

University of Groningen

## Planning of Combined Make-to-Order and Make-to-Stock Production

Beemsterboer, Bartholomeus Jacobus

**IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.**

*Document Version*

Publisher's PDF, also known as Version of record

*Publication date:*

2016

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*

Beemsterboer, B. J. (2016). *Planning of Combined Make-to-Order and Make-to-Stock Production*. [Thesis fully internal (DIV), University of Groningen]. University of Groningen, SOM research school.

### Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

### Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

# Chapter 5

## Two-product Storage-capacitated Inventory Systems: a Technical Note

*Abstract.* This paper considers a two-product inventory model with limited storage capacity and constant demand rates. We aim at finding an ordering policy that minimizes the cost per time unit. In the literature, several solution methods have been developed for this problem, but these are limited to very restrictive classes of policies. We consider a much more general class where the order quantity of one of the products is allowed to vary. These policies are still cyclic and easy to implement. Closed-form expressions are derived for determining the optimal order quantities. It is shown that savings of up to 25% are possible compared to existing approaches.

### 5.1 Introduction

The majority of inventory control literature is based on cost minimization and ignores capacity constraints. In real life, such constraints often do exist, either storage or production related. This paper focuses on storage related capacity restrictions, which may result from physical limitations or internal budget restrictions.

Many textbooks, including Hadley & Whitin (1963), suggest coping with storage capacity limitations through partitioning of the storage capacity. In this approach, each product gets its own share of the storage capacity. This approach is mainly referred to as the *Lagrange multiplier approach*, but the terms *independent cycle*, *independent solutions* and *fixed storage approach* are also used. The advan-

tage of this approach is that the negative effects of the capacity restriction can be spread evenly over the products. An important disadvantage is that the capacity is not used efficiently. Indeed, for deterministic demand, the average capacity usage is always 50 per cent.

Another approach is to use the same cycle time for all products. This approach is referred to as the *fixed cycle*, *pure cycle*, *rotation cycle* or *common cycle* approach. In this approach, a common cycle time is determined and all orders are phased within this cycle, such that the storage capacity is used more efficiently. The main problem is to determine this phasing, i.e. the sequencing of the products, usually called the *staggering* of the products. Homer (1966) was the first who solved this staggering problem to optimality. His result was rediscovered, independently, by Page & Paul (1976), Zoller (1977), and Hall (1988). The main advantage of this approach is that capacity is used more efficiently. An important disadvantage, however, is that forcing ordering cycles to become equal can be very costly.

Combining the two main approaches has also been suggested. Page & Paul (1976) provide a method that clusters the products. All products within a cluster have the same order interval, leading to an efficient use of the capacity, while the order intervals vary across the clusters. Anily (1991) provides another method that builds on the same rationale but determines the clusters in a different way. She shows that the performance of her clustering method does not exceed some lower bound on the costs by a factor larger than  $\sqrt{2}$ .

Besides combinations of the two main approaches, several authors generalized the methods. Gallego et al. (1996) generalized the fixed cycle approach to a powers-of-two policy. Order quantities remain fixed. The authors provide heuristics that determine ordering policies. An even more general approach is used by Hartley & Thomas (1982). They consider a two-product model in which they allow each product to be ordered several times per cycle, though still in fixed amounts. They provide an optimal solution procedure in a companion paper (Thomas & Hartley, 1983). Murthy et al. (2003) and Boctor (2010) consider the problem of offsetting the replenishment cycles by integer multiples of some base period, and use the result that, if the integer multiples of two items are not relatively prime, it is possible to offset their cycles such that the peaks of their inventory cycles never coincide over an infinite time horizon. Murthy et al. propose a heuristic for this framework, which is improved by Boctor.

Several authors provide structural insights. Anily (1991) provides a lower

bound on the costs of inventory policies with constant order quantities and Gallego et al. (1996) prove that this bound holds even for varying quantities. Finally, Gallego et al. (1992) prove that the problem of determining an optimal ordering policy subject to a storage capacity restriction is strongly NP-complete.

All the above discussed contributions consider policies with constant order quantities. Some other contributions do focus on varying order quantities. Hariga & Jackson (1995) formulate a nonlinear program that minimizes holding and ordering costs by selecting order quantities and the overall cycle length. They provide a heuristic solution procedure and conditions under which fixed order quantities are optimal. However, throughout the entire paper, they assume that the sequence of orders is given, which, in fact, constitutes the most challenging part of the determination of an optimal ordering policy.

