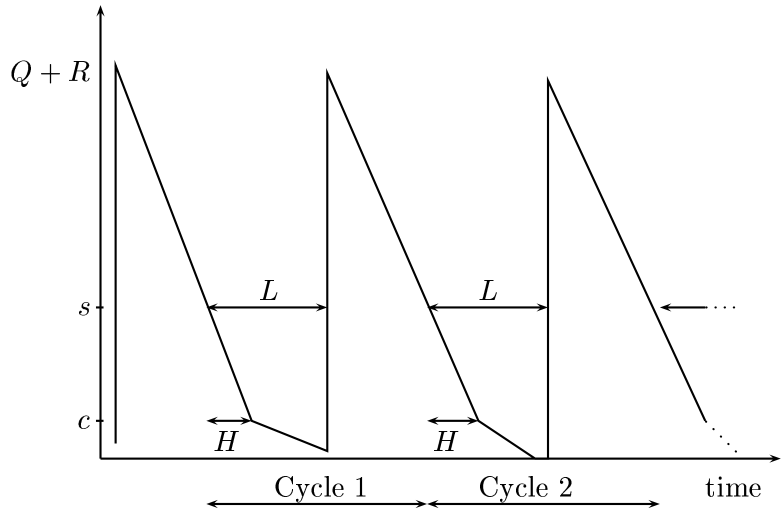

Figure 1: The inventory process with $c < s$.

We can now derive an expression for the average cost in a (c, s, Q) inventory system. The total cost consists of holding costs, shortage costs, and ordering costs. The approach we follow is to derive first the expected cost during a cycle and then calculate the expected cycle length. Using the renewal–reward theorem we obtain an expression for the average cost. The expressions are found by conditioning on the values of H and R. The average cost is denoted, respectively, $TC^{s<c}(c, s, Q)$ and $TC^{s \geq c}(c, s, Q)$ depending on whether $s < c$ or $s \geq c$.

2.4.2 Optimization

Due to the complexity of the average cost formula it has not been possible to derive an explicit expression for the optimal policy. The optimization procedure is therefore based on enumeration and bounding.

For a fixed Q we derive an upper bound on the optimal reorder point s. The bound is based on the following conjecture that we have not been able to prove : "The optimal reorder point s will be less than or equal to the optimal reorder level for the model without a critical level". Now for all s between 0 and the upper bound we calculate the best value of c. First we evaluate all critical levels between 0 and $s - 1$ using the average cost function $TC^{c<s}(c, s, Q)$. The average cost function $TC^{c \geq s}(c, s, Q)$ is for fixed values of s and Q either convex or concave in c, depending on the underlying model and the values of s and Q. If the average cost function is convex, c is found in the global minimum, which can be found explicitly. Otherwise the optimal value of c is found in either $c = s$ or $c = Q - 1$. The best critical level for each s can then be specified. This method will lead to the optimal values of (c, s).

In many practical situations the order size Q is prespecified. However, if one also wants to determine the optimal value of Q, one can use a local search algorithm with the Economic Order Quantity as a starting solution. Numerical experiments have indicated that the average cost function is quasi-convex in Q.

2.4.3 Numerical results

We compare the rationing policy with the best non-rationing policy for different combinations of demand rate, lead times, stockout cost and order cost. In summary the conclusions are: When the difference in stockout cost between the low-priority class and the high-priority class is significant, the cost reduction obtained by using a rationing policy is substantial. The influence of the demand rate is not that significant but systems where the majority of the demand is low–priority demand give the highest cost reductions. Since demand is Poisson, increasing the lead time increases the variance of the lead time demand and therefore the cost reduction increases with the lead time. For the majority of the investigated problems the cost reduction is between 1% and 6% compared with the best non–rationing policy.

To the best of our knowledge, we are the first to investigate rationing policies with a critical level above the reorder point. These policies are optimal for models where the stockout cost of the low–priority class is low compared to the holding cost and the order cost. In such models we would like to satisfy some of the low–priority demand but not all of it, since this would lead to very high holding or order costs. Huge cost reductions (up to 40 %) can therefore be obtained in these cases.

We conclude that the rationing policy can have two different effects on the optimal reorder level and replenishment order size, depending on whether the critical level is below or above the reorder level. In the first case, the critical level policy reduces the safety stock needed. Significant cost reductions can be obtained if the stockout cost of high–priority demand is considerably larger than the stockout cost of low–priority demand. If the critical level is at or above the reorder point, then the rationing policy will reduce the average holding cost, by rejecting a great part of the low–priority demand. This is in particular advantageous if the cost of rejecting low–priority demand is small (compared to the holding cost rate) or if the fixed order cost is high.

2.5 Inventory rationing with several demand classes

We will now look at the case with several demand classes and stochastic lead times. The analysis and the numerical results are based on Melchiors [35], to which we refer for further details. When dealing with stochastic lead times, it often suffices to focus on the distribution of the lead time demand, rather than the lead time itself. For our purpose this distribution is, however, not sufficient. Whether a demand is satisfied or not depends on when the demand occurs, and therefore we have to decompose the lead time demand into two random variables: the lead time and the Poisson demand.

The approach used in the previous section is not suitable, since we would have to condition on up to $n-1$ hitting times plus the length of the lead time to evaluate a given policy. This can only be done by a numerical integration technique which would be very cumbersome. Instead the problem is formulated as a Markov decision model where the decisions are allowed to depend on the inventory level and, if the inventory level is below the reorder level, the time since the replenishment order was issued. We show that the optimal policy is a time-remembering policy and that the critical levels for the case of constant lead times are decreasing in time. Since a time-remembering policy can be difficult to implement in practice, we also show how to find a good simple critical level policy.

We note that stochastic lead times in a rationing environment are also considered by Ha [20, 21], as mentioned earlier.

2.5.1 The model

As in the previous section we analyse the rationing policy in the setting of an (s, Q) inventory policy, with $s < Q$, ensuring at most one outstanding order at a time. The time when an order is outstanding is discretized to obtain a finite number of time epochs, each representing a small subinterval of length $1/N$. The stochastic lead time is then approximated by the probability mass function $f(t)$, which is the probability of a lead time of t subintervals. The true lead time may be continuous, but if N is high the error incurred is negligible. We assume that there exists an integer M such that $\sum_{t=0}^{M} f(t) = 1$ (for unbounded distributions we choose M such that the probability of a lead time of more than M subintervals is negligible). Based on $f(t)$ we can calculate the lead time hazard function $H(t)$, that denotes the probability of an arrival just prior to subinterval t, given that no order has arrived prior to subinterval $t-1$.

$$H(t) = \frac{f(t)}{\sum_{r=t}^{M} f(r)} \text{ for } t = 0, 1, \ldots, M-1$$

and $H(M) = 1$.

Assuming that s and Q are fixed, we 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. 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

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

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 and no order is outstanding, and at the beginning of each subinterval when one order is outstanding. At 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 if $a > 0$, 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. Since we do not allow backlogging we set $a = 0$ in states where the inventory level is zero.

The number N determines the length of each subinterval and 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 [11]). A Bernoulli process is a sequence of independent trials with an outcome that is either zero or one. Each of the subintervals can be viewed as a trial where the outcome is one, if a demand we are willing to satisfy occurs, and zero otherwise. The assumption of at most one demand per subinterval is not restrictive. If the policy is implemented in practice, it can handle more than one demand per subinterval. The approximation considerably simplifies 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 difference between N and M is subtle: N determines the length of a subinterval, while M determines the maximum number of subintervals in a lead time.

We 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 t/N and $(t+1)/N$, satisfy demand from classes 1 to $l(i,t)$.

The considered policy is not necessarily a critical level policy. To be a critical level policy it must satisfy

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

and

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

This means that, for each class $j \geq 2$ and for all t, there exists a unique critical level $c_j(t) = \max\{i | l(i,t) < j\} (= 0 \text{ if } l(1,t) \geq j)$. 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 where class j will not be served. 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. Observe that, if $l(i,t)$ is a constant function of t for all i, then the policy is a simple critical level policy.
For a given policy we can specify the transition probabilities, the expected time between two decision epochs and the expected one-step cost. Based on these, the long-run average cost per unit time (henceforth referred to as *cost* for simplicity) of using such a policy is found using the renewal-reward theorem (see e.g. Tijms [53]).

2.5.2 Optimization

We present two different policies. An optimal time-remembering policy and a good simple policy. The optimal policy is found by solving the so–called average cost optimality equations (see Tijms [53]). These are solved by a tailor-made policy-iteration algorithm, that utilizes the structure of the Markov chain. Based on the average cost optimality equations, we can derive the following theorem which considerably simplifies the search for the optimal policy.

Theorem. *For an optimal rationing policy the following properties are true:*

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

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

c *If the lead time is constant, then the critical levels are decreasing in the time t, i.e. the optimal actions satisfy*

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

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 neighbor search starting e.g. with the Economic Order Quantity, computed by considering the deterministic version of the problem where all demand classes are aggregated to one. For each value of Q, the optimal values of $k(i)$, $l(i,t)$ and s are found by the policy-iteration algorithm. The optimization procedure has been implemented in Pascal, and is very efficient.
Since the optimal policy can be difficult to implement in practice, it is relevant to find good simple policies for the case of several demand classes as well. 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 $(\boldsymbol{c}, s, Q)$.
For a fixed Q we enumerate all s in $[0, Q-1]$. For each s we find a 'good' vector of critical levels by a method based on the method suggested by Dekker, Hill & Kleijn [13]. 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 best vector obtained so far, and continue until no further improvements can be made. Each policy is evaluated by the method derived in the previous section.
The value of Q is determined by a neighbor search starting in the Economic Order Quantity. As for the optimal policy the minimum costs have been quasi–convex in Q in all our numerical tests.

2.5.3 Numerical results

Now let us illustrate the structure of the rationing policies by means of a numerical example, and, furthermore, give some guidelines for when to use rationing policies. To illustrate the optimal time-remembering policy consider the following example: $n = 4$, $K = 100$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 2$, $\lambda_4 = 7$ and $\pi_1 = 1000$, $\pi_2 = 40$, $\pi_3 = 12.5$, $\pi_4 = 5$. The lead time is constant with length $L = 1$ time unit, and the holding cost is $h = 1$.

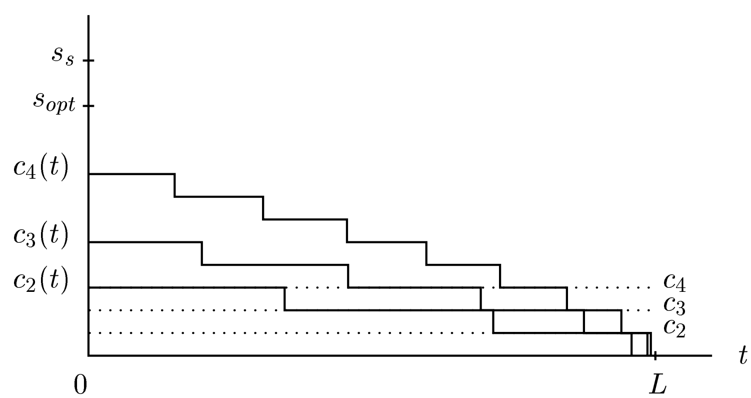

Figure 2: The critical levels of the optimal policy $c_2(t)$, $c_3(t)$ and $c_3(t)$ and the constant critical levels of the simple policy c_2, c_3, c_4 as function of the time t.

The reorder point of the optimal policy is found to be $s_{opt} = 11$ and the order size is $Q_{opt} = 48$. The cost of the optimal policy is 50.72. The best simple policy is $(\boldsymbol{c}, s_s, Q_s) = (1, 2, 3, 13, 48)$ with cost 51.79, which is a difference of 2.1%. Figure 2 shows the critical levels of both the simple and the optimal policy and illustrates furthermore the advantage of the optimal policy. In the beginning of the lead time the optimal policy rejects the demand classes 2,3 and 4 at a higher level than the simple policy does, and by the very end of the lead time, the optimal policy rejects no demand at all. That is, the optimal policy dominates the simple one in two situations: When demand is high in the beginning of the lead time, and when low-priority demand 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 inventory rationing policies can be applied to any inventory system with several demand classes. However, some systems may benefit more from rationing than others, and therefore we try to give some guidelines as to when the rationing policies are particularly beneficial, based on the numerical examples by Melchiors [35]

As expected we find rationing to be particularly important when the stockout costs of the different demand classes differ significantly, i.e. more than a factor of 10. With respect to the ordering cost, we find that rationing has the greatest impact when the ordering costs are low. This is mainly because the measure of performance is cost per unit time, and it is only during the lead time (except for the cases where the critical levels are above the reorder point) that rationing is used. As the ordering cost increases, the length of the inventory cycle increases, and thereby the cost difference per unit time decreases. The structure of the demand also has an impact on the general performance. The highest benefits are found when the majority of the demand is of low priority. When the majority of the demand has high priority, rationing will only seldom be used and consequently the relative benefits are smaller. As the general level of demand increases, the benefits of rationing increase as well. Since demand is Poisson, a higher demand mean is equivalent to a higher demand variance, and therefore the need for safety stock increases. Naturally, the same effect occurs when we increase the variability of the lead time. When we are uncertain about the delivery time of an order, the rationing policy reduces the need for otherwise high safety stocks. For the investigated examples the cost reduction obtained using the optimal policy rather than a non-rationing policy varies between 0 and 13%.

With respect to the difference between the simple and the time-remembering policy we have found relative differences between 0-3%. In general, the difference between the two policies is highest when the lead time is either constant or predictable in some sense. For example, either 'normal' with high probability or long with a small probability. When the lead time is predictable we can make full use of the information of time. Contrary to this, in cases where the length of the lead time is very uncertain, the value of time information is small. In these cases the optimal policy will do no better than a policy that disregards information about time, and therefore the optimal policy approaches a simple policy. Since the optimal policy is much more complicated to implement, we recommend to use the simple policy in most cases. However, for the case of constant or predictable lead times, the cost difference between the simple and the optimal policy is around 2-3% and in some cases this may be significant.

2.6 Restricted time-remembering policies

So far, we have described two kinds of policies, i.e. simple critical level policies and time-remembering policies. In practice a combination of the two could be more appealing. We analyse a policy where the rationing decisions are only based on whether the remaining lead time is say, short or long. We can thus improve the performance of the simple policy by taking time into account, and still have a policy that is easy to implement in practice. As illustrated the cost difference between simple and optimal rationing policies is particularly high in cases with constant lead time. In those cases the value of information of time is high compared to cases with stochastic lead times where a simple policy that neglects information of time is almost as good as the optimal time-remembering policy. Consequently, we focus only on the case with constant lead times. The section is based on Melchiors [36].

Define a *restricted time-remembering*(RTR) policy to be a time-remembering policy where the critical levels are restricted to be constant over intervals of the lead time. These intervals are defined by the policy variables and must cover the entire lead time. The number of intervals determines how simple the RTR policy is. If there is only one interval, the RTR policy is identical to the simple critical level policy, and if there is M intervals, it is identical to the optimal rationing policy. Obviously the performance of the RTR policy increases when the number of intervals is increased.

Assume that we restrict the policy to be constant over m intervals of time, and let $c_{j,\tau}$ denote the critical level of class j in interval number τ for $1 \leq j \leq n$ and $\tau = 1, 2, \ldots, m$. Interval τ consists of the subintervals $\{t_{\tau-1}, t_{\tau-1}+1, \ldots, t_\tau - 1\}$ with $0 = t_0 \leq t_1 \leq \cdots \leq t_m = M$, with $t_1, t_2, \ldots, t_{m-1}$ being decision variables.

Note that we require the set of intervals to be identical for all demand classes. A set of critical levels uniquely defines all values of $l(i,t)$:

$$l(i,t) = \max\{j | c_{j,\tau} \leq i,\ t \in \{t_{\tau-1}, t_{\tau-1}+1, \ldots, t_\tau - 1\}\}$$

Furthermore we have the option of rejecting a demand even before an order is placed, as specified by $k(i)$. Demands rejected before an order is placed, will not be satisfied during the lead time either, and we can therefore use the critical level of interval 1, to determine the values of $k(i)$:

$$k(i) = \max\{j | c_{j,1} \leq i\}$$

In this way we can evaluate any rationing policy by the method presented in Section 2.5. However, we cannot use a policy-iteration algorithm or a value-iteration algorithm to find the optimal RTR policy. Instead, we optimize the policy by a neighbor search based on the following empirical observations:

a The optimal critical levels increase (or remain the same) as the reorder point decreases.

b The optimal value of t_τ decreases (or remains the same) as s decreases for all τ.

c The optimal critical levels $c_{j,\tau}$, $c_{j,\tau+1}$ increase (or remain the same) as t_τ decreases for all τ .

Although it appears difficult to prove these observations, all our computations support them, and our optimization algorithm have found a policy equal to the optimal RTR policy (determined by a full enumeration approach) in all our numerical tests. Observations 1 and 2 can be explained as follows: When we reduce the safety stock available during the lead time we will (for fixed value of all t_τ) on average hit the critical levels earlier. The remaining stock (reserved for demand of higher priority) must therefore last for a longer period. Consequently, the optimal critical levels increase (or remain the same). Since we hit the new critical levels earlier, the optimal value of t_τ will decrease when the reorder point decreases, since we are interested in choosing the value of t_τ close to the expected hitting times. Observation 3 is illustrated in Figure 3. When the value of t_τ decreases, the critical levels covering period $\tau+1$ will increase since they must now also cover the interval $[t_\tau^{new}, t_\tau^{old}]$ where the remaining lead time is longer and therefore the safety-stock needed will be higher. Similarly, the observation holds for interval τ: Here the interval is reduced and since we now do not have to cover the interval $[t_\tau^{new}, t_\tau^{old}]$, we can increase our critical levels to obtain a better coverage of the interval $[t_{\tau-1}, t_\tau^{new}]$.

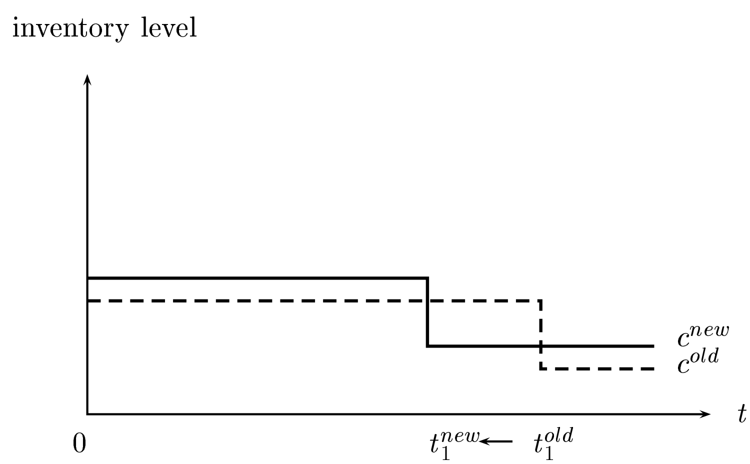

Figure 3: Illustration of how the optimal restricted critical levels increase (for both intervals) as the time t_1 decreases from t_1^{old} to t_1^{new}.

Based on these observations we construct an algorithm for finding the optimal RTR policy. First, we investigate how to separate short and long lead times in general, or more precisely, how to choose t_1 in the case $m = 2$. We note that there is almost nothing gained by policies with low value of t_1. In general, the need for critical levels is in the last part of the lead time. This is where we are likely to run out of stock and therefore this is where we can benefit the most from different critical levels. The best values are found around $t_1 = 0.75M$.

Next, we investigate how the choice of m influences the performance of the RTR policy. Our results show that 51% of the gap between the simple policy ($m = 1$) and the optimal policy ($m = M$) is covered by increasing m from 1 to 2. By increasing m from 2 to 3 we can further narrow the gap by additional 20%, but the benefits of increasing m further are very small. Thus by setting $m = 2$ (or 3) we can capture the essence of the time-remembering policy and still have a policy that is fairly easy to operate in practice.

2.7 Conclusion

In this section we have shown how to find simple, restricted time-remembering and optimal rationing policies for an (s, Q) inventory model with lost sales, and up-to several demand classes. Compared with a non-rationing policy the use of a rationing policy can lead to significant cost reductions. Since the optimal policy can be difficult to implement in practice, we have suggested the simple critical level policy and the restricted time-remembering policy which are both easy to find and easy to implement in practice.

While the methods developed may not be directly applicable to practical cases like Arla Foods or Multipart, the underlying ideas of the analysis and the structure of the policies can form a solid foundation for tailor-made applications.

3 Capacity rationing

3.1 Introduction and Background

In this section we apply the rationing policies developed for the make-to-stock inventory system to a make-to-order production system. Make-to-order systems are employed when it is not possible (or cost efficient) to produce products before they are demanded. Typically this applies to systems where each product is unique or requires significant customization. Rather than keeping products on inventory, such systems have a capacity to produce which means that whenever a demand occurs, some of this capacity can be allocated to satisfy the demand. Capacity may be an asset easily adjustable to meet demand, but in many cases capacity is fixed, at least on a short term, and therefore the capacity on hand is often not enough to meet demand. Whenever demand for capacity can be divided into classes either due to priority, criticality or profitability, decisions on which demands to accept and which to reject must be taken. It is particularly important to be able to meet high priority/profitability demands when they occur, and therefore it can make good sense to ration the capacity on hand.

The importance of using admission control for make-to-order system to control the performance of the system has been recognized by several authors. Hendry and Kingsman [23, 24] build a hierarchical system to control lead times in make-to-order companies. This is achieved by using a customer enquiry system based on the amount of the total backlog in the system. If this amount exceeds a maximum limit determined by management, orders are rejected or extra capacity is purchased. Balakrishnan, Sridharen and Patterson [8] consider capacity rationing for a make-to-stock system with two demand classes. There is a fixed amount of capacity to be allocated to demand during the selling season.

Essential to the problem characterization is the perishability of the capacity; unused capacity represents no value. On the other hand, it is also important not to allocate too much capacity to demand classes with low profitability if this means that there will be no capacity left for demand classes with high profitability. This aspect is central in the problem faced by airlines or hotels, known as perishable asset revenue management (PARM, see Weatherford and Bodily [59]). The capacity held by airlines is the actual number of seats on a given flight. Customers are typically divided into business and tourist class, paying different fares for essentially the same service. What distinguish PARM from the make-to-order system is that customers who require a flight or a hotel room Saturday, will (often) not accept a flight/room Friday or Sunday, whereas in the make-to-order system we can use capacity of adjacent periods to satisfy demand and in that sense we can store our capacity or put it on backorder.

We consider a make-to-order system with several job classes in a multi-period setting with rolling horizon. Jobs arrive stochastically and must be either accepted or rejected upon arrival. After acceptance we must decide in which periods to process the job. Every job is described by the amount of capacity required, a due date and a profit per workload unit, which are all known and deterministic. The modelling of different demand classes is central for the problem formulation. We assume that prices and due dates are fixed and non-negotiable; instead, we allow rejection of arriving jobs. Another approach is suggested by Johansen [27], who presents a job-shop model where, at every job arrival, a price is calculated based on the current state of the system and the workload of the incoming job. Our approach assumes that prices and capacity are fixed and focus on the short term management of the congested system. The objective is therefore to maximize the expected net profit, rather than to minimize the exptected costs, as in the remainder of the thesis.

We analyse two models: Model A is a simplified model, where it does not matter when a job is processed as long as it meets its due date. Model B is a more general model we need not meet the required due date; instead, there are penalty costs per period late and holding costs from the process has begun until the due date of the job. We present a general framework that provides a decision tool for both models. The strength of the framework lies in its coverage of a wide range of problems, and its capacity to solve these to near-optimality. However, the requirement of computation time and memory is high and increasing in the length of the planning horizon and with the number of different job types. For larger and more complex problems it seems unlikely that an optimal policy can be found, due to "the curse of dimensionality". Our approach is therefore to design heuristic policies that can be found even for complex real-life problems by the use of simulation. These simple policies are benchmarked against the near-optimal policy on smaller problems (where optimization is possible), and are shown to have a fine performance. The section is based on Melchiors [34].

3.2 Modelling

We consider a make-to-order system receiving jobs from a set of customers. Each job type j is described by its workload W_j, a desired due date DD_j, and the profit p_j per unit workload.

There is a holding cost hp_j per unit workload per period from the day processing of a type j job is begun until its due date. If processing is finished after the due date a penalty cost of πp_j per unit workload per period late is incurred.

The time unit is a period (representing a day or a week for example) which is divided in T sub-periods (hours of the day or weekdays). Incoming jobs can arrive in any sub-period, but cannot begin processing before the next period. We let $f(j,t)$ denote the probability that job j arrives when there is t sub-periods remaining in a period. We assume that there can arrive at most one job per sub-period.

We consider a rolling horizon with a length of N periods. The current period is period 0, and the first period a job can be processed in is period 1. The length of the planning horizon is fixed, and we can only accept jobs that can be processed within this horizon.

When a job is received, it must either be accepted or rejected. We assume that all jobs can be processed over several periods without additional costs (besides the holding costs). The allocation of workload is done upon the acceptance of the job. It is not possible to change this allocation. We assume that a minimum unit of workload exists. The capacity of the system in each period is C workload units.

The problem is formulated as a Markov decision process. Let $(t, \boldsymbol{x}) = (t, x_1, x_2, \ldots, x_N)$ be the state of the system. Let t denote the remaining sub-periods of the period and x_i denote the free capacity in period i. The process $\{t, \boldsymbol{x}\}_k, k = 1, 2, \ldots$ is a discrete time Markov chain with state space $S = \{0, 1, \ldots, T\} \times \{0, 1, \ldots, C\}^N$. In all states $(t, \boldsymbol{x})$ with $t > 0$ we decide which jobs we will accept and how to allocate their workload. States $(0, \boldsymbol{x})$ are artificial states where no job can arrive, representing the end of a period. Let $\boldsymbol{\alpha}$ be a vector of components α_n which is the workload allocated to period n for a job. An allocation for an accepted job with workload W must satisfy

$$0 \leq \alpha_n \leq x_n \text{ for all } n = 1, 2, \ldots, N$$

and

$$\sum_{n=1}^{N} \alpha_n = W.$$

We let the non-allocation with $\alpha_n = 0$ for all n denote a rejected job. Let $\mathcal{A}(\boldsymbol{x}, j)$ denote the set of feasible allocations, including the non-allocation, for job j in state $(t, \boldsymbol{x})$.

The model is similar in structure to a vehicle insurance example given by Tijms [53], and we can therefore use a similar analysis. We find the near-optimal policy, that maximizes average profit per period by a value-iteration algorithm. Let $c(\boldsymbol{\alpha}, j)$ be the net profit of a job j with allocation $\boldsymbol{\alpha}$, and let $v_i(t, \boldsymbol{x})$ be the maximum value of the expected future profit obtained i periods and t sub-periods from the end of the horizon in state $(t, \boldsymbol{x})$. $v_i(t, \boldsymbol{x})$ is found by the recursion

$$\begin{aligned} v_i(t, \boldsymbol{x}) &= \sum_{j \in J} f(j, t) \max_{\boldsymbol{\alpha} \in \mathcal{A}(t, \boldsymbol{x}, j)} \Big[c(\boldsymbol{\alpha}, j) + v_i(t-1, \boldsymbol{x} - \boldsymbol{\alpha}) \Big] \\ &\quad \text{for } 1 \leq t \leq T, \forall \boldsymbol{x} \geq \mathbf{0}, \forall i > 0 \qquad (3) \\ v_i(0, x_1, x_2, \ldots, x_N) &= v_{i-1}(T, x_2, x_3, \ldots, x_N, C) \; \forall \, x_1, x_2, \ldots, x_N \geq 0, \; \forall i > 0 \\ v_0(T, \boldsymbol{x}) &= 0 \; \forall \boldsymbol{x} \geq \mathbf{0}. \end{aligned}$$

By solving this system, we can obtain a near-optimal policy specifying which jobs to accept and how to allocate them, for any job j and any state $(t, \boldsymbol{x})$.

The disadvantage of the near-optimal policy is that it is difficult to implement in practice, and even more important, for complex problems its computation is intractable. We therefore suggest simple policies, characterized by a few policy variables which are easy to implement in practice. For complex real-life problems these variables can be optimized by simulation. The simple policies cannot be found directly by the value-iteration algorithm. But the value-iteration algorithm can be used to evaluate the simple policies which, in combination with a search mechanism, is used to find good simple policies.

In the two following sections we show how to find near-optimal policies and design simple policies for Model A and Model B, respectively.

3.3 Model A

In this section we consider the special case of the general model where $\pi = \infty$ and $h = 0$. The problem when a job arrives is whether it should be accepted or not and when to process it. Here we are not influenced by holding and penalty costs forcing us to process the job near the desired due date, we are free to process the job whenever we prefer, as long as the due date is met.

The net profit of a job of type j with feasible allocation $\boldsymbol{\alpha}$ equals

$$c(\boldsymbol{\alpha}, j) = p_j W_j$$

if the job is accepted, and $c(\mathbf{0}, j) = 0$ if the job is rejected. Instead of searching over all possible allocations, we consider two allocation heuristics: The first algorithm, FIFO(first in first out), allocates such that everything is processed as soon as possible. The second algorithm, MI(marginal improvement), is a greedy near-optimal method, based on marginal improvements of the expected profit function $v_i(t, \boldsymbol{x})$. Using the MI algorithm we can solve (3) to find the near-optimal policy.

If we use the FIFO allocation method, we can reduce our mathematical model significantly. Instead of keeping track of available capacity for each day, we can keep track of the total amount of accepted workload waiting to be processed. The state of the inventory is (t, x) where t denotes remaining sub-periods of the period and x denotes, instead of a vector of free capacity, the sum of all accepted workload not yet processed. The policy found by this method is called rationing(FIFO).

We investigate three heuristics: a simple policy, a selective policy, and a non-rationing policy. We divide the jobs into job classes based on the profit the jobs provide. Let $\tilde{p}_i$ be the profit per workload unit of a job from class i. Let m be the number of job classes and order the classes such that $\tilde{p}_i > \tilde{p}_{i+1}$ all $i < m$. The simple policy is based on the reduced mathematical model. We assign a critical level c_i to every job class i. A job j is accepted only if the resulting accepted workload $x + W_j$ does not exceed the critical level of its class. If a job is accepted it is scheduled to be processed as soon as possible (FIFO). We use a search routine similar to that of Melchiors [35] to find a good critical levels.

The selective policy accepts all jobs from class $1, 2, \ldots, i$, and is thus a reduction in the set of jobs classes, rather than a rationing policy. By evaluating the selective policy for different values of i, we can find the best selective policy. Finally, the non-rationing policy accepts all jobs that can be processed within their due date.

We investigate several numerical examples. Our measure of performance is the net profit obtained by the heuristics as percentage of the net-profit obtained by using the near-optimal(MI) policy.

In general, the performance of the rationing(FIFO) policy is very good, indicating that the benefit of leaving idle periods for flexibility is low. The simple policy is almost as good as the rationing(FIFO) policy. The performance of the selective policy is found to be 95-96% on average but for one case, as low as 83.8%. The non-rationing policy is in general very poor, with an average performance around 90% and a worst performce of 42.5%.

3.4 Model B

Now, let us consider the more general case where each job carries a positive holding cost hp_j per unit workload per period until its due date and a penalty cost πp_j per unit workload per period late. Typically, penalty cost must be paid for the entire workload for every period the last unit of workload is late. Holding costs are charged for the entire workload of the job from the period the job is begun until its due date. This corresponds to a situation where the cost of materials needed to process the job constitutes the major part of the value of the final product. Consider a job with j and an allocation $\boldsymbol{\alpha}$. Let $n_{min} = \min\{n|\alpha_n > 0\}$ and $n_{max} = \max\{n|\alpha_n > 0\}$. The net profit earned by accepting the job and choosing allocation $\boldsymbol{\alpha}$ is the profit minus the holding and penalty costs:

$$c(\boldsymbol{\alpha}, j) = p_j W_j - hp_j W_j (DD_j - n_{min})^+ - \pi p_j W_j (n_{max} - DD_j)^+$$

If the job is rejected, $c(\mathbf{0}, j) = 0$.

As observed in Section 4, the benefit of leaving idle capacity for future demand is only marginal, and therefore, once a job has begun processing, we will finish it as fast as possible. We use an allocation algorithm called BSP (best starting period), where the allocation for a job j has the following structure:

$$(0, \ldots, 0, \alpha_n, \alpha_{n+1}, \ldots, \alpha_{n+\tau_n}, 0, \ldots, 0)$$

with $\sum_{k=n}^{n+\tau_n} \alpha_k = W_j$, where τ_n denotes the number of periods needed to process the workload when processing starts in period n. For every allocation we find $c(\boldsymbol{\alpha}, j) + v_i(t - 1, \boldsymbol{x} - \boldsymbol{\alpha})$ and choose the starting period n with the highest computed sum. In this way the near-optimal policy is computed.

For Model B we consider a simple policy that works the following way: Each job class is assigned a critical horizon, which is a period in the planning horizon. Whenever a job arrives, it is accepted if there is sufficient capacity to process it within the critical horizon of the job class. The motivation for this criteria is to avoid using capacity on low-profit jobs, as long as there is a reasonable probability that it can be used to satisfy high-profit jobs. We consider two allocation methods, FIFO and minimum cost(MC). The MC allocation is an allocation that minimizes holding and penalty costs, within the critical horizon.

The critical horizon policy is optimized in the same way as the critical level policy for Model A. We also consider a selective policy and a non-rationing policy.

The performance of the heuristic policies are explored for small-scale numerical problems, where the computation of the near-optimal policy(BSP) is possible.

In general the performance is good. The average profit for the simple policy (FIFO) is 98.2% with a worst case performance of 84.7%. The performance of the simple policy(MC) is on average 94.2% with a worst case performance of 82.6%. The average performance of the selective policy is 90.9% with a worst case performance of 72.9%, and finally the average performance of the non-rationing policy is 81.9% with a worst case performance over the investigated examples of 51.2%. The FIFO allocation dominates the MC allocation, on average, even when holding cost is as high as 5% per period. Thus, the main objective is to process everything as soon as possible in order to maximize throughput, regardless of the holding costs.

3.5 Conclusion

In this section we have investigated a periodic make-to-order system with limited capacity where jobs of different profitability arrive in a stochastic manner. We focus on systems where the average demand for capacity exceeds the available capacity which leads to situations where jobs must be rejected. By using simple or near-optimal rationing policies we can significantly increase total expected profit compared with a non-rationing approach. In our numerical tests we have found examples where profits are almost doubled by the use of rationing policies.

The derivation of the near-optimal policy is only computationally tractable for smaller problems and therefore we consider simple policies whose performance is shown, numerically, to be a few percent below the near-optimal policy. The examples investigated are very small indeed, but we expect the simple policies to be good also for problems of a more realistic size, where the computation of the near-optimal policy is intractable. The simple policies are characterized by their simplicity and their small number of policy variables. For larger problems we therefore suggest to use simulation for evaluation of policies and then perform a local search to find good values of the policy variables.

4 The Joint Replenishment Problem

4.1 Introduction

In this section we study the stochastic joint replenishment problem, i.e. the problem of coordinating an inventory system with several items, all replenished by the same supplier. What characterizes the problem is that the cost of ordering consists of a major ordering cost plus a minor ordering cost for every item participating in the order. The problem can be found in many settings: Typically, a supplier offers a range of products which all may come in several different shapes, colors, sizes or configurations. For such situations, the major ordering cost consists primarily of costs of administration and transportation, while the minor ordering cost consists of handling costs associated with the delivery of a specific item.

Another example described by the problem is the following: A company sells a product at several locations, all replenished by the same supplier. The replenishment of the product at all locations can be coupled into one order that arrives at a transshipment center or a break-bulb point, after which it is distributed to the respective locations. The major ordering cost is the cost of transportation between the supplier and the transshipment center, and the minor ordering cost is the cost of the last part of the transportation to the specific locations. In both problems, the replenishment of the items should be coordinated in order to utilize the economies of scope. The literature has suggested several coordination policies, of which in particular two classes of policies, the can-order policy (originally suggested by Balintfy [9]) and the periodic replenishment policy (introduced by Atkins and Iyogun[3]), have received considerable attention. Under the regime of a can-order policy all items follow an (s, c, S) policy: When the inventory position is below the must-order level s, an order is placed to bring the inventory position up to S. Moreover, every item has a can-order level c. Whenever another item has reached its must-order level, any item with inventory position at or below its can-order level is included in the order.

A good periodic replenishment policy is suggested by Viswanathan [57], who analyses a periodic-review $P(s, S)$ policy. The inventory system is reviewed with an interval of t time units. At every review an (s, S) policy is applied to every item, such that any item with inventory level below s is included in the order. The review interval t is a policy variable, which must be the same for all items.

In accordance with the numerical results of Viswanathan [57], it seems to be generally believed that the periodic replenishment policy performs better than the can-order policy, as for example expressed in the well-known textbook by Silver, Pyke and Peterson [49]. In this section we demonstrate that this conclusion does not hold in general. First of all, we suggest a new method for computing can-order policies, which improves the general performance of the can-order policies. Secondly, we demonstrate that which of the two classes to prefer, depends in particular on the underlying demand process. Inventory systems facing steady demand with small variation will do fine with a periodic replenishment policy, while systems characterized by more irregular demand sizes, are better controlled by a can-order policy, which can react fast to sudden changes in the inventory position.

We proceed as follows: In the next section we provide a literature review and discuss the traditional method for computing can-order policies. In Section 4.3 we present the method for improving the can-order policy called the compensation approach. This method is applied to a can-order policy for a continuous-review system in Section 4.4 and a can-order policy for a periodic-review system in Section 4.5. Finally, in Section 4.6 we give our conclusions. The content is based on Melchiors [33] and Johansen & Melchiors [29], which are referred to for detailed analysis and numerical results.

4.2 Background

We consider an inventory system with n items facing stochastic demand. Besides holding and backorder costs there is a fixed cost of ordering, consisting of a major ordering cost K, plus a minor ordering cost k_i for every item i participating in the order. Since all demands are backordered, and thus satisfied eventually, the variable ordering cost is not included in the model.

A way to model this problem is to formulate an n-dimensional Markov chain, and use a value-iteration algorithm to find the optimal joint replenishment policy. Such an approach is, however, computationally intractable even for small-scale problems, and, moreover, the optimal policy has a non-simple structure (Ignall [26]), which means that the implementation of the policy will be very cumbersome. Ohno et al. [42, 41] present an improved policy-iteration algorithm for finding the optimal problem. While they succeed in finding the optimal policy, they do not overcome the "curse of dimensionality", and their approach can, in reality, be used for systems with 2–4 items only.

Another approach is to solve n single-item problems independent of each other, neglecting the correlation between the items. This approach works if the major cost K is very small (and is indeed optimal if $K = 0$), but otherwise better methods are needed.

Silver [48] introduces the principle of decomposition to model the correlation between the items. The idea is to decompose the original problem into n sub-problems, one for each item. Item i has normal replenishment opportunities with major and minor ordering cost $K + k_i$ occurring whenever the inventory level reaches s, and discount opportunities with only minor ordering cost k_i, whenever another item places a normal order. The process of discount opportunities is in general very complicated and moreover, not independent of the demand process for item i. Silver suggests to approximate this process by a Poisson process with rate μ_i, which is assumed to be independent of item i. This facilitates a simple analysis of the model. Moreover, Zheng [62] proves that the can-order policy is optimal for a single-item inventory system with Markovian discount opportunities and Poisson demand. The rate μ_i is calculated based on the rates, β_j, of order placements from other items. In each iteration of the optimization algorithm, all single-item problems are solved, after which the rates μ_i are updated. This is repeated until the optimal policies converge or start cycling.

Silver applies this to a continuous-review model with Poisson demand and uses a simple method to determine the values of the policy variables. Federgruen, Groenevelt & Tijms [15] also consider a continuous-review model but assume compound Poisson demand and apply a policy-iteration algorithm to find better can-order policies. Both papers are based on the same principle of decomposition.

Van Eijs [55] argues that the assumption of Poisson discount opportunities leads to poor results, and suggests a can-order policy where the can-order level c is always equal to

$S-1$, for inventory systems where the major ordering cost is high, compared with the average minor ordering costs. For such a policy, whenever an item places an order, all other items join the order. He minimizes holding and ordering costs subject to a service level constraint, and finds the optimal $(s, S-1, S)$ policy. A disadvantage of this approach is that all items have to follow a $(s, S-1, S)$ policy, which means that it is not suitable for problems where some items have high minor ordering cost and others have low minor ordering cost.

Schultz & Johansen [46] show by simulation that the empirical waiting times between successive discount opportunities do not appear to come from an exponential distribution. They formulate a model where the time between two consecutive discount opportunities for each item i is Erlang-r distributed, which appears to give a better fit. The shape parameter r is found by simulation. Some of the discount opportunities refer to discount opportunities generated by the item itself and are modelled as fictitious. However, it is evident that the fictitious discount opportunities are not independent of item i as assumed.

The above mentioned references all assume that demands for different items are independent. Liu and Yuan [32] analyse a can-order policy for a system where the demand for different items are correlated.

Periodic-review policies were first analysed by Atkins & Iyogun [3], who suggest a modified periodic-review policy (MP) where each item i order up to R_i each time the inventory is reviewed. The review interval T_i is restricted to be an integer multiple of a base period. This policy is dominated by the $P(s, S)$ policy of Viswanathan [57] mentioned in our introduction. Pantumsinchai [43] analyses a QS policy (Originally suggested by Renberg and Planche [45]) where an order is placed when the total system demand since the last order placement exceeds Q. Item i orders up to S_i. The performance of the QS policy is comparable to that of the MP policy. Forsberg [16] considers a generalization of this policy, where each item moreover has a must-order point, which can trigger an order.

Models where the joint replenishment problems is combined with routing/transportation problems are considered by Viswanathan and Mathur [58] and Qu, Bookbinder and Iyogun [44]. The deterministic joint replenishment problem has also received considerable attention. Procedures for determining optimal and heuristic policies are described in Goyal and Satir [18]. More recently, Wildeman, Frenk and Dekker [60] described an efficient method for determining the optimal solution.

4.3 The compensation approach

In particular in systems with high major ordering cost, the can-order policy seems to have a poor performance. A possible explanation for this is that the item considering to place an order balances the sum of the major and the minor ordering cost with the expected shortage cost for the item. Whenever an item places an order, other items receive a discount opportunity which may reduce their costs. However, in the original decomposition, the item placing the order does not take this into consideration when deciding whether to place an order or not. By calculating δ_j, the average benefit per discount opportunity for all items j, we can compensate an item i placing an order by only charging $K - \Delta_i$, where

$$\Delta_i = \sum_{j \neq i} \delta_j.$$

In this way, the implied effects of placing an order are included when deciding when to place an order. This leaves the question of how to find δ_j. The benefit of a discount opportunity changes with the inventory position of an item. If no demands have occurred since the last replenishment, we are not interested in joining a order, and the value is zero. If we are just about to place an order ourselves, the value of the discount opportunity is close to $K - \Delta_i$. We use the *relative values* of a policy to determine δ_j. The relative value $v(x)$ is the difference in expected long-run total cost of having an inventory position of x rather than the order-up-to level S. The relative values are typically used for optimization procedures, but here we use them to obtain information about the value of a discount opportunity. In

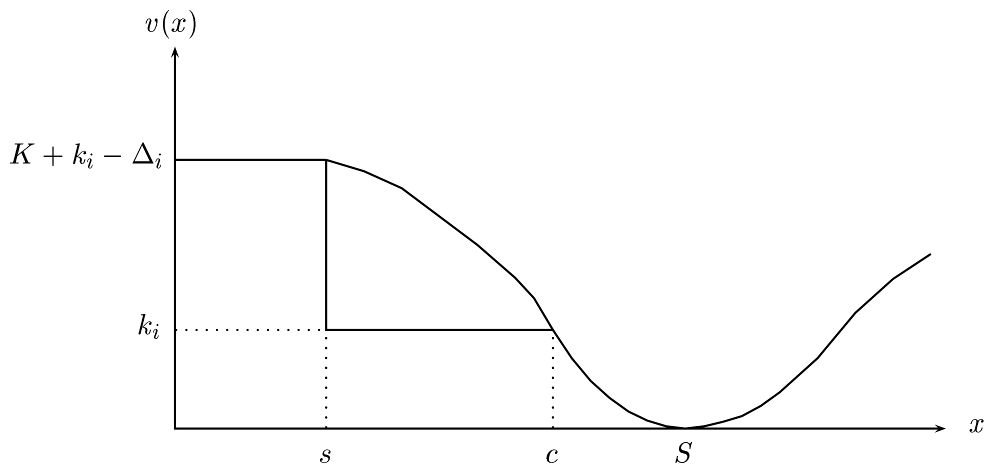

Figure 4: The relative values $v(x)$ as a function of the inventory position x for the optimal policy.

Figure 4 we have depicted the relative values of an optimal policy. An item can benefit from discount opportunities occurring while its inventory position is below c. If a discount opportunity occurs while the system is in state x, we will accept it (since $x \leq c$), and our inventory position will rise to S. Since the relative value of state S (by definition) is 0, the benefit of the discount opportunity is the positive amount $v(x) - k$. The expected benefit

δ_j is found by finding the expected value of $v(x) - k$ per generated discount opportunity. In the next sections we apply this approach to a continuous-review and a periodic-review can-order policy.

4.4 A continuous-review can-order policy

In this section we assume that demand is reviewed continuously and that orders can be placed at any point in time. Other methods for computing can-order policies are based on this assumption and this allows us to compare the compensation approach with these. We use the principle of decomposition with compensation, and thus first show how to solve a single-item problem with Markovian discount opportunities and derive δ_j and β_j, and then show how to combine these in the decomposition algorithm.

4.4.1 Modelling

The inventory system consists of n items, where item i faces Poisson demand with rate λ_i. Holding costs are charged at a rate $h_i > 0$ per unit per unit time. Demand not satisfied immediately is backlogged and shortage costs are charged at a rate of p_i per unit per unit time. Each unit backlogged moreover incurs a time independent cost of π_i. There is a constant lead time of L_i time units. The fixed ordering cost consists of a major cost K, and a minor cost k_i for each item i joining the order.

Let the decision epochs be the demand epochs and the arrivals of the discount opportunities. The state of the system at each decision epoch is described by the inventory position x. Since we have two independent Poisson processes with rate λ_i and μ_i the merged process is a Poisson process with rate $\lambda_i + \mu_i$. The probability of a decision epoch being generated by a demand [discount opportunity] is $\lambda_i/(\lambda_i + \mu_i)$ [$\mu_i/(\lambda_i + \mu_i)$]. At each decision epoch we can decide to place an order.

We can now find the transition probabilities and the expected cost charged to a decision epoch (the lead time is incorporated by a shift in time).

Based on these we can find the long-run average cost per unit time of a given policy. The inventory process is regenerative with regeneration point S. Define a cycle to be the time between two regeneration epochs. By renewal-reward theory we can then find the long-run average cost of a policy, by dividing the expected cost incurred in a cycle by the expected cycle length (see e.g. Tijms [53]). Let $z(x)$ be expected cost incurred up to the next regeneration point starting in state x, and let $y(x)$ be the expected time until we reach the next regeneration point, starting in state x. The functions $z(x)$ and $y(x)$ are easily computed by a recursion scheme. Let g_i be the long-run average cost for item i of using policy (s, c, S),

$$g_i = \frac{z(S)}{y(S)}. \tag{4}$$

Next, we need to find the expected rate β_i of order placements from item i. The probability that the item places an order in a cycle equals $(\frac{\lambda_i}{\lambda_i+\mu_i})^{c-s}$, the probability of $c-s$ consecutive demands with no discount opportunity in between. To find the average number of orders placed by the item per unit time, we divide by the expected length of the cycle

$$\beta_i = \left(\frac{\lambda_i}{\lambda_i + \mu_i}\right)^{c-s} \Big/ y(S). \tag{5}$$

Now let us calculate the expected gain of a discount opportunity. Let

$$v(x) = z(x) - g_i y(x)$$

be the relative value of inventory position x, with g_i being the cost of the policy found by (4).

Let J be the random variable denoting the number of demands occurring from the time when the can-order level is reached until the first discount opportunity occurs. The probability of $J = j$ is the probability of j consecutive demands followed by a discount opportunity. The expected gain of the discount opportunity is found by conditioning on the value of J.

$$V = \frac{\mu_i}{\lambda_i + \mu_i} \sum_{j=0}^{c-s-1} \left(\frac{\lambda_i}{\lambda_i + \mu_i}\right)^j \Big(v(c-j) - k_i\Big).$$

We can only benefit from one discount opportunity per cycle and therefore, to find the expected gain δ_i of the discount opportunity per discount opportunity, we divide by the expected number $\mu y(S)$ of discount opportunities occurring in a cycle, i.e.

$$\delta_i = \frac{V}{\mu_i y(S)}. \tag{6}$$

Naturally, we are interested in optimizing the performance of the single-item system. We can either use a policy-iteration algorithm to optimize the policy variables or use the algorithm of Zheng [62] to find the optimal can-order policy with the major ordering cost specified as $K - \Delta_i$.

The decomposition procedure works as follows: We initialize by setting $\delta_j = 0$ and β_j to a small but positive amount for all j. In each iteration of the procedure, we solve the single-item problem for each item i with values of Δ_i and μ_i given by

$$\Delta_i = \sum_{j \neq i} \delta_j$$

and

$$\mu_i = \sum_{j \neq i} \beta_j.$$

After solving the problem for item i, the values of δ_i and β_i are updated by (6) and (5). This iteration is repeated until the policy variables are either unchanged or start cycling between two or several solutions. This typically happens within 10-50 iterations. In the event of cycling the best policy is found by evaluating the policies in the cycle and choosing the one with the lowest cost. An estimate of the total system cost is found by $\sum_i g_i$. Since the decomposition procedure is based on approximations, we also evaluate the optimal policy by simulation, to give a fair comparison with other policies.

4.4.2 Numerical results

We first test the compensation can-order policy on a set of data provided by Atkins and Iyogun [3]. The $P(s, S)$ policy has lower cost when the major ordering cost is high, but when K is low the compensation can-order policy is the policy with the lower cost. The results show that the compensation can-order policy is in general better than the FGT can-order policy. In 23 out of 25 investigated cases, the cost of the compensation can-order policy is lower than those of the FGT can-order policy, in some cases by up to 25%. Compared with the Erlang can-order policy (Schultz & Johansen [46]) the compensation can-order policy is better in cases with a low major ordering cost, whereas the Erlang can-order policy gives lower costs on examples with high major ordering cost. However, these costs are still higher than those of the $P(s, S)$ policy.

Secondly, we consider a set of examples, introduced by Viswanathan [57], where the holding and penalty cost have been increased by a factor of 100. This corresponds to an increase in the demand rate and thereby the demand variation by a factor of 100. Viswanathan [57] concludes that even in these cases the $P(s, S)$ policy dominates the FGT can-order policy. We find that the compensation can-order is better than the $P(s, S)$ policy in all of the examples, including the examples where K is high. The average cost difference is only 0.75%, but the conclusion that the $P(s, S)$ policy dominates the can-order policy for these high variation examples is not true.

4.5 A periodic-review can-order policy

Next, we analyse a can-order policy for a periodic-review inventory system. The assumption of continuous review can be justified by the recent developments of access to point of sale information and electronic data interchange (EDI). However, while the access to information may be continuous, there is often only a limited number of replenishment opportunities, for example, once or maybe twice a day. For many systems a periodic-review model will therefore provide a better representation.

4.5.1 Modelling

We assume a periodic system where every period typically may represent one day. We assume that period demands are stochastic and stationary variables and let λ_i denote the probability of a positive demand, and $\phi_i(x)$ the probability mass function of the demand size for item i in any period.

Demand not satisfied immediately is backordered, incurring a fixed cost π_i per unit backordered. At the end of every period, a holding cost of h_i per unit in stock and a time dependent cost p_i per unit backordered is incurred. There is a constant lead time of L_i periods for each item i. The fixed cost of ordering consists of a major cost K plus a minor ordering cost k_i for every item i participating in the order.

For the periodic model several items can trigger an order at the same time, and therefore we need another approach to model the discount opportunities. We approximate the process of discount opportunities by a Bernoulli process with outcome 1, if a discount order opportunity occurs and 0 otherwise. Successive outcomes are assumed to be independent of each other, which means that the probability of a discount opportunity does not depend on discount opportunities of the past, and the process can therefore be seen as a discrete version of the Poisson process.

We employ the principle of decomposition with compensation, and solve first the single-item problem, similarly with the continuous-review case. Let μ_i be the probability that at least one other item, but item i, places an order in a period, and let Δ_i be the expected value of other items' benefit of an discount opportunity generated by item i.

Let the state, x_n, of the system at the end of period n, be the inventory position. Under the regime of a can-order policy $\{x_n\}_{n\geq 0}$ is a Markov chain with regeneration point S. Let the inventory cycle be the time between two consecutive visits in the regeneration point. As for the continuous-review policy, we can find the long-run average cost of a policy as the expected cost incurred during a cycle divided by the expected length of a cycle.

Let us first state the transition probabilities and the expected cost function, excluding costs of ordering. In any state x where we do not place an order, we jump to state $x - j$ with probability $\phi(j)$ for all $j \geq 0$. However, if we place or join an order, we jump immediately to state S.

The cost assigned to an inventory position x in period τ, is the expected cost incurred in period $\tau + L + 1$, given that the inventory position is x by the end of period τ. The first period that can be influenced by our decision is period $\tau + 1$ if the lead time is zero and period $\tau + L + 1$ if the lead time is L. Consequently, this shift in time assigns the relevant cost to each period. We find the expected cost by conditioning on the lead time demand and on the one-period demand.

The cost of placing an order is determined in the following way. If $s < x \leq c$ and a discount opportunity has occurred, we can join this, incurring only the minor ordering cost k_i. If $x \leq s$ and no other item is placing an order, the major ordering cost $K - \Delta_i$ is incurred. However, if several items are placing an order simultaneously, which is possible in the periodic model, but not in the continuous model, they must share the major ordering cost

$K - \Delta_i$. We model this by dividing the major ordering cost with the expected number of items with inventory position below s.

Let $z(x)$ be the expected cost incurring until we reach the regeneration point next time, starting in state x. Similarly, let $y(x)$ be the expected time until the regeneration point is reached, starting in state x. We find $z(x)$ and $y(x)$ in a recursive manner, and obtain the cost for item i as

$$g_i = \frac{z(S)}{y(S)}.$$

The relative value of state $v(x)$ is defined as for the continuous-review system and is given by

$$v(x) = z(x) - g_i y(x)$$

We now derive β_i, the average number of orders placed per period, and δ_i, the expected benefit of a discount opportunity. Let $m(x)$ denote the expected number of times we visit a state x during an inventory cycle. We can enter a state x from state $x+j$ after a demand of j units, $j > 0$. However, after a period with zero demand, we will remain in state x, which will count as an additional visit. The expected number of orders placed per period, equals the expected number of times we visit a state x with $x \leq s$ per cycle, divided by the cycle length $y(S)$.

$$\beta_i = y(S)^{-1} \sum_{x=-\infty}^{s} m(x) \tag{7}$$

Under the assumption that the discount opportunity process is independent of the item under consideration, the expected benefit per inventory cycle can be found by conditioning on the inventory position x. The expected number of periods where we are in state x is $m(x)$ and therefore the expected gain per cycle is

$$V = \sum_{x=s+1}^{c} m(x)(v(x) - k).$$

We can only benefit from one discount opportunity per cycle, and, therefore, to find the expected gain δ_i of the discount opportunity per discount opportunity, we divide by $\mu_i y(S)$, the expected number of discount opportunities occurring in a cycle, i.e.

$$\delta_i = \frac{V}{\mu_i y(S)}. \tag{8}$$

We apply a tailor-made policy-iteration algorithm to find a near-optimal can-order policy for the single-item problem.

The decomposition procedure works as follows: Let (s_i^k, c_i^k, S_i^k) be the policy found for item i in the k'th iteration of the decomposition algorithm. The algorithm is initialized by setting $\delta_i = 0$ for all i. β_i is set equal to a small but positive amount. In the k'th iteration of the algorithm we perform the following for each item i: First, we compute Δ_i and μ_i by

$$\Delta_i = \sum_{j \neq i} \delta_j$$

and

$$\mu_i = 1 - \prod_{j \neq i} (1 - \beta_j)$$

Then we solve the single-item problem based on these values and update the values of β_i and δ_i by (7) and (8).

The algorithm terminates when $(s_i^k, c_i^k, S_i^k) = (s_i^{k-1}, c_i^{k-1}, S_i^{k-1})$, or when the algorithm starts cycling between a set of policies. The algorithm terminates in approximately 20-40 iterations. Due to the approximate nature of our approach the computed cost given by $\sum_i g_i$ is only an approximation. In order to compare the policy with other policies we simulate the found policy. This cost is denoted simulated cost. In case of cycling we choose the policy with the lowest computed cost.

4.5.2 Numerical results

We test the performance of the can-order policy on a 12-item example, and investigate the influence of different lead times, ordering cost, holding costs and shortage costs. Moreover, we investigate how the demand structure influences performance by considering two types of demand structure: First, a system where demand is fairly irregular, being $0, 1$, or 2, in most periods, with occasional highs of 15 units every once in a while. And, in contrast to this, a system with a more regular demand distribution with the demand size being 0,1, 2, or 3. In both systems all holding, penalty and minor ordering costs are equal for all items, and so are the lead times. The examples can be thought of as systems with 12 fairly identical products, perhaps only differing in their color or their taste. What differs is typically the demand rate. Some colors or variants may be more needed than others. We have chosen an example where the probability of a positive demand λ_i varies from 0.15 to 0.7 over the 12 items.

We compare the can-order policy with the $P(s, S)$ policy of Viswanathan [57] and with a periodic can-order policy calculated by the original decomposition approach, i.e. with compensation. This policy is denoted FGT.

The most important factor of influence is found to be the demand structure. The costs of the $P(s, s)$ policy is on average 10.3% higher than those of the compensation policy and the FGT policy has cost 1.3% higher in 27 examples with irregular demand sizes. The compensation policy has the lowest cost in all 27 examples. When demand is regular the performance of the $P(s, S)$ policy is much better. The two policies are equally good, while the FGT policy on average has costs 3% above the costs of the can-order and the $P(s, S)$

policy. The impact of the other factors are marginal for the examples with irregular demand. For the regular demand examples, we find that the periodic $P(s, S)$ policy is best when the major ordering cost is high and the can-order policy is best when the major ordering cost is low.

4.6 Conclusion

Our studies of the joint replenishment problem have revealed that the can-order policy by no means can be regarded as inferior to the periodic ordering policies. The strongest arguments for periodic ordering policies are that the assumption of continuous review is not realistic in practice, and that the can-order policy is not as good as the periodic ordering policies. We have shown that the can-order policy, if computed by the compensation approach, is much better than its reputation. By the compensation approach is it possible to obtain a much richer model formulation and the added complexity does not seem to increase computation times.

Moreover, we have suggested a periodic-review can-order policy. For this policy we have found that there can be significant cost savings, in particular for systems with irregular demand patterns.

5 Lost sales in multi–echelon inventory problems

5.1 Introduction

In the basic inventory model is it assumed that the demand and supply processes are external processes which cannot be controlled. There are, however, systems where this assumption is inadequate. Many retail chains consists of one (or several) central warehouses each serving a set of retailers in their geographical region. The decisions at the warehouse clearly influence the supply process at the retailers, and this should be taken into consideration when deciding how to control the inventory at the warehouse. This system is a two-stage or a two-echelon inventory system, but it is easy to imagine distribution systems with several stages.

By optimizing the supply chain as a whole, it is possible to obtain an improved performance compared with a system where each stage of the supply chain is optimized as an independent single-stage system. This is the fundamental observation behind what is known as Supply Chain Management. In an efficient supply chain, the parties share information about demand, supply and even costs, in order to optimize the performance of the supply chain.

Evaluation and optimization of control policies for such inventory systems have attracted massive interest in the literature. See, for example, Axsäter [4] for an overview. There are in principle two types of control policies: centralized policies and decentralized policies. Both types of policies are computed using information of the entire supply chain. However, under the regime of a decentralized policy, ordering decisions at an installation can depend only on the inventory position at the local retailer, while a centralized policy can take information of inventory positions at all installations into account. While the centralized policies in general yield better results, the decentralized policies are easier to implement in practise.

In this section we analyse a model for a one-warehouse, multiple-retailer inventory system. Demand occurs only at the retailers and follows independent Poisson processes. The retailers replenish their stocks from the central warehouse, which, in turn, replenishes its stock from an outside supplier. All lead times are assumed to be constant. All installations use decentralized $(S-1, S)$-policies with continuous review. It is assumed that backlogging of customer demand is not allowed, which means that all demands not satisfied immediately are lost.

In the existing literature dealing with multi-echelon inventory control, the prevalent assumption is that complete backlogging of orders is allowed in case of stockouts. For example, Axsäter [5] shows how to exactly evaluate the performance for different (R, nQ)-policies when the retailers face compound Poisson demand and inventories are continuously reviewed. Cachon [10] gives an exact method for the periodic review case with identical retailers.

In some situations the assumption of complete backlogging may not be so realistic. For example, it may be more representative to model stockouts as lost sales when retailers are in a competitive market and customers can easily turn to another firm to purchase the good. Supermarket and grocery chains are obvious examples of systems where customers are lost rather than backlogged, if they cannot have their demand filled immediately.

The only other paper, to our knowledge, considering lost sales in a multi–echelon environment is Nahmias and Smith [40]. Their model differs from ours in several important aspects. First, they consider periodic review batch order policies. The model considered is more general since they consider partial lost sales. This means that, with probability u, demand not satisfied immediately, is lost, and with probability $1-u$ it is satisfied later by a special order. For the model to be tractable, they assume instantaneous deliveries from the warehouse to the retailers.

For single-echelon inventory models the lost sales assumption is more common. The exact cost for a single-level inventory system facing Poisson demand and fixed lead times was first given by Hadley and Whitin [22]. Smith [50] demonstrates how to evaluate and find optimal $(S-1, S)$-polices for an inventory system with zero replenishment costs and general distributed stochastic lead times. Hill [25] shows that for this model the $(S-1, S)$-policy is not the optimal policy, and Johansen [28] suggests a modified base-stock policy,

which is better than the pure base-stock policy.
Our analysis departs in one of the most widely known multi-echelon inventory models, the METRIC model developed by Sherbrooke [47]. In its original setting, it is assumed that stockouts at the retailers are completely backlogged. We demonstrate how the METRIC model can be modified to handle the lost sales case. Our approach gives an approximate model, which is quite simple and efficient from a computational point of view. Simulation experiments indicate that the performance is very good. The section is based on Andersson and Melchiors [2] which are referred to for a detailed analysis and numerical results.

5.2 Modelling and optimization

The inventory system under consideration consists of one central warehouse and N retailers, facing Poisson customer demand with rate λ_i. No backlogging is allowed at the retailers. Consequently, a customer that arrives at a retailer out of stock, will be a lost sale for the retailer. When stockouts occur at the warehouse, all demands from the retailers are fully backlogged and the backorders are filled according to a FIFO-policy. The transportation time between the warehouse and a given retailer is assumed to be constant as well as the transportation time from the external supplier to the warehouse. The cost of a replenishment is assumed to be zero or negligible compared to the holding and stockout costs. The external supplier is assumed to have infinite capacity, which means that the replenishment lead time for the central warehouse is constant. All installations use $(S-1, S)$ base-stock policies with continuous review, with S_0 being the base-stock level of the warehouse and S_i the base-stock level of retailer i. Units held in stock at both the warehouse and the retailers incur holding costs per unit and time unit. Moreover, a fixed penalty cost per lost customer is incurred at the retailers.
We present a model for the considered inventory system which can be used to evaluate the long-run average cost for different policies within the class of $(S-1, S)$-policies. The objective is to find the policy that minimizes the long-run average cost.
For the backorder case the exact cost of such a system can be derived by observing that any unit ordered by a retailer i, is used to fulfill the S_ith demand. The cost can then be derived by conditioning on the arrival time of the S_ith demand (which is Erlang distributed) and the arrival of the ordered unit (see Axsäter [4]). In a lost sales environment the corresponding observation is that any unit ordered by a retailer i, is used to fulfill the $S_i + X_i$th demand, where X_i is a random variable denoting the number of lost sales incurred at the retailer during the replenishment lead time. X_i is obviously very complex and we therefore focus on a heuristics rather than on the exact solution.
The analysis of this system has many similarities with the analysis in Sherbrooke [47]. The lost sale case is, however, more complicated. In the backorder case, all customers arriving

at the retailers generate demands at the warehouse immediately at the arrival epoch, since all retailers use continuous review $(S-1, S)$-policies. Consequently, the warehouse faces a Poisson process with intensity $\lambda_0 = \lambda_1 + \lambda_2 + \cdots + \lambda_N$. For the lost sales case this is not true. When backordering is not allowed, customer demands can be lost due to stockouts at the retailers. Therefore the demand at the warehouse is not a Poisson process. Moreover, the demand rate and thereby the cost at the warehouse depend on the order-up-to levels of the retailers.

We first show how to evaluate the costs at the retailers given a certain replenishment lead time provided by the warehouse. We then show how to calculate the costs at the warehouse given the demand intensity from the retailers. Finally, we introduce an iterative procedure, from which we obtain the average costs of the inventory system.

Given a replenishment lead time, $\bar{L}$, the number of outstanding orders towards the central warehouse at a retailer, is equal to the occupancy level in an $M/G/S/S$ queue, with S servers, each with general distributed service times and no queueing allowed. If the service times are independent random variables with mean $\bar{L}$, we can find the steady-state distribution for the occupancy level by Erlang's loss formula. The stochastic lead times are evidently not independent in our case, but if we disregard this correlation we can approximate the number of outstanding orders with a Poisson distribution. This is the idea behind the METRIC approximation. Based on this, Smith [50] derives the exact cost of a $(S-1, S)$ lost sales single-stage inventory system with generally distributed lead times. He also shows that the cost is a convex function of the S and therefore the optimal value of S can be found by local search.

For each retailer i and an order-up-to level S_i, we let $q_i^{S_i}(S_i)$ denote the probability that a demand is lost and thereby not ordered at the warehouse. The demand rate at the warehouse, given the order-up-to levels of the retailers, is therefore

$$\Lambda = \sum_{i=1}^{N} \lambda_i (1 - q_i^{S_i}(S_i)) \tag{9}$$

The demand process is not a Poisson process: If, for example, the base–stock level at a retailer is one, the smallest interval between two successive demands from that retailer will be the retailers lead time. We ignore this and approximate the demand process at the warehouse with a Poisson process with mean Λ. The derivation of the cost at the warehouse is identical to that for the pure backorder model. For an order-up-to level at the warehouse we can moreover calculate the average delivery delay, due to stockouts at the warehouse by Little's formula. This plus the fixed transportation time to a retailer, gives the replenishment lead time, $\bar{L}$.

We can now establish the solution procedure. Let $TC^*(S_0)$ be the minimal cost of the inventory system given a value of S_0. $TC^*(S_0)$ is not convex in S_0 and therefore we perform

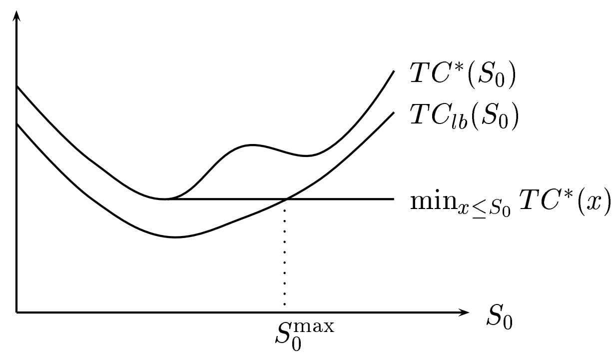

Figure 5: Illustration of the abortion criteria. The search for the optimal S_0 is aborted at $S_0^{\max}$.

an enumeration over S_0 starting with $S_0 = 0$, bounded from above by an abortion criteria. We show that $TC_{lb}(S_0)$ is a lower bound for $TC^*(S_0)$ and that $TC_{lb}(S_0)$ furthermore is convex in S_0. The search for the optimal value of S_0 is aborted when

$$\min_{x \le S_0} TC^*(x) \le TC_{lb}(S_0)$$

as illustrated in Figure 5.

For each value of S_0, we calculate the value of $\bar{L}_i$ and optimize the base-stock levels, S_i, for all retailers i. We now update the new value of Λ by (9) and repeat the calculations until all values of S_i are unchanged. This usually happens in 2 or 3 iterations.

5.3 Numerical results

Since our heuristic is based on approximations it is of interest to simulate the system to see how good they are. We consider 36 different test examples with five identical retailers. The method performs rather well for all the considered problems. It seems that we mostly tend to underestimate the total cost, especially in the problems with high stockout cost at the retailers. This is due to the METRIC approximation, where the stochastic lead times are replaced by their averages, when evaluating the costs for the retailers. On average the method underestimates the total cost with 1.1 %.
Since the examples investigated have identical retailers we can find the optimal base-stock policy for each example by extensive simulation. We can then compare the cost of the optimal policy with the cost of the policy suggested by the heuristic. The increase in cost by using the policies obtained by our method is only 0.40 %, on average. In 14 of the 36 examples the policy suggested by the heuristic is optimal. In 16 of the 22 problems where we fail to find the true optimal policy the method merely underestimates the order-up-to level at the warehouse by a single unit. In one problem the warehouse order-up-to level

is underestimated by two units. In the other 5 problems where the optimal policy is not found, the method tends to allocate more stock to the retailers and less stock to the warehouse than it is optimal from a cost perspective.

5.4 Conclusion

In conclusion, the method developed is very efficient and simple. Numerical results also indicate that the performance is quite good.
To the best of our knowledge, we are the first to deal with lost sales in a continuous review multi-echelon inventory setting. Moreover, the original backorder METRIC-model [47] is one of the most widely used multi-echelon inventory models. Our lost sales generalization makes the policy evaluation a bit more complex, since we have to use an iterative procedure to obtain the cost. Still, the model is rather simple and easy to implement. Moreover, in many practical situations the lost sales assumption is a reasonable way to model stockouts. Therefore our technique is also relevant for practitioners.

6 Conclusion

Mathematical models for inventory systems have been known for about a century. Earlier models were limited by little data availability and only very little computer power, if any. Today, there is easy access to data, and ample computer power. This means that we can compute policies for much more complicated inventory systems, and thereby obtain a better representation of the supply chain. This thesis consists of several small steps towards efficient supply chain management. Let us shortly sum up the contributions: In Melchiors, Dekker and Kleijn [37], Melchiors [35] and Melchiors [36] we analyse and discuss rationing policies for inventory systems with several demand classes. This area has not received much attention in the literature and since companies today are creating closer relationships with their customers, efficient control policies are needed. In Melchiors [34] we apply the rationing policies to a make-to-stock system with several demand classes. Melchiors [33] and Johansen & Melchiors [29] present a new approach for computing can-order policies for the joint replenishment problem which is applied to both a continuous-review and a periodic-review inventory system. Our model gives a better system representation which leads to lower system costs. In Andersson and Melchiors [2] we compute base-stock policies for a two-echelon inventory system with lost sales. Most multi-echelon models are based on the assumption of full backlogging, which is unrealistic in many settings.

As mentioned earlier our ambition has not been to provide a model for a full-scale supply chain. We have rather chosen to focus on interesting sub-systems of the supply chain. We have attempted to focus on the essential characteristics of a system, but the methods will typically need customization to fit a real system. In many cases our analysis will provide a solid foundation for further analysis. If the deviation from our model is significant, our analysis may, however, not be applicable. But the structure of the policies suggested will still be useful, as will our recommendations for when these policies should be preferred.

Today, the distance between what practitioners want and what research can provide is as small as ever. However, the gap between the models that practitioners use and the models researchers develop is still surprisingly significant. To narrow this gap is an important step towards efficient supply chain management, and the real challenge for practitioners as well as researchers. We hope that this thesis is a step in this direction.

References

[1] American Airlines Annual Report . The art of managing yields. Technical report, 1987.

[2] J. Andersson and P. Melchiors. A two-echelon inventory model with lost sales. *International journal of Production economics*, 69(3):307–315, 2001.

[3] D. Atkins and P.O. Iyogun. Periodic versus 'can–order' policies for coordinated multi–item inventory systems. *Management Science*, 34:791–796, 1988.

[4] S. Axsäter. Continuous review policies for multi–level inventory systems with stochastic demand. In S.C. Graves, A.H.G. Rinnooy Kan, and P. Zipkin, editors, *Handbooks in OR & MS, Vol. 4*, pages 175–197. Elsevier Science Publishers B.V., North–Holland, 1993.

[5] S Axsäter. Exact analysis of continuous review (R, Q)–policies in two–echelon inventory systems with compound Poisson demand. *Operations Research*, 48:686–696, 2000.

[6] S. Axsäter. *Inventory Control*. Kluwer Academic Publishers, 2000.

[7] 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.

[8] N. Balakrishnan, V. Sridharan, and J.W. Patterson. Rationing capacity between two product classes. *Decision Sciences*, 27(2):185–214, 1996.

[9] J.L. Balintfy. On a basic class of multi-items inventory problems. *Management Science*, 10(2):287–297, 1964.

[10] G.P. Cachon. Exact evaluation of batch-ordering inventory policies in two–echelon supply chains with periodic review. *Operations Research*, 49(1):79–98, 2001.

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

[12] 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.

[13] 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.

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

[15] A. Federgruen, H. Groenevelt, and H.C. Tijms. Coordinated replenishments in a multi–item inventory system with compound Poisson demands. *Management Science*, 30:344–357, 1984.

[16] R. Forsberg. A joint replenishment strategy for inventory systems with Poisson demands. *Lund University, Sweden*, 1996.

[17] 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.

[18] S.K. Goyal and A.T. Satir. Joint replenishment inventory control: deterministic and stochastic models. *European Journal of Operational Research*, 38:2–13, 1989.

[19] S.C. Graves, A.H.G. Rinnooy Kan, and P. Zipkin. *Logistics of production and inventory, Handbooks in OR & MS, Vol. 4.* Elsevier Science Publishers B.V., 1993.

[20] 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.

[21] A.Y. Ha. Stock rationing in an $M/E_k/1$ make-to-stock queue. *Management Science*, 46(1):77–87, 2000.

[22] G. Hadley and T.M. Whitin. *Analysis of Inventory Systems.* Prentice–Hall, Englewood Cliffs, NJ, 1963.

[23] L.C. Hendry and B.G. Kingsman. Job release: Part of a hierarchical system to manage manufacturing lead times in make-to-order companies. *Journal of the Operational Research Society*, 42(10):71–883, 1991.

[24] L.C. Hendry and B.G. Kingsman. Customer enquiry management: Part of a hierarchical system to control lead times in make-to-order companies. *Journal of the Operational Research Society*, 44(1):61–70, 1993.

[25] R.M. Hill. On the suboptimality of $(S-1, S)$ lost sales inventory policies. *International Journal of Production Economics*, 59(1-3):377–385, 1999.

[26] E. Ignall. Optimal continuous review policies for two product inventory systems with joint setup costs. *Management Science*, 15:278–283, 1969.

[27] S.G. Johansen. Transfer pricing of a service department facing random demand. *International Journal of Production Economics*, 46–47:351–358, 1996.

[28] S.G. Johansen. Pure and modified base–stock policies for the lost sales inventory system with negligible set–up costs and constant lead times. *Int. J. Production Economics*, 71:391–399, 2001.

[29] S.G. Johansen and P. Melchiors. The can-order policy for the periodic-review joint replenishment problem. *Department of Operations Research, University of Aarhus*, 2001.

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

[31] M.J. Kleijn. *Demand differentiation in inventory systems.* Thesis Publishers, Amsterdam, 1998.

[32] L. Liu and W.-M. Yuan. Coordinated replenishments in inventory systems with correlated demands. *European Journal of Operational Research*, 2000.

[33] P. Melchiors. Calculating can-order policies for the joint replenishment problem by the compensation approach. *Department of Operations Research, University of Aarhus*, 2000.

[34] P. Melchiors. Rationing of a congested multi-period make-to-order system. *Department of Operations Research, University of Aarhus*, 2001.

[35] P. Melchiors. Rationing policies for an inventory model with several demand classes and stochastic lead times. *Department of Operations Research, University of Aarhus*, 2001.

[36] P. Melchiors. Restricted time remembering policies for the inventory rationing problem. *Department of Operations Research, University of Aarhus*, 2001.

[37] P. Melchiors, R. Dekker, and M.J. Kleijn. Inventory rationing in an (s, Q) inventory model with two demand classes and lost sales. *Journal of the Operational Research Society*, 51(1):111–122, 2000.

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

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

[40] S. Nahmias and S.A. Smith. Optimizing inventory levels in a two-echelon retailer system with partial lost sales. *Management Science*, 40:582–596, 1994.

[41] K. Ohno and T. Ishigaki. A multi-item continuous review inventory system with compound Poisson demand. *Mathematical Methods of OR*, 2001.

[42] K. Ohno, T. Ishigaki, and T. Yoshii. A new algorithm for a multi-item periodic review inventory system. *Mathematical Methods of OR*, 39:349–364, 1994.

[43] P. Pantumsinchai. A comparison of three joint ordering inventory policies. *Decision Science*, 23:111–127, 1992.

[44] W.W. Qu, J.H. Bookbinder, and P. Iyogun. An integrated inventory–transportation system with modified periodic policy for multiple products. *European Journal of Operational Research*, 115:254–269, 1999.

[45] B. Renberg and R. Planche. Un modele pour la gestion simultanee des n articles d'un stock. *Revue fr. Inf. Rech. Op*, Serie Bleue(6):47–59, 1967.

[46] H. Schultz and S.G. Johansen. Can-order policies for coordinated inventory replenishment with Erlang distributed times between ordering. *European Journal of Operational Research*, 113:30–41, 1999.

[47] C.C. Sherbrooke. METRIC: A multi-echelon technique for recoverable item control. *Operations Research*, 16:122–141, 1968.

[48] E.A. Silver. A control system for coordinated inventory replenishment. *Int. J. Prod. Res.*, 12:647–670, 1974.

[49] E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling.* John Wiley & Sons, New York, 1998.

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

[51] S.R. Tayur, R. Ganeshan, and M. Magazine. *Quantitative models for supply chain management.* Kluwer Academic Publishers, 1999.

[52] 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.

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

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

[55] M.J.G. van Eijs. On the determination of the control parameters of the optimal can–order policy. *Mathematical Methods of Operations Research*, 39:289–304, 1994.

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

[57] S. Viswanathan. Periodic review (s, S) policies for joint replenishment inventory systems. *Management Science*, 43(10):1447–1454, 1997.

[58] S. Viswanathan and K. Mathur. Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. *Management Science*, 43(3):294–312, 1997.

[59] L.R. Weatherford and S.E. Bodily. A taxamony and research overview of perishable-asset revenue management: Yield management, overbooking and pricing. *Operations Research.*, 40:831–844, 1992.

[60] R.E. Wildeman, J.B.G. Frenk, and R. Dekker. An efficient optimal solution method for the joint replenishment problem. *European Journal of Operational Research*, 99:433–444, 1997.

[61] R.Q. Zhang and M.J. Sobel. Inventory policies for systems with stochastic amd deterministic demand. *Operations Research*, 49(1):157–162, 2001.

[62] Y.S. Zheng. Optimal control policy for stochastic inventory systems with Markovian discount opportunities. *Operations Research*, 42(4):721–738, 1994.

[63] P.H. Zipkin. *Foundations of Inventory Management.* Mc Graw Hill, 2000.

Inventory rationing in an (s, Q) inventory model with lost sales and two demand classes

Philip Melchiors* Rommert Dekker† Marcel J. Kleijn†

January 2000

Journal of the Operational Research Society 51(1) 111–122
Reprinted with permision of Palgrave Publishers Ltd.

Abstract

Whenever demand for a single item can be categorized into classes of different priority, an inventory rationing policy should be considered. In this paper we analyse a continuous review (s, Q) model with lost sales and two demand classes. A so-called critical level policy is applied to ration the inventory among the two demand classes. With this policy, low–priority demand is rejected in anticipation of future high–priority demand whenever the inventory level is at or below a prespecified critical level. For Poisson demand and deterministic lead times, we present an exact formulation of the average inventory cost. A simple optimization procedure is presented, and in a numerical study we compare the optimal rationing policy with a policy where no distinction between the demand classes is made. The benefit of the rationing policy is investigated for various cases and the results show that significant cost reductions can be obtained.

Keywords: Inventory, rationing, lost sales, two demand classes.

1 Introduction

In most of the literature on inventory models it is assumed that all demand for a single item is equally important. However, in practice, the demand for an item can often be categorized

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

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

into classes of different priority. Consider, for example, the spare parts inventory in the airline industry. Most airlines have a contractual agreement with a company that supplies them with spare parts whenever an aircraft is grounded at the airport due to failure of some equipment. In the contractual agreement it is stated that the company promises e.g. a 98% service level to the airline. Beside these key customers, the spare parts inventory company may also satisfy demand from other airlines. These airlines are usually considered to be of lower priority, and their orders will only be satisfied if the inventory level is high enough, so that the 98% service level for the key customers is not endangered. Another situation where demand for a single item may have different priorities occurs in multi-echelon inventory systems with emergency orders (see e.g. Chiang & Gutierrez [1]). In many of these systems emergency orders are placed at the lowest echelon whenever the stock level is low or when customer demand is backordered. As a result, at the higher echelon two types of demand are faced: emergency orders and normal replenishment orders. Whenever the higher echelon has insufficient stock to meet both types of demand, priority will be given to the emergency orders. Finally, we mention an example that can be found in an assembly–to–order system, where a component may be used for several end–products. If these end–products yield different profits to the firm, then demand for this component may be categorized into classes of different priority.

A simple way of operating inventory systems with two demand classes is to use a rationing policy that reserves part of the stock for high priority demand by rejecting low–priority demand when stock is below a certain critical level. Henceforth, we refer to such a policy as a *critical level policy* and we will restrict ourselves to policies where the critical level is independent of the remaining lead time. However, such information, if available, could lead to improved policies. For example, if the inventory manager knows that a replenishment order will arrive soon, it may be optimal to satisfy low–priority demand even though the inventory level is below the critical level. A disadvantage of operating a policy that takes into account information about remaining lead times is that it is much more difficult to implement in practice.

In this paper we will consider a critical level policy in the context of a continuous review (s, Q) inventory model with lost sales. In some practical situations, a customer demand is handled in another way, e.g. through another supplier, if it cannot be delivered from stock on hand. Hence, at the inventory system, this demand may be viewed as a lost sale. The stockout cost in this case represents the additional cost for expediting the customer order. To the best of our knowledge, this model has not been analysed in the literature so far. However, some closely related models have appeared and an overview of them is presented in the next section. The two following sections deal with the derivation of the average inventory cost in a continuous review (s, Q) inventory model with lost sales and two demand classes. For Poisson demand and fixed lead times we derive an expression for the average inventory cost, including holding, shortage and ordering costs. In the fifth section the optimization of the policy parameters is discussed, and the sixth section illustrates the model by means of some numerical examples. The main conclusions are presented in the last section of this paper.

2 Related literature

In this paper two areas in inventory control theory are combined, i.e. continuous review (s, Q) inventory models with lost sales, and inventory rationing. In this section an overview of related literature in both areas is presented.

The (s, Q) model with lost sales was first discussed by Hadley & Whitin [7]. They derived an exact formulation of the average inventory cost for an (s, Q) policy with Poisson demand and constant deterministic lead times, under the assumption that at most one order is outstanding. They also presented an easy approximation of the average cost and developed an iterative procedure to optimize the policy parameters, which has become the standard textbook approach (see e.g. Silver and Peterson [12] and Tersine [13]). More recently, Johansen & Thorstenson [8] formulated and solved the same model as a semi–Markov decision model.

Inventory rationing was first introduced by Veinott [16], who proposed a critical–level policy for a periodic review model with n demand classes and zero lead time in a backorder environment. This model was also analysed by Topkis [15], and for two demand classes by Kaplan [9] and Evans [5]. The first contribution in a continuous review inventory model was made by Nahmias & Demmy [11]. They analysed an (s, Q) inventory model with two demand classes, Poisson demand, backordering, a fixed lead time and a critical level policy, under the assumption that there is at most one outstanding order. 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. In their analysis they used the notion of the *hitting time* of the critical level, i.e. the time that the inventory level reaches the critical level. Conditioning on this hitting time, it is possible to derive approximate expressions for the service levels. Observe that the model presented by Nahmias & Demmy [11] is the one most closely related to the model we present in this paper. However, we assume lost sales and find expected holding, shortage and ordering costs, which enable us to optimize the parameters of the policy such that the costs are minimized. 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 main result was the derivation of (approximate) expressions for the fill rates for both demand classes. The results of Nahmias & Demmy [11] were generalized by Moon & Kang [10]. They considered an (s, Q) model with compound Poisson demand, and derived (approximate) expressions for the fill rates of the two demand classes.

Rationing policies in a lost sales environment have not received much attention. Cohen, Kleindorfer & Lee [2] consider a periodic review (s, S) inventory system where all demands in each period are collected, and by the end of each period the inventory is used to satisfy high–priority demand first, and the remaining inventory is then made available for low–priority demand. Hence, they did not consider a critical–level policy. Recently, Ha [6] analysed a lot–for–lot lost sales model with n demand classes and Poisson demand. He assumed exponentially distributed lead times and modelled the system as a single–product $M/M/1/S$ queue (Tijms [14]) with state–dependent service times. This enabled him to prove optimality of the lot–for–lot critical–level policy. Dekker, Hill & Kleijn [3] analysed the same model with a general lead time distribution. They modelled the system as an $M/M/S/S$ queue (Tijms [14]) and developed efficient methods to determine the optimal policy. Since they restricted themselves to policies which are independent of the remaining lead time, the optimality of the critical level policy could not be guaranteed for generally distributed lead times.

3 Notation and preliminaries

In this section we introduce the notation that will be used throughout this paper. We consider an inventory model with two demand classes, each with unit Poisson demand with arrival rate λ_1 for high–priority demand and λ_2 for low–priority demand. The cost of not satisfying a demand from demand class j is denoted by π_j, $j = 1, 2$, with $\pi_1 > \pi_2 > 0$. All demand not satisfied immediately is assumed to be lost. The fixed ordering cost is K and there is a fixed lead time of L time units. The unit holding cost per time unit is denoted by $h > 0$.

The (s, Q) policy extended with a critical level is denoted as a (c, s, Q) policy, which operates as follows: whenever the inventory level drops to the reorder level s, a replenishment order of size Q is placed which arrives after L time units. Demand from both classes is satisfied whenever the inventory level exceeds the critical level c, otherwise only high–priority demand (class 1) is satisfied from stock on hand and low–priority demand is lost. Following Hadley & Whitin [7] and Nahmias & Demmy [11], we will restrict ourselves to policies in which there is at most one outstanding order. In a lost sales environment, the condition that $s < Q$ is sufficient to enforce that at most one order is outstanding. This means that in contrast to Nahmias & Demmy [11] (where the assumption of only one outstanding order is an approximation, due to the backorder environment) our results are exact. In principle, the critical level c is unbounded, but for the model to be tractable we need to require that $c < Q$. Nahmias & Demmy [11] only consider policies where the critical level is below the reorder point, and to our knowledge nobody has considered the situation where $c \geq s$, although the benefit of such policies, as we shall show later, can be significant. In order to be able to derive an expression for the average cost, we need some additional notation. Let $X(t)$ denote the physical inventory level at time t, and let $\{X(t), t \geq 0\}$ be the corresponding stochastic process. The restriction $Q > s$ ensures that $\{X(t), t \geq 0\}$ is a regenerative process with regenerative epochs each time the inventory level reaches the reorder level s and a replenishment order is placed. Define a cycle as the time between two consecutive regenerative epochs. Then our process can be split into independent and identically distributed renewal cycles. Using the renewal–reward theorem (see e.g. Tijms [14]) we know that the average cost per time unit equals the expected cost during a cycle divided by the expected length of a cycle. In case the inventory policy satisfies the condition $c < s$, we let H be a random variable denoting the *hitting time* of the critical level, i.e. the time from placing a replenishment order (or the time when the inventory level 'hits' the reorder level s) until the time where the inventory level 'hits' the critical level c. Since the total demand from both classes follows a Poisson distribution with parameter $\lambda := \lambda_1 + \lambda_2$, it readily follows that H is Erlang distributed with parameters $(s - c)$ and λ. Figure 1 illustrates the inventory process over a cycle for

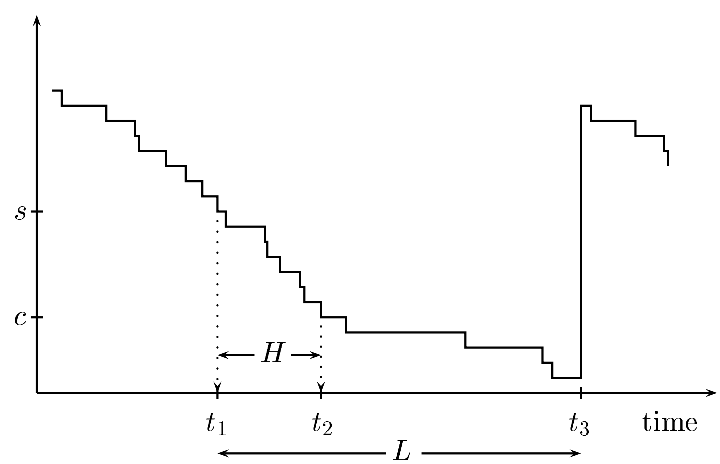

Figure 1: The inventory process with $c < s$. At t_1 an order is placed. At t_2 the inventory level 'hits' the critical level c and at t_3 the order arrives.

a (c, s, Q) policy with $c < s$. Furthermore, we define R as the random variable denoting the inventory level just before a replenishment order arrives. Let $D_j(t)$, $j = 1, 2$, be a random variable denoting the demand from demand class j during t time units, and let $D(t) := D_1(t) + D_2(t)$ be the total demand from both classes during t time units. We can find the distribution of R as

$$\mathbb{P}(R=i)=\begin{cases}\mathbb{P}(D_1(L-H)\geq c) & \text{for } i=0\\ \mathbb{P}(D_1(L-H)=c-i) & \text{for } 0<i\leq c\\ \mathbb{P}(D(L)=s-i) & \text{for } c<i\leq s\end{cases} \tag{1}$$

For a (c,s,Q) policy with $c\geq s$ we reach the critical level before we place an order and the hitting time H is therefore not defined. The distribution of R is then simply

$$\mathbb{P}(R=i)=\begin{cases}\mathbb{P}(D_1(L)\geq s) & \text{for } i=0\\ \mathbb{P}(D_1(L)=s-i) & \text{for } 0<i\leq s\end{cases} \tag{2}$$

We are now able to calculate the expected cost of a (c,s,Q) policy. This will be done in the next section.

4 Deriving the average cost

In this section an expression for the average cost in a (c,s,Q) inventory system will be derived. The total cost is composed of inventory holding and shortage costs, and ordering costs. The approach we follow is to derive first the expected cost during a cycle and then calculate the expected cycle length. Using the renewal–reward theorem we obtain an expression for the average cost. We divide the analysis into two parts: first we consider policies with $c<s$, thereafter we discuss the situation where $c\geq s$.

Average cost for $c<s$

We first consider the case where $c<s$, so that the inventory level hits the critical level after a replenishment order is placed. In this case we introduce the hitting time H and we may find the expected number of stockouts per cycle $B_j^{c<s}$ for demand class j, $j=1,2$, by conditioning on this hitting time. Given a hitting time t, the conditional value of $B_2^{c<s}$ is simply equal to the expected value of the demand from class 2 until the next order arrives $(L-t)$ time units later. The conditional value of $B_1^{c<s}$ is equal to $\mathbb{E}_{D_1(L-t)}[D_1(L-t)-c]^+$, which is the expected value of the demand from class 1 during $(L-t)$ that exceeds the stock c reserved for this class. To ease notation let $\mathbb{E}_{D_j(L-H),H}[f(D_j(L-H),H)]$ be the expected value of f where expectations are taken first with respect to the demand $D_j(L-t)$ given the event $(H=t)$, and then with respect to H. We can now find the unconditional values of $B_j^{c<s}$ $j=1,2$.

$$B_1^{c<s} = \mathbb{E}_{D_1(L-H),H}[D_1(L-H)-c]^+ \tag{3}$$
$$B_2^{c<s} = \mathbb{E}_{D_2(L-H),H}[D_2(L-H)] \tag{4}$$

Since the distributions of D_1, D_2 and H have been determined it is not difficult to calculate B_1 and B_2. In Appendix 1 we give the results.

Our derivation of $B_2^{c<s}$ is identical to that of Nahmias & Demmy [11], whereas we present a faster method for calculating $B_1^{c<s}$.
The total expected shortage cost per cycle is given by

$$TSC^{c<s} = \pi_1 B_1^{c<s} + \pi_2 B_2^{c<s}$$

The holding cost incurred during a cycle is the sum of the holding cost incurred on each inventory level visited during the cycle. The cost incurred on one inventory level i is simply the number of units i times the unit holding cost per time unit, h, times the expected time spent on the level. It is a well–known fact that (see e.g. Tijms [14], p. 24)

> "given the occurrence of n arrivals in $(0,t)$, the n arrival epochs are statistically indistinguishable from n independent observations taken from the uniform distribution on $(0,t)$".

The expected time spent on each inventory level reached during a period of length t with n demands is therefore $t/(n+1)$ if the time interval does not end with a demand (e.g. when a replenishment arrives), and t/n if the time interval does end with a demand. We will split the holding cost up in two parts. The holding cost $HC_1^{c<s}$ incurred during the lead time, and the holding cost $HC_2^{c<s}$ incurred in the remaining part of the cycle. The holding cost incurred in the lead time depends on whether we hit the critical level during the lead time, and if so, whether the inventory is depleted during the remaining lead time. If the total lead time demand $D(L)$ is less than $s-c$ then the holding cost incurred is

$$h\mathbb{E}_{D(L)}\left[\sum_{i=s-D(L)}^{s} i \cdot \frac{L}{D(L)+1}\right] \tag{5}$$

If we hit the critical level (i.e. $D(L) \geq s-c$), we divide the holding cost in the holding cost before and after the hitting time H. The expected holding cost incurred before the hitting time is

$$h\mathbb{E}_{H}\left[\sum_{i=c+1}^{s} i \cdot \frac{H}{s-c}\right] \tag{6}$$

If after the lead time the inventory is not depleted, that is $D_1(L-H) < c$, the expected holding cost incurred is

$$h\mathbb{E}_{D_1(L-H),H}\left[\sum_{i=c-D_1(L-H)}^{c} i \cdot \frac{L-H}{D_1(L-H)+1}\right] \tag{7}$$

If the inventory is depleted during the remaining part of the lead time, the expected arrival time of the last demand satisfied is $\frac{c}{D_1(L-H)+1}(L-H)$ time units after the hitting time.

The expected holding cost incurred is therefore

$$h\mathbb{E}_{D_1(L-H),H}\left[\sum_{i=1}^{c} i \cdot \frac{(L-H)c/(D_1(L-H)+1)}{c}\right] \tag{8}$$

We have now described the holding cost incurred during the lead time as a function of the random variables H, $D(L)$ and $D_1(L-H)$. Since their distributions are known, we can determine the expected holding cost incurred during the lead time. In Appendix 1 a complete derivation of the expected holding cost $HC_1^{c<s}$, suitable for implementation, is presented.

We will now find the expected holding cost $HC_2^{c<s}$ incurred in the remaining part of the cycle. Observe that after a replenishment order arrives, the inventory level is $R+Q$. The expected holding cost incurred while the inventory level drops to s is the sum of the holding cost incurred on each level, and since we have unit demand the expected time spent on each level is $1/\lambda$. Taking expectations with respect to R yields

$$HC_2^{c<s} = h\mathbb{E}_R\Big[\sum_{i=s+1}^{Q+R} i \cdot \frac{1}{\lambda}\Big]$$

The total expected holding cost is

$$THC^{c<s} = HC_1^{c<s} + HC_2^{c<s}$$

All we need now to derive an expression for the average inventory cost is the expected length of a cycle, which is the lead time plus the expected length of the period where the inventory is reduced from $R+Q$ to s. Hence the expected length is given by

$$LoC^{c<s} = L + \frac{Q + \mathbb{E}[R] - s}{\lambda}$$

and the average cost of a (c, s, Q) policy with $c < s$ is given by

$$TC^{c<s}(c,s,Q) = \frac{TSC^{c<s} + THC^{c<s} + K}{LoC^{c<s}}$$

4.1 Average cost for $c \geq s$

In the model developed above only rationing policies with $c < s$ were considered. In this section we will find the expected cost of a policy with $c \geq s$. For such policies we will start rejecting demand from demand class 2 before we place an order. The analysis is similar to the one in the previous section and we will adopt the same notation. Let $\tau := \inf\{t \geq 0 : D_1(t) \geq c - s\}$ denote the time between the start of rejecting low–priority demand and placing a replenishment order. See Figure 2.

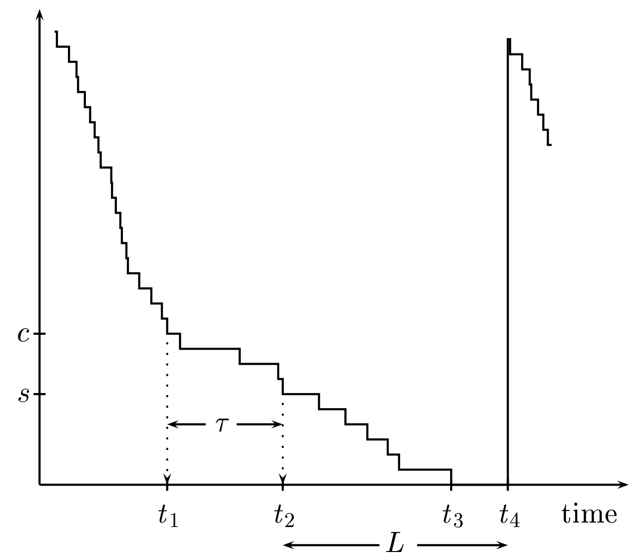

Figure 2: The inventory process with $c \geq s$. At t_1 the inventory 'hits' the critical level. At t_2 the order is placed. The inventory is depleted at t_3 and at t_4 the order arrives.

The expected number of stockouts for demand class 1 and 2 is

$$\begin{aligned} B_1^{c\geq s} &= \mathbb{E}_{D_1(L)}[D_1(L) - s]^+ \\ B_2^{c\geq s} &= \mathbb{E}_{D_2(L+\tau),\tau}[D_2(L+\tau)] \end{aligned}$$

By observing that $\mathbb{E}\tau = (c-s)/\lambda_1$ we obtain by the memoryless property of the Poisson process that $B_2^{c\geq s} = \lambda_2(L + (c-s)/\lambda_1)$. The calculation of $B_1^{c\geq s}$ is straightforward. The expected total stockout cost per cycle is

$$TSC^{c\geq s} = \pi_1 B_1^{c\geq s} + \pi_2 B_2^{c\geq s}$$

To calculate the expected holding cost during a cycle, we divide the holding cost in three parts. The expected holding cost incurred is found by the same principles used in the previous section. Let $HC_3^{c\geq s}$ be the expected holding cost that incur as the inventory level drops from $Q+R$ to c and let $HC_2^{c\geq s}$ be the expected holding cost incurred while the inventory level drops from c to s.

$$\begin{aligned} HC_3^{c\geq s} &= h\mathbb{E}_R\Big[\sum_{i=c+1}^{Q+R} i \cdot \frac{1}{\lambda}\Big] \\ HC_2^{c\geq s} &= h\sum_{i=s+1}^{c} i \cdot \frac{1}{\lambda_1} \end{aligned}$$

Finally let $HC_1^{c\geq s}$ be the expected holding incurred during the lead time. By conditioning on whether the inventory is depleted or not and using the same reasoning as in (8) we obtain

$$
\begin{aligned}
HC_1^{c\geq s} &= h\mathbb{E}_{D_1(L)}\Bigg[\mathbf{1}_{\{D_1(L)<s\}}\sum_{i=s-D_1(L)}^{s} i\cdot\frac{L}{D_1(L)+1} \\
&\quad +\mathbf{1}_{\{D_1(L)\geq s\}}\sum_{i=1}^{s} i\cdot\frac{Ls/(D_1(L)+1)}{s}\Bigg]
\end{aligned}
$$

The total expected holding cost is now given by

$$
THC^{c\geq s} = HC_1^{c\geq s} + HC_2^{c\geq s} + HC_3^{c\geq s}
$$

It is easy to calculate the total expected holding cost (see Appendix 2).
The expected length of a cycle is

$$
LoC^{c\geq s} = L + \frac{Q+\mathbb{E}[R]-c}{\lambda} + \frac{c-s}{\lambda_1}
$$

Hence, the total cost of a (c, s, Q) policy with $c \geq s$ is given by

$$
TC^{c\geq s}(c,s,Q) = \frac{TSC^{c\geq s} + THC^{c\geq s} + K}{LoC^{c\geq s}}
$$

We have now concluded the analytic derivations of the expected cost of the rationing policies. Using the results presented in Section 4.1 and 4.2, we obtain that the average cost $TC(c, s, Q)$ of a (c, s, Q) inventory policy is given by

$$
TC(c,s,Q) = \begin{cases} TC^{c<s}(c,s,Q) & \text{if } c < s \\ TC^{c\geq s}(c,s,Q) & \text{if } c \geq s \end{cases}
$$

5 Optimization

Due to the complexity of the average cost formula it has not been possible to derive an explicit expression for the optimal policy. The optimization procedure is therefore based on enumeration and bounding.
Assume that the order size Q is given, and denote the associated optimal values of c and s by $c^*(Q)$ and $s^*(Q)$. To obtain an upper bound on the value of $s^*(Q)$ we conjecture that $s^*(Q)$ will be less than or equal to the optimal reorder level for the model without a critical level (which is equivalent to our model with $c = 0$).
Although we cannot give a formal proof of this, we give an intuitive explanation. When the critical level is positive, the average demand rate during the lead time will decrease. Since the main purpose of using a reorder level is to cover the lead time demand, and the optimal reorder level is increasing with the total lead time demand, we conjecture that the optimal reorder level $s^*(Q)$ is decreasing with the critical level c. We note that all our numerical experiments supported this conjecture.

The inventory model without a critical level is identical to a simple lost sales (s, Q) model with demand rate $\lambda := \lambda_1 + \lambda_2$ and lost sales cost $\pi := (\lambda_1\pi_1 + \lambda_2\pi_2)/\lambda$. Instead of determining the optimal reorder point for this model we will use the reorder point following from the Hadley–Whitin heuristic which is an upper bound on $s^*(Q)$, as shown in the following lemma.

Lemma 1. *Let $\tilde{s}(Q)$ be the reorder point obtained by using the Hadley-Whitin heuristic and let $\bar{s}(Q)$ be the optimal reorder point in the simple lost sales model. For a fixed value of Q it follows that $\tilde{s}(Q) \geq \bar{s}(Q)$. Furthermore, $\tilde{s}(Q)$ is found as the solution to*

$$\tilde{s}(Q) = min\ \{s \geq 0 : \mathbb{P}(D(L) \geq s+1) \leq \frac{h}{h + \pi\lambda/Q}\}$$

Proof. The average cost of the simple (s, Q) policy is given by (see Hadley & Whitin [7])

$$TC_0(s, Q) = \frac{[K\lambda/Q + h[(Q+1)/2 + s - \lambda L] + (h + \frac{\pi\lambda}{Q})\mathbb{E}[D(L) - s]^+]}{(Q + \mathbb{E}[D(L) - s]^+)/\lambda}$$

Since Q is fixed we can write this as

$$TC_0(s, Q) = \frac{f(s)}{Q/\lambda + g(s)} \text{ where } g(s) \geq 0$$

Following the heuristic we approximate the average cycle length by Q/λ, as if there were no stockouts, and obtain

$$\tilde{TC}_0(s, Q) = \frac{f(s)}{Q/\lambda}$$

The reorder level $\tilde{s}(Q)$ that minimizes $\tilde{TC}_0(s, Q)$ surely minimizes $f(s)$ too. Thus for any $y > 0$ we have

$$\begin{aligned} TC_0(\tilde{s}(Q) + y, Q) &\geq \frac{f(\tilde{s}(Q))}{Q/\lambda + g(\tilde{s}(Q)) + y)} \\ &\geq \frac{f(\tilde{s}(Q))}{Q/\lambda + g(\tilde{s}(Q))} \\ &= TC_0(\tilde{s}(Q), Q) \end{aligned}$$

establishing the upper bound. Observe that the second inequality is a result of $g(s)$ being decreasing. The reorder level $\tilde{s}(Q)$ is found as (see Hadley & Whitin [7])

$$\tilde{s}(Q) = \min\{s \geq 0 : \mathbb{P}(D(L) \geq s+1) \leq \frac{h}{h + \pi\lambda/Q}\}$$

□

By Lemma 1 and our previous conjecture, we obtain that $\tilde{s}(Q)$ is an upper bound on the optimal reorder level $s^*(Q)$. In our computational experiments we experienced that it is possible to end up in local minima when searching for $s^*(Q)$. Hence, we suggest enumeration over all values between 0 and $\tilde{s}(Q)$.

We also suggest enumeration to determine the optimal critical level. Given the reorder level s, we evaluate all critical levels between 0 and $s-1$ using the average cost function $TC^{c<s}(c,s,Q)$. Let c' be the value which gives the minimum cost, i.e.

$$c' = \text{argmin } \{TC^{c<s}(c,s,Q) : 0 \leq c < s\}$$

Regarding $c \geq s$ it can easily be proved that for fixed values of s and Q, the average cost function is either convex or concave in c, depending on the underlying model and the values of s and Q. The critical level that minimizes the average cost function for $c \geq s$ is denoted by

$$c'' = \text{argmin } \{TC^{c\geq s}(c,s,Q) : s \leq c < Q\}$$

If the average cost function is convex, c'' is found in the global minimum, which can be found explicitly. If the average cost function is concave, let $c'' = s$ if $TC^{c\geq s}(s,s,Q) < TC^{c\geq s}(Q-1,s,Q)$. Otherwise let $c'' = Q-1$. Finally, let the optimal critical level given the reorder level s and the order size Q be given by

$$c = \begin{cases} c' & \text{if } TC^{c<s}(c',s,Q) < TC^{c\geq s}(c'',s,Q) \\ c'' & \text{if } TC^{c<s}(c',s,Q) \geq TC^{c\geq s}(c'',s,Q) \end{cases}$$

In many practical situations the order size Q is prespecified. However, if one also wants to determine the optimal value of Q, one can use a local search algorithm with the economic order quantity as a starting solution. Numerical experiments have indicated that the average cost function is unimodal in Q.

6 Numerical results

In this section we will investigate the performance of the rationing policy discussed in the previous sections. As a performance measure we will use the cost reduction CR of using the optimal (c,s,Q) rationing policy compared to the best possible (s,Q) policy. Hence, CR is defined as

$$CR := \frac{\min\{TC_0(s,Q) : s \geq 0, Q > s\} - \min\{TC(c,s,Q) : Q > c \geq 0, Q > s \geq 0\}}{\min\{TC_0(s,Q) : s \geq 0, Q > s\}}$$

To determine $\min\{TC_0(s,Q) : s \geq 0, Q > s\}$ we used an enumeration approach similar to the one described in the previous section. Alternatively, one may use the method described in Johansen & Thorstenson [8].

From computational experiments it appeared that the critical level may influence the optimal reorder level s and the replenishment order size Q in two ways: if $c < s$ the main effect of the critical level is a reduction of the optimal reorder level s, whereas if $c \geq s$ the main effect lies in the reduction of the optimal order size Q. In this section we will consider examples that lead to both types of rationing policies, and at the end of the section, try to describe what determines the structure of the optimal policy.

Example 1

In the first example, we consider an inventory system with the following characteristics: $L = 1$, $h = 1$, $K = 100$, $\lambda_1 = 1$, $\lambda_2 = 10$, $\pi_1 = 1000$, and $\pi_2 = 10$. In Table 1 we have calculated the optimal critical level policy and the optimal non-rationing (s, Q) policy for Example 1. Observe that all costs are average cost per time unit. We see that a cost

Policy	$(c, s, Q) = (2, 14, 48)$	$(s, Q) = (17, 48)$
Total cost	52.49	54.96
Holding cost	27.87	30.52
Shortage cost	2.09	1.55
Ordering cost	22.54	22.88
Cycle length	4.44	4.37
Number of Stockouts Class 1	0.000058	0.00032
Number of Stockouts Class 2	0.041	0.0032

Table 1: Comparison of the optimal (c, s, Q) policy and the optimal (s, Q) policy.

reduction of 4.5% is obtained when a critical level policy is applied. As expected, the effect of the rationing policy is a reduced reorder level, leading to a lower average holding cost. The average stockout cost increases, but is more than compensated by the decrease in the holding cost. The expected ordering cost decreases due to the increase in the expected length of a cycle. We can also observe that the rationing policy has a dramatic effect on the expected number of stockouts for demand class 2, which increases by 1200%, whereas the number of stockouts for demand class 1 is reduced with 80%.

We have performed some variations of this example to show how the optimal policy is influenced by changes in the parameter values, and to investigate under which conditions the gain of rationing is most significant. The results are presented in Table 2.

In Table 2 we see what happens when we change the stockout cost of demand class 1. For small values of π_1 the cost reduction is negligible, whereas for larger values of π_1 the cost reduction is quite significant. The opposite is true if we change the value of π_2. Hence, it seems reasonable to conclude that the greater the difference between π_1 and π_2, the greater the cost reduction obtained by applying a critical level policy. When $\pi_2 = 1$ an interesting phenomenon occurs, i.e. the structure of the policy changes. From Table 2

π_1	100	500	1000	5000	10000	100000
(c, s, Q)	(1,12,49)	(1,14,48)	(2,14,48)	(2,15,48)	(3,15,48)	(4,16,48)
(s, Q)	(13,49)	(16,48)	(17,48)	(19,49)	(20,49)	(23,49)
CR	0.0062	0.0305	0.0449	0.0700	0.0819	0.1086
π_2	1	5	10	25	50	100
(c, s, Q)	(16,4,24)	(3,11,48)	(2,14,48)	(1,16,48)	(1,17,48)	(0,18,48)
(s, Q)	(17,48)	(17,48)	(17,48)	(17,49)	(17,49)	(18,48)
CR	0.4973	0.0698	0.0449	0.0217	0.0106	0.0000
K	25	50	100	200	500	1000
(c, s, Q)	(2,15,25)	(2,15,35)	(2,14,48)	(2,13,67)	(2,11,106)	(24, 3,132)
(s, Q)	(18,25)	(18,34)	(17,48)	(16,68)	(15,107)	(15,150)
CR	0.0625	0.0548	0.0449	0.0374	0.0296	0.1268
L	0.5	1	1.5	2	2.5	3
(c, s, Q)	(1, 8,48)	(2,14,48)	(2,20,48)	(2,26,49)	(2,31,50)	(3,37,49)
(s, Q)	(10,48)	(17,48)	(24,48)	(30,49)	(37,49)	(43,49)
CR	0.0289	0.0449	0.0569	0.0650	0.0717	0.0788

Table 2: Cost reductions for variations of Example 1.

one can observe that for $\pi_2 \geq 5$ the optimal policies satisfy $c < s$, whereas for $\pi_2 = 1$ we obtain an optimal critical level policy with $c \geq s$. A similar observation can be made with respect to the fixed order cost K. If we increase K the cost reduction decreases because the average ordering cost constitutes a larger part of the total cost. Moreover, the optimal order size Q increases and the reorder level s decreases. For all $K \leq 500$ the optimal critical level is equal to 2. However, for $K = 1000$ the structure of the optimal policy changes and we get an optimal critical level of $c = 24$. Increasing the lead time L is, to some extent, equivalent to increasing λ_1 and λ_2 simultaneously. As L increases the cost reduction increases as well. Note that for this example it has not been possible to investigate problems with $L > 3$. In these cases it would be optimal to have $s > Q$, which is not covered by our analysis.

In Figure 3 we study in more detail the change in structure of the policy, or 'bang–bang' effect, for varying values of π_2 and K. We have calculated the optimal policy with respect to two different restrictions. The solid lines represent policies where $c \geq s$ and the dashed lines policies with $c < s$. For small values of K the restriction $c \geq s$ leads to policies that perform worse than the optimal (s, Q) policy. But as K increases, the optimal order size is increased too, and the cost of carrying inventory increases. When the fixed order

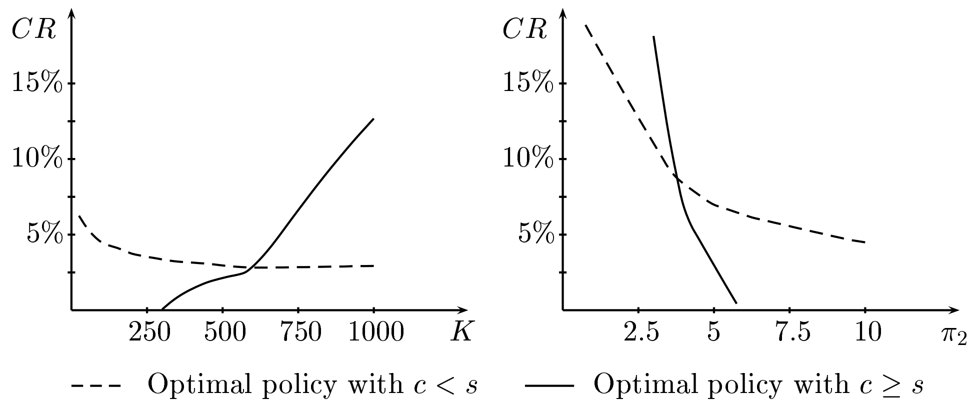

Figure 3: Cost reductions for policies with $c < s$ and policies with $c \geq s$ (Example 1).

cost exceeds a certain value (approximately 600 in this example) the cost of carrying inventory gets so high that it is optimal to reject some low–priority demand in order to reduce demand and thereby the holding cost. Another 'bang–bang' effect is found when we change the cost of rejecting low–priority demand. As seen in Table 2 the cost reduction increases as π_2 decreases, and Figure 3 illustrates that as the cost of rejecting low–priority demand gets low enough, the cost reduction of using policies with $c \geq s$ increases rapidly, and the structure of the optimal policy changes. In Example 2 we will discuss policies with $c \geq s$ in more detail.

To investigate the effect of changing the demand rates we have calculated the cost reduction for 400 combinations of λ_1 and λ_2 with all other parameters fixed (see Figure 4). The

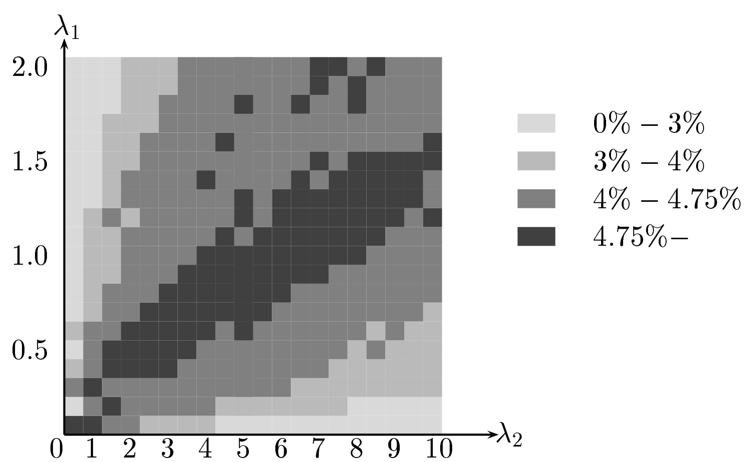

Figure 4: Cost reduction for Example 1 obtained by rationing for different values of λ_1 and λ_2.

classification is chosen in order to equalize the size of the areas. No rigorous conclusion is possible, but it seems clear that the cost reduction is strongly connected to the demand ratio λ_1/λ_2. If this ratio is greater than one, that is the majority of the demand is considered to be of high priority, the cost reduction is very small. This is also the case if the demand ratio is smaller than 1/20. In between it appears that the largest cost reduction is obtained for demand ratios between 1/4 and 1/10. This observation differs from the observations made by Ha [6], who concluded that the greatest cost reduction, in an $(S-1, S)$ model, is obtained for demand ratios around one.

Example 2

In this example, we will turn our attention to rationing policies with $c \geq s$. There is no way to guarantee that the optimal rationing policy has $c \geq s$. However, in order to favour the policies with $c \geq s$ we can increase the fixed order cost and the unit holding cost, or lower the stockout cost of demand class 2. For the second example the following parameter values are used: $L = 1$, $h = 2$, $K = 200$, $\lambda_1 = 1$, $\lambda_2 = 5$, $\pi_1 = 500$, and $\pi_2 = 6$. The optimal critical level policy, the optimal (s, Q) policy, and the corresponding costs for Example 2 are reported in Table 3. Again, all costs are average cost per time unit.

Policy	$(c, s, Q) = (12, 3, 28)$	$(s, Q) = (9, 36)$
Total cost	60.76	78.68
Holding cost	21.41	43.13
Shortage cost	23.97	2.36
Ordering cost	15.38	33.18
Cycle length	13.00	6.03
Number of stockouts class 1	0.0018	0.0045
Number of stockouts class 2	3.85	0.13

Table 3: Comparison of the optimal (c, s, Q) policy and the optimal (s, Q) policy.

We see that a considerable cost reduction of 22.8% is obtained. It is very interesting to see that the cost allocation in the optimal critical level policy is very different from the allocation in the standard (s, Q) policy. The average holding and ordering costs are both reduced with about 50% while the average shortage cost has increased with a factor 10. This is caused by the fact that the expected cycle length is doubled because we reject on average 3.85 demands from demand class 2 per time unit. The expected holding and ordering cost per cycle is more or less unchanged, so the reduction per time unit is mainly due to the longer expected cycle length. The expected number of stockouts per cycle for demand from demand class 1 hardly changes. However, since the cycle length is doubled, the average stockout cost per time unit for demand class 1 is reduced.

Also for Example 2 we have analysed the effect of variations in the parameter values on the optimal policy and the cost reduction. The results are shown in Table 4.

π_1	40	100	500	1000	5000	10000
(c, s, Q)	(10, 0,28)	(11, 1,28)	(12, 3,28)	(12, 3,28)	(13, 4,28)	(14, 5,28)
(s, Q)	(0,35)	(5,36)	(9,36)	(10,36)	(12,36)	(13,35)
CR	0.1896	0.2046	0.2278	0.2344	0.2470	0.2470
π_2	2	4	6	10	25	50
(c, s, Q)	(15, 3,19)	(14, 3,24)	(12, 3,28)	(6, 3,34)	(1, 8,36)	(1, 9,36)
(s, Q)	(9,36)	(9,36)	(9,36)	(9,36)	(9,36)	(9,36)
CR	0.4439	0.3307	0.2278	0.0664	0.0203	0.0073
K	50	100	200	400	1000	1500
(c, s, Q)	(2,7,18)	(3, 7,23)	(12, 3,28)	(18, 3,34)	(30, 2,47)	(37, 2,54)
(s, Q)	(10,18)	(9,26)	(9,36)	(8,50)	(7,79)	(7,96)
CR	0.0978	0.1364	0.2278	0.3108	0.4038	0.4353

Table 4: Cost reductions for variations of Example 2.

By changing the value of the parameters π_1, π_2 and K, we see that high cost reductions can be obtained by applying a critical level policy. The cost reduction CR increases when π_1 increases, but the optimal policies remain more or less the same. As π_2 gets very small, the advantage of using the rationing policy increases rapidly. Note that while the optimal (s, Q) policy does not change at all, the (c, s, Q) policy is sensitive to changes in π_2. The 'bang–bang' effect that occurs is similar to the one observed in Example 1 (see Figure 3). If the fixed order cost K increases, the optimal replenishment order size Q increases as well, both for the (s, Q) policy and the (c, s, Q) policy. However, for the latter policy this increase is limited due to the increasing level of c, thus part of the holding cost is replaced by additional stockout cost.

In Figure 5 we observe that there is no clear relation between the cost reduction CR and the demand ratio λ_1/λ_2, as was the case for Example 1. It seems like CR decreases as λ_1 increases. When the share of high–priority demand in the total demand increases, the influence of the rationing policy declines, which explains the dependence of the cost reduction with λ_1. The dependence with λ_2 is more complicated. The largest cost reductions are found for values of λ_2 between 2 and 6. It is obvious that when the demand rate approaches zero there will be no gain of rationing. On the other hand, if the demand rate gets very high, the cost of rejecting demand will increase so that it is not profitable to

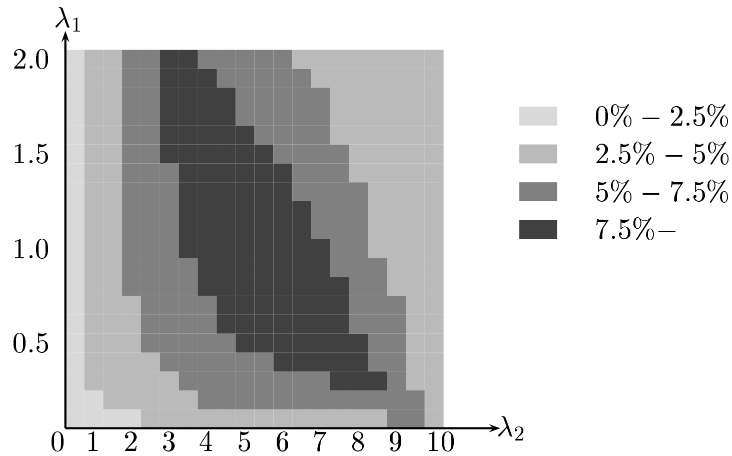

Figure 5: Cost reduction for Example 2 obtained by rationing for different values of λ_1 and λ_2.

exchange holding cost for stockout cost. Finally, we observe that for all combinations of λ_1 and λ_2, the cost reduction is substantial.

We will conclude this section by investigating what determines the structure of the optimal policy. We have previously seen that the values of the fixed order cost and the stockout cost for demand class 2 have great influence on the structure of the optimal policy. To obtain further insight we have determined the structure of the optimal policy for a number of different parameter values. The effect of changing the demand rate or the stockout cost of demand class 1 turned out to be negligible. More interesting is the effect of the parameter values connected with demand class 2 with respect to the structure of the optimal policy. Figure 6 shows how the structure of the optimal policy depends on the values of K, λ_2 and π_2. Observe that the area above the curve corresponds to optimal policies satisfying $c \geq s$, whereas the area below is associated with optimal policies satisfying $c < s$.
The effect of K and π_2 on the structure of the optimal policy are as expected: if K is small and π_2 is large then the optimal policy will satisfy $c < s$. We also see that the demand rate of demand class 2 significantly influences the structure of the optimal policy. If the demand rate for class 2 is relatively low, then the optimal policy is more likely to satisfy $c \geq s$. This can be explained by observing that a policy with $c \geq s$ implies that most demand from class 2 is lost which does not lead to high lost sales cost if λ_2 is small.

7 Conclusions and further research

In this paper we have discussed an (s, Q) inventory model with lost sales and two demand classes. We have introduced a lead time independent rationing policy, i.e. the (c, s, Q)

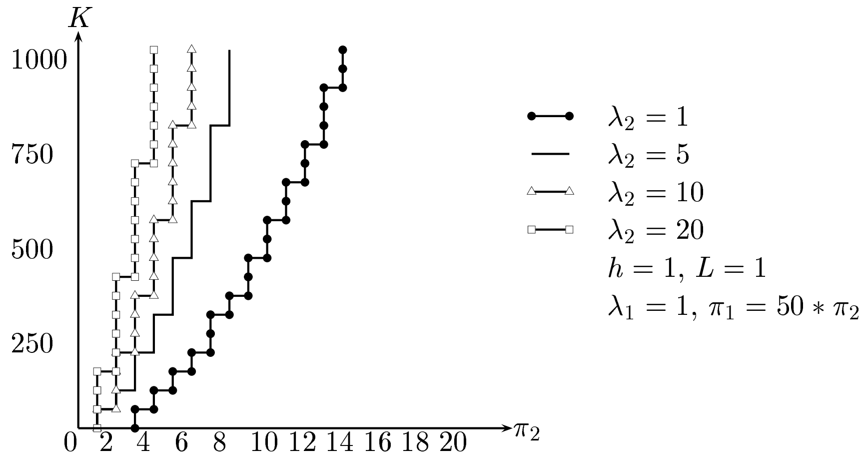

Figure 6: The structure of the optimal policy.

policy. This so-called critical level policy reserves part of the inventory for high-priority demand, i.e. if the inventory level is at or below the critical level c, low-priority demand is rejected in anticipation of future high–priority demand. We have derived an exact expression for the average cost of (c, s, Q) policies, that satisfy $0 \leq c < s < Q$ or $0 \leq s \leq c < Q$. We have shown that this rationing policy can have two different effects on the optimal reorder level and replenishment order size, depending on whether the critical level is below or above the reorder level. In cases where the optimal policy has $c \geq s$ the rationing policy will in general reduce the average holding cost by rejecting a great part of the low–priority demand. This type of policy is usually optimal if the cost of rejecting low–priority demand is small (compared to the holding cost rate), or if the fixed order cost is high. The cost reduction obtained by using a rationing policy in these cases is in general higher than it is for cases, where a policy with a critical level below the reorder point is optimal. In our examples cost reductions up to 50 % were recorded. In cases where the optimal policy has $c < s$, the critical level policy reduces the safety stock needed. Significant cost reductions can be obtained if the stockout cost of high–priority demand is considerably larger than the stockout cost of low–priority demand. However, the maximum cost reduction we observed was 10% which was observed in a situation where the stockout costs differed by a factor of 10000.

Although the lead time independent (c, s, Q) policy is easy to understand and implement in practice, it may be cost effective to consider a lead time dependent policy. If the inventory level is below the critical level, and a low–priority customer arrives, it may be optimal to deliver this demand anyway given that a replenishment order will arrive soon.

We are currently investigating a model with several demand classes, where the rationing decision is allowed to depend on both the inventory level and, during the lead time, the time since the replenishment order was placed.

Appendix 1: Derivation of average cost for $c < s$

To simplify the notation let

$$\begin{aligned} p_j(i) &= \mathbb{P}(D_j(L) = i) \text{ for } j = 1,2 \text{ and } i = 0,1,2,\ldots \\ p(i) &= \mathbb{P}(D(L) = i) \text{ for } i = 0,1,2,\ldots \end{aligned}$$

Moreover let $f_H(t)$ denote the pdf of the hitting time H. It is easy to find an expression for $B_2^{c<s}$, i.e.

$$\begin{aligned} B_2^{c<s} &= \mathbb{E}_{D_2(L-H),H}[D_2(L-H)] \\ &= \int_0^L f_H(t)\lambda_2(L-t)dt \\ &= \int_0^L e^{-\lambda t}\lambda^{s-c}\frac{t^{s-c-1}}{(s-c-1)!}(L-t)\lambda_2 \, dt \\ &= \frac{\lambda_2}{\lambda}(\lambda L - s + c)\Big[1 - \sum_{j=0}^{s-c-1} p(j)\Big] + \frac{\lambda_2}{\lambda}\frac{e^{-L\lambda}(\lambda L)^{s-c}}{(s-c-1)!} \end{aligned}$$

which is equivalent to the result obtained by Nahmias & Demmy [11].
To find $B_1^{c<s}$ we need the distribution of $D_1(L-H)$. By conditioning on the hitting time and using the binomial expansion for $(L-t)^i$ we obtain

$$\begin{aligned} \mathbb{P}(D_1(L-H) = i) &= \int_0^L f_H(t) \cdot P(D_1(L-t) = i)dt \\ &= \int_0^L e^{-\lambda t}\lambda^{s-c}\frac{t^{s-c-1}}{(s-c-1)!}e^{-\lambda_1(L-t)}\lambda_1^i\frac{(L-t)^i}{i!}dt \\ &= \sum_{k=0}^{i}\frac{e^{-\lambda_1 L}L^{i-k}(-1)^k}{(i-k)!k!}\frac{(k+s-c-1)!}{(s-c-1)!}\frac{\lambda^{s-c}\lambda_1^i}{\lambda_2^{k+s-c}}\Big[1 - \sum_{j=0}^{k+s-c-1} p_2(j)\Big] \\ &= \sum_{k=0}^{i}(-1)^k A(i,k)\Big[1 - \sum_{j=0}^{k+s-c-1} p_2(j)\Big] \end{aligned}$$

with

$$\begin{aligned} A(i,0) &:= \frac{e^{-\lambda_1 L}L^i\lambda^{s-c}\lambda_1^i}{i!\lambda_2^{s-c}} \\ A(i,k) &:= A(i,k-1) \cdot \frac{(k+s-c-1)\cdot(i-k+1)}{L \cdot k \cdot \lambda_2} \end{aligned} \qquad (9)$$

Hence, we find

$$\begin{aligned} B_1^{c<s} &= \mathbb{E}_{D_1(L-H),H}[D_1(L-H)-c]^+ \\ &= \sum_{i=c+1}^{\infty} \mathbb{P}(D_1(L-H)=i)\ (i-c) \end{aligned}$$

Nahmias & Demmy [11] suggest that the integral is solved using numerical integration. However, this is a slow procedure, whereas the expression developed above is exact and fast.

The holding cost incurred during the lead time has previously been found as a function of the random variables H, $D(L)$ and $D_1(L-H)$. We will now find the expected holding cost $HC_1^{c<s}$ incurred during the lead time, by conditioning on these variables. For $D(L) < s-c$ we apply (5), for $D(L) \geq s-c$ (which is equivalent with $0 < H < L$) we apply (6) and either (7) or (8).

$$\begin{aligned} HC1^{c<s} &= h \sum_{j=0}^{s-c-1} p(j) \Big(\sum_{i=s-j}^{s} i \cdot \frac{L}{j+1} \Big) \\ &+ h \int_0^L f_H(t) \Bigg[\sum_{i=c+1}^{s} i \cdot \frac{t}{s-c} \\ &+ \sum_{j=0}^{c} \mathbb{P}(D_1(L-t)=j) \Big(\sum_{i=c-j}^{c} i \cdot \frac{L-t}{j+1} \Big) \\ &+ \sum_{j=c+1}^{\infty} \mathbb{P}(D_1(L-t)=j) \Big(\sum_{i=1}^{c} i \cdot \frac{(L-t)c/(j+1)}{c} \Big) \Bigg] dt \end{aligned}$$

Exploiting the properties of $\sum_{i=a}^{b} i$ yields

$$\begin{aligned}
HC_1^{c<s} &= h\sum_{j=0}^{s-c-1} p(j)(s-j/2)L + h\int_0^L f_H(t)\frac{c+1+s}{2}t\ dt \\
&+h\int_0^L f_H(t)\sum_{i=0}^{c}\mathbb{P}(D_1(L-t)=i)(c-i/2)(L-t)dt \\
&+h\int_0^L f_H(t)\sum_{i=c+1}^{\infty}\mathbb{P}(D_1(L-t)=i)\frac{c+1}{2}\cdot(L-t)\frac{c}{i+1}dt \\
&= h\sum_{j=0}^{s-c-1} p(j)(s-j/2)L \\
&+h\frac{s^2-c^2+s-c}{2\lambda}\Big[1-\sum_{i=0}^{s-c}p(i)\Big] \\
&+h\sum_{j=0}^{c}(c-j/2)\sum_{k=0}^{j+1}(-1)^k B(k,j)\Big[1-\sum_{i=0}^{s-c-1+k}p_2(i)\Big] \\
&+h\sum_{j=c+1}^{\infty}\frac{c+1}{2}\frac{c}{j+1}\sum_{k=0}^{j+1}(-1)^k B(k,j)\Big[1-\sum_{i=0}^{s-c-1+k}p_2(i)\Big]
\end{aligned}$$

with

$$\begin{aligned}
B(j,0) &:= e^{-\lambda_1 L}\frac{L^{j+1}\lambda^{s-c}\lambda_1^j}{j!\lambda_2^{s-c}} \\
B(j,k) &:= B(j,k-1)\cdot\frac{(s-c-1+k)\ \cdot(j+2-k)}{L\cdot\lambda_2\cdot k}
\end{aligned} \tag{10}$$

The expected holding cost incurred in the remaining part of the cycle is easily found, i.e.

$$\begin{aligned}
HC_2^{c<s} &= h\mathbb{E}_R\Big[\sum_{i=s+1}^{Q+R} i\cdot\frac{1}{\lambda}\Big] \\
&= h\mathbb{E}_R\Big[\frac{Q+R+s+1}{2}\cdot\frac{Q+R-s}{\lambda}\Big] \\
&= h\frac{Q^2+2Q\mathbb{E}[R]+\mathbb{E}[R^2]+Q+\mathbb{E}[R]-s-s^2}{2\lambda}
\end{aligned}$$

The first two moments of the random variable R are easily found from (1). The total expected holding cost is

$$THC^{c<s} = HC_1^{c<s} + HC_2^{c<s}$$

The expressions developed in this appendix are valid for all combinations of parameters. However, during implementation, numerical problems can arise when evaluating $\mathbb{P}(D_1(L-H))$ and $H_1^{c<s}$. If $\lambda_2 < \lambda_1$ the terms in(9) and (10) get very big, causing representation problems, and the integrals should be solved using numerical integration.

Appendix 2: Derivation of the expected holding cost for $c \geq s$

The expected holding cost is divided in three parts.

$$
\begin{aligned}
HC_3^{c\geq s} &= h\mathbb{E}_R\Big[\sum_{i=c+1}^{Q+R} i \cdot \frac{1}{\lambda}\Big] \\
&= h\frac{Q^2 + 2QE[R] + E[R^2] + Q + E[R] - c - c^2}{2\lambda} \\
HC_2^{c\geq s} &= h\sum_{i=s+1}^{c} i \cdot \frac{1}{\lambda_1} \\
&= h\frac{c+s+1}{2}\frac{c-s}{\lambda_1}
\end{aligned}
$$

For $HC_1^{c\geq s}$ we condition on whether the inventory is depleted or not.

$$
\begin{aligned}
HC_1^{c\geq s} &= h\mathbb{E}_{D_1(L)}\Bigg[\mathbf{1}_{\{D_1(L)<s\}} \sum_{i=s-D_1(L)}^{s} i \cdot \frac{L}{D_1(L)+1} \\
&\quad + \mathbf{1}_{\{D_1(L)\geq s\}} \sum_{i=1}^{s} i \cdot \frac{Ls/(D_1(L)+1)}{s}\Bigg] \\
&= h\sum_{i=0}^{s-1} p_1(i)L(s-i/2) + h\sum_{i=s}^{\infty} p_1(i)\frac{s+1}{i+1}\frac{s}{2}L
\end{aligned}
$$

Hence, we have

$$THC^{c\geq s} = HC_1^{c\geq s} + HC_2^{c\geq s} + HC_3^{c\geq s}$$

References

[1] C. Chiang and G. J. Gutierrez. A periodic review inventory system with two supply modes. *European Journal of Operational Research*, 94(3):527–547, 1996.

[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] 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] R. Dekker, M.J. Kleijn, and P.J. De Rooij. A spare parts stocking system based on equipment criticality. *International Journal of Production Economics*, 56-57:463–472, 1998.

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

[6] 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.

[7] G. Hadley and T.M. Whitin. *Analysis of Inventory Systems.* Prentice–Hall, Englewood Cliffs, NJ, 1963.

[8] S.G. Johansen and Thorstenson A. Optimal and approximate (Q, r) inventory policies with lost sales and gamma-distributed lead time. *International Journal of Production Economics*, 30–31:179–194, 1993.

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

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

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

[12] E.A. Silver and R. Peterson. *Decision Systems for Inventory Management and Production Planning.* John Wiley & Sons, New York, 1985.

[13] R.J. Tersine. *Principles of Inventory and Materials Management.* North-Holland, New York, 1988.

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

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

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

Rationing policies for an inventory model with several demand classes and stochastic lead times

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

FEBRUARY 2001

Abstract

In this paper we analyse an (s, Q) inventory model with unit Poisson demand, several demand classes, lost sales and stochastic lead times. 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 elapsed 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. In general the performance of the simple policy is almost as good as that of the optimal policy. Depending on the underlying parameters of the model, the cost reduction compared with a non-rationing policy is typically 5-10%.

Keywords: Inventory, rationing, Markov processes, several demand classes, stochastic lead times.

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. However, different airlines may 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 is given by Axsäter, Kleijn and de Kok [1], who analyse a two–echelon inventory model, where the warehouse faces demand from several retailers. If the warehouse cannot satisfy a retailer demand immediately, the demand is expedited and satisfied by an emergency order directly from the outside supplier. Since the costs of expedition and backordering are different from retailer to retailer, the warehouse are rationed to minimize costs.

The inventory system considered is controlled by a rationing policy specified by critical levels. For each demand class, except the one with the 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 [17] analyses a model with several demand classes and zero lead time, and introduces the concept of critical levels. Topkis [16] 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, Zhang and Duenyas [4] 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 at 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 for 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. Dekker, Hill and Kleijn [3] consider a lot-for-lot inventory system with several demand classes, and 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. Simple critical level policies for an (s, Q) inventory model are first analysed by Nahmias and Demmy [12], 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 [11] 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 closely related to the present one is that of Melchiors, Dekker and Kleijn [9], which presents 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 that of Teunter and Klein Haneveld [14], presenting 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. Stochastic lead times in a rationing environment are considered by Ha [5, 6], who analyses an $(S-1, S)$ inventory/production model where the production times are assumed to be exponential and Erlang distributed, respectively. These assumptions facilitate an analysis based on queuing theory, but may appear unrealistic in many cases.

Typically, it is not the lead time itself but the lead time demand that is of interest. Many inventory systems are analysed based on the distribution of the lead time demand. For our purpose this distribution is not sufficient. Whether a demand is satisfied or not depends on when the demand occurs, and therefore we have to decompose the lead time demand into two random variables: the lead time and the Poisson demand.

We analyse the (s, Q) inventory model with several demand classes, lost sales and stochastic lead times, and by the use of the 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, in the case of constant lead times, that the critical levels are decreasing in time. 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. The performance of the simple policy is very good, and the difference in cost between the simple and the optimal policy is, only in a few cases, higher than 2 percent.

The time-remembering policy allows actions to depend on the time elapsed since the order was placed, which is a continuous variable. To obtain a finite state space, we therefore discretize the lead time. The demand is essentially Poisson. However, during lead time we approximate the demand with a Bernoulli process, such that the demand at every time epoch, is either one or zero. This approximation is good, as long as distance between the time epochs is small. A similar approach is used by Johansen and Thorstenson [8] to find optimal emergency order policies.

We do not restrict the lead time to follow a phase-type distribution or a parametric distribution. The discretized lead time allows us to describe the stochastic lead time by a probability mass function. This function can, e.g., be found by empirical observations. If the number of observations is small, one can apply a kernel density approximation to smooth out the observations (see Strijbosch and Heuts [13], who also argue that this gives better results than fitting a parametric function to the observations). Stochastic lead times for a similar model without rationing are treated by several authors, e.g. Mohebbi and Posner [10] who evaluate an (s, nQ) inventory system with continuous review and lost sales, where lead times follow phase type distributions.

Our paper distinguishes itself by considering multiple demand classes, stochastic lead times and time-remembering policies. Although we, in most cases, will recommend the use of simple rationing policies, it is the analysis of the time-remembering policies that facilitates evaluation of the simple rationing policy with multiple demand classes and stochastic lead times. Moreover, the optimal policy serves as a benchmark for the simple policy.

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 the optimal time-remembering policy, and in Section 4 we present a heuristic for finding good simple policies. In Section 5 we investigate the performance of the rationing policies by means of some numerical examples and, finally, some concluding remarks 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 Poisson demand with rate λ_j. All demand not satisfied immediately is assumed to be lost (or expedited). The classes are distinguished by their stockout 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 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 there is a fixed ordering cost K. The unit holding cost per time unit is $h > 0$.

The time when an order is outstanding is discretized to obtain a finite number of time epochs, each representing a small subinterval of length $1/N$. The stochastic lead time is then approximated by the probability mass function $f(t)$, which is the probability of a lead time of t subintervals. The true lead time is continuous, but if N is high this error is negligible. We assume that there exists an integer M such that $\sum_{t=0}^{M} f(t) = 1$ (for unbounded distributions we choose M such that the probability of a lead time of more than M subintervals is negligible). Based on $f(t)$ we can calculate the lead time hazard function $H(t)$ that denotes the probability of an arrival just prior to subinterval t, given that no order has arrived prior to subinterval $t-1$.

$$H(t) = \frac{f(t)}{\sum_{r=t}^{M} f(r)} \text{ for } t = 0, 1, \ldots, M-1$$

and $H(M) = 1$.

We analyse the rationing policy in the context of an (s, Q) policy with reorder point s and order size Q where $Q > s$. This condition and the lost sales assumption ensure that at most one order is outstanding at any time (in the analysis in Nahmias and Demmy [12] the assumption of only one outstanding order is an approximation, due to the backorder environment). 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. 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

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

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 at the beginning of each subinterval when one order is outstanding. At 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 if $a > 0$, 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 shall 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 determines the length of each subinterval and 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 an outcome that is either one or zero. Each of the subintervals can be viewed as a trial where the outcome is one if a demand that we are willing to satisfy occurs, and zero otherwise. The probability of outcome one depends on the chosen action and is $p_1(a) = \Lambda_a/N$. Also let $p_0(a) = 1 - p_1(a)$ denote the probability of outcome zero. The assumption of at most one demand per subinterval has only to do with the analysis. If the policy is implemented in practice, it can handle more than one demand per subinterval, and the assumption of at most one demand per subinterval is therefore not restrictive. The approximation considerably simplifies 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 three different states. With probability $H(t)$ an order arrives and we jump to state $i+Q \in S_0$. With probability $(1-H(t))p_0(a)$ we jump to state $(i,t+1)$ and with probability $(1-H(t))p_1(a)$ we jump to state $(i-1,t+1)$. In states $(0,t) \in S_1$ we jump either to state Q or to state $(0,t+1)$ since backlogging is not allowed.

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

$$\tau_{i,t} = 1/N \text{ for } (i,t) \in S_1,$$

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 action a is chosen, is

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

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

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

Note that we make a small numerical 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 t/N and $(t+1)/N$, satisfy demand from classes 1 to $l(i,t)$.

The considered policy is not necessarily a critical level policy. To be a critical level policy it must satisfy

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

and

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

This means that, for each class $j \geq 2$ and for all t, there 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 were introduced by Melchiors, Dekker and Kleijn [9], who also describe 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 [15]) 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 at 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 formulas

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

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

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

and

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

can be found by recursion for $t = M-1, M-2, \ldots, 0$ 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 [15]). 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. *For an optimal rationing policy the following properties are true:*

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

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

c *If the lead time is constant, then the critical levels are decreasing in the time t, i.e. the optimal actions satisfy*

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

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 neighborhood search starting e.g. with the Economic Order Quantity, computed by considering the deterministic version of the problem where all demand classes are aggregated to one. For each value of Q, the optimal values of $k(i)$, $l(i,t)$ and s are found by the policy-iteration algorithm. The optimal policy is denoted R_{opt}. The optimization procedure has been implemented in Pascal, and is very efficient.

4 The simple policy

The optimal policy can be difficult to implement in practice. In this section we shall therefore 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

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

and

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

Let $g(\boldsymbol{c}; s, Q)$ denote the cost of the simple policy $(\boldsymbol{c}; s, Q)$. The order size Q is found by neighborhood search starting with the Economic Order Quantity. 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 and 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 $j = n$ and $k = 1$. Let $\boldsymbol{c}^{k+1} = \boldsymbol{c}^k + \boldsymbol{e}_j$. If $g(\boldsymbol{c}^{k+1}; s, Q) < g(\boldsymbol{c}^k; s, Q)$ set $j := j-1$ and $k := k+1$, 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 best vector obtained sofar, and continue until no further improvements can be made.

The backwards enumeration over s is chosen because we have observed that the best $\boldsymbol{c}$–vector increases as s decreases, which means 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). We note that the procedure is not guaranteed to find the optimal $\boldsymbol{c}$–vector. In particular when there are critical levels above the reorder point, we have found that the search method may fail to find the best policy. In those cases we therefore calculate an alternative solution initialized by the critical levels from the optimal solution where defined, instead of the vector of zeroes, i.e. $\boldsymbol{c}^1 = (c_2(-), c_3(-), \ldots, c_n(-))$. The simple policy is then found as the policy with the lower cost of the two.

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 would then 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 shall first investigate an example described by Melchiors, Dekker and Kleijn [9], 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. In a numerical study we have found that the difference in cost between evaluating using $N = 10000$ and $N = 100000$ is within 0.004%. We do not, however, use $N = 10000$ when we find the optimal and the simple policies. 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 $N = 500$. Thereafter, the found policy is evaluated using $N = 10000$. The cost of a policy R is denoted $\gamma(R)$.

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

$$\lambda^{non} = \sum_{i=1}^{n} \lambda_i \text{ and } \pi^{non} = \sum_{i=1}^{n} \pi_i \lambda_i / \Lambda_n.$$

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

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

In all our numerical test we let $h = 1$ define the monetary unit and let the mean lead time $E(L) = 1$ define the time unit.

Example 1

In the first example, we consider an inventory system with two demand classes and the following characteristics: A constant lead time, $L = 1$, $K = 100$, $\lambda_1 = 1$, $\lambda_2 = 10$, $\pi_1 = 1000$, 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

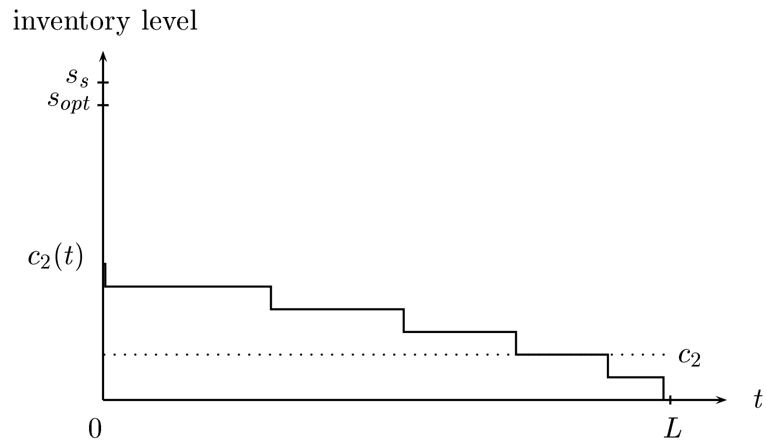

Figure 1: The critical levels $c_2(t)$ of the optimal policy and c_2 of the simple policy for Example 1 (n=2).

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 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 at the end of the lead time we will not reject demand class 2 at all. That is, the optimal policy dominates the simple policy in two situations: when the demand in the beginning of the lead time is high, and when demand from class 2 appears at 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 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 dividing demand class 2 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 basically the same as in the original example. The way the two examples

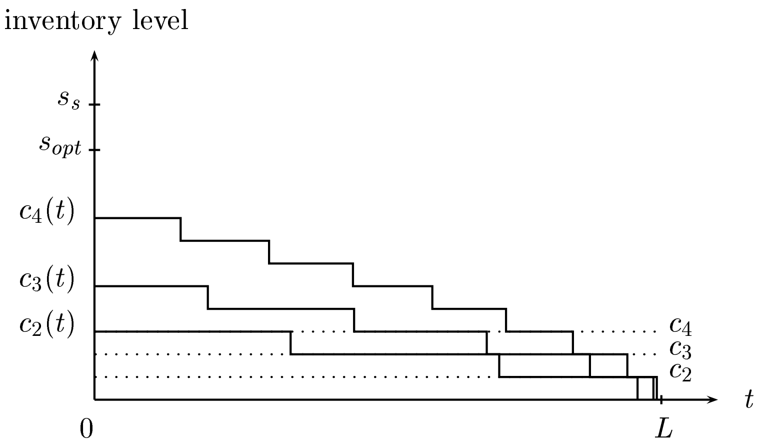

Figure 2: The critical levels of the optimal and the simple policies for Example 1 (n=4).

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 at 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.

Example 2

We now investigate the influence of the underlying parameters and different lead time distributions. We consider two types of lead time distributions: a cut-off normal distribution, truncated at zero, and a discrete two-point distribution with a 95% probability of a lead time close to the mean (L_1), and a 5% probability of a high lead time (L_2). Let $F(t)$ be the distribution function of a normal distribution with mean μ and variance σ^2. Since our time is discrete, we set

$$f(t) = \frac{F(t) - F(t-1)}{1 - F(0)} \text{ for } 0 \leq t \leq M$$

for the cut-off normal distribution. The distribution is truncated at 0 to avoid negative lead times. The mode of the lead time, μ, is adjusted to ensure that the mean remains 1, for all values of lead time variation.

For the two-point distribution we choose L_1, and L_2 such that the mean lead time remains 1 as well. The lead time variance is denoted $V(L)$

We let the arrival rates λ_i be composed of two parameters. A general level of demand, r, and a class dependent arrival rate $\tilde{\lambda_i}$, such that $\lambda_i = r\tilde{\lambda_i}$. Similarly we let π_i be composed of a general level of stockout cost and a class dependent cost. We consider the following base case: $K = 200$, $(\tilde{\lambda_1}, \tilde{\lambda_2}, \tilde{\lambda_3}, \tilde{\lambda_4}) = (1, 1, 2, 6)$. $(\pi_1, \pi_2, \pi_3, \pi_4) = (300, 90, 30, 9)$, $r = 1.5$. The lead time is cut-off normal with $V(L) = 0.64$.

Tables 1, 2 and 3 show a number of variations over this base case. For every case we report the best simple policy, the best non-rationing policy, the cost reduction obtained by using the simple policy, and finally the cost reduction obtained by using the optimal policy.

$(\pi_1, \pi_2, \pi_3, \pi_4)$	$(c_2, c_3, c_4; s, Q)_s$	$(s, Q)_{non}$	$CR_s(\%)$	$CR_{opt}(\%)$
(5,4,3,2)	(17,23,33; 1, 63)	(1,73)	13.24	13.25
(100,30,10,3)	(2,8,44;16, 80)	(25,86)	10.30	10.38
(1000,200,40,10)	(3,10,23;34, 85)	(47,86)	9.97	10.01
(15,12,9,6)	(0,1,6; 9, 83)	(10,83)	0.67	0.70
(300,90,30,9)	(2,7,19;28, 86)	(37,87)	6.97	7.01
(3000,600,120,30)	(3,9,20;45, 86)	(57,85)	8.01	8.06
(25,20,15,10)	(0,2,6;17, 84)	(17,86)	0.60	0.61
(500,150,50,15)	(2,7,18;34, 86)	(42,87)	6.20	6.23
(5000,1000,200,50)	(2,8,19;50, 86)	(61,85)	7.32	7.38
K	$(c_2, c_3, c_4; s, Q)_s$	$(s, Q)_{non}$	$CR_s(\%)$	$CR_{opt}(\%)$
50	(2,7,17;35, 48)	(43,48)	9.26	9.30
100	(2,7,18;32, 63)	(40,64)	8.17	8.21
200	(2,7,19;28, 86)	(37,87)	6.97	7.01
300	(2,8,20;26,102)	(35,104)	6.28	6.31
500	(2,8,21;23,129)	(33,131)	5.44	5.46

Table 1: Cost reductions for different stockout and ordering costs.

In Table 1 we first investigate three different levels of stockout cost and three different class dependent levels. In this way we represent a fair range different stockout costs combinations. Somewhat against intuition the benefits of rationing are decreasing as the level of the stockout costs increase. For the lowest values of stockout cost we see a high cost reduction of 13 %. In this example the critical levels are above the reorder point, i.e. we start to reject demand before we place an order. Thus the lowest demand class only serves to reduce inventories to avoid high holding cost in the beginning of an order cycle. The benefits of this can be quite significant. This is investigated more thoroughly in Melchiors, Dekker and Kleijn [9]. For higher values of stockout cost this phenomenon does not occur, because the cost of rejecting the demand exceeds the potential benefit.

For those examples we can also observe that a higher spread in stockout costs leads to higher benefit of rationing.

$(\tilde{\lambda_1}, \tilde{\lambda_2}, \tilde{\lambda_3}, \tilde{\lambda_4})$	$(c_2, c_3, c_4; s, Q)_s$	$(s, Q)_{non}$	$CR_s(\%)$	$CR_{opt}(\%)$
(1,1,2,6)	(1,4,11;20, 84)	(31,87)	8.87	8.89
(4,1,1,4)	(4,11,21;30, 85)	(41,87)	8.52	8.56
(2,2,3,3)	(2,7,19;28, 86)	(37,87)	6.97	7.01
(3,3,2,2)	(3,11,24;33, 86)	(41,86)	5.54	5.57
(6,2,1,1)	(6,16,31;41, 85)	(45,86)	3.45	3.48
r	$(c_2, c_3, c_4; s, Q)_s$	$(s, Q)_{non}$	$CR_s(\%)$	$CR_{opt}(\%)$
0.25	(0,2,12; 4, 32)	(5,34)	4.39	4.41
1.00	(2,5,13;18, 69)	(24,69)	6.06	6.11
1.50	(2,7,19;28, 86)	(37,87)	6.97	7.01
2.00	(3,9,24;39,100)	(51,101)	7.61	7.64
3.00	(4,14,35;62,126)	(79,127)	8.49	8.52

Table 2: Cost reductions for different class allocations and demand levels r

With respect to the ordering cost it appears that rationing has the greatest impact when the ordering costs are low. This is because the measure of performance is cost per unit time, and it is only during the lead time (except for the cases where the critical levels are above the reorder point) that rationing is used. As the ordering cost increases, the length of the inventory cycle increases and thereby the cost difference per unit time decreases.

In Table 2 we look at the impact of different demand structures. First we see that rationing has the greatest impact when the majority of the demand has low priority. When the majority of the demand has high priority, rationing will only seldom be used and the relative benefits are therefore smaller. As the general level of demand r increases the benefits of rationing increase as well. Since demand is Poisson, a higher demand mean is equivalent to a higher demand variance, and therefore the need for safety stock increases.

In Table 3 we see the impact of the lead time distribution. In general, rationing is more important when the lead time variance is high, which is what we would expect. When we are uncertain about the delivery time of an order, the rationing policy reduces the need for otherwise high safety stocks. For the two-point distribution a strange phenomenon occurs: as the lead time variance increases, the found reorder point of the non-rationing policy decreases and ends up being 33% lower than that of the simple policy. The best explanation for this is that it is better to be well off in 95% of the order cycles and then have serious stockouts in the remaining 5%, than having too much stock in 95% of the cycles and still face some stockouts in the remaining 5%. Instead, an increase in the order

$V(L)$			$(c_2, c_3, c_4; s, Q)_s$	$(s, Q)_{non}$	$CR_s(\%)$	$CR_{opt}(\%)$
0			(0,2,4;20,79)	(21,80)	1.29	2.59
0.21			(1,4,10;24, 82)	(28,82)	3.52	3.72
0.48			(2,6,16;27, 84)	(34,85)	5.70	5.78
0.64			(2,7,19;28, 85)	(37,87)	6.97	7.01
0.74			(2,8,21;29, 86)	(39,88)	7.69	7.73
0.81			(3,9,21;29, 87)	(39,90)	8.16	8.17
L_1	L_2	$V(L)$	$(c_2, c_3, c_4; s, Q)_s$	$(s, Q)_{non}$	$CR_s(\%)$	$CR_{opt}(\%)$
0.95	2.0	0.05	(1,4,7;21,82)	(23,83)	3.45	4.75
0.89	3.0	0.21	(5,7,10;23,86)	(21,95)	6.66	8.61
0.84	4.0	0.47	(8,10,13;25,90)	(20,105)	8.58	10.66
0.82	4.5	0.64	(10,12,15;27,91)	(19,110)	9.21	11.31
0.79	5.0	0.84	(11,13,16;27,94)	(18,115)	9.73	11.80

Table 3: Cost reductions for cut-off normal distributed and two-point distributed lead times for different values of lead time variance.

size is experienced.

For all examined examples, except the one with constant lead time and the ones with two-point distributed lead times, the difference between the optimal and the simple policy is very small. This is because the value of the time information depends on the distribution of the lead time. When lead times are deterministic we can make full use of the information. Although the two-point distribution is not deterministic the information of time is still very important, which is why the difference between the simple and the optimal policy increases as the variance increases. However, in the cut-off normal lead time cases where the lead time variance is high, the value of time information is very small. In these cases the optimal policy will do no better than a policy that disregards information about time, and therefore, as the variance of the cut-off normal distributed lead time increases, the optimal policy approaches the simple policy. Since the optimal policy is much more complicated to implement, we therefore recommend to use the simple policy in most cases. However, for the case of constant lead times and two point distributed lead times the cost difference between the simple and the optimal policy is 1-3% and in these cases in particular the optimal rationing policy should be considered.

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, stochastic lead times and several demand classes. Using

Markov decision theory, we have found the optimal rationing policy and shown that it is a critical level policy. Moreover, we have constructed an algorithm for finding a good simple policy. 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. Compared to a non-rationing policy, it is possible to obtain significant cost reductions. The disadvantage of rationing is the potential loss of goodwill incurred when unsatisfied customers realize that later-arriving customers may have their demands fulfilled. However, in most situations customers are aware of which class they belong to and accept the corresponding service level. Moreover, the situations where it is possible to divide customers into classes, are mostly characterized by a buying process where sales are made over the phone, by EDI or via mail correspondence, and the risk of different customers meeting each other is therefore very small.

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 [15] 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 $v(i,t)$, $w(i)$ and g.

$$w(i) = \min_a \left\{ C_i(a) - g\tau_i(a) + \sum_{j \in S_0} P_{(i),(j)}(a) w(j) + \sum_{(j,r) \in S_1} P_{(i),(j,r)}(a) v(j,r) \right\} \quad (5)$$
$$\text{for } i \in S_0$$

$$v(i,t) = \min_a \left\{ C_{i,t}(a) - g\tau_{i,t} + \sum_{(j,r) \in S_1} P_{(i,t),(j,r)}(a) v(j,r) + \sum_{j \in S_0} P_{(i,t),(j)}(a) w(j) \right\} \quad (6)$$
$$\text{for } (i,t) \in S_1$$

Here $P_{(\cdot),(\cdot)}$ are the transition probabilities of which most are zero. 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 $v(i,t)$ and $w(i)$. 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) for $i > Q$ as $w(i) = \min_a\{J(i,a)\}$ with

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

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 $v(i,M) = w(i+Q)$ for $i = 0, 1, \ldots, Q-1$. For $i = 0$ and $t = M-1, M-2, \ldots, 0$, the values are easily found since we can only choose $a = 0$,

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

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

$$\begin{aligned} v(i,t) &= \min_a\Big\{(1-H(t))(C_{i,t}(a) - g\tau_{i,t} + p_0(a)v(i,t+1) \\ &\quad +p_1(a)v(i-1,t+1)) + H(t)w(Q+i)\Big\} \\ &= (1-H(t))\min_a\Big\{G_t(i,a)\Big\} + H(t)w(Q+i) \end{aligned}$$

where $G_t(i,a) = C_{i,t}(a) - g\tau_{i,t} + p_0(a)v(i,t+1) + p_1(a)v(i-1,t+1)$. It is easy to show that $G_t(i,a)$ is convex in a. Moreover, the action a that minimizes $G_t(i,a)$ 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) of the theorem, by setting $\tilde{a} = l(i-1,t)$. If the lead time is constant we can furthermore use part c) by setting $\tilde{a} = \max\{l(i-1,t), l(i,t+1)\}$. 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 of iterations (typically 4-6).

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,

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

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

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

and

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

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 $t = M$ we have

$$\begin{aligned}
& v(i+1,M) - 2v(i,M) + v(i-1,M) \\
&= w(i+Q+1) - 2w(i+Q) + w(i+Q-1) \\
&= J(i+Q+1;k(i+Q+1) - w(i+Q) - J(i+Q;k(i+Q) + w(i+Q-1) \qquad (11) \\
&\geq J(i+Q+1;k(i+Q+1) - w(i+Q) - J(i+Q;k(i+Q+1) + w(i+Q-1) \\
&= c(i+Q+1;k(i+Q+1)) - g/\Lambda_{k(i+Q+1)} - (c(i+Q;k(i+Q+1)) + g/\Lambda_{k(i+Q+1)}) \\
&\geq 0
\end{aligned}$$

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

$$\begin{aligned}
& v(i+1,r) - 2v(i,r) + v(i-1,r) \\
&= (1-H(r))\Big(G_r(i+1,l(i+1,r)) - G_r(i,l(i,r)) - G_r(i,l(i,r)) + G_r(i-1,l(i-1,r))\Big) \\
&\quad +H(r)\Big(w(i+Q+1) - 2w(i+Q) + w(i+Q-1)\Big)
\end{aligned}$$

Since $H(r)$ must be between zero and one, $(1-H(r))$ and $H(r)$ are greater than or equal to zero, and by (11) the sum of the three last terms is positive as well. We therefore only need to show that

$$G_r(i+1,l(i+1,r)) - G_r(i,l(i+1,r)) - G_r(i,l(i-1,r)) + G_r(i-1,l(i-1,r))$$

is positive. By the definition of $l(i,t)$ we get

$$\begin{aligned}
&G_r(i+1,l(i+1,r)) - G_r(i,l(i,r)) - G_r(i,l(i,r)) + G_r(i-1,l(i-1,r)) \\
\geq\ & G_r(i+1,l(i+1,r)) - G_r(i,l(i+1,r)) - G_r(i,l(i-1,r)) + G_r(i-1,l(i-1,r)).
\end{aligned}$$

and since the holding and the penalty costs cancel out together with g/N, this equals

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

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 for the case of constant lead times. This means that $H(t) = 0$ for all $t < M$ and $H(M) = 1$. By (8) this is the equivalent to

$$v(i+1,t+1) - v(i,t+1) \geq v(i+1,t) - v(i,t) \text{ for } i = 1,2,\ldots,Q-2 \tag{12}$$

for all t. We note that (12) is equivalent to the definition of a two-dimensional supermodular function (see e.g. Heyman and Sobel [7]). First we need to prove

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

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

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

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

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

Suppose inductively that (12) is true for $t = M, M-1, \ldots, r+1$. Now

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

by (12) 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] A.Y. Ha. Stock rationing in an $M/E_k/1$ make-to-stock queue. *Management Science*, 46(1):77–87, 2000.

[7] D.P. Heyman and M.J. Sobel. *Stochastic models in operations research, vol. 2*, chapter 8. McGraw Hill inc., 1984.

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

[9] P. Melchiors, R. Dekker, and M.J. Kleijn. Inventory rationing in an (s, Q) inventory model with two demand classes and lost sales. *Journal of the Operational Research Society*, 51(1):111–122, 2000.

[10] E. Mohebbi and M.J.M. Posner. A continuous–review inventory system with lost sales and variable lead time. *Naval Research Logistics*, 45:259–278, 1998.

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

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

[13] L.W.G. Strijbosch and R.M.J. Heuts. Modelling (s, Q) inventory systems: Parametric versus non-parametric approximations for the lead time demand distribution. *European Journal of Operational Research*, 63:86–101, 1992.

[14] 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.

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

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

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

Restricted time-remembering policies for the inventory rationing problem

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

May 2001

Abstract

We analyse 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 higher priority. We focus on time-remembering policies where the critical levels are allowed to depend on the time since the actual outstanding order (if any) was issued. In particular we introduce a new class of policies called restricted time-remembering (RTR) policies. We divide the lead time in m intervals of time and restrict the critical levels of the RTR policy to be constant over each interval of time. This policy is much easier to implement than the optimal rationing policy, where the critical levels can change continuously over time, and in a numerical study we show that the performance of the RTR policy is close to that of the optimal policy, even when the number of intervals is 2 or 3.

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

1 Introduction

Traditional inventory literature deals with the problem of how to replenish an inventory facing deterministic or stochastic demand. It is usually assumed that there is a cost of holding inventory, a fixed ordering cost (perhaps zero) and a requirement of a certain service level, or a specification of a stockout cost for unsatisfied demand. In this paper we will take a closer look at the demand process and how it influences our inventory decisions. The prevalent assumption in the inventory literature is that demand may be deterministic or stochastic following some known or unknown distribution, but the demand is homogeneous. This means that, from a cost or service-level perspective it is of no influence which demand is satisfied and which is not, in case of stockouts. This assumption is in many cases a good one. However, today companies are creating closer relationships with their suppliers who in turn need to provide these key-customers with the service they require. Simultaneously, the supplier faces demand from regular customers who may not be willing to pay for an increased level of service. In many cases a company can therefore divide their customers into demand classes of different priority.

There does not exist much literature on inventory control with several demand classes. In the recent textbooks of Silver, Pyke & Petersen [7] and Zipkin [11] the topic is not even mentioned.

In this paper we will discuss an extension of the inventory rationing problem presented in Melchiors [4]. The inventory rationing problem arises when customers for a single product can be divided into several classes of different importance. Assume for example that customers can be divided into high-priority and low-priority customers. In order to provide a high service level for high-priority customers, the inventory manager must maintain a high safety stock to protect the inventory from stockouts. However, by doing so, low-priority customers will receive the same, unnecessarily high service level. Alternatively, the inventory manager could ration his inventory. This could for example be done by using a critical level policy. The critical level policy introduced by Nahmias & Demmy [6], specifies a critical level for each demand class. A demand is only satisfied whenever the inventory level is above the critical level for the demand class. In this way it is possible to reserve stock for possible future high-priority demand. Melchiors, Dekker and Kleijn [5] analyse a critical level policy for an (s, Q) policy with two demand classes. The critical level policy is, however, not the optimal rationing policy. For example if it is known that a replenishment is about to arrive, there is no need to reject a low priority demand even if the inventory level is below the critical level. Melchiors [4] analyses an (s, Q) inventory rationing model with several demand classes and stochastic lead time, where actions can be taken based on information about both inventory level and elapsed lead time. The optimal policy, a so-called time-remembering policy, is indeed a critical level policy. For the case of constant lead times, the critical levels can be shown to be decreasing in time. The advantage of the optimal policy is, however, also its disadvantage. Since the critical levels change over time, it is only easily implementable in highly computerized implementations. In other applications a much more simple policy is needed.

In this paper we analyse a policy where the rationing decisions are based only on whether the remaining lead time is, say, short or long. In this way we can improve our performance by taking time into account, and still have a policy that is easy to operate in practice. Melchiors [4] illustrates that the cost difference between simple and optimal rationing policies is particularly high in cases with constant lead time. In those cases the value of information of time is high compared with cases with stochastic lead times where a simple policy that neglects information of time is almost as good as the optimal time-remembering policy. Consequently, we will in this paper focus only on the case with constant lead times.

For other references that discuss modelling of several demand classes we refer to Topkis [10], Teunter and Klein Haneveld [8], Dekker, Hill and Kleijn [2] and Ha [3].

In the next section we introduce a class of rationing policies called restricted time-remembering policies, and present the method for evaluating rationing policies given by Melchiors [4]. In Section 3 we present the optimization algorithm and conclude the paper with some numerical results in Section 4, and some concluding remarks in Section 5.

2 Evaluating rationing policies

We consider a continuous-review inventory model with n demand classes. Class j faces unit Poisson demand with rate λ_j. All demand not satisfied immediately is assumed to be lost. The classes are distinguished by their stockout cost π_j, and we rank the classes such that $0 < \pi_n < \pi_{n-1} < \cdots < \pi_1$.

For each replenishment order there is a fixed ordering cost K, and a constant lead time of L time units. The unit holding cost per time unit is $h > 0$. We will analyse 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. The rationing policy is evaluated by the method developed in Melchiors [4] which we include here, adjusted to the case of constant lead times.

Assuming that s and Q are fixed, the problem is formulated 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. 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 [1]). 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 $p_1(a) = \frac{L}{N}\sum_{j=1}^{a}\lambda_j$. 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. In our numerical results, we have used $N = 500$ which ensures that all values of $p_1(a)$ are less than or equal to 0.03.

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 $i > 0$ and $t < N$, 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 $i + Q \in S_0$.

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

$$C_i(a) = \frac{1}{\sum_{j=1}^{a}\lambda_j}[hi + \sum_{j=a+1}^{n}\lambda_j\pi_j] \text{ for } i \in S_0.$$

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

$$C_{i,t}(a) = \frac{L}{N}[hi + \sum_{j=a+1}^{n}\lambda_j\pi_j] \text{ for } (i,t) \in S_1.$$

Note that $\frac{1}{\sum_{j=1}^{a}\lambda_j}$ and $\frac{L}{N}$ are the expected time between two decisions epochs when there is no and one order outstanding, respectively. 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. The rationing policy is described by $k(i)$ and $l(i,t)$. Let $k(i)$ denote the action taken in states $i \in S_0$ and $l(i,t)$ denote the action taken in states $(i,t) \in S_1$.

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 [9]) 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

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

and

$$Y(i) = \frac{1}{\sum_{j=1}^{k(i)} \lambda_j} + Y(i-1) \text{ for } i \in S_0. \tag{2}$$

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 consider the states $(i,t) \in S_1$. 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

$$\begin{aligned} z(i,t) &= C_{i,t}(l(i,t)) + p_0(l(i,t))z(i,t+1) \\ &\quad + p_1(l(i,t))z(i-1,t+1) \text{ for } 0 < i \le s \\ z(0,t) &= \frac{L}{N}\sum_{j=1}^{n} \lambda_j \pi_j + z(0,t+1) \end{aligned}$$

and

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

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 (1) and (2) for $i = s+1, s+2, \ldots, Q$. The cost of the policy is

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

3 Restricted time-remembering policies

So far, the literature has described two kinds of policies, i.e. simple critical level policies and time-remembering policies. In practice a combination of the two could be more appealing.

Define a *restricted time-remembering*(RTR) policy to be a time-remembering policy where the critical levels are restricted to be constant over intervals of the lead time. These intervals are defined by the policy variables and must cover the entire lead time. The number of intervals determines how simple the RTR policy is. If there is only one interval, the RTR policy is identical to the simple critical level policy, and if there is N intervals, it is identical with the optimal rationing policy. Obviously the performance of the RTR policy increases when the number of intervals are increased.

We assume that customers are divided into n demand classes and that the lead time can be divided in m intervals of time. Let $c_{j,\tau}$ denote the critical level of class j in interval number τ for $1 \leq j \leq n$ and $\tau = 1, 2, \ldots, m$. Interval τ consists of the subintervals $\{t_{\tau-1}, t_{\tau-1}+1, \ldots, t_\tau - 1\}$ with $0 = t_0 \leq t_1 \leq \cdots \leq t_m = N$, with $t_1, t_2, \ldots, t_{m-1}$ being decision variables.

We restrict the set of intervals to be identical for all demand classes. A set of critical levels uniquely defines all values of $l(i,t)$:

$$l(i,t) = \max\bigl\{j | c_{j,\tau} \leq i,\ t \in \{t_{\tau-1}, t_{\tau-1}+1, \ldots, t_\tau - 1\}\bigr\}$$

We have furthermore the option of rejecting a demand even before an order is placed, as specified by $k(i)$. Demands rejected before an order is placed, will not be satisfied during the lead time either, and we can therefore use the critical level of interval 1, to determine the values of $k(i)$:

$$k(i) = \max\{j | c_{j,1} \leq i\}$$

In this way we can evaluate any rationing policy by the method presented in Section 2. However, we cannot use a policy-iteration algorithm or a value iteration algorithm to find the optimal RTR policy. Instead, we optimize the policy by a neighbor search based on the following empirical observations.

1. The optimal critical levels increase (or remain the same) as the reorder point decreases.
2. The optimal value of t_τ decreases (or remains the same) as the reorder point decreases for all τ.
3. The optimal critical levels $c_{j,\tau}$, $c_{j,\tau+1}$ increase (or remain the same) as t_τ decreases for all τ.

Although it appears difficult to prove these observations, all our computations support them, and our optimization algorithm has found a policy equal to the optimal RTR policy (determined by a full enumeration approach) in all our numerical tests. Observations 1 and 2 can be explained as follows: When we reduce the safety stock available during the lead time, we will (for fixed value of all t_τ) on average hit the critical levels earlier. The remaining stock (reserved for demand of higher priority) must therefore last for a longer period. Consequently, the optimal critical levels increase (or remain the same). Since we hit the new critical levels earlier, the optimal value of t_τ will decrease when the reorder point decreases, since we are interested in choosing the value of t_τ close to the expected hitting times. Observation 3 is illustrated in Figure 1. When the value of t_τ decreases, the critical levels covering period $\tau + 1$ will increase, since they must now also cover a period of time $[t_\tau^{new}, t_\tau^{old}]$ where the remaining lead time is longer and therefore the safety-stock needed will be higher. Similarly, the observation holds for period τ: Here the interval is reduced and since we now do not have to cover the period $[t_\tau^{new}, t_\tau^{old}]$, we can increase our critical levels to obtain a better coverage of the period $[t_{\tau-1}, t_\tau^{new}]$.

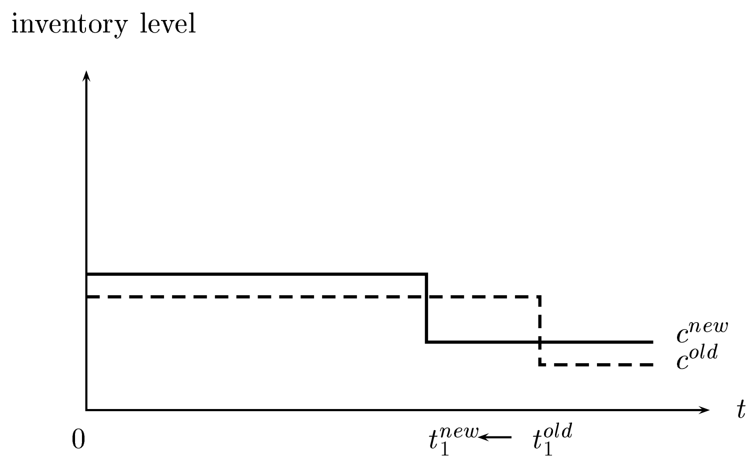

Figure 1: Illustration of how the optimal restricted critical levels increase (for both intervals) as the time t_1 decreases from t_1^{old} to t_1^{new}.

Based on these observations we can construct an algorithm, similar with that of Melchiors[4], for finding the optimal RTR policy. The algorithm is based on a improvement procedure `increase_c`, which we describe first :

`increase_c`: For a fixed value of Q, s and $(t_1, t_2, \ldots, t_{m-1})$ we find the best critical level vector by successive increments. Let $j = n$. In each iteration, first increase $c_{j,m}$ by one, then increase $c_{j,m-1}$ by one, etc., until either $c_{j,1}$ has been increased or the costs have increased. Now, reduce j by one and repeat this iteration until either $j = 2$ or the costs have increased. At each step we only increase $c_{j,\tau}$ if $c_{j,\tau} < c_{j+1,\tau}$. Moreover, if the increment of $c_{j,\tau}$ has led to an increase in costs, we reduce $c_{j,\tau}$ by one before we continue. The algorithm is repeated until all critical levels are unchanged.

The algorithm for finding the optimal RTR policy performs a neighbor search in the order size Q, starting with an order size found either by adopting the EOQ formula or by using that of the optimal rationing policy (by the method of Melchiors[4]). Let Δ be the stepsize used for the search of optimal values of $(t_1, t_2, \ldots, t_{m-1})$. For each Q we initialize by setting all $c_{j,\tau} = 0$ and all $t_\tau = N$ for all $\tau = 1, 2, \ldots, m-1$. We search for the optimal value of s in the interval $[0, Q-1]$. Initially, we set $s = Q-1$, after which we reduce the reorder level one by one.

For each reorder level s, we improve the time separators $(t_1, t_2, \ldots, t_{m-1})$ and the critical levels for $s+1$, using the observation that the optimal time separators decrease and the optimal critical levels increase. First we reduce t_1 with Δ, and apply `increase_c`. If costs are reduced, we reduce t_2 with Δ and apply `increase_c` etc. until t_{m-1} has been reduced or costs have increased. We only reduce t_τ if $t_\tau > t_{\tau-1} + \Delta$, and increase t_τ with Δ, if the reduction of t_τ have led to increased costs. This is repeated until all $(t_1, t_2, \ldots, t_{m-1})$ are unchanged. In this way the critical levels are increased little by little as s decreases and the values of $(t_1, t_2, \ldots, t_{m-1})$ decrease. The optimal RTR policy is given by the value of Q and s that gives the lowest cost and the corresponding values of $(t_1, t_2 \ldots, t_{m-1})$ and $c_{j,\tau}$ for all j, τ.

When N is high (representing a fine approximation of the Poisson process), we can reduce computation time by using a stepsize greater than 1, while searching for the best values of $(t_1, t_2, \ldots, t_m)$. In this paper we have used a stepsize of $\Delta = 0.05N$ which is equal to 25 subintervals. The increase in cost, by doing so, is found to be negligible.

4 Numerical results

In this section we investigate the performance of the restricted time-remembering policy. Our performance measure is the relative cost difference between the optimal rationing policy, as calculated by Melchiors [4], and the restricted time-remembering policy, sometimes measured by the percentage of the gap between the simple and the true optimal policy covered by the RTR policy. First, we investigate how to separate short and long lead times in general, or more precisely how to choose t_1 in the case $m = 2$ for the set of problem parameters described below. In Figure 2 we have depicted the average percentage of gap filled by using an RTR policy where the value of t_1 is considered to be given, rather than a policy variable, for varying values of t_1. We note that there is almost nothing gained by policies with low value of t_1. In general, the need for critical levels is in the last part of the lead time. This is where we are likely to run out of stock and therefore this is where we

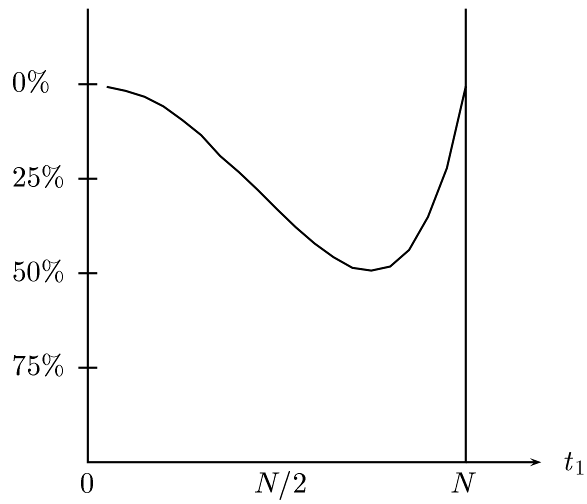

Figure 2: Percentage of gap filled (average over 27 examples), when t_1 is not a policy parameter.

can benefit the most from different critical levels. From the figure we can see that the best values are found around $t_1 = 0.75N$. However, the policies where we can incorporate a 'just before replenishment arrives change' corresponding to high values of t_1, also perform rather well.

Next, we look at the performance of the RTR policies when we optimize all parameters. We consider a base case with $n = 4$ demand classes. All examples have $h = 1$ and $L = 1$, defining the unit time and the monetary unit, respectively. Furthermore, all examples have $K = 100$. We consider three different levels of demand. High demand ($\sum_j \lambda_j = 15$), medium demand ($\sum_j \lambda_j = 10$) and low demand ($\sum_j \lambda_j = 5$). For each of these we consider three different demand allocations. $\boldsymbol{\alpha}^1 = (1/2, 1/4, 1/8, 1/8)$, $\boldsymbol{\alpha}^2 = (1/4, 1/4, 1/4, 1/4)$, $\boldsymbol{\alpha}^3 = (1/8, 1/8, 1/4, 1/2)$. Where for example $\boldsymbol{\alpha}^3$ refers to an allocation where 50% of the demand is from class 4, 25% of the demand from class 3, and classes 1 and 2 represent 12.5% of the demand each. Stockout costs for the four classes are $\boldsymbol{\pi}^1 = (100, 50, 25, 10)$, $\boldsymbol{\pi}^2 = (1000, 500, 100, 10)$, $\boldsymbol{\pi}^3 = (10000, 1000, 100, 10)$.

First, we investigate how the choice of m influences the performance of the RTR policy. The results are found in Table 1. The case $m = 1$ is identical to a simple critical level policy. From the results we see that 51% of the gap is covered when we increase m from 1 to 2. By increasing m from 2 to 3 we can further narrow the gap by additional 20%, but the benefits of increasing m further are very small. Thus, by setting $m = 2$ (or 3), we can capture the essence of the time-remembering policy and still have a policy that is fairly easy to operate in practice.

m	CD	bridged gap
1	1.4%	0%
2	0.70 %	51%
3	0.41 %	71%
4	0.29 %	79%
5	0.27 %	80%

Table 1: The performance of the RTR policy for different values of m. Average over 27 examples.

In Table 2 we report the best policy and the (relative) cost difference CD between the policy and the optimal rationing policy for the simple policy and the RTR policy with $m = 2$. We have also reported the gap bridged by using the RTR policy.

For the investigated examples we observe that the structure of the policies satisfies $c_{j,1} \geq c_j^{simple} \geq c_{j,2}$ as expected. With repects to costs, it appears that the CD of the simple and the RTR policy is lowest when demand is low. Since demand is Poisson, a low demand rate is equivalent to a low demand variance, which means that we rarely end up in the extreme situations covered by the optimal policy. The bridged gap also appears to be lower when the demand rate is low. The allocation of the demand does not appear to have much influence on the performance, but it seems like the bridged gap is slightly higher for demand allocation $\boldsymbol{\alpha}^3$

As $\boldsymbol{\pi}$ increases we see that the cost differences increase, since the importance of using the optimal policy instead of a heuristic increases. This does not seem to influence the bridged gap.

5 Conclusion

We have introduced a new class of policies for the inventory rationing problem, the so-called restricted time-remembering policies. Although much more simple in structure, we have found that the performance of the policies is quite good compared with the optimal rationing policy, even when the number of intervals where the critical levels must be constant are low. We can therefore enjoy the benefit of a time-remembering policy and still have a policy that is easy to implement in practice.

References

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

[2] R. Dekker, J.B.G. Frenk, M.J. Kleijn, and A.G. de Kok. On the newsboy model with a cutoff transaction size. Technical Report 9736/A, Econometric Institute, Erasmus University Rotterdam, The Netherlands, 1997.

[3] A.Y. Ha. Stock rationing in an $M/E_k/1$ make-to-stock queue. *Management Science*, 46(1):77–87, 2000.

[4] P. Melchiors. Rationing policies for an inventory model with several demand classes and stochastic lead times. *Department of Operations Research, University of Aarhus*, 2001.

[5] P. Melchiors, R. Dekker, and M.J. Kleijn. Inventory rationing in an (s, Q) inventory model with two demand classes and lost sales. *Journal of the Operational Research Society*, 51(1):111–122, 2000.

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

[7] E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling.* John Wiley & Sons, New York, 1998.

[8] 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.

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

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

[11] P.H. Zipkin. *Foundations of Inventory Management.* Mc Graw Hill, 2000.

demand	$\boldsymbol{\pi}$	$\boldsymbol{\alpha}$	Simple $(c_1, c_2, c_3, c_4), s, Q$	CD	RTR(m=2) $(c_{\cdot,1})(c_{\cdot,2}), t_1, s, Q$	CD	Bridged gap
Low	$\boldsymbol{\pi}^1$	$\boldsymbol{\alpha}^1$	(0,0,1,3), 8,32	0.38	(0,1,2,4)(0,0,1,2),0.7N, 8,32	0.17	0.55
		$\boldsymbol{\alpha}^2$	(0,0,0,2), 7,33	0.67	(0,0,1,3)(0,0,0,1),0.75N, 7,33	0.25	0.63
		$\boldsymbol{\alpha}^3$	(0,0,0,1), 6,33	0.84	(0,0,1,3)(0,0,0,1),0.7N, 6,33	0.27	0.68
	$\boldsymbol{\pi}^2$	$\boldsymbol{\alpha}^1$	(0,0,2,5),10,33	0.79	(0,1,3,7)(0,0,1,3),0.7N,10,33	0.34	0.57
		$\boldsymbol{\alpha}^2$	(0,0,1,5), 9,33	1.24	(0,1,3,6)(0,0,1,3),0.6N, 9,33	0.49	0.60
		$\boldsymbol{\alpha}^3$	(0,0,1,3), 8,33	1.37	(0,0,1,4)(0,0,0,2),0.75N, 8,33	0.62	0.54
	$\boldsymbol{\pi}^3$	$\boldsymbol{\alpha}^1$	(0,1,3,7),12,32	0.87	(0,2,5,8)(0,1,2,4),0.65N,12,32	0.39	0.55
		$\boldsymbol{\alpha}^2$	(0,1,2,5),11,32	1.69	(0,2,4,7)(0,1,2,4),0.65N,10,33	1.04	0.38
		$\boldsymbol{\alpha}^3$	(0,0,2,4), 9,33	1.97	(0,1,3,5)(0,0,1,3),0.7N, 9,32	0.85	058
Medium	$\boldsymbol{\pi}^1$	$\boldsymbol{\alpha}^1$	(0,0,1,3),15,46	0.80	(0,2,3,5)(0,0,1,2),0.75N,14,46	0.39	0.51
		$\boldsymbol{\alpha}^2$	(0,0,1,2),14,47	0.97	(0,1,3,6)(0,0,1,2),0.7N,13,46	0.51	0.47
		$\boldsymbol{\alpha}^3$	(0,0,0,2),12,47	1.10	(0,0,1,3)(0,0,0,1),0.8N,12,46	0.38	0.65
	$\boldsymbol{\pi}^2$	$\boldsymbol{\alpha}^1$	(0,0,2,6),18,46	1.02	(0,2,5,10)(0,0,2,5),0.7N,17,46	0.65	0.36
		$\boldsymbol{\alpha}^2$	(0,0,1,5),17,46	1.57	(0,1,4,8)(0,0,1,4),0.7N,16,46	0.69	0.56
		$\boldsymbol{\alpha}^3$	(0,0,1,4),15,46	1.81	(0,0,2,6)(0,0,1,3),0.7N,14,46	0.88	0.51
	$\boldsymbol{\pi}^3$	$\boldsymbol{\alpha}^1$	(0,1,4,8),20,46	1.53	(0,4,7,12)(0,1,4,6),0.75N,19,46	0.97	0.36
		$\boldsymbol{\alpha}^2$	(0,1,3,7),18,46	2.08	(0,2,5,8)(0,1,2,4),0.75N,18,46	1.22	0.41
		$\boldsymbol{\alpha}^3$	(0,0,2,5),16,46	2.42	(0,1,3,7)(0,0,1,3),0.8N,15,46	1.26	0.48
High	$\boldsymbol{\pi}^1$	$\boldsymbol{\alpha}^1$	(0,0,2,4),21,57	0.80	(0,2,4,7)(0,0,1,2),0.8N,21,56	0.45	0.44
		$\boldsymbol{\alpha}^2$	(0,0,1,3),20,56	1.00	(0,1,3,6)(0,0,1,2),0.8N,19,57	0.56	0.44
		$\boldsymbol{\alpha}^3$	(0,0,0,2),19,56	1.07	(0,0,2,4)(0,0,0,1),0.85N,18,57	0.48	0.55
	$\boldsymbol{\pi}^2$	$\boldsymbol{\alpha}^1$	(0,0,2,7),25,56	1.24	(0,3,7,12)(0,0,2,6),0.75N,24,56	0.69	0.44
		$\boldsymbol{\alpha}^2$	(0,0,2,6),23,56	1.83	(0,1,4,9)(0,0,1,4),0.8N,23,56	1.00	0.44
		$\boldsymbol{\alpha}^3$	(0,0,1,4),21,57	2.08	(0,0,3,7)(0,0,1,3),0.75N,20,57	1.02	0.51
	$\boldsymbol{\pi}^3$	$\boldsymbol{\alpha}^1$	(0,2,5,10),27,56	1.61	(0,4,8,13)(0,1,4,6),0.75N,26,57	0.96	0.40
		$\boldsymbol{\alpha}^2$	(0,1,3,8),25,56	2.42	(0,3,6,10)(0,1,3,5),0.75N,24,56	1.35	0.44
		$\boldsymbol{\alpha}^3$	(0,1,2,5),23,56	2.67	(0,2,4,8)(0,0,2,4),0.8N,21,57	1.31	0.51

Table 2: For each set of parameters we report the best simple policy, the best RTR policy(m=2), the cost difference $CD = (cost_{-}cost_{optimal})/cost_{optimal}$ and the gap bridged by the RTR policy $=(cost_{RTR} - cost_{optimal})/(cost_{simple} - cost_{optimal})$

Rationing in a congested multi-period make-to-order system

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

July 2001

Abstract

We analyse a make-to-order system with jobs arriving in a stochastic manner. We assume a periodic model where there is a fixed amount of capacity available in each period. There are holding costs for jobs in process and penalty costs for jobs finished after their due date. Jobs have different profitability. The objective is to maximize the long-run average profit minus holding and penalty costs.

We present a near-optimal rationing policy where jobs are accepted or rejected based on the properties of the job and the current state of the system. The computation of the near-optimal policy is intractable for realistically sized problems. We therefore suggest two simple policies which are easy to find and easy to implement in practice. The simple policies are benchmarked against the near-optimal policy on low dimensional problems in a numerical study and are shown to have a fine performance.

Keywords: Capacity rationing, Markov processes, job classes, scheduling.

1 Introduction

Management of capacity is an important issue in almost every imaginable setting and today's labour scarcity has made its mathematical modelling a pertinent challenge attracting widespread interest. An individual's capacity, conceived as a 24-hour span, must be managed to satisfy the needs of professional and private life. The capacity of a consultancy group can be measured by daily working hours, and that of a manufacturing tool by the number of products produced weekly. Capacity may be an asset easily adjustable to meet demand (e.g. by buying another machine or hiring an extra working group), but in many cases capacity is fixed, at least on a short term, and therefore the capacity on hand is often not enough to meet demand. Consequently, all demands cannot be met. Whenever demand for capacity can be divided into classes either due to priority, criticality or profitability, decisions on which demands to accept and which to reject must be taken.

An important aspect when managing capacity is time. Some jobs need to be processed immediately, while others may not be due earlier than a few days or even weeks from the time where we need to accept or reject the job. By postponing the processing of a job until it is due, we can obtain flexibility to accept incoming jobs that require prompt attention. Also, jobs produced early take up space (in case of physical products), or may become outdated. On the other hand, currently available capacity, if not used today, will be lost tomorrow. When to process an accepted job is consequently not straightforward.

Mathematical modelling of such problems is in general very complicated. To obtain optimal decision rules is virtually impossible from a computational point of view and, moreover, the structure of the optimal policy would be very complex, thus making it difficult to implement in practice. It is therefore of great importance to construct simple, tractable models that capture essential system properties, and simple policies for accepting, rejecting and allocating demand for capacity.

This paper considers a make-to-order system with several job classes in a multi-period setting with a rolling horizon. Jobs arrive stochastically and must be either accepted or rejected upon arrival. After acceptance we must decide in which periods to process the job. Every job is described by the amount of capacity required, a due date and a profit per workload unit, which are all known and deterministic. We assume that the average demand for capacity exceeds the capacity on hand. On a longer term, strategic decisions to increase capacity or increase prices to make demand match capacity, should be taken. Our approach assumes that prices and capacity are fixed and focuses on the short term management of the congested system. We divide every period into sub-periods in which there can be at most one demand. In this way every decision regarding acceptance and allocation of a job, can be done with respect to the properties of the one job on hand and the current state of the system. The latter is represented by the unused capacity in every period in the planning horizon. After job acceptance and allocation we reduce the unused capacity according to the chosen allocation. Keeping track of the list of accepted jobs hence becomes superfluous, which significantly simplifies the model.

We analyse two models: Model A is a simplified model, where it does not matter when a job is processed as long as it meets its due date. Model B is a more general model where we do not need to meet the required due date; instead, there are penalty costs per period the job is late and holding costs from the processing has begun until the due date of the job. We present a general framework that provides a decision tool for both models. The strength of the framework lies in its coverage of a wide range of problems, and its capacity to solve these to near-optimality. However, the requirement of computation time and memory is high, and increasing in the length of the planning horizon and the number of different job types. For more complex problems it therefore seems unlikely that an optimal policy can be found, due to "the curse of dimensionality".

Model A is evidently more simple to analyse, and by restricting ourselves to a simple decision policy, we are able to represent the system by a much more simple mathematical model. This facilitates fast optimization, also for problems of higher dimension. We also consider the use of two heuristic policies and compare them with the near-optimal policy. For Model B, it is difficult to reduce the model, and computation times are therefore longer. We design heuristic policies that can be found even for complex real-life problems by the use of simulation. These simple policies are benchmarked against the near-optimal policy on smaller problems (where optimization is possible), and are shown to have a fine performance.

In Section 2 we provide a literature review of related research and in Section 3 we describe the general framework in detail. In Section 4 and 5 we show how to solve Model A and B, respectively. Numerical results are provided in both sections to illustrate the performance of the different policies. All policies are also compared with a non-rationing policy where all jobs are accepted if possible. Concluding remarks are given in Section 6.

2 Literature review

The importance of using admission control for make-to-order systems to control the performance of the system has been recognized by several authors. Hendry and Kingsman [8, 9] build a hierarchical system to control lead times in make-to-order companies. This is achieved by using a customer enquiry system based on the amount of the total backlog in the system. If this amount exceeds a maximum limit determined by management, orders are rejected or extra capacity is purchased. Once accepted, the jobs enter the job-pool and are later released to the shop floor. The released backlog length is controllable, and the authors show the relation between this length and the shop floor throughput time, which is the measure of performance. An option for high-priority jobs which are allowed to skip queues, is included in the analysis, but demand differentiation is not their primary concern. Balakrishnan, Sridharen and Patterson [2] consider capacity rationing for a make-to-stock system with two demand classes. There is a fixed amount of capacity to be allocated to demand during the selling season. By a decision-theory-based approach they calculate, at any given time during the season, how much capacity to reserve for future high-priority demand. Remaining unused capacity can be used to satisfy low-priority demand. The performance of the decision tool is evaluated by simulation. Being a one-period model it does not include the option of allocating demand for capacity to different periods, and only two different demand classes are considered. Balakrishnan, Sridharen and Patterson [1, 15] make further investigations of capacity rationing. Common for all three papers is, however, that they use a non-rationing policy that accepts all demand as the only benchmark for the rationing policy. In the highly congested systems they consider, we conjecture that a simple policy that rejects all low-profit jobs, would have a performance close to that of the optimal policy.

Essential to the problem characterization is the perishability of the capacity; unused capacity of yesterday represents no value today. On the other hand, it is also important not to allocate too much capacity to demand classes with low profitability if this means that there will be no capacity left for demand classes with high profitability. This aspect is central in the problem faced by airlines or hotels, known as perishable asset revenue management (PARM, see Weatherford and Bodily [17]). The capacity held by airlines is the actual number of seats on a given flight. Customers are typically divided into business and tourist class, paying different fares for essentially the same service. Bitran and Mondschein [3] consider an application to the hotel industry with several classes of customers and rooms, allowing upgrading and requests for multiple nights. The decision tool is based on a linear maximization program where the stochastic customer arrivals are replaced with their expected values. They compare the heuristic with an upper bound, found by assuming full information about future demand and find that the heuristic lies within 2% of the upper bound. What distinguishes PARM from the make-to-order system is that customers who require a flight or a hotel room Saturday, will (often) not accept a flight/room Friday or Sunday, whereas in the make-to-order system we can use capacity of adjacent periods to satisfy demand and in that sense we can store our capacity or put it on backorder.

The issue of several demand classes is treated in other settings than PARM. For make-to-stock or inventory systems, demand classes are distinguished by their stockout cost, and the objective is to minimize holding and penalty cost rather than maximize profit. A stream of research deals with rationing in a batch ordering environment: Nahmias and Demmy [14] analyse a simple critical level policy in a continuous review setting with Poisson demand and two demand classes. Whenever the inventory level is below the critical level, only high-priority demand is satisfied. In this way low-priority demand is rejected in order to meet possible future high-priority demand. Melchiors, Dekker and Kleijn [13] show how to optimize such a system under the assumption of lost sales. Optimal time-remembering policies and the extension to several demand classes is considered by Melchiors [12], who allows rationing decisions to depend both on the inventory level and on the time elapsed since the outstanding order (if any) was issued. For systems where the cost of ordering is small or negligible, a base-stock policy is often used instead of batch-ordering. Ha [6, 7] analyses base-stock policies with rationing in a production environment under the assumption of exponential and Erlang distributed production times, respectively. In this paper we apply the ideas for constructing simple critical level policies to the make-to-order problem.

The modelling of different demand classes is central for the problem formulation. We assume that prices and due dates are fixed and non-negotiable; instead, we allow rejection of arriving jobs. Johansen [10] presents a job-shop model where, at every job arrival, a price is calculated based on the current state of the system and the workload of the incoming job. If this price is less than the reservation price of the customer, the job is accepted, otherwise the job is lost. The state space of this model is the amount of accepted workload waiting to be processed, similarly to the model presented in Section 4 of this paper. Simple pricing strategies where only the size of the job determines the price are also considered. ElHafsi [5] provides a decision tool for a manufacturing system to quote lead times and prices for incoming orders, based on the actual congestion level of the system and the expected operating cost.

The make-to-order problem is naturally related to queuing theory. Both are characterized by the arrival of a job, a potential waiting time, followed by a service time. However, our model assumes that service times vary, but are known at the arrival of the job, while most queuing models assume that the variability is due to a stochastic service time, which is first revealed when the service is completed. Consequently, it cannot be taken into consideration when deciding whether to accept or reject a job. It also seems that an analysis of the multi-period make-to-order system based on queuing theory would be very difficult.

3 The general model

We consider a make-to-order system receiving jobs from a set of customers. Each job type j is described by its workload W_j, a desired due date DD_j, and the profit p_j per unit workload. The profit equals the price minus all costs of labor and materials necessary to process the job, but does not include holding and penalty costs. The arrival of jobs follows a stochastic process, which we will specify later. We let J denote a set of job types. Eventually some job is rejected, either because of insufficient capacity or due to a rationing decision. We assume that the cost of rejecting a customer (loss of goodwill etc.) is zero. A positive rejection cost can easily be incorporated by adding it to the profit of an accepted job, instead of adding a negative profit when a job is rejected.

There is a holding cost hp_j per unit workload per period from the day processing of a type j job is begun until its due date. If processing is finished after the due date a penalty cost of πp_j per unit workload per period late is incurred. We note that there are other ways in which these cost could have been specified. If the penalty costs are a result of overtime production, these costs are typically not dependent of the profit of the job. If, however, the penalty cost is based on a contractual agreement it seems reasonable that a high-profit job has a high penalty for late delivery. Also, in many situations customers will be more than happy to have their job finished early. This is analysed as a special case in Section 4. However, it is easy to imagine situations where customers do not want their products before they need them. Indeed, this is one of the key elements of the Just-In-Time philosophy.

The time unit is a period (representing a day or a week for example) which is divided in T sub-periods (hours of the day or weekdays). Incoming jobs can arrive in any sub-period, but cannot begin processing before the next period. We consider a rolling horizon with a length of N periods. The current period is period 0, and the first period a job can be processed in is period 1. The length of the planning horizon is fixed, and we can only accept jobs that can be processed within this horizon.

When a job is received, it must either be accepted or rejected. We assume that all jobs can be processed over several periods without additional costs (besides the holding costs). The allocation of workload is done upon the acceptance of the job. We assume that it is not possible to change this allocation. Our reason for this, albeit, restrictive assumption, is the modelling of our state-space. By postponing the allocation or allowing reallocation (which is the same) we have to keep track of the list of accepted jobs and their properties. By allocating the workload of a job upon its arrival, we only need to keep track of the available capacity in the system. Specification of decision policies is much easier to formulate based on available system capacity, than on a list of jobs waiting to be allocated. There is, however, no doubt that the system that allows postponing of allocation, will have a better performance than the system we analyse, simply because we can wait until last minute before deciding what to process in the following period, and never make allocation plans for more than the first period in the planning horizon. A study that explores the cost of not being able to reallocate would be very interesting, but is beyond the scope of this paper. In practice, if the scheduled date where processing begins is near, orders for materials are placed and setup of machines may have begun, and consequently rescheduling will be very expensive, which must also be taken into consideration.

We assume that there exists a minimum unit of workload. This unit may represent one hour or one day, either for a person or for a work group. The capacity of the system in each period is C workload units. Furthermore, we assume that the workload of each job can be expressed as an integral multiple of this unit. We investigate two different models:

- MODEL A : $\pi = \infty$ and $h = 0$. This is a simplified situation, where it is irrelevant when a job is processed as long as it meets the required due date.
- MODEL B : The general model as described above with $\pi < \infty$ and $h > 0$.

A job that can be processed within the planning horizon, or if $\pi = \infty$ before its due date, is called a feasible job. Naturally, we can only accept feasible jobs. However, as discussed earlier there are circumstances under which we would be better off by rejecting a feasible job. We use one of the following policies to specify whether arriving jobs should be accepted or rejected.

- Non-rationing policy: Accept all feasible jobs.
- Selective policy: Accept all feasible jobs if their profit per unit workload exceeds a fixed policy-specific value. Acceptance does not depend on the actual state of the system.
- Simple rationing policy: Acceptance is based on the actual state of the system and the properties of the incoming job. However, the decision criteria must be a simple one. We suggest two different simple policies for Model A and B, respectively.
- Near-optimal rationing policy: Jobs are accepted based on their properties and the actual state of the system.

The problem is formulated as a Markov decision process. Let t denote the number of remaining sub-periods of period 0 and let x_i denote the unused capacity in period i. The state of the system is then described by $(t, \boldsymbol{x})$, where $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$. Moreover, the process $\{t, \boldsymbol{x}\}_k, k = 1, 2, \ldots$ is a discrete time Markov chain with state space $S = \{0, 1, \ldots, T\} \times \{0, 1, \ldots, C\}^N$. In all states $(t, \boldsymbol{x})$ with $t > 0$ we decide which jobs we can accept and how to allocate their workload. States $(0, \boldsymbol{x})$ are artificial states where no job can arrive, representing the end of a period. Let $\boldsymbol{\alpha}$ be a vector of components α_n, which is the workload allocated to period n for a job. An allocation for an accepted job with workload W must satisfy

$$0 \leq \alpha_n \leq x_n \text{ for all } n = 1, 2, \ldots, N$$

and

$$\sum_{n=1}^{N} \alpha_n = W.$$

We let the non-allocation with $\alpha_n = 0$ for all n denote a rejected job. Let $\mathcal{A}(\boldsymbol{x}, j)$ denote the set of feasible allocations, including the non-allocation, for job j in state $(t, \boldsymbol{x})$. From all states $(t, x_1, x_2, \ldots, x_N)$ with $0 < t \leq T$ and all jobs j with allocation $(\alpha_1, \alpha_2 \ldots, \alpha_N)$ we jump to state $(t-1, x_1 - \alpha_1, x_2 - \alpha_2, \ldots, x_N - \alpha_N)$. In the states $(0, x_1, x_2, \ldots, x_n)$, there are no jobarrivals and we jump to the beginning of the next period, to state $(T, x_2, x_3, \ldots, x_n, C)$. Remaining unused capacity in period one is lost.

Let $f(j, t)$ denote the probability that job j arrives in state $(t, \boldsymbol{x})$. We assume that there can only arrive one job per sub-period. If the arrival process is a Poisson process we can approximate the real arrival process by a Bernoulli process, by choosing the number of sub-periods such that the probability of more than one incoming job per sub-period is negligible (see e.g. Çinlar [4]). The situation where no job arrives is represented by $j = 0$, included in J. Consequently, $\sum_{j \in J} f(j, t) = 1$ for all t. For technical reasons, we furthermore assume that $f(0, t) > 0$ for all t. Let $c(\boldsymbol{\alpha}, j)$ be the profit minus the holding and penalty cost for a job j and an allocation $\boldsymbol{\alpha}$. The net profit functions for models A and B are specified in Sections 4 and 5.

Our model is similar to the vehicle insurance example given by Tijms [16]. He considers a model where accidents occur during the months of a year. If the motorist has made no claims by the end of the year, he is rewarded with a lower premium the following year. The premium is assumed only to depend on the number of claims made during the year and the premium of the previous year. The decision of whether to claim an accident or not, will then depend on the size of the claim, the claim history of the current year and the premium of the previous year. A value-iteration algorithm is used to find optimal no-claim limits for the example. The model is similar to ours in the sense that accidents (demands) occur during the months (the sub-periods) and claims (acceptance) must be made when the accidents occur. By the end of the year (period) the consequences of the actions taken occur. The motorist finds himself with a lower premium if he has made no claims, and similarly our make-to-stock system will have more free capacity, if we have rejected jobs during the sub-periods. The vehicle insurance model balances the cost of an un-claimed accident with the benefit of a lower premium in the following year, while our model balances the loss in profit by rejecting a job with the benefit of more free capacity in the following periods.

We find a policy, that gives the maximal average net profit per period by a value-iteration algorithm (see Tijms [16]). Let $v_i(t,\boldsymbol{x})$ be the maximum expected future net profit obtained i periods and t sub-periods from the end of the horizon in state $(t,\boldsymbol{x})$. $v_i(t,\boldsymbol{x})$ is found by the recursion

$$\begin{aligned} v_i(t,\boldsymbol{x}) &= \sum_{j\in J} f(j,t) \max_{\boldsymbol{\alpha}\in\mathcal{A}(\boldsymbol{x},j)} \Big[c(\boldsymbol{\alpha},j) + v_i(t-1,\boldsymbol{x}-\boldsymbol{\alpha})\Big] \\ &\quad \text{for } 1 \le t \le T, \forall \boldsymbol{x} \ge \mathbf{0}, \forall i > 0 \qquad (1) \\ v_i(0,x_1,x_2,\ldots,x_N) &= v_{i-1}(T,x_2,x_3,\ldots,x_N,C)\ \forall\ x_1,x_2,\ldots,x_N \ge 0,\ \forall i > 0 \\ v_0(T,\boldsymbol{x}) &= 0\ \forall \boldsymbol{x} \ge \mathbf{0}. \end{aligned}$$

How to solve the maximization problem is model specific and will be discussed in the following sections.

The Markov chain is obviously periodic. But as for the vehicle insurance example, we can convert it into a aperiodic model by only considering the states $(T,\boldsymbol{x})$. In this way we regard what happens during the sub-periods as a black box, and focus on the system only in the beginning of a period, in the states $(T,\boldsymbol{x})$ for all $\boldsymbol{x}$. In line with this, we have chosen to index the value functions only by the subscript i, instead of the strictly speaking more correct $iT+t$. The difference lies in the notation only, and therefore we have chosen to use the simpler of the two.

Define m_i and M_i by

$$m_i = \min_{\boldsymbol{x}} \left(v_i(T, \boldsymbol{x}) - v_{i-1}(T, \boldsymbol{x}) \right)$$

and

$$M_i = \max_{\boldsymbol{x}} \left(v_i(T, \boldsymbol{x}) - v_{i-1}(T, \boldsymbol{x}) \right).$$

The Markov chain is aperiodic and since $f(0,t) > 0$, the Markov Chain cannot have two disjoint closed sets. By Theorems 3.4.1 and 3.4.2 of Tijms [16] we then obtain that the maximal long-run average net profit g per period satisfies $m_i \leq g \leq M_i$ for all $i \geq 1$, and furthermore, there exist finite constants $\kappa > 0$ and $0 < \beta < 1$ such that $|M_i - m_i| \leq \kappa\beta^i$.

In each iteration i of the algorithm, for all states $(t, \boldsymbol{x})$, we need to solve the maximization problem for all jobs $j \in J \setminus \{0\}$. In this way we make a list of how to treat an incoming job j in state $(t, \boldsymbol{x})$. For every job j in every state $(t, \boldsymbol{x})$ we use an algorithm to find a near-optimal allocation $\boldsymbol{\alpha}$ of the job (under the assumption that it will be accepted). This algorithm seeks to maximize $c(\boldsymbol{\alpha}, j) + v_i(t-1, \boldsymbol{x} - \boldsymbol{\alpha})$ subject to $\boldsymbol{\alpha} \in \mathcal{A}(\boldsymbol{x}, j)$. If the largest computed sum is greater than $v_i(t-1, \boldsymbol{x})$, the job is accepted and otherwise the job is rejected. The search algorithm is described for each model in the following sections. $\mathcal{A}(\boldsymbol{x}, j)$ consists of the empty allocation, representing the rejection of a job, and the set of allocations that can be found by the search algorithm. Due to the size of the decision space we cannot in general guarantee to find the optimal allocation, and therefore we refer to our policy as a near-optimal policy.

In iteration i, $i > 1$, after all values $v_i(t, \boldsymbol{x})$ are calculated, we compute m_i and M_i. If $(M_i - m_i)/m_i < \epsilon$, where ϵ is a small positive number, the algorithm terminates, otherwise another iteration is performed. The expected profit is found as $(m_i + M_i)/2$. For the computations in this paper we have used $\epsilon = 0.0001$.

A disadvantage of the near-optimal policy is that it is difficult to implement in practice. A simple policy for example specifying that jobs are only accepted if they can meet their due date, or if their profit is higher than a certain amount, would be much easier to implement. The value-iteration algorithm cannot be used directly to find simple policies. Under the regime of a simple policy, the decisions are bound by the parameters of the simple policy, and we cannot therefore maximize the expected net profit $v_i(t, \boldsymbol{x})$ independently in each state. The algorithm can, however, be used to evaluate any specified policy. Given a policy that, for any state of the system $(t, \boldsymbol{x})$ and any incoming job j, specifies an allocation $\boldsymbol{\alpha}(\boldsymbol{x}, j)$, we simply set $\mathcal{A}(\boldsymbol{x}, j) = \{\boldsymbol{\alpha}(\boldsymbol{x}, j)\}$, such that there is only one element in $\mathcal{A}(\boldsymbol{x}, j)$. In this way we can evaluate simple policies. To find good simple policies we therefore first design the structure of the simple policy (e.g. only allow jobs with a profit above some minimum level). The value-iteration algorithm does not perform any maximization since there is only one prespecified action in $\mathcal{A}(\boldsymbol{x}, j)$; it only evaluates a given policy. Therefore we apply a local search algorithm to determine the best value of the policy variables.

In the two following sections we show how to find near-optimal policies and design simple policies for Model A and Model B.

4 Model A

In this section we consider the special case of the general model where $\pi = \infty$ and $h = 0$. The problem when a job arrives is whether it should be accepted or not and when to process it. Here we are not influenced by holding and penalty costs forcing us to process the job near the desired due date, i.e. we are free to process the job whenever we prefer, as long as the due date is met.

The net profit of a job of type j with feasable allocation $\boldsymbol{\alpha}$ equals

$$c(\boldsymbol{\alpha}, j) = p_j W_j$$

if the job is accepted, and $c(\mathbf{0}, j) = 0$ if the job is rejected. Now, we consider how to solve the maximization problem (1) by finding the best allocation of workload. Consider the maximization problem in iteration i in state $(t, \boldsymbol{x})$, where we are to allocate a feasible job with workload W and due date DD. If the job is infeasible, we choose the non-allocation. If there is no limitation on capacity, the number of ways to allocate W units of workload to DD periods is equal to the number of ways to place $DD - 1$ separators, in a string consisting of W units of workload and $DD - 1$ separators which is given by the binomial $\binom{W+DD-1}{DD-1}$. With $W = 10$ and $DD = 5$ this gives 1001 combinations and if both the number of periods and the number of workload units is 10, there is 92387 possible combinations. In practice many of these would not be feasible due to limited capacity, and most of them would be far from optimal. Instead of searching among all possible allocations we therefore consider two allocation heuristics: The first algorithm, FIFO(first in first out), allocates such that everything is processed as soon as possible. Let $\boldsymbol{\alpha}_{FIFO}(\boldsymbol{x}, j)$ be the FIFO allocation for job j given a vector of free capacity $\boldsymbol{x}$.

The second algorithm, MI(marginal improvement), is a greedy near-optimal method based on marginal improvements of the expected profit function $v_i(t, \boldsymbol{x})$. Let $\boldsymbol{y} = \boldsymbol{x}$ and let $\boldsymbol{e_n}$ be a vector of zeroes except for the n'th entry which is 1. In each iteration of the algorithm we allocate a unit of workload to the period n ($y_n > 0$), with the highest value of $v_i(t-1, \boldsymbol{y} - \boldsymbol{e_n})$, and let $\boldsymbol{y} := \boldsymbol{y} - \boldsymbol{e_n}$. This iteration is repeated until all workload is allocated. The near-optimal allocation is given by $\boldsymbol{\alpha} = \boldsymbol{x} - \boldsymbol{y}$. Since we cannot guarantee that we find the optimal allocation (this would require $v_i(t, \boldsymbol{x})$ being a concave function), we call the policy found when using the MI allocation a near-optimal policy.

The MI algorithm is constructed with the purpose of having the option of leaving idle periods to increase flexibility. The benefit of leaving idle periods for future high-profit demand, may not offset the loss in profit incurred when the idle periods remain idle and capacity is lost. In that case the performance of the FIFO allocation would be just as good. To prove this theoretically we would have to show that

$$v_i(t,\ldots,x_n-1,\ldots,x_m,\ldots) > v_i(t,\ldots,x_n,\ldots,x_m-1,\ldots)$$

for all $n \neq m$ with $x_n, x_m > 0$ and for all $i > 0,\ 1 \leq t \leq T$. Or in words: The expected future profit is highest if a unit of workload is allocated to the first period with available capacity. As we shall see in our numerical results, this is not true in general.

If we use the FIFO allocation method, we can reduce our mathematical model significantly. Instead of keeping track of available capacity for each day, it suffices to keep track of the total amount of accepted workload waiting to be processed. The state of this system is (t,x), where t denotes remaining sub-periods of period 0 and x denotes, instead of a vector of unused capacity, the sum of all accepted workload not yet processed. The process $\{t,x\}_k,\ k = 1,2,\ldots$ is a discrete Markov chain with state space $\{0,1,\ldots,T\} \times \{0,1,\ldots,C\cdot N\}$, since we can at most accept C units of workload per period in the planning horizon. In states (t,x) with $t > 0$, if a job j is accepted we jump to state $(t-1, x+W_j)$, and if no jobs are accepted we jump to state $(t-1,x)$. In states $(0,x)$ we jump to $(T,(x-C)^+)$. $(x-C)^+$ equals $x-C$, if $x-C$ is positive, and zero otherwise. A job j is feasible only if $x + W_j \leq C\cdot DD_j$.

The model is solved by a value-iteration algorithm. Let $w_i(t,x)$ be the maximum expected net profit obtained i periods and t sub-periods from the end of the horizon in state (t,x). Let a be the decision variable, being W_j if a job j is accepted and 0 otherwise.

$$\begin{aligned}
w_i(t,x) &= \sum_{j\in J} f(j,t) \max_{a\in\{0,W_j\}} \Big[p_j a + w_i(t-1,x+a) | x+a \leq C\cdot DD_j\Big] \\
&\quad \text{for } 0 < t \leq T,\ 0 \leq x \leq C\cdot N \text{ and } \forall i > 0 \qquad (2) \\
w_i(0,x) &= w_{i-1}(T,(x-C)^+) \text{ for } 0 \leq x \leq C\cdot N \text{ and } \forall i > 0 \\
w_0(T,x) &= 0\ \forall x
\end{aligned}$$

If we consider only states (T,x) the Markov chain is aperiodic as argued in Section 3. The convergence properties therefore also apply to this model as well. Due to the reduced state space, the computation time of this model is much smaller. The optimal policy found by this reduced model is called rationing(FIFO) and is identical to the policy found using the FIFO alloaction in the full-dimensional model.

Regardless of which model is used, the found policy may be difficult to implement in practice since the rationing decision depends on which sub-period the system is in when a job arrives. For practical implementations this may not be desirable, and consequently we consider how to design and optimize simple policies.

4.1 Heuristic policies

We investigate two heuristics: A simple policy and a selective policy. When analysing these policies we will use the FIFO allocation rather than the MI allocation, since a simple policy cannot be based on the $v_i(t, \boldsymbol{x})$ function. Moreover, this allows the use of the reduced state-space model. We divide the job types into job classes based on the profit the jobs provide. Let J_i be the set of job types in class i, and let $\tilde{p}_i$ be the profit of a job from class i. Let m be the number of job classes and order the classes such that $\tilde{p}_i > \tilde{p}_{i+1}$ for all $i < m$. We investigate a critical level policy, where a critical level c_i is assigned to every job class i. A job type j is accepted only if the resulting accepted workload $x + W_j$ does not exceed the critical level of its class. If a job is accepted, it is scheduled to be processed as soon as possible. In this way we obtain a controlled backlog length similar to that of Hendry and Kingsman [9].

The value-iteration algorithm is used to evaluate a given policy as described in the previous section. We use a search routine similar to that of Melchiors [12] to find a good simple policy. Let $\boldsymbol{c} = (c_2, c_3, \ldots, c_m)$ be a vector of critical levels. Initialize by setting $c_j = C \cdot N$ for $2 \leq j \leq m$, which corresponds to a situation with no rationing (since we can only accept jobs that can be processed within our planning horizon). Let $k = m$. In each iteration of the algorithm we reduce c_k by one and evaluate the new policy. If the net profit increases, we let $k := k - 1$. This is repeated until the net profit decreases or until $k = 2$ (we will not reject a feasible class 1 job). Set $\boldsymbol{c}$ equal to the best of the critical level vectors evaluated so far and repeat the process with $k = m$ again, until no improvement can be found.

While searching we can make use of the upper bound M_i on the profit function in the i'th iteration when evaluating a policy. Let g' denote the net profit of the best policy found so far. If in any iteration i of the value-iteration algorithm, $M_i < g'$, the evaluation can be abandoned, since the profit of the policy under evaluation is at most M_i.

We also consider the use of an even simpler policy: a selective policy. Since we mainly are concerned with congested make-to-order systems, a good heuristic would be to be more selective when accepting jobs, and only accept jobs of certain job classes, or in other words reduce the set of job types. This would mean that fewer high-profit jobs would be rejected. If a job from class i is accepted, so should a job from class $i - 1$, and we can therefore by a simple search, using the evaluation method described previously, determine from which classes to accept jobs. The selective policy is not a real rationing policy, but simply a reduction of the set of job types, where after all remaining feasible jobs are accepted. It is possible to extend this class of policies by allowing randomized policies, such that a job is accepted only with a certain class dependent probability. This is, however, not studied in this paper. Finally, we also consider the use of non-rationing policies for comparison. The non-rationing policy specifies to accept every job that can be processed within its due date, regardless of the profit it provides. Accepted jobs are allocated by the FIFO principle.

4.2 Numerical results

In this section we analyse the performance of the different rationing and allocation policies. Our measure of performance is the average profit obtained by either the rationing(FIFO), the simple, the selective or the non-rationing policy compared with the profit obtained by using the near-optimal policy found using the MI allocation. Let $E(W)$ be the expected demand per period.

$$E(W) = \sum_{t=1}^{T} \sum_{j \in J} f(j,t) W_j$$

The availability of capacity β defined as the ratio of total capacity per period to the expected demand is

$$\beta = C/E(W)$$

We consider a range from $\beta = 0.4$, representing systems where the availability of capacity is low, to $\beta = 1$ representing systems where the capacity equals the expected demand. When the capacity of the system exceeds the expected demand, almost all jobs should be accepted and as β increases, the benefit of rationing policies becomes marginal.

Naturally, the profit obtained from the different job classes is important for the performance of the system; if all jobs provide the same profit, rationing cannot improve the performance of the system. Another factor that influences the performance of the system, is the fraction of total workload demand with high profit. Let

$$\gamma_i = E(W_i)/E(W),$$

denote the fraction of class i demand, where $E(W_i) = \sum_{t=1}^{T} \sum_{j \in J_i} f(j,t) W_j$ is the expected demand from class i per period.

In this section we will investigate five examples all with three job classes, described in Table 1. Example A1 corresponds to a situation where customers that require a small lead time pay more than customers accepting a longer lead time. In example A2, class one and three have been interchanged, and in examples A3, A4 and A5 the workload of the jobs are low with a loose due date, low and with a tight due date or high with a loose due date, respectively. In each example there are 5 different job types.

The capacity per period is $C = 3$, the length of the planning horizon is $N = 5$ periods, and every period is divided in $T = 8$ sub-periods. The availability of capacity β is set to 0.4, 0.55, 0.7, 0.85 and 1. We consider five different sets of profits, such that

$$(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3) \in \{(1.6, 1.2, 1), (2.2, 1.4, 1), (2.8, 1.6, 1), (3.4, 1.8, 1), (4, 2, 1)\}$$

j	A1 W_j, DD_j, JC_j	A2 W_j, DD_j, JC_j	A3 W_j, DD_j, JC_j	A4 W_j, DD_j, JC_j	A5 W_j, DD_j, JC_j
1	1,1,1	3,4,1	1,5,1	1,1,1	4,5,1
2	2,3,2	4,5,1	2,5,1	2,2,1	5,5,1
3	3,4,2	2,3,2	2,5,2	2,2,2	4,5,2
4	3,4,3	3,4,2	1,5,3	1,1,3	4,5,3
5	4,5,3	1,1,3	2,5,3	2,2,3	5,5,4

Table 1: Description of job workload(W_j), duedate(DD_j) and jobclass(JC_j) for examples A1, A2, A3, A4 and A5

Finally, we investigate five different distributions of demand, covering situations where jobs have primarily low profit to situations where jobs have primarily high profit.

$$(\gamma_1, \gamma_2, \gamma_3) \in \{(0.1, 0.3, 0.6), (0.2, 0.3, 0.5), (0.3, 0.4, 0.3), (0.5, 0.3, 0.2), (0.6, 0.3, 0.1)\}$$

This gives a total of $5 \times 5 \times 5 = 125$ different cases for each example. We assume that all jobs types in a job class i have uniform probability, such that $f(j,t) = f(k,t)$ for all job types $k, j \in J_i$. Moreover, we let the demand be stationary during every period, i.e. $f(j,t) = f(j,s)$ for all $j \in J$ and for all t, s. The equations defining $\beta, \gamma, E(W_i)$ and $E(W)$ then determine unique values of $f(j,t)$ for all t and all $j \in J$. In Table 2

	A1	A2	A3	A4	A5
rationing(FIFO)	97.1 (91.8)%	99.3 (96.9)%	100 (100)%	99.9 (99.6)%	100 (100)%
simple	96.0 (90.9)%	99.1 (96.9)%	99.9 (99.7)%	98.5 (92.3)%	99.8 (99.4)%
selective	94.7 (83.8)%	98.6 (94.8)%	96.0 (85.6)%	96.7 (86.9)%	96.3 (84.7)%
non-rationing	80.7 (42.5)%	98.3 (91.0)%	89.1 (69.2)%	93.1 (73.8)%	91.7 (72.5)%

Table 2: Average net profit for each heuristic as percentage of the near-optimal net profit found by the MI allocation for 5 examples of job type combinations described in Table 1. The worst case performance found in the 125 cases is reported in parenthesis for each heuristic.

we report the profit obtained for each example for each heuristic, as percentage of the profit of the near-optimal policy as found by the MI allocation. The value reported is the average performance over the 125 different combinations of capacity availability, profit and workload distribution. We moreover report the worst performance found. Except for Example A2 the results show that rationing in general is very useful compared with a non-rationing approach. The difference between the near-optimal policy and the selective policy is 4-5 % on average but can be as high as 16.2%. The non-rationing policy is in general very poor, with an average performance around 90% and a worst performance of 42.5%. In general the performance of the rationing(FIFO) policy is very good, indicating that the benefit of leaving idle periods for flexibility is low. The performance of the simple policy is almost equally good. Except for example A1, the simple policy is very close to the near-optimal policy, and due to its simplicity it is well-suited for practical implementation. In example A1 the properties of the jobs are such that it is important to reserve idle capacity in period 1 in order to meet the profitable class one jobs. This is expensive if we restrict ourselves to the simple policy, where we cannot reject demand for period 1 capacity in the early sub-periods of period 0, and accept similar demands by the end of period 0, if no class one demands have occurred.

The reason behind the relative poor benefit of rationing in example A2, is the high flexibility of the class one customers, which means that there almost always is available capacity for class one jobs,

The impacts of the different underlying factors are reported in Table 3 for example A1 as average values over all cases. The rationing(FIFO) policy is best when available

β	0.4	0.55	0.70	0.85	1
Rationing(FIFO)	99.0 %	98.0 %	97.1 %	96.1 %	95.3 %
Simple	97.5 %	96.5 %	95.9 %	95.3 %	94.8 %
Selective	96.6 %	95.1 %	94.1 %	93.7 %	93.9 %
Non-rationing	63.2 %	74.7 %	84.0 %	89.5 %	92.0 %
$(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3)$	(1.6,1.2,1)	(2.2,1.4,1)	(2.8,1.6,1)	(3.4,1.8,1)	(4,2,1)
Rationing(FIFO)	97.3 %	97.0 %	97.0 %	97.0 %	97.1%
Simple	96.6 %	96.0 %	95.8 %	95.7 %	95.8 %
Selective	95.7 %	94.5 %	94.3 %	94.4 %	94.5 %
Non-rationing	90.7 %	84.2 %	79.4 %	75.9 %	73.2 %
$(\gamma_1, \gamma_2, \gamma_3)$	(0.1,0.3,0.6)	(0.2,0.3,0.5)	(0.3,0.4,0.3)	(0.5,0.3,0.2)	(0.6,0.3,0.1)
Rationing(FIFO)	94.6 %	95.5 %	97.0 %	98.9 %	99.4 %
Simple	93.5 %	93.8 %	95.6 %	98.2 %	98.8 %
Selective	90.6 %	92.0 %	94.5 %	97.8 %	98.5 %
Non-rationing	81.4 %	75.9 %	77.4 %	82.0 %	86.8 %

Table 3: Example A1: Average net profit as percentage of the near-optimal net profit for the rationing(FIFO), the simple, the selective and the non-rationing policy, as function of available capacity β, profits $(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3)$ and distribution of demand $(\gamma_1, \gamma_2, \gamma_3)$.

capacity is low and when most jobs have high profits. When demand significantly exceeds the capacity, it is too expensive to reserve idle periods, which means that the FIFO allocation is a good choice.

Let us now compare the rationing policies with the non-rationing policy and the selective policy. As expected, the benefit of rationing compared with non-rationing is higher when there is less capacity available; when capacity is plenty, we should accept most jobs. As the difference in profit increases, so does the the benefit of rationing compared with the non-rationing policy, which seems very intuitive. With respect to the distribution of demand, rationing compared to non-rationing seems equally important regardless of whether the majority of the jobs have high or low profit. Compared with the selective policy, however, rationing is particularly important when the demand primarily consists of low-profit jobs. This corresponds to a situation where high profit jobs occur only once in a while. A selective policy will either only accept these jobs and shut down low-profit production or accept all jobs, while a rationing policy would be able to accept most low-profit jobs, while reserving a fraction of the capacity for possible future high-profit jobs. It is therefore in particular in those situations that the use of rationing policies is important.

5 Model B

In this section we consider the more general case where each job carries a positive holding cost hp_j per unit workload per period until its due date, and a penalty cost πp_j per unit workload per period late.

Every state is represented by the vector $(t, x_1, x_2, \ldots, x_N)$. Typically, penalty cost must be paid for the entire workload for every period the last unit of workload is late. According to the specific situation the holding cost can be specified in different ways. Here, holding costs are charged for the entire workload of the job from the period the job is begun until its due date. This corresponds to a situation where the cost of materials needed to process the job constitutes the major part of the value of the final product. Consider a job with j and an allocation $\boldsymbol{\alpha}$. Let $n_{min} = \min\{n|\alpha_n > 0\}$ and $n_{max} = \max\{n|\alpha_n > 0\}$. The net profit earned by accepting the job and choosing allocation $\boldsymbol{\alpha}$ is the profit minus the holding and penalty costs:

$$c(\boldsymbol{\alpha}, j) = p_j W_j - hp_j W_j (DD_j - n_{min})^+ - \pi p_j W_j (n_{max} - DD_j)^+$$

If the job is rejected, $c(\mathbf{0}, j) = 0$. Our framework allows us to calculate near-optimal policies which are, however, difficult to implement in practice and, moreover, computationally demanding if not intractable for larger problems. These are therefore primarily used as a benchmark to heuristic policies that are found easily, even for larger problems, and are much easier to implement in practice.

Let us briefly discuss the allocation methods of Section 4. The FIFO allocation neglects the holding cost, but is besides that expected to perform fairly well. The MI allocation has a drawback since it is based on local neighbor search. Consider a case where $W = 3$, $DD = 4$ and $\boldsymbol{x} = (0,0,0,2,5)$. Initially MI would most likely allocate 2 units to period 4 in order to avoid penalty costs. However, we cannot avoid processing the third unit late. In step 2 of the algorithm we would try to re-allocate one of the two units to period 5. This will not lead to a reduction in holding and penalty costs, and depending on the values $v_i(t,0,0,0,0,5)$ and $v_i(t,0,0,0,1,4)$, we will not move further in this direction. However, knowing that the job cannot be processed in due time, we can reduce our holding cost by assigning the entire workload to period 5, without increasing the penalty costs.

We therefore use another allocation algorithm BSP (best starting period). As observed in Section 4, the benefit of leaving idle capacity for future demand is only marginal, and therefore once a job has begun processing, we will finish it as fast as possible.

An allocation for a job j will then be on the form

$$(0,\ldots,0,\alpha_n,\alpha_{n+1},\ldots,\alpha_{n+\tau_n},0,\ldots,0)$$

with $\sum_{k=n}^{n+\tau_n} \alpha_k = W_j$, where τ_n denotes the number of periods needed to process the workload when processing starts in period n. For each value of n we will consider two allocations: one where $\alpha_k = x_k$ for $k = n, n+1, \ldots, n+\tau_n - 1$, and $\alpha_{n+\tau_n} = W_j - \sum_{k=n}^{n+\tau_n-1} x_k$, and one where $\alpha_k = x_k$ for $k = n+1, n+2, \ldots, n+\tau_n$ and $\alpha_n = W_j - \sum_{k=n+1}^{n+\tau_n} x_k$. The two combinations are checked for all values of n with $n + \tau_n \leq N$. $\mathcal{A}(\boldsymbol{x}, j)$ thus consists of these two combinations for all values of n with $n + \tau_n \leq N$ plus the non-allocation, corresponding to a rejected job . For every allocation we find $c(\boldsymbol{\alpha}, j) + v_i(t-1, \boldsymbol{x} - \boldsymbol{\alpha})$ and choose the allocation with the highest expected net profit.

When applying the value-iteration algorithm we find that the algorithm needs a considerable number of iterations to converge. Typically, 50-75 iterations are needed.

5.1 Heuristic policies

The complexity of the optimal policy and the high computation times and memory requirements lead us to consider a simple policy which may not be optimal, but instead both simpler to use and easier to find. We consider the use of a policy that works the following way: Each job class is assigned a critical horizon which is a period in the planning horizon. We accept an arriving job, if there is sufficient capacity to process it within the critical horizon of the class to which the job belongs to. The motivation for this criteria is to avoid using capacity on low-profit jobs, as long as there is a reasonable probability that it can be used to satisfy high-profit jobs. If we neglect the influence of holding costs, the critical horizon should therefore be chosen as the period in which the expected profit from future high-profit jobs (i.e. the profit multiplied with the probability of the arrival of at least one high-profit job before the beginning of a period), equals the profit of the job on hand. The optimal critical horizon can be calculated every time a job was on hand, taken all available information into account, but then our requirements of a simple policy will not be met. Instead we will, for each class, compute the fixed value of the critical horizon, that maximize the long-run average net profit per time unit. Then the current state of the system is not used to determine the value of the critical horizons, but to determine whether there is sufficient capacity within the critical horizon to accept the job on hand.

The critical horizon policy is in structure very similar to the simple policy suggested for Model A. For that policy, since all accepted jobs are processed as soon as possible by the FIFO allocation, the state of the system is fully described by the amount of unprocessed workload. When jobs are not processed as soon as possible, we cannot tell, based on the amount of accepted workload only, whether we should accept a low-profit job or not. If there is available capacity in the beginning of the horizon the probability that the capacity can be used to satisfy high-profit job is low, and if the available capacity is at the end of the horizon, the probability is high. This aspect is captured by the critical horizon policy.

Having decided whether or not to accept an incoming job, we need to allocate the accepted workload. Scheduling of accepted jobs is among others treated by Lee and Choi [11] who analyse a deterministic job scheduling problem, where a set of jobs are scheduled in order to minimize early-tardy penalty costs, relative to a job specific due date. Their problem is that of sequencing several jobs; in our model we only need to schedule one job, which is much easier.

Here we consider two allocation algorithms: MC (Minimum cost) and FIFO. The MC algorithm finds the allocation that minimizes the holding and penalty costs, under the restriction that every job must be processed within its critical horizon. Once we begin processing a job, we must continue until it is finished. Idle periods can only lead to increased holding or penalty costs.

If it is possible to process the job in due time, we initially choose the allocation that starts as close to the due date as possible and still finishes in time. If the job cannot be finished in due time, we initially choose the allocation that finishes as close to the due date as possible. We may improve the initial allocation by postponing the processing, if the reduction in holding cost can offset the increase in penalty cost. For example if the allocation of a job with workload 2 and due date 6 is $(1, 0, 0, 0, 0, 1, 0)$, it may be worthwhile choosing $(0, 0, 0, 0, 0, 1, 1)$ even though the job will be late. We therefore successively postpone processing within the critical horizon, and pick the allocation with the lowest cost.

Since the MC allocation policy neglects the perishability of capacity, we also consider the use of a FIFO allocation. We note that when using the FIFO allocation, we could just as well have chosen the critical level policy of the previous section, rather than the critical horizon policy, since the state of the system is then fully described by the amount of accepted unprocessed workload.

The critical horizon policy is optimized in the same way as the critical level policy of Section 4. Initially, the critical horizon is set to N for all job classes. In each iteration, the critical horizon for class m is reduced by one, then that of class $m-1$ is reduced by one, etc., until the net profit decreases, or the critical horizon of class 2 has been reduced. This iteration is repeated until no improvements can be found.

During the search for the critical horizon policy, we evaluate several different policies. The evaluation of a policy can, as discussed in the previous section, be abandoned in iteration i of the value-iteration algorithm, if the upper bound M_i of the net profit of the policy is below the net profit of the so far best found policy. In this way we reduce the computation time of the simple policy.

For Model B we also consider a non-rationing policy and a selective policy where only jobs with a profit above some minimum level are accepted. As in Section 4 we apply a local search to find the best selection of classes to satisfy.

5.2 Numerical results

We analyse an example with 3 job classes and 9 job types. The capacity per period, is

j	W_j	DD_j	JC_j
1	1	2	1
2	2	6	1
3	4	4	1
4	1	1	2
5	2	3	2
6	4	5	2
7	1	4	3
8	2	2	3
9	4	7	3

Table 4: Description of workload(W_j), due date(DD_j) and jobclass (JC_j) for jobtype j in the numerical example of Model B.

$C = 2$, the length of the planning horizon is $N = 8$ periods, and every period is divided in $T = 8$ sub-periods.

The tightness of capacity β are set to 0.4, 0.55, 0.7, 0.85 and 1. The profits are

$$(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3) \in \{(1.6, 1.2, 1), (2.2, 1.4, 1), (2.8, 1.6, 1)(3.4, 1.8, 1), (4, 2, 1)\}$$

and the demand distributions are

$$(\gamma_1, \gamma_2, \gamma_3) \in \{(0.1, 0.3, 0.6), (0.2, 0.3, 0.5), (0.3, 0.4, 0.3), (0.5, 0.3, 0.2), (0.6, 0.3, 0.1)\}.$$

Furthermore, we consider 5 different combinations of holding and penalty costs:

$$(h, \pi) \in \{(0.005, 0.05), (0.01, 0.05), (0.01, 0.05), (0.05, 0.10), (0.05, 0.15)\}$$

Each parameter value is represented in 125 different cases for which the average value of the profit over near-optimal profit, as found by the BSP allocation, is found for each heuristic. These average values are reported in Table 5, where Simple(FIFO) refers to the critical horizon policy computed using FIFO allocation, and Simple(MC) refers to the critical horizon policy computed with the minimum cost algorithm. The MC allocation algorithm is in general not as good as the FIFO allocation, and for the selective policy and the non-rationing policy, we have therefore reported the results for the FIFO allocation only. The average net profit for the simple policy (FIFO) is 98.2% with a worst case

β	0.4	0.55	0.70	0.85	1
Simple(FIFO)	96.7 %	98.2 %	98.6 %	98.7 %	98.6 %
Simple(MC)	93.0 %	93.9 %	94.3 %	94.6 %	95.1 %
Selective	88.7 %	88.4 %	89.4 %	92.7 %	95.5 %
Non-rationing	66.7 %	74.6 %	82.9 %	90.3 %	95.0 %
(h, π)	(0.005,0.05)	(0.01,0.05)	(0.01,0.10)	(0.05,0.10)	(0.05,0.15)
Simple(FIFO)	99.3 %	99.4 %	98.1 %	97.7 %	96.4 %
Simple(MC)	94.7 %	95.3 %	91.9 %	95.7 %	93.2 %
Selective	93.0 %	93.5 %	89.3 %	91.7 %	87.2 %
Non-rationing	85.2 %	86.2 %	77.1 %	85.2 %	75.7 %
$(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3)$	(1.6,1.2,1)	(2.2,1.4,1)	(2.8,1.6,1)	(3.4,1.8,1)	(4,2,1)
Simple(FIFO)	97.9 %	98.2 %	98.3 %	98.3 %	98.3 %
Simple(MC)	93.9 %	94.1 %	94.2 %	94.3 %	94.4 %
Selective	90.8 %	90.8 %	90.8 %	91.0 %	91.2 %
Non-rationing	86.1 %	83.4 %	81.4 %	79.9 %	78.7 %
$(\gamma_1, \gamma_2, \gamma_3)$	(0.1,0.3,0.6)	(0.2,0.3,0.5)	(0.3,0.4,0.3)	(0.5,0.3,0.2)	(0.6,0.3,0.1)
Simple(FIFO)	98.6 %	98.6 %	98.7 %	98.1 %	97.0 %
Simple(MC)	94.2 %	94.4 %	94.7 %	94.2 %	93.4 %
Selective	87.9 %	90.2 %	90.4 %	93.0 %	93.2 %
Non-rationing	82.8 %	81.3 %	81.0 %	81.5 %	82.8 %

Table 5: Average net profit for as percentage of the near-optimal net profit for the simple (FIFO), the simple (MC), the selective and the non-rationing policy as function of available capacity β, profits $(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3)$ and the distribution of demand $(\gamma_1, \gamma_2, \gamma_3)$.

performance of 84.7%. The performance of the simple policy(MC) is on average 94.2%

with a worst case performance of 82.6%. The average performance of the selective policy is 90.9% with a worst case performance of 72.9%, and finally the average performance of the non-rationing policy is 81.9% with a worst case performance over the investigated examples of 51.2%. That is, in one of the investigated cases the profit can be almost doubled by using the near-optimal rationing policy rather than the non-rationing policy. Although the average performance of the simple policy is good, a worst case performance of 84.7% is not satisfactory. The impacts of the different factors are similar to those found for Model A in Section 4: When dealing with congested systems it is of high importance to employ a rationing policy of some kind, the loss in profit by using a non-rationing policy can be very high. Secondly, when dealing with such systems, the main objective is to avoid rejecting jobs, by simply processing as soon as possible by the FIFO principle, neglecting the holding cost incurred. Even when holding cost is as high as 5% per period it is better on average to use a FIFO allocation.

6 Conclusion

In this paper we have investigated a periodic make-to-order system with limited capacity, where jobs of different profitability arrive in a stochastic manner. We focus on systems where the average demand for capacity exceeds the available capacity which leads to situations where jobs must be rejected. By using simple or near-optimal rationing policies we can significantly increase the expected net profit, compared with a non-rationing approach. In our numerical tests we have found examples where profits are almost doubled by the use of rationing policies.

The derivation of the near-optimal policy is only computationally tractable for smaller problems and therefore we consider simple policies whose performance is shown, numerically, to be a few percent below the near-optimal policy. The examples investigated are very small indeed, but we conjecture that the simple policies will have a good performance also for problems of a realistic size, where the computation of the near-optimal policy is intractable. The simple policies are characterized by their simplicity and their small number of policy variables. For larger problems we therefore suggest to use simulation for evaluation of policies and then perform a local search to find good values of the policy variables.

Although, the simple policies suggested in this papers have a fine average performance, there is still room for improvement. In particular, future research should be focused on finding better allocation policies, perhaps including options for re-allocation, which seems reasonable in many settings.

References

[1] N. Balakrishnan, J.W. Patterson, and V. Sridharan. Robustness of capacity rationing policies. *European Journal of Operational Research*, 115:328–338, 1999.

[2] N. Balakrishnan, V. Sridharan, and J.W. Patterson. Rationing capacity between two product classes. *Decision Sciences*, 27(2):185–214, 1996.

[3] G.R. Bitran and S.V. Mondschein. An application of yield management to the hotel industry considering multiple day stays. *Operations Research*, 43(3):427–433, 1995.

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

[5] M. ElHafsi. An operational decision model for lead-time and price quotation in congested manufacturing systems. *European Journal of Operational Research*, 126:355–370, 2000.

[6] 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.

[7] A.Y. Ha. Stock rationing in an $M/E_k/1$ make-to-stock queue. *Management Science*, 46(1):77–87, 2000.

[8] L.C. Hendry and B.G. Kingsman. Job release: Part of a hierarchical system to manage manufacturing lead times in make-to-order companies. *Journal of the Operational Research Society*, 42(10):71–883, 1991.

[9] L.C. Hendry and B.G. Kingsman. Customer enquiry management: Part of a hierarchical system to control lead times in make-to-order companies. *Journal of the Operational Research Society*, 44(1):61–70, 1993.

[10] S.G. Johansen. Transfer pricing of a service department facing random demand. *International Journal of Production Economics*, 46–47:351–358, 1996.

[11] C.Y. Lee and J.Y. Choi. A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weigths. *Computers and Operations Research*, 1995.

[12] P. Melchiors. Rationing policies for an inventory model with several demand classes and stochastic lead times. *Department of Operations Research, University of Aarhus*, 2001.

[13] P. Melchiors, R. Dekker, and M.J. Kleijn. Inventory rationing in an (s, Q) inventory model with two demand classes and lost sales. *Journal of the Operational Research Society*, 51(1):111–122, 2000.

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

[15] J.W. Patterson, N. Balakrishnan, and V. Sridharan. An experimental comparison of capacity rationing models. *International Journal of Production Research*, 1997.

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

[17] L.R. Weatherford and S.E. Bodily. A taxamony and research overview of perishable-asset revenue management: Yield management, overbooking and pricing. *Operations Research.*, 40:831–844, 1992.

Calculating can-order policies for the joint replenishment problem by the compensation approach

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

December 2000

Abstract

In this paper we consider the stochastic joint replenishment problem. We coordinate the replenishment of an inventory with several items, where the fixed cost of a replenishment consists of a major order cost for each order and a minor order cost for each item participating in the order. The literature has focused on two kinds of policies: can-order policies and periodic replenishment policies. Generally speaking the can-order policy performs better when the major ordering cost is relative low, and the periodic replenishment policies better when the major ordering cost is relative high. We present a new method for calculating can-order policies, which dominates existing methods on examples where the can-order policy performs better than the periodic replenishment policies. The method is based on a compensation approach, where an item placing an order receives a compensation from other items benefitting from the order opportunity.

Keywords : inventory, Markov decision processes, joint replenishment problem, can-order policy

1 Introduction

We consider an inventory system with several different items facing Poisson demands. The items are reviewed continuously and are replenished by the same supplier. The fixed cost of a replenishment consists of a major order cost and an item-dependent minor order cost for each item participating in the replenishment. For such an inventory system it is obvious that some kind of coordination should be used. The optimal joint replenishment policy can, theoretically, be found by solving a huge Markov decision model, but since the size of the state and the decision space grow exponentially with the number of different items, it seems intractable to solve the model for more than, say, 5 different items, even on computers of today. Ignall [4] solves the problem for two items, and finds that the optimal policy is in general unfortunately not a simple policy (i.e. one that can be described by a few variables per item). Instead of focusing on the optimal policy the literature has been analyzing simple policies with good performance, in particular the continuous-review can-order policy and the periodic replenishment policy.

Under the regime of a can-order policy, each item is controlled by three variables s, c and S. When the inventory position of item i reaches the must-order level s_i, a replenishment order is placed. At the same time any item j, with inventory position at or below its can-order level c_j, is included in the order. S_i is the order-up-to level of item i. Finding the optimal can-order policy seems intractable, but there have been several ways suggested for computing good can-order policies, the most famous being the one proposed by Federgruen, Groenevelt & Tijms [3].

In this paper we suggest a new way of calculating the variables of the can-order policy. Our policy does not dominate all other can-order policies on all problems examined. It does however dominate all other can-order policies on problems where the can-order policy is better than the periodic replenishment policy, which is the main conclusion of the paper.

In the next section the background of the can-order policy is discussed and we perform a small simulation study to illustrate that the can-order policy of Federgruen, Groenevelt & Tijms [3] can be far from optimal. In Section 3 the new method for calculating the can-order policy is described, after which we compare its performance with other can-order polices and the periodic replenishment policy on the examples of Atkins & Iyogun [1] and Viswanathan [10].

2 Background

Let us first describe the inventory control problem known as the stochastic joint replenishment problem. The inventory system consists of n items, where item i faces Poisson demand with rate λ_i. Holding costs are charged at a rate $h_i > 0$ per unit per unit time. Demand not satisfied immediately is backlogged and shortage costs are charged at a rate of p_i per unit per unit time. Each unit backlogged moreover incurs a time independent cost of π_i. There is a constant lead time of L_i time units. The fixed ordering cost consists of a major cost K, and a minor cost k_i for each item i joining the order. Since all demands have to be satisfied eventually, the variable order costs (or purchasing costs) are not included in the model formulation. Note that although two items i, j place a joint order and share the major ordering cost K, their replenishments do not arrive simultaneously if $L_i \neq L_j$.

The can-order policy was first introduced by Balintfy [2]. Assuming no lead time and identical items he calculates a good can-order policy. Silver [7] relaxes these restrictive assumptions and introduces the principle of decomposition: from time to time an item i is faced with an opportunity of a discount replenishment, namely when another item reaches its must-order level and places an order. Assuming this process of discount opportunities is independent of item i, the multi-item inventory problem can be decomposed into several single-item inventory problems, each with occasional opportunities for discount replenishments, and solved by successive iterations. For item i the discount opportunity process is generated by the order placements of all items but item i. Federgruen, Groenevelt & Tijms [3] find a can-order policy for such a system (generalized to compound Poisson demand) by solving the single-item problem with a policy-iteration algorithm, assuming that the discount opportunity process is Poisson. This is obviously an approximation, but it simplifies the analysis considerably. Moreover, Zheng [11], in a theoretical paper, proves that if the discount opportunity process is Poisson then the can-order policy is optimal. After the single-item problems for each item have been solved, the rate at which discount opportunities are generated is calculated and used in the next iteration. The procedure stops when the optimal policies are unchanged.

Van Eijs [9] argues that the assumption of Poisson discount opportunities leads to poor results, and suggests a can-order policy where the can-order level c is always equal to $S-1$, for inventory systems where the major ordering cost is high compared with the average of the minor ordering costs. For such a policy, whenever an item places an order, all other items join the order. He minimizes holding and ordering costs subject to a service level constraint and finds the optimal $(s, S-1, S)$ policy. A disadvantage of this approach is that all items have to follow the $(s, S-1, S)$ policy, which means that it is not suitable for problems where some items have high minor ordering cost and others have low minor ordering cost.

Schultz & Johansen [6] show by simulation (for a 12-item example introduced by Atkins & Iyogun [1]) that the empirical waiting times between successive discount opportunities do not appear to come from an exponential distribution. They formulate a model where the time between two consecutive discount opportunities for each item i is Erlang-r distributed, which appears to give a better fit. The shape parameter r is found by simulation. Some of the discount opportunities refer to discount opportunities generated by the item itself and are modelled as fictitious. However, it is evident that the fictitious discount opportunities are not independent of item i as assumed. Still, they find a can-order policy that in most cases outperforms the can-order policy of Federgruen, Groenevelt & Tijms [3], but only in few cases is better than the periodic replenishment policies.

Periodic replenishment policies were first analyzed by Atkins & Iyogun [1], who suggest a modified periodic review policy (MP) where each item i orders up to R_i each time the inventory is reviewed. The review interval T_i is restricted to be an integer multiple of a base period. Viswanathan [10] suggests a $P(s, S)$ policy, which dominates the MP policy, where a periodic (s_i, S_i) policy is used for item i. The review interval t is a policy variable and must be the same for all items. Pantumsinchai [5] analyzes a QS policy where an order is placed when the total system demand since the last order placement exceeds Q. Item i orders up to S_i. The performance of the QS policy is comparable to that of the MP policy.

Generally speaking the can-order policy performs better when the major ordering cost is relatively low, and the periodic replenishment policies better when the major ordering cost is relatively high. Since the can-order policy is a continuous-review replenishment policy and therefore can react faster to new information than the periodic replenishment policies, the can-order policy should intuitively perform better than the periodic replenishment policies.

There are two possible explanations of why this is not so: either the can-order policy is too simple or it is not the can-order policy itself but rather the way the policy variables is computed that leads to the poor performance.

To investigate this we consider an example with 12 identical items facing Poisson demand (see Table 1). We have in each of 6 cases computed the optimal can-order policy by

	Can-order policies						Periodic $P(s,S)$	
	Optimal		FGT		Compensation			
	s,c,S	cost	s,c,S	cost	s,c,S	cost	t,s,S	cost
$K=500$ $k=10$	7,27,28	1405	-7,33,39	1935	5,23,30	1428	1.37,24,27	**1404**
$K=500$ $k=50$	5,26,31	1698	-7,29,42	2074	4,21,34	1750	1.87,25,31	**1691**
$K=150$ $k=10$	8,23,24	1102	5,22,29	1196	8,19,25	1106	0.93,20,23	**1101**
$K=150$ $k=50$	6,26,29	1477	5,20,33	1544	8,17,30	1514	1.56,21,28	**1472**
$K=50$ $k=10$	10,18,23	**981**	9,18,24	988	10,16,23	987	0.75,18,22	984
$K=50$ $k=50$	9,15,27	**1377**	8,16,29	1387	9,15,28	1378	1.40,20,27	1408

Table 1: The policy variables and the average costs for the three can-order policies and the $P(s,S)$ policy. $n=12$, $h_i=h=6$, $\pi_i=\pi=30$, $L_i=L=1$ and $\lambda_i=\lambda=10$. $p_i=p=10$ in the two first examples, and $p_i=p=0$ in the remaining four.

simulation over a wide range of variables. FGT denotes the can-order policy suggested by Federgruen, Groenevelt & Tijms [3], Compensation denotes the new can-order policy computed as suggested in Section 3 of this paper, and $P(s,S)$ denotes the policy suggested by Viswanathan [10]. For each can-order policy we report the variables s,c,S and the average cost computed by simulation. For the $P(s,S)$ policy we report the optimal value of t,s,S and the associated cost. The lowest costs are typed in boldface.

As Viswanathan [10], we observe that the FGT can-order policy has a poor performance for examples with high major ordering cost. However, the results show that in cases with a low value of K ($K=50$) the optimal can-order policy is better than the periodic replenishment policy, and for moderate and high values of K ($K=150$ and $K=500$) the $P(s,S)$ policy gives the lowest cost, but the difference is very small. The example clearly illustrates that the FGT can-order policy can be far from the optimal can-order policy. Whether the can-order policy is too simple is a little bit more ambiguous. At least it cannot outperform the $P(s,S)$ policy. In the cases with high values of K/k we note that the optimal policy is of the $(s,S-1,S)$ type, as suggested by van Eijs [9].

3 The compensation approach

In the cases with high major ordering cost, the suggested must-order points for the FGT can-order policies in Table 1 are very low. This is because the item considering placing an order balances the sum of the major and the minor ordering cost with the expected shortage cost for the item. In reality other items will benefit from the replenishment too, but this aspect has not previously been included in the decomposition principle. To include this we extend the decomposition principle by compensating the item i placing the order with an amount Δ_i, representing the expected value of other items' benefit from the discount opportunity, generated by the order placement. This means that when we consider the single-item problem i, the inputs are the process of discount opportunities generated by other items and the compensation amount Δ_i. We maintain the assumption of Poisson discount opportunities viewed from item i, and denote the rate of arrivals by μ_i

We first describe the the single-item model and show how to compute the compensation amount Δ_i. Afterwards we describe the decomposition procedure in its new form.

3.1 The single-item model

To ease notation we omit the index i in this section. We look at a single-item inventory system with Poisson discount opportunities. Since the demand also is Poisson, the problem can be formulated as a semi–Markov decision process.

Let the decision epochs be the demand epochs and the arrivals of the discount opportunities. The state of the system at each decision epoch is described by the inventory position x. Since we have two independent Poisson processes with rate λ and μ the merged process is a Poisson process with rate $\lambda + \mu$. The probability of a decision epoch being generated by a demand [discount opportunity] is $\lambda/(\lambda + \mu)$ [$\mu/(\lambda + \mu)$]. At each decision epoch we can decide to place an order. Zheng [11] proves that the can-order policy is optimal for such a system and we therefore only need to consider policies of the can-order type. For such an (s, c, S) policy the corresponding actions are to place an order of size $S - x$ when $x \leq s$, in decision epochs generated by a demand, and to place an order of size $S - x$ when $x \leq c$, in decision epochs generated by a discount opportunity. In all other states no order is placed.

We can now specify the transition probabilities of the system. In states $x > c$, we do not need to register the discount opportunities, and we jump to state $x - 1$ with probability one. In states x, with $s < x \leq c$, we jump to state $x - 1$ with probability $\lambda/(\lambda + \mu)$, and to state S with probability $\mu/(\lambda + \mu)$. In states x, with $x \leq s$, we jump to state S with probability one. The expected time between two decision epochs is $1/\lambda$ in states $x > c$, $1/(\lambda + \mu)$ in states $s < x \leq c$, and 0 in states $x \leq s$.

The lead time is incorporated by a standard shift in time (see e.g. Federgruen, Groenevelt

& Tijms[3]). The expected cost charged to a decision epoch when $x > c$ is

$$C(x) = \sum_{j=-\infty}^{x} \mathbb{P}(D = x - j) * \Big(\mathbf{1}_{j>0}[hj/\lambda] + \mathbf{1}_{j\le 0}[-pj/\lambda + \pi]\Big)$$

where $\mathbb{P}(D = x - j)$ denotes the probability of a lead time demand of $x - j$ units and $\mathbf{1}_{j>0}$ denotes the indicator function, which is one when $j > 0$ and zero otherwise. The expected cost charged to a decision epoch when $s < x \le c$ is $\lambda/(\lambda+\mu)C(x)$. The adjustment of the holding rate and the backorder rate is due to the change in expected time to next decision epoch. The stockout cost π is also adjusted, since for $x \le c$ it is only with probability $\lambda/(\lambda+\mu)$ that another demand is backordered before the next decision epoch. Finally we have to add the minor ordering cost k, if a discount opportunity occur in states $s < x \le c$, and the major and the minor ordering cost $K - \Delta + k$ in states $x \le s$.

Based on this we can find the long–run average cost per unit time of a given policy. The inventory process is regenerative with regeneration point S. Define a cycle to be the time between two regeneration epochs. By renewal–reward theory we can then find the average cost of a policy by dividing the expected cost incurred in a cycle by the expected cycle length (see e.g. Tijms [8]). Let $z(x)$ be expected cost incurred up to the next regeneration point starting in state x, and let $y(x)$ be the expected time until we reach the next regeneration point, starting in state x. Note that $z(s) = K - \Delta + k$ and

$$z(x) = \frac{\lambda}{\lambda+\mu}\left[C(x) + z(x-1)\right] + \frac{\mu}{\lambda+\mu}k \text{ for } s < x \le c.$$

Hence, it is easy to find $z(x)$ recursively for $x = s+1, s+2, \ldots, c$. For $x = c+1, c+2, \ldots, S$ we find $z(x)$ by

$$z(x) = C(x) + z(x-1).$$

The $y(x)$ values are calculated similarly,

$$y(x) = 1/(\lambda+\mu) + \frac{\lambda}{\lambda+\mu}y(x-1) \text{ for } s < x \le c,$$

initialized with $y(s) = 0$. The remaining values are found by

$$y(x) = 1/\lambda + y(x-1) \text{ for } c < x \le S.$$

Let g be the average cost of using policy (s, c, S),

$$g = \frac{z(S)}{y(S)}. \tag{1}$$

Next, we need to find the expected rate β_i of order placements from item i. The probability that the item places an order in a cycle equals $(\frac{\lambda}{\lambda+\mu})^{c-s}$, which is the probability of $c - s$

consecutive demands with no discount opportunity in between. To find the average number of orders placed by the item per unit time, we divide by the expected length of the cycle

$$\beta_i = \left(\frac{\lambda}{\lambda+\mu}\right)^{c-s} \Big/ y(S). \tag{2}$$

Now let us calculate the expected gain of a discount opportunity. Let

$$v(x) = z(x) - gy(x)$$

be the relative value of inventory position x, with g being the cost of the policy found by (1). The relative value $v(x)$ is the difference in expected long-run total cost of having an inventory position of x rather than the order-up-to level S. The relative values are typically used for optimization procedures, but here we use them to obtain information about the value of a discount opportunity.

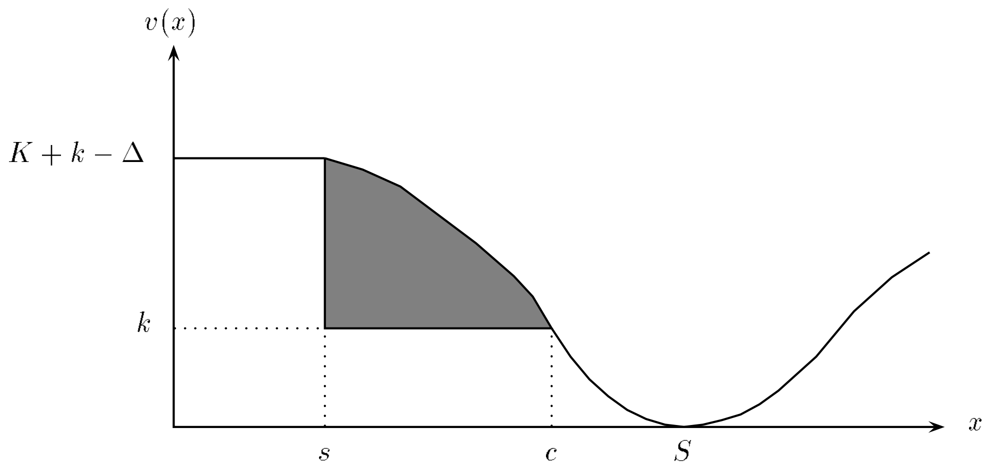

Figure 1: The relative values $v(x)$ as a function of the inventory position x for the optimal policy.

In Figure 1 we have depicted the relative values of the optimal policy. An item can benefit from discount opportunities occurring while its inventory position is below c. The relative value of being in state x is $v(x)$. If a discount opportunity occurs while the system is in state x, we will accept it (since $x \leq c$), and our inventory position will rise to S. Since the relative value of state S (by definition) is 0, the benefit of the discount opportunity is the positive amount $v(x) - k$.

Let J be the random variable denoting the number of demands occurring from the time when the can-order level is reached until the first discount opportunity occurs. The probability of $J = j$ is the probability of j consecutive demands followed by a discount opportunity. The expected gain of the discount opportunity is found by conditioning on the value of J.

$$V = \frac{\mu}{\lambda+\mu} \sum_{j=0}^{c-s-1} \Big(\frac{\lambda}{\lambda+\mu}\Big)^j \Big(v(c-j)-k\Big).$$

We can only benefit from one discount per cycle and therefore, to find the expected gain δ_i of the discount opportunity per discount opportunity, we divide by the expected number $\mu y(S)$ of discount opportunities occurring in a cycle, i.e.

$$\delta_i = \frac{V}{\mu y(S)}. \tag{3}$$

This can be done for any can-order policy. Naturally we are interested in optimizing the performance of the system and therefore we use the algorithm of Zheng [11] to find the optimal can-order policy (under the assumption of Poisson discount opportunities) with the major ordering cost specified as $K - \Delta$. We then use (2) and (3) to find β_i and δ_i, which are used in the decomposition procedure.

3.2 The decomposition procedure

To find the variables of the can-order policy for all items, we use the following decomposition procedure. The procedure is initialized by setting $\Delta_i = 0$ and β_i to a small but positive amount for all i.

In each iteration of the procedure we solve the single-item problem for each item i with values of Δ_i and μ_i given by

$$\Delta_i = \sum_{j \neq i} \delta_j$$

and

$$\mu_i = \sum_{j \neq i} \beta_j.$$

After solving the problem for item i the values of δ_i and β_i are updated by (2) and (3). This iteration is repeated until the policy variables values either are unchanged or start cycling between two or several solutions. This typically happens within 10-50 iterations. In the event of cycling the best policy is found by evaluating the policies in the cycle and choosing the one with the lowest cost. The introduction of the Δ_i-values does not lead to a significantly higher number of iterations compared with the traditional iteration scheme without compensation.

An estimate of the total system cost is found by $\sum_i g_i$. Since the decomposition procedure is based on approximations we also evaluate the optimal policy by simulation. We have found that the estimate can differ by up to 10% from the cost found by simulation and therefore we will in general only report the cost found by simulating the optimal policy.

				$P(s,S)$	Erlang	FGT	Compensation	
i	k_i	L_i	λ_i	s_i, S_i	s_i, c_i, S_i	s_i, c_i, S_i	s_i, c_i, S_i	δ_i
1	10	0.2	40	33,37	10,35,40	8,34,46	9,26,38	10.90
2	10	0.5	35	40,45	20,42,48	17,43,54	19,36,47	9.98
3	20	0.2	40	31,37	10,33,41	8,34,49	9,24,41	9.05
4	20	0.1	40	27,33	5,28,36	4,27,44	5,19,36	8.51
5	40	0.2	40	29,37	9,30,40	8,29,53	9,22,46	6.86
6	20	1.5	20	43,50	31,45,51	23,46,58	28,42,55	6.73
7	40	1.0	20	31,46	20,33,48	14,33,50	18,30,47	5.75
8	40	1.0	20	31,46	20,33,48	14,33,50	18,30,47	5.75
9	60	1.0	28	42,65	30,46,66	25,44,69	27,40,64	6.46
10	60	1.0	20	30,48	20,32,50	13,32,53	18,29,50	4.97
11	80	1.0	20	29,50	20,32,55	12,31,55	17,28,52	5.07
12	80	1.0	20	29,50	20,32,55	12,31,55	17,28,52	5.07
Computed Cost				**2267**	2377	2620	2268	
Simulated Cost					2288.7 ±0.6	2365.5 ±1.2	2313.0 ±0.8	

Table 2: Base-case example. Data as well as optimal policies, computed costs and simulated costs are reported. The simulated costs are found by simulating the policies found by the respective methods. Other data are : $K = 150$, $p = 0$, $\pi = 30$. For the $P(s,S)$ policy $t = 0.557$.

3.3 Numerical results

Reverting to Table 1, we see that, by using the compensation method, we obtain the result that the costs of the can-order policies lie closer to those of the optimal can-order policies. The $P(s,S)$ policy dominates the compensation can-order policy in most of the cases but the difference between the costs of the two policies is less than 2% on average for the 5 cases where the $P(s,S)$ policy dominates. For comparison the difference in average cost between the FGT policy and the $P(s,S)$ policy is on average 15%.

The following numerical results are variations of a 12-item base-case introduced by Atkins and Iyogyn [1]. In Table 2 we report the parameters of the 12 items together with the policy variables found for the $P(s,S)$ policy and for the can-order policy using the Erlang, FGT and compensation method, respectively. For the compensation policy we also report the optimal values of δ_i for each item i. For the base-case the $P(s,S)$ policy has the best performance followed by the Erlang, compensation and the FGT can-order policy. As we can see the computed cost of the compensation can-order policy is closer to the simulated cost, compared with the FGT can-order policy, however the difference is not negligible and therefore for, all our numerical results, we only report the cost obtained by simulation of the found policy.

Besides the base-case we have also compared the compensation can-order policy with the

$P(s,S)$ policy on some of the examples presented by Viswanathan [10]. These experiments are reported in Tables 3 and 4. The values of k_i, λ_i and L_i are as in Table 2 for all examples, except for the 5 problems with only 8 items in Table 3. The costs of the MP policy and the $P(s,S)$ policy are obtained from Viswanathan [10] where possible. The costs of the Erlang policy are obtained from Schultz & Johansen [6]. For the 25 examples with relatively low values of h and p (Table 3) the $P(s,S)$ policy is in general the best policy. The can-order policy is better in 5 examples, all characterized by a low value of K.

The results show that the compensation can-order policy is in general better than the FGT can-order policy. In all cases but two, the cost of the compensation can-order policy is lower than those of the FGT can-order policy, in some cases by up to 25%. Compared with the Erlang can-order policy the compensation can-order policy is better in cases with a low major ordering cost, whereas the Erlang can-order policy gives lower costs on examples with a high major ordering cost. However, these costs are still higher than those of the $P(s,S)$ policy.

In Table 4 we report a set of examples, introduced by Viswanathan [10], where h and p have been increased by a factor of 100. This corresponds to an increase in the demand rate and thereby the demand variation by a factor of 100. Viswanathan [10] concludes that even in these cases the $P(s,S)$ policy dominates the FGT can-order policy. In the table we report the cost of the $P(s,S)$ policy and the compensation can-order policy, respectively, for 36 examples. The compensation can-order policy is better than the $P(s,S)$ policy in all of the examples, including the examples where K is high. The average cost difference is only 0.75%, but the conclusion that the $P(s,S)$ policy still dominates the can-order policy for these high variation examples is not true.

Our conclusion is therefore that the $P(s,S)$ policy should be used on problems where it turns out that the ordering process for the best can-order policy follows a regular pattern, e.g. because the demand variation is low. In these cases we can be better off using a real periodic policy, since we eliminate the uncertainty of not knowing exactly when the next order will be placed. This will reduce the holding cost, which offsets the cost of having a longer reaction time (which is small when demand variability is low).

For problems where the time between two consecutive replenishments is more unpredictable the loss in reaction time is much more significant and we would be better off using the can-order policy computed by the compensation approach.

References

[1] D. Atkins and P.O. Iyogun. Periodic versus ‘can–order’ policies for coordinated multi–item inventory systems. *Management Science*, 34:791–796, 1988.

[2] J.L. Balintfy. On a basic class of multi-items inventory problems. *Management Science*, 10(2):287–297, 1964.

Problem parameters		(R,T)	$P(s,S)$	Erlang	FGT	Compensation
	$K=20$	1094	1066	1034	1031	**1029**
$\pi=30$	$K=50$	1135	1121	1125	1122	**1109**
$p=0$	$K=100$	1198	**1185**	1197	1236	1211
$h=2$	$K=150$	1244	**1241**	1253	1322	1289
	$K=200$	1289	**1285**	1307	1395	1355
	$K=250$	1330	**1327**	*	1467	1410
	$K=20$	907	886	878	873	**872**
$\pi=0$	$K=50$	944	**934**	957	957	944
$p=30$	$K=100$	1005	**992**	1023	1058	1034
$h=2$	$K=150$	1046	**1043**	1068	1134	1106
	$K=200$	1088	**1085**	1113	1205	1159
$\pi=0$	$K=50$	1570	**1530**	1540	1558	1542
$p=30$	$K=100$	1676	**1623**	1632	1694	1660
$h=6$	$K=150$	1733	**1706**	1725	1821	1747
	$K=200$	1803	**1778**	1810	1926	1821
8-Item	Exp. 1	1559	**1559**	*	1625	1637
Problem	Exp. 2	1615	**1615**	*	1678	1683
Table 5 of	Exp. 3	1664	**1664**	*	1873	1794
Atkins &	Exp. 4	1544	**1542**	*	1611	1579
Iyogun [1]	Exp. 5	1264	1264	*	1267	**1234**
	$K=20$	2564	2325	*	2320	**2314**
	$K=50$	2626	**2427**	*	2462	2434
$\pi=0$	$K=100$	2811	**2562**	*	2687	2571
$p=30$	$K=150$	2872	**2684**	*	2887	2702
$h=20$	$K=200$	2985	**2792**	*	3047	2820

* Observations not reported

Table 3: Costs for the (R,T) policy, the $P(s,S)$ policy, the Erlang, the FGT and the Compensation can-order policy for the Atkins & Iyogun data set

Problem parameters	$K = 20$		$K = 100$		$K = 500$	
	$P(s, S)$	Compen-sation	$P(s, S)$	Compen-sation	$P(s, S)$	Compen-sation
$p = 1000$, $h = 200$	18689	**18532**	19675	**19523**	22317	**22224**
$p = 1000$, $h = 600$	34210	**33991**	35597	**35373**	39124	**39009**
$p = 1000$, $h = 1000$	43461	**43241**	44976	**44749**	48959	**48822**
$p = 5000$, $h = 200$	26468	**26216**	27787	**27433**	30971	**30592**
$p = 5000$, $h = 600$	57738	**57374**	59467	**59098**	64277	**63778**
$p = 5000$, $h = 1000$	81101	**80597**	83199	**82619**	88770	**88060**
$p = 10000$, $h = 200$	29538	**29284**	30908	**30553**	34312	**33849**
$p = 10000$, $h = 600$	67589	**67106**	69445	**68866**	74634	**73959**
$p = 10000$, $h = 1000$	97756	**97052**	100053	**99350**	106153	**105157**
$p = 20000$, $h = 200$	32428	**32136**	33866	**33398**	37463	**36885**
$p = 20000$, $h = 600$	76918	**76216**	78922	**78333**	84265	**83323**
$p = 20000$, $h = 1000$	113698	**112947**	116194	**115345**	122725	**121596**

Table 4: Costs for the $P(s, S)$ policy and the compensation can-order policy for the data set of Viswanathan [10] with high values of penalty and holding cost rates.

[3] A. Federgruen, H. Groenevelt, and H.C. Tijms. Coordinated replenishments in a multi–item inventory system with compound Poisson demands. *Management Science*, 30:344–357, 1984.

[4] E. Ignall. Optimal continuous review policies for two product inventory systems with joint setup costs. *Management Science*, 15:278–283, 1969.

[5] P. Pantumsinchai. A comparison of three joint ordering inventory policies. *Decision Science*, 23:111–127, 1992.

[6] H. Schultz and S.G. Johansen. Can-order policies for coordinated inventory replenishment with Erlang distributed times between ordering. *European Journal of Operational Research*, 113:30–41, 1999.

[7] E.A. Silver. A control system for coordinated inventory replenishment. *Int. J. Prod. Res.*, 12:647–670, 1974.

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

[9] M.J.G. van Eijs. On the determination of the control parameters of the optimal can–order policy. *Mathematical Methods of Operations Research*, 39:289–304, 1994.

[10] S. Viswanathan. Periodic review (s, S) policies for joint replenishment inventory systems. *Management Science*, 43(10):1447–1454, 1997.

[11] Y.S. Zheng. Optimal control policy for stochastic inventory systems with Markovian discount opportunities. *Operations Research*, 42(4):721–738, 1994.

The can-order policy for the periodic-review joint replenishment problem

Søren Glud Johansen* and Philip Melchiors
Department of Operations Research, University of Aarhus,
Ny Munkegade, Building 530, 8000 Aarhus C, Denmark.

June 2001

Abstract

In this paper we study the stochastic joint replenishment problem. In contrast to what appears to be the general belief, we show that the class of periodic replenishment policies does not outperform the class of can-order policies for this problem.

We present a method, based on Markov decision theory, to calculate near-optimal can-order policies for a periodic-review inventory system. Our numerical study shows that the can-order policy behaves as good as, if not better than, the periodic replenishment policies. In particular, for examples where the demand is irregular, we find cost differences up to 15% in favor of the can-order policy.

Keywords : inventory, joint replenishment problem, can-order policy, Markov decision process

1 Introduction

In this paper we study the stochastic joint replenishment problem, i.e. the problem of coordinating an inventory system with several items, all replenished by the same supplier. The literature has suggested several coordination policies, of which in particular two classes of policies, the can-order policy (originally suggested by Balintfy [2]) and the periodic replenishment policy (introduced by Atkins and Iyogun [1]), have received considerable attention. Under the regime of a can-order policy all items follow an (s, c, S)-policy: When the inventory level is below the must-order level s, an order is placed to bring the inventory position up to S. Moreover, every item has a can-order level c. Whenever another item has reached its must-order level, any item with inventory position at or below its can-order level is included in the order. A good periodic replenishment policy is suggested

*Corresponding author. Tel.:+45 8942 3547 ; fax: +45 8613 1769 ; e-mail: sgj@imf.au.dk

by Viswanathan [12], who analyses a periodic-review $P(s, S)$ policy, with t units of time between each review. At every review an (s, S) policy is applied to each item, such that any item with inventory level below s is included in the order. The review interval t is a policy variable, which must be the same for all items.

In accordance with the numerical results of Viswanathan [12], it seems to be generally believed that the periodic replenishment policy performs better than the can-order policy, as for example expressed in the well-known textbook by Silver, Pyke and Peterson [10].

In this paper we demonstrate that for problems with little demand variation, the periodic replenishment policy has a fine performance, but it does not outperform the can-order policy. For problems with high demand variation, we find that the can-order policy is a much better choice, achieving cost reductions up to 15% compared with the periodic replenishment policy.

We analyse a can-order policy for a periodic-review inventory system. In the existing literature, the can-order policy has been studied under the assumption of continuous review, which can be justified by the recent developments of access to point of sale information and Electronic Data Interchange. However, while the access to information may be continuous, there is often only a limited number of replenishment opportunities, say, once or maybe twice a day. For many systems a periodic-review model will therefore provide a better representation.

We apply the ideas normally used for the optimization of continuous-review can-order policies to the periodic-review model and use Markov decision theory to find good can-order policies. In our numerical studies we investigate the performance of the can-order policy and find that in many cases the can-order policy is significantly better than the $P(s, S)$ policy, primarily the situations with high demand variation, whereas in situations with low demand variation the two policies are equally good.

The paper proceeds as follows: In Section 2 we first introduce the underlying assumptions and notation and relate the periodic-review can-order policy to the continuous-review can-order policy. Then the joint replenishment problem is decomposed into several single-item problems, which are used in the decomposition algorithm. We show how to construct this algorithm and how to solve the single-item problems. Finally, in Section 3 we provide some numerical results and, in Section 4, our conclusions.

2 Modelling

2.1 Notation and background

We consider an inventory system with n items. We assume a periodic model where every period typically represents one day. We assume that period demands are stochastic and stationary variables and let $\phi_i(x)$ be the probability mass function of the demand for item

i in any period. Let $\phi_i^t(x)$ be the t-fold convolution of $\phi_i(x)$. Let D_i and LTD_i denote demand per period and demand during lead time, respectively.

Demand not satisfied immediately is backordered, incurring a fixed cost π_i per unit backordered. At the end of every period, a holding cost of h_i per unit in stock and a time dependent cost p_i per unit backordered is incurred. There is a constant lead time of L_i periods for each item i. The fixed cost of ordering consists of a major cost K plus a minor ordering cost k_i for every item i participating in the order. Since all demands are backordered, and thus satisfied eventually, the variable ordering cost is not included in the model.

A way to model this problem is to formulate it as an n-dimensional Markov chain, and use a value iteration algorithm to find the optimal joint replenishment policy. Such an approach is computationally intractable even for small-scale problems, and moreover, the optimal policy has a non-simple structure (Ignall [4]), which means that the implementation of the policy will be very cumbersome. Ohno et al. [6, 7] present an improved policy-iteration algorithm for finding the optimal problem. While they succeed in finding the optimal policy, they do not overcome the "curse of dimensionality", and their approach can in reality be used for systems with 2–4 items only.

Another approach is to solve n single-item problems independent of each other, neglecting the correlation between the items. This approach works if the major cost K is very small (and is indeed optimal if $K = 0$), but otherwise better methods are needed.

Silver [9] introduces the principle of decomposition to model the interaction between the items. The idea is to decompose the original problem into n sub-problems, one for each item. Item i has normal replenishment opportunities with major and minor ordering cost $K + k_i$ occurring whenever the inventory level reaches s, and discount opportunities with only minor ordering cost k_i, whenever another item places a normal order. The process of discount order opportunities is in general very complicated and moreover, not independent of the demand process for item i. Silver suggests to approximate this process by a Poisson process with rate μ_i, which is assumed to be independent of item i. This facilitates a simple analysis of the model. Moreover, Zheng [13] proves that the can-order policy is optimal for a single-item inventory system with Markovian discount opportunities and Poisson demand. The rate μ_i is calculated based on the rates, β_j, of order placements from other items. For the continuous-review model the superposition of the $n - 1$ Poisson processes is also a Poisson process with rate $\mu_i = \sum_{j \neq i} \beta_j$. In each iteration of the optimization algorithm, the single-item problem is solved, after which the rates μ_i are updated. This is repeated until the rates μ_i converge or start cycling.

Silver applies this to a continuous-review model with Poisson demand and uses a simple method to determine the values of the policy variables. Federgruen, Groenevelt & Tijms [3] also consider a continuous-review model but assume compound Poisson demand and apply a policy-iteration algorithm to find better can-order policies. Both papers are based on the same principle of decomposition. Schultz & Johansen [8] show that the decomposition approach can be improved by assuming Erlang distributed inter-arrival times between the discount opportunities.

Another way to improve the principle of decomposition is suggested by Melchiors [5] for a can-order policy with continuous review. Whenever an item places an order, other items receive a discount opportunity which may reduce their cost. However, in the original decomposition, the item placing the order does not take this into consideration when deciding whether to place an order or not. By calculating the average benefit per discount opportunity δ_j for all items j, we can compensate an item i placing an order by only charging $K - \Delta_i$, where

$$\Delta_i = \sum_{j \neq i} \delta_j. \tag{1}$$

In this way, the implied effects of placing an order is included when deciding when to place an order.

In this paper we use the extended decomposition principle introduced by Melchiors [5] to solve the periodic-review joint replenishment problem. Let us first look at the implications of using a periodic-review model. The continuous-review model enjoys the property that there can be only one event occurring at a time. Once an item hits its must-order level, it is therefore the only item below its must-order level. For the periodic model several items can trigger an order at the same time, and therefore we need another approach to model the discount opportunities. We approximate the process of discount opportunities by a Bernoulli process with outcome 1, if a discount order opportunity occurs and 0 otherwise. Successive outcomes are assumed to be independent of each other, which means that the probability of a discount opportunity does not depend on discount opportunities of the past, and the process can therefore be seen as a discrete version of the Poisson process.

Let β_j be the fraction of time, where item j has an inventory position below s_j. We assume (as an approximation) that the ordering process of item i is independent of the ordering process of item j for all $j \neq i$. Let ψ_i be the random variable denoting the number of items, not counting item i, with inventory position below their must-order point. Viewed from item i, we can then find the probability that at least one item, besides item i, places an order:

$$\begin{aligned} \mu_i &= \mathbb{P}(\psi_i > 0) = 1 - \mathbb{P}(\psi_i = 0) \\ &= 1 - \prod_{j \neq i} (1 - \beta_j) \end{aligned} \tag{2}$$

We can also derive the expected number of items placing an order in a period, given that at least one item places an order:

$$\begin{aligned} E(\psi_i|\psi_i > 0) &= \frac{E(\psi_i)}{\mathbb{P}(\psi_i > 0)} \\ &= \frac{1}{\mu_i}\sum_{j\neq i}\beta_j \end{aligned} \tag{3}$$

We first show how to solve the periodic-review single-item model for item i given a value of compensation Δ_i, under the assumption of Bernoulli generated discount opportunities with rate μ_i. We also show how to derive δ_i and β_i given a can-order policy. We then use these in the decomposition algorithm to solve the joint replenishment problem.

2.2 The single-item model

Let us first describe the sequence of events in a period. At the beginning of the period outstanding orders arrive. Demands occur during the period, and demands that cannot be satisfied immediately are backlogged. At the end of the period, costs are assigned based on the amount of physical inventory, the number of shortages in the current period, and the current amount of backlogged demands. An order can then be placed or, if another item is placing an order, we can join this order. Orders with zero lead time arrive instantly and can be used in the following period.

For convenience we omit the index i in the remainder of this subsection. Let the state, x_n, of the system at the end of period n, be the inventory position. Under the regime of a can-order policy $\{x_n\}_{n\geq 0}$ is a Markov chain with regeneration point S. Let the inventory cycle be the time between two consecutive visits in the regeneration point. Using results from Markov decision theory, we can find the average long-run cost of a policy as the expected cost incurred during a cycle divided by the expected length of a cycle (see Tijms [11]). Let us first state the transition probabilities and the expected cost function, excluding costs of ordering.

In any state x where we do not place an order, we jump to state $x - j$ with probability $\phi(j)$ for all $j \geq 0$. However, if we place or join an order, we jump immediately to state S.

Let $C(x)$ be the expected cost incurred in period $\tau + L + 1$ if the state in period τ is x. The first period that can be influenced by our decision is period $\tau + 1$ if the lead time is zero, and period $\tau + L + 1$ if the lead time is L. Consequently, this shift in time assigns the relevant cost to each period. To find the expected cost we condition on the lead time demand, which has probability mass function $\phi^L(j)$:

$$\begin{aligned} C(x) &= \sum_{d=0}^{x}\phi^L(d)\left(\sum_{j=0}^{x-d}\phi(j)h(x-d-j) + \sum_{j=x-d+1}^{\infty}\phi(j)(j\pi - p(x-d-j))\right) \\ &\quad + \sum_{d=x+1}^{\infty}\phi^L(d)\sum_{j=0}^{\infty}\phi(j)(j\pi - p(x-d-j)) \text{ for } x > 0 \end{aligned}$$

and $C(x) = E(D)\pi - p(x - E(LTD) - E(D))$ for $x \leq 0$. By the end of a period we have three options. We can either place an order, join an order (if possible), or decide to leave the inventory position unchanged. With probability μ, another item has placed an order which we will join if $s < x \leq c$. Otherwise, if $x \leq s$ and no discount opportunity is available, we place an order and if $x > s$, no order is placed. Whenever we join an order, our ordering cost is k, and whenever we place an order, our ordering cost is $K - \Delta + k$. In both cases we order up to inventory position S, and, consequently, we jump immediately to state S, and assign the cost $C(S)$.

All we need now is to specify what happens when $x \leq s$ and another item places an order. Since another item is placing an order we could join this order and only assign the minor ordering cost k. However, consider a situation where two items simultaneously place an order. Both items use the discount opportunity and consequently none of them is assigned the major ordering cost $K - \Delta$. Instead, the items placing an order should share the major ordering cost. On average, the number of items that will share the ordering cost if an order is placed, is $E(\psi|\psi > 0) + 1$, since we know that at least one other item has placed an order. The resulting ordering cost can therefore be approximated with

$$k + (K - \Delta)/(E(\psi|\psi > 0) + 1).$$

The minor (item specific) ordering cost is not shared, but the compensation Δ is. We can find the expected ordering cost when $x < s$, by conditioning on whether ψ is zero or positive:

$$\begin{aligned} &\mathbb{P}(\psi > 0)\left(k + \frac{K - \Delta}{(E(\psi|\psi > 0) + 1)}\right) + (1 - \mathbb{P}(\psi > 0))\,(k + K - \Delta) \\ &= k + (K - \Delta)\theta \end{aligned}$$

with

$$\theta = 1 - \mathbb{P}(\psi > 0)\frac{E(\psi)}{E(\psi) + \mathbb{P}(\psi > 0)}. \tag{4}$$

Consider an example where $\mathbb{P}(\psi > 0) \approx 1$ and $E(\psi) = 5$. If an item is placing an order, the cost will then be the minor ordering cost plus a share $\theta \approx 1/6$ of the major ordering cost, which seems reasonable.

Let $z(x)$ be the expected cost incurring until we reach the regeneration point next time, starting in state x. Similarly, let $y(x)$ be the expected time until the regeneration point is reached, starting in state x. We can find $z(x)$ and $y(x)$ in a recursive manner by the following formulas:

$$z(x) = \begin{cases} C(x) + \sum_{j=0}^{\infty} \phi(j) z(x-j) & c < x \\ \mu k + (1-\mu)\left(C(x) + \sum_{j=0}^{\infty} \phi(j) z(x-j)\right) & s < x \leq c \\ k + (K - \Delta)\theta & x \leq s \end{cases} \tag{5}$$

$$y(x) = \begin{cases} 1 + \sum_{j=0}^{\infty} \phi(j) y(x-j) & c < x \\ (1-\mu)\left(1 + \sum_{j=0}^{\infty} \phi(j) y(x-j)\right) & s < x \leq c \\ 0 & x \leq s \end{cases} \tag{6}$$

Note that we jump immediately to state S without assigning cost when an order is placed. Using the renewal–reward theorem, the average cost of a policy (s, c, S) is then given by

$$g = \frac{z(S)}{y(S)}. \tag{7}$$

The values $z(x)$ and $y(x)$ can be found recursively, by setting $z(x) = k + \theta(K - \Delta)$ and $y(x) = 0$ for all $x \leq s$, and then compute $z(s+1), z(s+2), \ldots, z(S)$ and $y(s+1), y(s+2), \ldots, y(S)$ by (5) and (6).

We apply a tailor-made policy-iteration algorithm to find a near-optimal can-order policy. Let

$$v(x) = z(x) - g y(x)$$

be the relative value of an inventory position x, with g being the cost of the policy found by (7). The relative value $v(x)$ is the difference in expected long-run total cost of having an inventory position of x rather than the order-up-to level S.

The semi–Markov version of Theorem 3.2.1 in Tijms [11] tells that an optimal policy, i.e. one that minimizes the cost of running the system, can be found by solving the so–called average cost optimality equations

$$v(x) = \min\{v_1(x), v_2(x), v_3(x)\} \text{ for all } x \tag{8}$$

with

$$\begin{aligned} v_1(x) &= \theta(K - \Delta) + k + v(S) \\ v_2(x) &= (1-\mu)\left(C(x) - g + \sum_{j=0}^{\infty} \phi(j) v(x-j)\right) + \mu(k + v(S)) \\ v_3(x) &= C(x) - g + \sum_{j=0}^{\infty} \phi(j) v(x-j)), \end{aligned}$$

in the unknowns $v(x)$, g and S. The three functions represent the three actions: Place an order, do not place an order but accept discount opportunities, and neither place an order nor accept discount opportunities.

For a single-item inventory system with Markovian discount opportunities and Poisson demand a policy of the can-order type is optimal (Zheng [13]). When demands are compound this is not necessarily true. We will, however, restrict our search for policies to policies of the can-order type. The must-order level, s, is determined such that $s+1$ is the lowest inventory position, where we do not want to place an order. The can-order level, c, is determined such that $c+1$ is the lowest inventory position, where we do not wish to join a discount opportunity. This means that our policy is not necessarily optimal, and therefore we refer to it as a near-optimal policy.

We solve the set of equations by a policy-iteration algorithm. The initial value of g can be found by evaluating an arbitrary policy. In each iteration we use g to find an improved policy by solving the equations with respect to $v(x)$ and S (as explained below). Let g' be the cost of the new policy found by the minimizing actions. If $g' = g$, we have solved (8) and thereby found the near-optimal policy. Otherwise, we set g equal to g' and perform another iteration with the new value of g. The iteration scheme converges within a finite (and small) number of iterations.

Consider an iteration where we want to solve (8) for a given scalar g. Let $v(S) = 0$ and consider the must-order point s. For very small x it will always be optimal to place an order, so we are looking for the lowest x where we do not want to place an order in state $x+1$. The must-order point is found by

$$s = \min\{x | v_1(x+1) > v_2(x+1)\}, \tag{9}$$

using that $v(x) = v_1(x)$ for all $x \leq s$. Let $v(s) = \theta(K - \Delta) + k$. Set $x = s$ and increase x while computing the values $v(x)$ by

$$v(x) = v_2(x).$$

We are looking for the first inventory position where we do not want to use a discount opportunity if we had one, i.e.

$$c = \min\{x \geq s | v_2(x+1) > v_3(x+1)\}. \tag{10}$$

It is easy to show that by this definition, $v(x) \geq k$ for all $x \leq c$. For $x > c$ we do not use the discount opportunity, and therefore the relative values are given by

$$v(x) = v_3(x)$$

for all $x > c$. We now consider choosing the optimal order–up–to level S. Since we want to minimize the relative values of each state, we must chose S such that $v(S)$ is minimized. For the continuous-review Poisson demand model, $C(x)$ is quasi-convex and therefore the minimization can be done by a neighbor search. This cannot be guaranteed for the compound demand case and therefore we use enumeration to find the value of S that minimizes $v(S)$.

$$S = \operatorname{argmin}\{v(x) | x > c\} \tag{11}$$

Now, compute the values $y(x)$ using (6) for the found policy and we can find the cost g'.

$$g' = \frac{v(S) + gy(S)}{y(S)} = \frac{v(S)}{y(S)} + g$$

If $v(S) = 0$ then $g' = g$ and we have solved (8), with the near-optimal policy being specified by (9), (10) and (11). Otherwise, $g' < g$ and we have found an improved policy and another iteration is performed using the new value $g := g'$.

In Figure 1 we show the relative values for the optimal solution to a single-item problem (item 12 in Example A in Section 3). The figure furthermore illustrates how the policy variables s, c and S are determined.

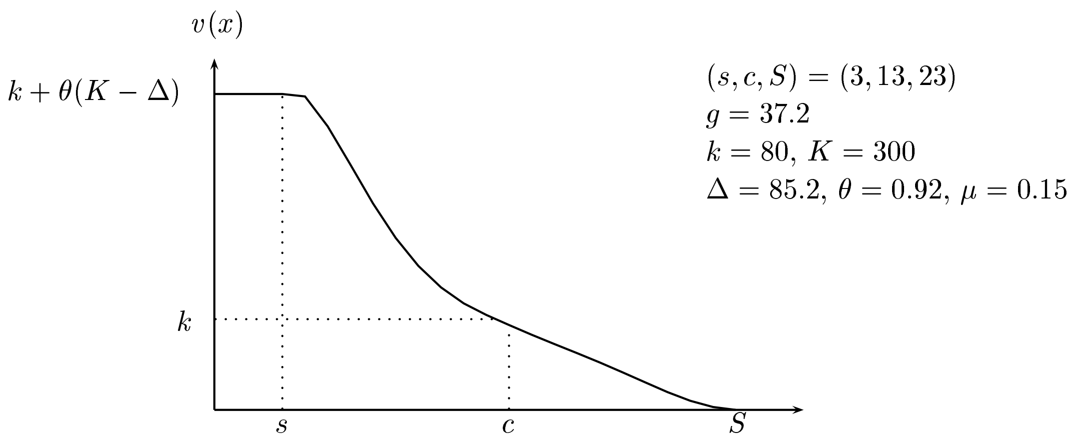

Figure 1: The relative values $v(x)$ and the optimal values of s, c and S for item 12 of Example A.

Given a can-order policy, we can derive β, the expected number of orders placed per period, and δ, the expected benefit of a discount opportunity. Let $m(x)$ denote the expected number of times we visit a state x during an inventory cycle. We can enter a state x from state $x + j$ after a demand of j units, $j > 0$. However, after a period with zero demand, we will remain in state x, which will count as an additional visit. We find $m(x)$ by the use of renewal theory (see e.g. Tijms [11]). Naturally, $m(S) = 1/(1 - \phi(0))$, since every inventory cycle begins in state S. For $x = S - 1, S - 2, \ldots, c + 1$, $m(x)$ can be found by the recursion

$$m(x) = (1 - \phi(0))^{-1} \sum_{j=1}^{S-x} \phi(j) m(x + j).$$

For $x = c, c-1, \ldots, s+1$,

$$m(x) = (1-\phi(0)(1-\mu))^{-1}\left(\sum_{j=1}^{c-x}(1-\mu)\phi(j)m(x+j) + \sum_{j=c+1-x}^{S-x}\phi(j)m(x+j)\right),$$

and, finally, for $x \leq s$

$$m(x) = \sum_{j=s+1-x}^{c-x}(1-\mu)\phi(j)m(x+j) + \sum_{j=c+1-x}^{S-x}\phi(j)m(x+j)).$$

Let $M = \sum_{x=-\infty}^{S} m(x)$, i.e. M is the average number of visits into some state per inventory cycle. Since we spend one period per visit, except the visit into the state in which we place an order, which is exactly once every inventory cycle, M is equal to $y(S)+1$. The expected number of orders placed per period, equals the expected number of times we visit a state x with $x \leq s$ per cycle, divided by the cycle length $y(S)$.

$$\beta = y(S)^{-1}\sum_{x=-\infty}^{s} m(x) \qquad (12)$$

An item benefits from discount opportunities occurring while its inventory position is below c. The relative value of being in state x is $v(x)$. If a discount opportunity occurs while the system is in state x, we will accept it when $x \leq c$, and our inventory position will rise to S. Since the relative value of state S is 0, the benefit of the discount opportunity is the positive amount $v(x) - k$, as illustrated in Figure 1. If $x > c$, we do not join the order and consequently the benefit is zero.

Under the assumption that the discount opportunity process is independent of the item under consideration, the expected benefit per inventory cycle can be found by conditioning on the inventory position, x. The number of periods where we are in state x is $m(x)$, and therefore the expected gain per cycle is

$$V = \sum_{x=s+1}^{c} m(x)(v(x)-k).$$

Note that we can only benefit from one discount opportunity per cycle, and, therefore, to find the expected gain δ of the discount opportunity per discount opportunity, we divide with $\mu y(S)$, the expected number of discount opportunities occurring in a cycle, i.e.

$$\delta = \frac{V}{\mu y(S)}. \qquad (13)$$

2.3 The decomposition algorithm

Based on the method for the single-item problem, we construct an algorithm for solving the joint replenishment problem. The idea of the algorithm is to successively solve the single-item problems, and update the rate of orders placed β_i for all items, until the found can-order policies converge. As demonstrated by Schultz & Johansen [8], there is a non-trivial risk that the algorithm starts cycling between different policies, rather than converging to a single one, and there seems to be no way to ensure convergence. Let (s_i^k, c_i^k, S_i^k) be the policy found for item i in the k'th iteration of the decomposition algorithm. The algorithm is initialized by setting $\delta_i = 0$ for all i. β_i is set equal to a small but positive amount, in this paper $\beta_i = 0.02$ is used for all i. Also, in the first iteration we evaluate an arbitrary policy, (s_i^0, c_i^0, S_i^0), to find an initial g for the policy-iteration algorithm. In the k'th iteration of the algorithm we perform the following for each item i:

- First, we compute Δ_i by (1) , μ_i by (2), and θ_i by (3) and (4).
- Then we solve the single-item problem based on these values and update the values of δ_i and β_i by (12) and (13).

As initialization for the policy-iteration algorithm we evaluate the policy $(s_i^{k-1}, c_i^{k-1}, S_i^{k-1})$ with the new values of Δ_i and μ_i to find the initial g_i. The algorithm terminates when $(s_i^k, c_i^k, S_i^k) = (s_i^{k-1}, c_i^{k-1}, S_i^{k-1})$, or when the algorithm starts cycling between a set of policies. The algorithm terminates in approximately 20-40 iterations. Due to the approximate nature of our approach, the cost given by $\sum_i g_i$ is only an approximation. In order to compare the policy with other policies, we simulate the found policy. Its cost is denoted the simulated cost. In case of cycling we choose the policy with the lowest computed cost.

3 Numerical results

In this section we compare the periodic can-order policy with the $P(s, S)$ policy suggested by Viswanathan [12]. The $P(s, S)$ policy for the periodic model is computed by restricting the search for the review interval t to an integral number of periods. For the continuous model there are no restrictions on t. We use a neighbor search to determine the optimal value of t, initialized by $t = 1$. The computation times of the can-order policy and the $P(s, S)$ policy are comparable, with the can-order policy being the faster when the optimal value of t is high. The cost computed as suggested by Viswanathan [12] provides an upper bound for the total costs, and in order to make a fair comparison the optimal $P(s, S)$ policy is simulated as well as the can-order policies. In order to reduce variance, the same stream of random numbers is used to generate demand data for all policies. We simulate 10 batches, each consisting of 100000 periods.

We also investigate the effect of using the compensation policy derived in this paper, compared with a policy in line with that of Federgruen, Groenevelt & Tijms [3], where the rate of order opportunities is the only input to the single-item model. This policy is found by setting Δ_i equal to zero for all i in all iterations of the decomposition algorithm.

				Compensation		FGT	$P(s,S)$ $(t=7)$
i	k_i	L_i	λ_i	δ_i	s,c,S	s,c,S	s,S
1	10	2	0.8	16.70	1,16,20	1,17,21	15,19
2	10	5	0.7	16.20	3,19,23	3,20,24	18,22
3	20	2	0.8	11.74	1,15,21	1,16,22	12,20
4	20	1	0.8	11.48	0,11,19	0,14,20	9,18
5	40	2	0.8	9.10	1,12,22	1,14,22	10,21
6	20	15	0.4	4.08	7,20,26	6,21,26	20,25
7	40	10	0.4	2.60	4,13,22	3,14,23	13,22
8	40	10	0.4	2.69	4,13,22	3,14,23	13,22
9	60	10	0.56	5.47	6,19,28	6,20,28	18,27
10	60	10	0.4	2.46	3,13,23	3,13,23	12,22
11	80	10	0.4	2.57	3,13,23	3,13,24	11,23
12	80	10	0.4	2.58	3,13,23	3,13,24	11,23
Computed cost					448.9	462.8	459.3
Simulated cost					**427.8** ± 0.5	432.2 ± 0.5	457.7 $\pm$ 0.7

Table 1: Example A. Data as well as optimal policies for the 12 items,plus computed and simulated cost (with 95% confidence intervals) for the compensation, FGT and $P(s,S)$ policy.

The resulting policy is denoted FGT. In all numerical experiments, the lowest cost of the three policies is typed in boldface.

We investigate three examples with 12 items. Example A is based on the standard example introduced by Atkins & Iyogun [1]. The example is modified in order to fit the periodic model in the following way: The Poisson demand is replaced by a Bernoulli demand process, where the probability λ_i of a positive demand from item i in each period is proportional to the arrival rate of item i in the standard example. In our modification, the demand sizes are either 1,2 or 15, while the standard example assumes unit sized demands. The probability mass function is $\phi_i(0) = 1 - \lambda_i$, $\phi_i(1) = 0.8\lambda_i$, $\phi_i(2) = 0.18\lambda_i$ and $\phi_i(15) = 0.02\lambda_i$ for all i. We set $K = 300$, and for all items i, $h_i = 2$, $p_i = 20$ and $\pi_i = 30$. The lead time is converted into periods by dividing the time unit of the standard example into 10 periods. The values of λ_i, L_i and the minor ordering cost k_i are given in Table 1 for all items i, together with the compensation policy, the FGT policy and the $P(s,S)$ policy. For the compensation policy we have reported the found values of δ_i for illustration. We also report the computed cost, and the simulated cost with 95% confidence intervals, for the three policies.

For the example the simulated cost of the $P(s,S)$ policy is 7% above the simulated cost of the compensation policy. The FGT and the compensation policy are almost identical for this example and so are their costs. We observe that the computed cost differs significantly from the simulated cost for the two can-order policies, whereas the upper bound of Viswanathan [12] in this example is very tight.

In Table 2 we have reported a set of variations over this example. More specifically we have examined different values of ordering cost $K \in \{100, 300, 500\}$ and for all items i, different values of $p_i \in \{10, 20, 30\}$ and $h_i \in \{1, 2, 3\}$. Among the 27 examples, the compensation policy is best in all but one. On average, the simulated cost of the FGT policy is 1.2 % higher than and the simulated cost of the $P(s, S)$ policy is 5.7% higher than the simulated cost of the compensation policy. There appears to be no systematic effect of the underlying model parameters K, h and p. Intuitively, the can-order policies should perform better when K is low, since we then approach the system with independent items, while the $P(s, S)$ policy should be better when K is high, as observed for the continuous models (Viswanathan [12] and Melchiors [5]). This is not the case in this example.

Next, we analyse Examples B and C to obtain further insight. Example B is a system where the demand is fairly irregular, being $0, 1$, or 2 in most periods with occasional highs of 15 units every once in a while, as in Example A. The probability of a positive demand in a period is $\lambda_i = 0.1 + 0.05i$. The probability mass functions of the demand size is given by $\phi_i(0) = 1 - \lambda_i$, $\phi_i(1) = 0.8\lambda_i$, $\phi_i(2) = 0.18\lambda_i$ and $\phi_i(15) = 0.02\lambda_i$ for all i. Example C has a more regular demand distribution with $\phi_i(0) = 1 - \lambda_i$, $\phi_i(1) = 0.65\lambda_i$, $\phi_i(2) = 0.25\lambda_i$ and $\phi_i(3) = 0.1\lambda_i$ and the same rate of demand λ_i as Example B. The average demand size is 1.46 and 1.45 in the two examples, respectively. The examples can be thought of as systems with 12 fairly identical products, perhaps only differing in their color or their taste. Holding and penalty costs are therefore identical for all items and so are the minor ordering costs and the lead times. What differs is typically the demand rate. Some colors or variants may be more needed than others. We have chosen an example with demand rates from 0.15 to 0.7 for the 12 items. In contrast to Example A we fix the penalty costs and consider different values of the lead time instead. The remaining parameters are $K \in \{100, 300, 500\}$ and for all items, $k = 25$, $p = 20$, $\pi = 30$, $L \in \{2, 4, 6\}$ and $h \in \{1, 2, 3\}$.

In Example B, the simulated costs of the $P(s, s)$ policy is on average 10.3% higher than those of the compensation policy. The performance of the FGT policy is better, having only 1.3% higher costs than the compensation can-order policy on average. In all 27 cases the compensation policy has the lowest cost. As in Example A, there seems to be no clear relation between the ordering cost K and the cost ratio. With respect to the lead time, it seems like the performance of the $P(s, S)$ improves as the lead time increases. Typically, the benefit of the can-order policies is that they are able to react faster to demand, and

			Compensation	FGT	P(s,S)	$P(s,S)$/Compensation
$K = 100$	$h = 1$	$p = 10$	**233.7**	235.5	246.6	1.06
		$p = 20$	**257.7**	259.2	270.5	1.05
		$p = 30$	**271.1**	273.6	284.5	1.05
	$h = 2$	$p = 10$	**330.2**	330.6	349.8	1.06
		$p = 20$	**393.1**	393.5	415.9	1.06
		$p = 30$	428.2	**427.8**	452.6	1.06
	$h = 3$	$p = 10$	**386.2**	388.1	404.7	1.05
		$p = 20$	**478.4**	480.1	508.6	1.06
		$p = 30$	**541.8**	542.3	575.9	1.06
$K = 300$	$h = 1$	$p = 10$	**260.1**	263.2	273.9	1.05
		$p = 20$	**281.7**	285.5	297.9	1.06
		$p = 30$	**297.2**	300.9	313.1	1.05
	$h = 2$	$p = 10$	**375.5**	381.6	392.0	1.04
		$p = 20$	**427.8**	432.2	457.7	1.07
		$p = 30$	**459.5**	463.7	490.6	1.07
	$h = 3$	$p = 10$	**432.7**	440.9	451.8	1.04
		$p = 20$	**523.9**	528.7	564.2	1.08
		$p = 30$	**584.1**	588.4	631.3	1.08
$K = 500$	$h = 1$	$p = 10$	**279.8**	285.0	291.9	1.04
		$p = 20$	**301.7**	307.3	317.5	1.05
		$p = 30$	**316.6**	324.2	335.5	1.06
	$h = 2$	$p = 10$	**407.4**	413.5	419.0	1.03
		$p = 20$	**456.0**	462.4	485.1	1.06
		$p = 30$	**485.5**	491.8	516.6	1.06
	$h = 3$	$p = 10$	**472.5**	492.6	484.2	1.02
		$p = 20$	**566.0**	575.7	601.3	1.06
		$p = 30$	**617.6**	624.8	665.2	1.08

Table 2: Example A. Simulated costs for the investigated policies for various values of ordering cost K, holding cost h and backorder cost p. We also report the cost ratio between the $P(s,S)$ policy and the compensation policy.

			Compensation	FGT	P(s,S)	$P(s,S)$/Compensation
		$L=2$	**186.0**	186.9	207.4	1.11
	$h=1$	$L=4$	**207.7**	208.7	225.5	1.09
		$L=6$	**223.0**	223.6	237.5	1.07
		$L=2$	**250.8**	252.2	278.8	1.11
$K=100$	$h=2$	$L=4$	**294.3**	295.8	320.0	1.09
		$L=6$	**328.2**	329.1	352.3	1.07
		$L=2$	**291.7**	292.9	319.3	1.09
	$h=3$	$L=4$	**345.6**	347.1	373.5	1.08
		$L=6$	**393.5**	394.3	420.0	1.07
		$L=2$	**211.7**	214.5	243.3	1.15
	$h=1$	$L=4$	**229.0**	232.4	254.0	1.11
		$L=6$	**243.7**	247.3	262.8	1.08
		$L=2$	**292.1**	294.9	330.4	1.13
$K=300$	$h=2$	$L=4$	**328.4**	331.7	366.9	1.12
		$L=6$	**357.7**	360.8	391.8	1.10
		$L=2$	**334.9**	339.9	377.8	1.13
	$h=3$	$L=4$	**384.8**	391.2	427.8	1.11
		$L=6$	**429.7**	434.3	469.4	1.09
		$L=2$	**232.7**	238.3	262.9	1.13
	$h=1$	$L=4$	**248.0**	252.6	272.2	1.10
		$L=6$	**260.8**	265.3	281.0	1.08
		$L=2$	**321.6**	330.8	365.7	1.14
$K=500$	$h=2$	$L=4$	**355.2**	362.2	396.2	1.12
		$L=6$	**381.1**	386.8	418.0	1.10
		$L=2$	**369.5**	382.2	417.1	1.13
	$h=3$	$L=4$	**418.6**	428.7	464.9	1.11
		$L=6$	**459.6**	467.7	503.3	1.09

Table 3: Example B. Simulated costs for the investigated policies for various values of ordering cost K, holding cost h and lead time L. We also report the cost ratio between the $P(s,S)$ policy and the compensation policy.

			Compensation	FGT	P(s,S)	$P(s,S)$/Compensation
$K = 100$	$h = 1$	$L = 2$	**122.3**	123.6	124.8	1.02
		$L = 4$	**127.9**	128.5	130.6	1.02
		$L = 6$	**133.4**	134.9	135.4	1.01
	$h = 2$	$L = 2$	**176.6**	179.0	178.6	1.01
		$L = 4$	**187.6**	191.4	189.8	1.01
		$L = 6$	**199.2**	200.9	200.3	1.01
	$h = 3$	$L = 2$	**217.3**	219.9	219.5	1.01
		$L = 4$	**234.3**	237.9	236.9	1.01
		$L = 6$	**249.2**	252.6	251.1	1.01
$K = 300$	$h = 1$	$L = 2$	147.7	150.0	**147.2**	1.00
		$L = 4$	153.0	158.2	**151.9**	0.99
		$L = 6$	157.4	162.2	**156.2**	0.99
	$h = 2$	$L = 2$	210.8	217.3	**209.7**	0.99
		$L = 4$	220.2	227.7	**219.4**	1.00
		$L = 6$	228.6	236.1	**228.3**	1.00
	$h = 3$	$L = 2$	258.8	267.3	**257.3**	0.99
		$L = 4$	272.4	282.3	**271.2**	1.00
		$L = 6$	284.2	293.2	**284.1**	1.00
$K = 500$	$h = 1$	$L = 2$	164.5	171.7	**163.6**	0.99
		$L = 4$	170.6	178.6	**168.2**	0.99
		$L = 6$	174.3	182.1	**172.2**	0.99
	$h = 2$	$L = 2$	235.0	247.2	**232.9**	0.99
		$L = 4$	244.4	255.7	**241.8**	0.99
		$L = 6$	251.6	263.4	**249.7**	0.99
	$h = 3$	$L = 2$	287.0	302.5	**284.2**	0.99
		$L = 4$	301.4	313.7	**297.3**	0.99
		$L = 6$	311.0	327.3	**308.9**	0.99

Table 4: Example C. Simulated costs for the investigated policies for various values of ordering cost K, holding cost h and lead time L. We also report the cost ratio between the $P(s,S)$ policy and the compensation policy.

thereby reduce the number of periods without stock. When lead times are short, orders are often triggered by a high demand leaving the inventory depleted or with very little stock. In these situations the can-order policies are superior to the periodic replenishment policies. When lead times are longer the triggering of an order is determined by the size of expected lead time demand, and consequently the effect of a fast reaction is relatively smaller.

In Example C, the demand variation is very small and the performance of the $P(s, S)$ policy is much better. The average cost ratio is 1.00 while the cost ratio of the FGT policy to the compensation policy is 1.03. For this low-variance example the $P(s, S)$ policy has an acceptable performance, and we observe that it is better than the can-order policy on problems with high ordering cost.

4 Conclusion

In this paper we have demonstrated how to calculate can-order policies for a periodic model with stochastic demand. Contrary to the general belief as expressed for example in the textbook of Silver, Pyke and Petersen [10], the can-order policy seems to perform very well and our results show that it is not outperformed by the periodic replenishment policies. Indeed, on examples with irregular demand patterns, we have recorded cost differences up to 15% in favour of the can-order policy.

References

[1] D. Atkins and P.O. Iyogun. Periodic versus 'can–order' policies for coordinated multi–item inventory systems. *Management Science*, 34:791–796, 1988.

[2] J.L. Balintfy. On a basic class of multi-items inventory problems. *Management Science*, 10(2):287–297, 1964.

[3] A. Federgruen, H. Groenevelt, and H.C. Tijms. Coordinated replenishments in a multi–item inventory system with compound Poisson demands. *Management Science*, 30:344–357, 1984.

[4] E. Ignall. Optimal continuous review policies for two product inventory systems with joint setup costs. *Management Science*, 15:278–283, 1969.

[5] P. Melchiors. Calculating can-order policies for the joint replenishment problem by the compensation approach. *Department of Operations Research, University of Aarhus*, 2000.

[6] K. Ohno and T. Ishigaki. A multi-item continuous review inventory system with compound Poisson demand. *Mathematical Methods of OR*, 2001.

[7] K. Ohno, T. Ishigaki, and T. Yoshii. A new algorithm for a multi-item periodic review inventory system. *Mathematical Methods of OR*, 39:349–364, 1994.

[8] H. Schultz and S.G. Johansen. Can-order policies for coordinated inventory replenishment with Erlang distributed times between ordering. *European Journal of Operational Research*, 113:30–41, 1999.

[9] E.A. Silver. A control system for coordinated inventory replenishment. *Int. J. Prod. Res.*, 12:647–670, 1974.

[10] E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling.* John Wiley & Sons, New York, 1998.

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

[12] S. Viswanathan. Periodic review (s, S) policies for joint replenishment inventory systems. *Management Science*, 43(10):1447–1454, 1997.

[13] Y.S. Zheng. Optimal control policy for stochastic inventory systems with Markovian discount opportunities. *Operations Research*, 42(4):721–738, 1994.

A Two-Echelon Inventory Model With Lost Sales

Jonas Andersson
Division of Production Management,
Department of Industrial Engineering,
University of Lund,
Box 118, SE-22100 Lund,Sweden

Philip Melchiors*
Department of Operations Research,
University of Aarhus,
Ny Munkegade, Building 530,
8000 Aarhus C, Denmark.

February 2001

International Journal of Production Economics 69(3) 307–315
Reprinted with permision of Elsevier Science

Abstract

Almost all multi-echelon inventory models assume that demand not satisfied immediately can be backordered. In some situations this assumption may not be realistic. For example, it may be more representative to model stockouts as lost sales when the retailers are in a competitive market and customers can easily turn to another firm when purchasing the good. Assuming lost sales at the retailers, we consider a one warehouse several retailers inventory system. Using the well-known METRIC-approximation as a framework, we present a heuristic for finding cost effective base-stock policies. In a numerical study we find that the cost of the policies suggested by the heuristic is on average 0.40% above the cost of the $(S-1, S)$-optimal policy.

Keywords : Inventory, Multi-echelon, Lost sales, METRIC

1 Introduction

Consider a two-echelon inventory system with one central warehouse and an arbitrary number of retailers. See Figure 1. The retailers face customer demand and replenish their stocks from the central warehouse. The warehouse, in turn, replenishes its stock from an outside supplier. Evaluation and optimization of control policies for such inventory systems have attracted massive interest in the literature. See, for example, Axsäter [3]

*Corresponding author. Tel.:+45-89 42 35 36; fax: +45-86 13 17 69; e-mail: philip@imf.au.dk

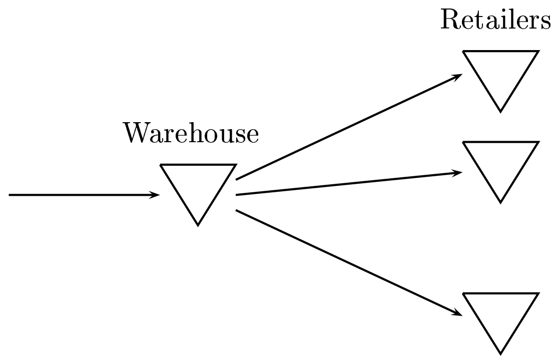

Figure 1: Multi-echelon inventory system

for an overview. In the existing literature dealing with multi-echelon inventory control the prevalent assumption is that complete backlogging of orders is allowed in case of stockouts. For example, Axsäter [4] shows how to exactly evaluate the performance for different (R, nQ)-polices when the retailers face compound Poisson demand and inventories are continuously reviewed. Cachon [5] gives an exact method for the periodic review case with identical retailers.

In some situations the assumption of complete backlogging may not be so realistic. For example, it may be more representative to model stockouts as lost sales when the retailers are in a competitive market and customers can easily turn to another firm when purchasing the good. The research dealing with multi-echelon inventory models has focused mainly on the backorder case, and the number of models dealing with lost sales is rather limited. The main reason for this is the added complexity of the lost sales case. Sherbrooke ([11] also argues that in system approaches where focus is on the simultaneous availability of several items (e.g. at least one of each spare part for an airplane), the assumption of lost sales does not make sense. However, Anupindi and Bassok [1] consider a periodic review two-echelon inventory system where a part of the unsatisfied sales at the retailers is lost. Since the transportation time between the manufacturer and the retailers is zero, the optimal order policy at each retailer is a base-stock policy. The manufacturer carries linear production cost and no holding cost. The retailers can agree to centralize their stocks and the problem considered is whether or not this will lead to an increase in total expected sales at the manufacturer. Nahmias and Smith [8] also consider lost sales in a multi-echelon environment in a paper more closely related to this paper. However, their model differs from ours in several important aspects. First, they consider periodic review batch order policies. The model is more general since they deal with partial lost sales. This means that, with probability u, demand not satisfied immediately, is lost, and with probability $1 - u$, it is satisfied later by a special order. Moreover, for the model to be tractable they assume instantaneous deliveries from the warehouse to the retailers.

For single-echelon inventory models the lost sales assumption is more common. The exact cost for a single level inventory system facing Poisson demand and fixed leadtimes was first given by Hadley and Whitin [6]. Smith [12] demonstrates how to evaluate and find optimal $(S-1, S)$-policies for an inventory system with zero replenishment costs and generally distributed stochastic leadtimes. Recently Hill [7] showed that for the lost sales case the $(S-1, S)$-policy is not necessarily optimal.

In this paper we analyze a model for a one warehouse, multiple retailer inventory system. Demand occurs only at the retailers and follows independent Poisson processes. All leadtimes are assumed to be constant. All installations use $(S-1, S)$-policies with continuous review. It is assumed that backlogging of customer demand is not allowed. The analysis departs in one of the most widely known multi-echelon inventory models, the METRIC-model developed by Sherbrooke [10]. In its original setting, it is assumed that stockouts at the retailers are completely backlogged. We demonstrate how the METRIC-model can be modified to handle the lost sales case. Our approach gives an approximate model which is quite simple and efficient from a computational point of view. Simulation experiments indicate that the performance is very good.

The outline of this paper is as follows: In Section 2 we give a detailed problem formulation and pose all assumptions. Section 3 gives the solution procedure. The numerical results are given in Section 4, and in Section 5 we give some conclusions and point out some possible directions for future research.

2 Problem Formulation

The inventory system under consideration consists of one central warehouse and an arbitrary number of retailers. The retailers face Poisson customer demand. No backlogging is allowed at the retailers. Consequently, the customers that arrive to a retailer that is out of stock will become lost sales for the retailer. When stockouts occur at the warehouse, all demands from the retailers are fully backlogged and the backorders are filled according to a FIFO-policy. The transportation time between the warehouse and a given retailer is assumed to be constant as well as the transportation time from the external supplier to the warehouse. The cost of a replenishment is assumed to be zero or negligible compared to the holding and stockout costs. The external supplier is assumed to have infinite capacity, which means that the replenishment leadtime for the central warehouse is constant. All installations use $(S-1, S)$-policies with continuous review. Units held in stock both at the warehouse and at the retailers incur holding costs per unit and time unit. Moreover, a fixed penalty cost per lost customer is incurred at the retailers. In this paper we present a model for the considered inventory system, which can be used to evaluate the long-run

average cost for different policies within the class of $(S-1, S)$-policies. The objective is to find the policy that minimizes the long-run average cost for the inventory system. Let us introduce the following notation:
N = the number of retailers,
λ_i = demand intensity at retailer i, $i = 1, 2, \ldots, N$,
L_i = transportation time for the deliveries from the warehouse to retailer i, $i = 1, 2, \ldots, N$,
L_0 =transportation time for the deliveries from the external supplier to the warehouse,
S_0 = order-up-to level at the warehouse,
S_i = order-up-to level at retailer i, $i = 1, 2, \ldots, N$,
h_0 = holding cost rate at the warehouse,
h_i = holding cost rate at retailer i, $i = 1, 2, \ldots, N$,
π_i = penalty cost for a lost sale at retailer i, $i = 1, 2, \ldots, N$.

We want to determine the total cost for the inventory system in steady state. Define
TC = total cost for the inventory system per time unit in steady state,
C_0 = cost per time unit for the warehouse in steady state,
C_i = cost per time unit for retailer i in steady state, $i = 1, 2, \ldots, N$.

Obviously,

$$TC = C_0 + \sum_{i=0}^{N} C_i. \tag{1}$$

Our objective is to determine a control policy, $S_0, S_1, .., S_N$ that minimizes the total cost, TC.

3 Solution Procedure

In this section we first demonstrate how the total cost for different control policies can be evaluated. For the backorder case the exact cost of the system can be derived by observing that any unit ordered by retailer i is used to fulfill the S_ith demand. The cost can then be derived by conditioning on the arrival time of the S_ith demand (which is Erlang distributed) and the arrival of the ordered unit (see Axsäter [3]). In a lost sales environment the corresponding observation is that any unit ordered by retailer i is used to fulfill the $S_i + X_i$th demand, where X_i is a random variable denoting the number of lost sales incurred at the retailer during the replenishment lead time. X_i is obviously very hard to characterize and we have therefore chosen to focus on a heuristic rather than on the exact solution.

The analysis has many similarities with the analysis in Sherbrooke [10]. However, our assumption of lost sales at the retailers destroys some of the nice properties valid for the backorder model. The analysis of the warehouse, e.g., becomes more complex for the lost sales case. In the backorder case, all customers arriving at the retailers generate demands at the warehouse immediately at the arrival epoch, since all retailers use continuous review $(S-1, S)$-policies. Consequently, the warehouse faces a Poisson process with intensity $\lambda_0 = \lambda_1 + \lambda_2 + \cdots + \lambda_N$. For the lost sales case this is not true. When backordering is not allowed, customer demands can be lost due to stockouts at the retailers. Therefore the demand at the warehouse is not Poisson process anymore.

Another important difference compared with the backorder case is that the order-up-to level S_i at retailer i, affects the costs at all retailers and at the warehouse. In the backorder case S_i only affects the local cost at retailer i, since the warehouse demand process is unaffected by the order-up-to levels at the retailers. For the lost sales case the order-up-to level affects the number of lost sales and consequently, the demand process at the warehouse is not independent of the policies at the retailers. Therefore the order-up-to level at a certain retailer affects the costs at all installations in the inventory system.

We will first show how to evaluate the costs at the retailers given a certain replenishment leadtime provided by the warehouse. We then show how to calculate the cost at the warehouse given the demand intensity from the retailers. Finally we introduce an iterative procedure from which we obtain the total cost for the inventory system.

3.1 Approximate retailer cost

As Sherbrooke [10] we use a queueing system analogy when evaluating the costs for the retailers. For a retailer where backlogging is allowed, the number of outstanding orders towards the central warehouse is the same as the occupancy level in an $M/G/\infty$ queue. Recall that the customer demand is Poisson and the replenishment leadtimes are stochastic, since orders can be delayed due to stockouts at the central warehouse. For this type of queue a famous theorem by Palm [9] states that the steady state occupancy level is Poisson distributed with mean λL, where λ is the arrival rate and L is the mean service time. Palm's theorem holds for i.i.d. service times. The stochastic leadtimes in our case are evidently not independent, but if we disregard this correlation we can approximate the number of outstanding orders with a Poisson distribution. This is the idea behind the METRIC-approximation.

When demand is lost, the queueing system of interest is an $M/G/S/S$ queue, with S servers, each with generally distributed service times and no queueing allowed. If the service times are independent random variables with mean $\bar{L}$, Erlang's loss formula states the steady-state distribution for the occupancy level as

$$q^S(j) = \frac{(\lambda \bar{L})^j / j!}{\sum_{n=0}^{S} (\lambda \bar{L})^n / n!} \text{ for } 0 \leq j \leq S$$

where $q^S(j)$ = the probability that j servers (out of S) are occupied in steady state. Following METRIC we approximate the number of outstanding orders with this distribution.

Suppose that the mean leadtime for retailer i is $\bar{L}_i$ and let $q_i^{S_i}(j)$ be the steady state probability of j outstanding orders given a desired base-stock level S_i. The expected number of lost sales per time unit is clearly $\lambda_i q_i^{S_i}(S_i)$ and the expected number of units in stock is

$$\sum_{j=0}^{S_i}(S_i - j)q_i^{S_i}(j) = S_i - [1 - q_i^{S_i}(S_i)]\lambda_i \bar{L}_i. \tag{2}$$

The total relevant cost for retailer i is therefore

$$C_i(S_i, \bar{L}_i) = \lambda_i \pi_i q_i^{S_i}(S_i) + h_i\Big(S_i - [1 - q_i^{S_i}(S_i)]\lambda_i \bar{L}_i\Big)$$

and the rate of demand from retailer i which is not lost is $(1 - q_i^{S_i}(S_i))\lambda_i$.

The derivation of the exact cost of a $(S-1, S)$ lost sales single stage inventory system with generally distributed leadtimes was first presented by Smith [12]. He also proves that $C_i(S_i, \bar{L}_i)$ is convex in S_i for fixed $\bar{L}_i$, which means that the optimal value can be found by a local search routine.

3.2 Approximate warehouse cost

In the backorder case the demand process at the warehouse is a Poisson process. In the lost sales case this is not the case. If, for example, the base-stock level at a retailer is one, the smallest interval between two successive demands from that retailer will be the retailer leadtime. We will ignore this and approximate the demand process at the warehouse with a Poisson process with mean Λ. Λ depends on how much demand is lost at the retailers and is determined as

$$\Lambda = \sum_{i=1}^{N} \lambda_i(1 - q_i^{S_i}(S_i)) \tag{3}$$

Since we have a fixed deterministic leadtime L_0, we can find the average holding cost incurred at the warehouse as a function of Λ and S_0.

$$C_0(S_0, \Lambda) = h_0 \sum_{j=0}^{S_0}(S_0 - j)\frac{(\Lambda L_0)^j}{j!}\exp(-\Lambda L_0)$$

We can also derive the mean delay due to stockouts at the warehouse by first calculating B_0, the average number of backorders at the warehouse.

$$B_0 = \sum_{j=S_0+1}^{\infty}(j - S_0)\frac{(\Lambda L_0)^j}{j!}\exp(-\Lambda L_0), \tag{4}$$

We then apply Little's formula to obtain the average delivery delay, B_0/Λ. The mean leadtime for retailer i is then

$$\bar{L_i} = L_i + B_0/\Lambda \quad (5)$$

Finally, we obtain the total cost from (1).

3.3 Overall solution procedure

We can now establish the solution procedure. The procedure enumerates over S_0. It can be shown that for a cost minimizing solution, S_0 can not be negative. See, for example, Axsäter [2]. Consequently our procedure starts with S_0=0. Moreover, S_0 is bounded from above by an abortion criteria. We need the following new notation:

$C_i^{\min}$ = $\min_{S_i} C_i(S_i, L_i)$ = minimum cost per time unit for retailer i in steady state when the leadtime, $\bar{L_i}$, is equal to the transportation time, L_i, $i = 1, 2, \ldots, N$.

$S_i(k)$ = order-up-to level at retailer i in iteration k.

$TC^*(S_0)$ = minimum value of TC given a fixed value of S_0.

Let us first consider two simple lemmas. The proofs can be found in the Appendix. The first lemma gives a lower bound for the retailer costs, and the second establishes two important properties for the warehouse cost.

Lemma 1. *$C_i^{\min}$ is a lower bound for the retailer cost, $C_i(S_i, \bar{L_i})$ for all S_i and any $\bar{L_i} > L_i$.*

Lemma 2. *$C_0(S_0, \lambda_0) \leq C_0(S_0, \Lambda)$, for all S_0 and all $\Lambda \leq \lambda_0$. Moreover, $C_0(S_0, \lambda_0)$ is convex in S_0.*

To construct an abortion criteria for the procedure, consider the cost function

$$TC_{lb}(S_0) = C_0(S_0, \lambda_0) + \sum_{i=0}^{N} C_i^{\min}.$$

By Lemma 1 and Lemma 2, $TC_{lb}(S_0)$ is a lower bound for the cost function, $TC^*(S_0)$. Moreover, since the cost function $TC_{lb}(S_0)$ is convex in S_0 the search over S_0 can be aborted when S_0 satisfies

$$\min_{x \leq S_0} TC^*(x) \leq TC_{lb}(S_0).$$

The abortion criteria is illustrated in Figure 2.

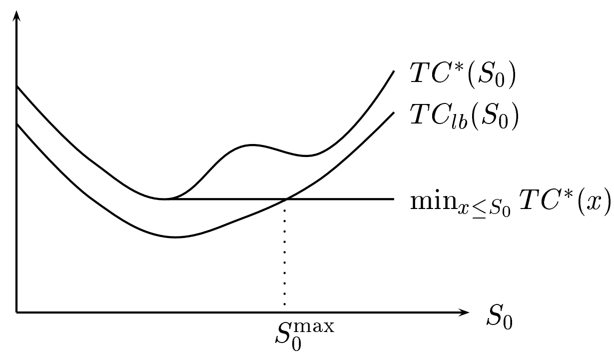

Figure 2: Illustration of the abortion criteria. The search for the optimal S_0 is aborted at $S_0^{\max}$.

The solution procedure can now be established as:

STEP 0: Set $S_0 = 0$ and $TC_{\min} = \infty$.

STEP 1: Set $k = 0$ and $\Lambda = \lambda_0$.

STEP 2: For each $i = 1, 2, \ldots, N$ calculate $\bar{L}_i$ by (4) and (5) and set $S_i(k) = \arg\min_S(C_i(S, \bar{L}_i))$.

STEP 3: If $k > 0$ and $S_i(k) = S_i(k-1)$ for all $i = 1, 2, \ldots, N$ then goto STEP 4, else calculate Λ by (3), set $k := k + 1$ and goto STEP 2.

STEP 4: Set $TC^*(S_0) = C_0(S_0, \Lambda) + \sum_{i=1}^{n} C_i(S_i(k), \bar{L}_i)$.
If $TC^*(S_0) < TC_{\min}$ then set $TC_{\min} = TC^*(S_0)$ and let $S_0^{opt} = S_0$ and $S_i^{opt} = S_i(k)$ for $i = 1, 2, \ldots, N$.
If $TC_{\min} < TC_{lb}(S_0)$ then STOP, else set $S_0 = S_0 + 1$ and goto STEP 1.

To show that $S_i(k)$ converges for fixed S_0 we conjecture that the optimal order-up-to level at the retailer either increases or remains the same when the expected leadtime between the warehouse and the retailer increases. The conjecture is very intuitive, and is supported by all our numerical tests. However, as of yet we have not been able to prove it formally. For each S_0 we initialize the procedure with $\Lambda = \lambda_0$. It is easy to show that a lower demand rate at the warehouse leads to a shorter expected leadtime from the warehouse to the retailers (remember S_0 is fixed). It now follows that $S_i(k+1) \leq S_i(k)$ for all i, k establishing the convergence since $S_i \geq 0$.

4 Numerical Results

In order to examine the effectiveness of the presented methodology we have performed a small numerical study. In total we consider 36 different test problems with five identical retailers. For each test problem we find the best order-up-to levels according to our method. We also obtain the approximate total holding and stock out costs for the inventory system. The accuracy of these results are then evaluated by simulation. Each simulation consists of 10 runs, each with a run length of 100 000 time units. The result is a confidence interval for the exact cost. We express the confidence limits on a 95% significance level. A comparison between the total cost given by our method and the total cost for the simulation gives an indication of how accurate our method is when estimating the total cost for the inventory system.

We also use simulation to determine the optimal policy for the system. The cost for this policy can then be compared with the cost for the policy determined by our technique, to obtain an estimate for the performance of the method when optimizing the ordering policies. The policy that we report as the optimal policy is the policy with the lowest average simulated cost, which is obtained by enumeration over combinations of S_0 and S_i (we only search within policies where the order-up-to levels are identical for the retailers). However, this policy does not necessarily dominate all the other policies when taking confidence intervals into consideration.

The problem data and results can be found in Table 1. We only report the optimal policy when it is different from the one obtained from our algorithm. From Table 1 we can see that our method performs rather well for all the considered problems. It seems that we mostly tend to underestimate the total cost, especially in the problems with high stockout costs at the retailers. This is due to the METRIC-approximation, where the stochastic leadtimes are replaced by their averages when evaluating the costs for the retailers. On average the method underestimates the costs with 1.1 %.

In 13 problems we can observe (on a significance level of 95%) that the method fails to find the optimal policy. In 9 more problems the policy suggested by our method does not have the lowest average cost according to the simulation runs. However, in these cases the deviations are not significant on a 95% confidence level. In comparison to the optimal policies obtained by simulation, the increase in costs by using the policies obtained by our method is only 0.40 %, on average. In 16 of the 22 problems where we fail to find the true optimal policy, the method merely underestimates the order-up-to level at the warehouse by a single unit. In one problem the warehouse order-up-to level is underestimated by two units. In the other 5 problems where the optimal policy is not found, the method tends to allocate more stock to the retailers and less stock to the warehouse than what is optimal from a cost perspective.

Finally it seems that our methodology performs better if the warehouse leadtime is small compared to the transportation time from the warehouse to the retailers. In the 12 problem instances with $L_i = 1.5$ the average cost increase, SC/CC is only 0.07%, whereas in the problems with $L_i = 0.5$, the corresponding figure is considerable higher, 0.67%. This behavior is due to the METRIC-approximation, where the stochastic replenishment leadtime facing a retailer is replaced by its mean value. If the constant transportation time to the retailers is large compared to the warehouse leadtime, the stochastic delivery delays tend to have less relative variation and consequently the impact of the METRIC-approximation will be smaller.

5 Conclusions and directions for future research

This paper presents a heuristic method for evaluation and optimization of $(S-1, S)$-policies for a one warehouse, multiple retailers inventory system. The evaluation technique uses the well-known METRIC-approximation as a framework. From a computational point of view the presented technique is very efficient and simple. Numerical results also indicate that the performance is quite good.

Up to our knowledge, no paper is yet published, which deals with lost sales in a continuous review multi-echelon inventory setting. Moreover, the original backorder METRIC-model [10] is one of the most widely used multi-echelon inventory models. Our lost sales generalization makes the policy evaluation a bit more complex, since we have to use an iterative procedure to obtain the cost. Still, the model is rather simple and easy to implement. Moreover, in many practical situations lost sales is a reasonable way to model stockouts. Therefore our technique is also relevant for practitioners.

In a research perspective our model can form a framework in which different generalizations can be considered as options for future research. For example, batch ordering policies and more general demand processes may be analyzed, still using the ideas presented in this paper. Generalizations to periodic review policies is also important. We are also interested in finding a formal proof of the conjecture in section 3. The derivation of an exact evaluation of costs seems to be a very difficult problem to solve. This is a real challenge for future research.

Appendix

Proof for Lemma 1

We need to show that

$$\min_{S_i} C_i(S_i, L_i) \leq \min_{S_i} C_i(S_i, \bar{L}_i) \text{ for } L_i \leq \bar{L}_i. \tag{6}$$

Let l_i be an arbitrarily chosen leadtime, where $L_i \leq l_i \leq \bar{L}_i$. Consider the cost $C_i(S_i, l_i)$, where S_i is set to its optimal value for each l_i. Obviously, $C_i(S_i, l_i) \leq C_i(S_i - 1, l_i)$ for each l_i such that $L_i \leq l_i \leq \bar{L}_i$. Start with $l_i = \bar{L}_i$ and let l_i be continuously lowered until we reach $l_i{=}L_i$, while S_i is set to its optimal value for each l_i. Since $C_i(S_i, l_i)$ is a continuous function of l_i for fixed S_i, it also is a continuous function of l_i when S_i is optimally chosen. Moreover, the fact that S_i minimizes the cost $C_i(S_i, l_i)$, implies that $C_i(S_i, l_i) \leq C_i(S_i - 1, l_i)$. Consequently, (6) follows if

$$C_i(S_i, l_i) \leq C_i(S_i - 1, l_i) \Rightarrow \frac{\partial C_i(S_i, l_i)}{\partial l_i} \geq 0 \text{ for } L_i \leq l_i \leq \bar{L}_i. \tag{7}$$

For notational reasons we omit the index i from all variables. It can be shown that

$$\frac{\partial C(S,l)}{\partial l} = -h\lambda(1 - q^S(S)) + \lambda(h\lambda l + \pi\lambda)(q^S(S-1) - q^S(S) + q^S(S)^2). \tag{8}$$

Moreover, $C(S,l) \leq C(S-1,l)$ implies that

$$\frac{h}{h\lambda l + \pi\lambda} \leq q^{S-1}(S-1) - q^S(S) \tag{9}$$

Let

$$A = \frac{1}{\lambda(h\lambda l + \pi\lambda)} \frac{\partial C_i(S,l)}{\partial l}. \tag{10}$$

From (8) we have that

$$A = \frac{h}{h\lambda l + \pi\lambda}\big(q^S(S) - 1\big) + q^S(S-1) - q^S(S) + q^S(S)^2). \tag{11}$$

(9) and (11) now give (recall that $q^S(S) < 1$)

$$\begin{aligned} A &\geq (q^{S-1}(S-1) - q^S(S))(q^S(S) - 1) + q^S(S-1) - q^S(S) + \big(q^S(S)\big)^2 \\ &= q^{S-1}(S-1)q^S(S) - q^{S-1}(S-1) + q^S(S-1) \\ &= q^{S-1}(S-1)q^S(S) - (q^{S-1}(S-1)q^S(S)) \\ &= 0. \end{aligned}$$

Note that $q^S(S) \leq 1$. Consequently, $A \geq 0$ and therefore (7) holds and the proof is complete.

Proof for Lemma 2

Since $\Lambda \leq \lambda_0$, we only need to show that $\frac{\partial C_0(S_0,\Lambda)}{\partial \Lambda} \leq 0$. The convexity of $C_0(S_0, \lambda_0)$ in S_0 follows, for example, from Axsäter [3].

$$\begin{aligned} \frac{\partial C_0(S_0,\Lambda)}{\partial \Lambda} &= -h_0 L_0 \exp(-\Lambda L_0)\left(S_0 + \sum_{j=1}^{S_0}(S_0 - j)\cdot\left(\frac{(\Lambda L_0)^j}{j!} - \frac{(\Lambda L_0)^{(j-1)}}{(j-1)!}\right)\right) \\ &= -h_0 L_0 \exp(-\Lambda L_0)\sum_{j=0}^{S_0-1}\frac{(\Lambda L_0)^j}{j!} \\ &\leq 0. \end{aligned}$$

References

[1] R. Anupindi and Y. Bassok. Centralization of stocks: Retailers vs. manufacturer. *Management Science*, 45:178–191, 1999.

[2] S. Axsäter. Simple solution procedures for a class of two–echelon inventory problems. *Operations Research*, 38:64–69, 1990.

[3] S. Axsäter. Continuous review policies for multi–level inventory systems with stochastic demand. In S.C. Graves, A.H.G. Rinnooy Kan, and P. Zipkin, editors, *Handbooks in OR & MS, Vol. 4*, pages 175–197. Elsevier Science Publishers B.V., North–Holland, 1993.

[4] S Axsäter. Exact analysis of continuous review (R, Q)–policies in two–echelon inventory systems with compound Poisson demand. *Operations Research*, 48:686–696, 2000.

[5] G.P. Cachon. Exact evaluation of batch-ordering inventory policies in two–echelon supply chains with periodic review. *Operations Research*, 49(1):79–98, 2001.

[6] G. Hadley and T.M. Whitin. *Analysis of Inventory Systems.* Prentice–Hall, Englewood Cliffs, NJ, 1963.

[7] R.M. Hill. On the suboptimality of $(S-1, S)$ lost sales inventory policies. *International Journal of Production Economics*, 59(1-3):377–385, 1999.

[8] S. Nahmias and S.A. Smith. Optimizing inventory levels in a two-echelon retailer system with partial lost sales. *Management Science*, 40:582–596, 1994.

[9] C. Palm. Analysis of the Erlang traffic formula for busy signal assignment. *Ericson Technics*, 5:39–58, 1938.

[10] C.C. Sherbrooke. METRIC: A multi-echelon technique for recoverable item control. *Operations Research*, 16:122–141, 1968.

[11] C.C. Sherbrooke. *Optimal inventory modeling of systems.* John Wiley & Sons, New York, 1992.

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

	λ_i	π_i	L_i	S	Calc cost	Sim cost	Spread	Opt pol	Cost	Spread	Cost dev
1	1	5	0.5	4,2	10.82	10.74	0.01				
2	1	5	1.0	2,3	11.99	12.03	0.02				
3	1	5	1.5	3,3	12.51	12.46	0.01				
4	1	25	0.5	5,3	15.86	16.15	0.03				
5	1	25	1.0	4,4	18.18	18.43	0.04	5,4	18.41	0.03	0.1%
6	1	25	1.5	3,5	19.94	20.16	0.06	4,5	20.12	0.04	0.2%
7	1	125	0.5	5,4	20.27	21.27	0.05	6,4	20.90	0.06	1.7%
8	1	125	1.0	5,5	23.64	24.16	0.06	6,5	24.09	0.06	0.3%
9	1	125	1.5	5,6	26.15	26.45	0.08				
10	2	5	0.5	8,3	15.43	15.41	0.02	9,3	15.33	0.02	0.5%
11	2	5	1.0	6,5	17.11	17.27	0.02	8,4	17.16	0.02	0.6%
12	2	5	1.5	5,6	18.23	18.34	0.02	7,5	18.29	0.03	0.3%
13	2	25	0.5	8,5	21.56	22.52	0.04	9,5	22.30	0.04	1.0%
14	2	25	1.0	9,6	24.96	25.35	0.06	10,6	25.33	0.05	0.1%
15	2	25	1.5	7,8	27.49	27.92	0.08	8,9	27.90	0.04	0.1%
16	2	125	0.5	9,6	26.82	28.56	0.12	10,6	28.15	0.07	1.5%
17	2	125	1.0	9,8	31.84	32.75	0.08				
18	2	125	1.5	10,9	35.49	36.07	0.11				
19	1	5	0.5	4,1	16.96	16.41	0.02				
20	1	5	1.0	2,2	17.55	17.51	0.02				
21	1	5	1.5	2,2	18.14	18.04	0.03				
22	1	25	0.5	4,3	27.50	27.99	0.03	5,3	27.96	0.02	0.1%
23	1	25	1.0	5,3	30.81	30.72	0.06	6,3	30.65	0.05	0.2%
24	1	25	1.5	4,4	33.07	33.11	0.05				
25	1	125	0.5	7,3	36.86	37.21	0.12	8,3	36.92	0.08	0.8%
26	1	125	1.0	4,5	42.75	43.64	0.14	7,4	43.19	0.17	1.1%
27	1	125	1.5	6,5	46.83	46.92	0.10				
28	2	5	0.5	6,3	24.21	24.42	0.02	8,2	24.14	0.03	1.2%
29	2	5	1.0	5,4	26.48	26.60	0.02	7,3	26.18	0.03	1.6%
30	2	5	1.5	6,4	27.61	27.43	0.02				
31	2	25	0.5	10,4	36.44	37.10	0.06	11,4	36.95	0.05	0.4%
32	2	25	1.0	10,5	42.85	42.91	0.12	11,5	42.85	0.07	0.1%
33	2	25	1.5	8,7	46.74	47.11	0.08				
34	2	125	0.5	11,5	47.24	48.92	0.08	13,5	48.21	0.12	1.5%
35	2	125	1.0	10,7	56.53	57.73	0.16	11,7	57.37	0.11	0.6%
36	2	125	1.5	11,8	63.77	64.19	0.19	12.8	64.05	0.10	0.2%

Table 1: Numerical results. The optimal policy is only reported when it is different than the policy suggested by our algorithm.

Summary

This thesis deals with computation of control policies for a number of different inventory systems. Inventories arise when demand and supply are not perfectly matched. Typically, there are economies of scale by ordering a batch of products, rather than ordering one unit for every demand. The supply may also be delayed by production or transportation times, and therefore we need to maintain an inventory to satisfy demands when they occur. Inventories are costly and therefore efficient control policies specifying how the inventory should be replenished, are needed.

In this thesis we show how different inventory systems can be modelled, and compute control policies for these systems. The focus is on the modelling and optimization of systems based on mathematical analysis, rather than the actual implementation of the control policies, and the thesis is therefore theoretical rather than practical. Although theoretical, all of our analyses is backed by numerical examples, where we illustrate the performance of the suggested policies. One of the research objectives has been to keep the suggested inventory policies simple. Simple policies are easier to understand and to implement, and in many cases the cost difference between a simple policy and an optimal policy is small. This can be verified by numerical examples.

We do not attempt to model a full-scale supply chain, but focus on a few sub-systems of the supply chain, where we find the available methods developed so far inadequate. The following systems are analysed:

- An inventory system with several demand classes.
- A make-to-order system with several demand classes.
- A multiple-item, single-supplier inventory system.
- A two-echelon inventory system.

This thesis consists of a survey of the results obtained, and seven scientific papers, which are described here:

Inventory rationing in an (s, Q) inventory model with lost sales and two demand classes. In this paper, which is the first in a series of three, we deal with the management of inventory systems with two demand classes. We suggest a critical-level policy which besides the reorder point s and the order size Q consists of a critical level c. Demands from the low-priority class are satisfied only if the inventory level is above c. In this way it is possible to reserve stock for future high-priority demand. We illustrate, by numerical examples, that costs may be reduced with up to 10%, compared with a non-rationing approach.

Rationing policies for an inventory model with several demand classes and stochastic lead times. We extend the analysis to cover several demand classes and stochastic

lead times, and we moreover generalize the class of policies to include the optimal time-remembering policies. Simple critical level policies have constant critical levels while the critical levels of the time-remembering policy are allowed to vary over time. The analysis is based on Markov decision theory and in a numerical study we analyze the difference between the simple critical level policy and the optimal policy and find differences between 0-3%.

Restricted time-remembering policies for the inventory rationing problem. We investigate a class of time-remembering policies which are restricted to be constant over intervals of time. These policies are simpler from a practical point of view. A neighbor search based on properties of the policy variables facilitates fast optimization and numerical results show that the policies perform well.

Rationing of a congested multi-period make-to-order system. In this paper we apply the rationing policies to a congested make-to-order system with several demand classes. In a make-to-order system products are tailored to the actual order or job, and it is impossible to keep inventory. The asset to be managed is instead the production capacity. Incoming jobs must be either accepted or rejected upon arrival, after which the workload of the job is allocated to the periods in the planning horizon. Simple and near-optimal policies are discussed and evaluated in a numerical study.

Calculating can-order policies for the joint replenishment problem by the compensation approach. We study the problem of coordinating an inventory system with several items with the same supplier. When an item places an order other items can join this order and thereby obtain a reduced cost of ordering. We introduce the compensation approach, in which the benefit of the discount opportunity of other items, is taken into account, when an item considers to place an order. The new method is used to compute improved so-called can-order policies for the joint replenishment problem.

The can-order policy for the periodic-review joint replenishment problem. The compensation approach is, in this paper, applied to a periodic-review can-order policy. We focus on examples with irregular demand, and show that the can-order policy, in opposition to the general belief, has a strong performance, often better than the periodic-ordering policies.

A two-echelon inventory model with lost sale. In the final paper of the thesis we discuss a system consisting of one warehouse and several retailers. The prevalent assumption for such systems is that demands not satisfied immediately are backordered, since this leads to a much easier analysis. However, in competitive markets, such demands may very well be lost, and we therefore develop an iterative algorithm which computes near-optimal base-stock policies for the system with lost sales.