Güder et al. (1995) also allow varying order quantities. They provide a myopic procedure that determines order quantities one by one, which in general results in a non-cyclic ordering policy. The order quantities that the procedure selects are based on the Economic Order Quantity as well as the available storage space. An upper bound on the optimality gap is not provided.

In this paper, we present an exact approach for the two product inventory model with a capacity constraint and varying order quantities. We provide closed-form expressions for optimizing order quantities and ordering moments for a very general class of ordering policies. We further derive theoretical and numerical results on the suboptimality of existing approaches, which turns out to be considerable in many situations.

This paper proceeds as follows. Section 5.2 describes the inventory system and introduces the concepts of simple and general cycles. In Section 5.3, structural properties of the optimal simple cycle policy are derived and used to determine the optimal timing and quantity of orders. Section 5.4 compares the cost performance of the optimal simple cycle policy to that of existing approaches. Section 5.5 shortly discusses general cycle policies. We end in Section 5.6 with a summary of the key findings and insights, and a discussion of research opportunities.

## 5.2 System description

Consider a two product, infinite time horizon inventory system. The products share a common storage resource with limited capacity. The products are numbered 1 and 2. Demand rates are deterministic and denoted by  $d_1$  and  $d_2$ . Demand is measured in *capacity units per time unit*, i.e. as the rate with which capacity decreases, to allow for normalization of the capacity to unity. Subscripts refer to the product number. Lead times are constant and backorders are not allowed.

The objective is minimizing costs per time unit by determining ordering *moments* and *quantities*. This average cost per time unit is denoted by  $C$  and based on a cost per order for each of the two products, which are denoted by  $A_1$  and  $A_2$ , respectively. We will focus purely on the ordering costs as we consider situations where ordering quantities are restricted by the limited available capacity rather than the need to avoid excessive holding costs.

Our attention is restricted to cyclic solutions. A *cyclic solution* is a solution in which ordering moments and quantities are described for a finite time interval, the *cycle*, and the inventory levels at the beginning and at the end of this cycle are equal. As a result, the ordering policy within the cycle can be repeated infinitely many times. The length of the cycle is denoted by  $T$ , which is a decision variable that will vary for different configurations. Cycles in which at least one of the products is ordered only once, which implies that its order quantity does not vary, will be called *simple cycles*. Cycles without any further restrictions will be referred to as *general cycles*.

## 5.3 Simple cycles

Let the *base product* refer to the product that is ordered once per cycle. If both products are ordered once, an arbitrary product can be selected as the base product. We will present the analysis for the case where product 1 serves as the basis, leading to the optimal policy of that type, and then 'copy' the result for the other case where product 2 is the base product.

The replenishment order of product 1 is referred to as the *product 1 order* and its order quantity is denoted by  $Q_1$ . Recall from Section 5.1 that the capacity is normalized to one and hence  $Q_1 \leq 1$ . The number of orders of product 2 during the cycle is denoted by  $m$ , which is restricted to be integer. The product 2 orders

are numbered according to their position in the cycle, starting from the product 1 order. Hence the  $i$ -th product 2 order after the product 1 order will be named *product 2 order  $i$*  and its quantity will be denoted by  $Q_{2,i}$ , for  $i = 1, \dots, m$ , where  $0 < Q_{2,i} \leq 1$ .

The above properties and notations lead to the following expression for the cycle time  $T$  of simple cycles.

$$T = \frac{Q_1}{d_1} = \frac{\sum_{i=1}^m Q_{2,i}}{d_2}. \quad (5.1)$$

The cost per time unit, which should be minimized, is

$$C = \frac{A_1 + mA_2}{T}. \quad (5.2)$$

It is easy to see that, for fixed  $m$ , maximization of  $T$  implies minimization of  $C$ . In turn, (5.1) shows that maximizing the order quantities maximizes  $T$ . This observation suggests that (a) the products should be ordered only when their stock is empty, and (b) the order quantities fill all remaining capacity, i.e. order quantities are of maximum size. In the next subsection, we show that optimal solutions indeed always satisfy properties (a) and (b) and we derive expressions for optimal order quantities by applying these.

### 5.3.1 Deriving optimal order quantities

Let us assume that we have decided on the base product (numbered 1) and the number,  $m$ , of orders per cycle for product 2. In the next subsection, we will show how to select the base product and the value of  $m$  optimally, in order to find the overall best simple cycle policy.

The following theorem states that all orders must be of maximum size. We remark that Hariga & Jackson (1995) obtain the same results under the objective of minimizing the storage capacity, but not under cost minimization as considered here.

**Theorem 5.1.** *Under an optimal simple cycle policy, every order is of maximum size, i.e. uses up all spare storage capacity when it arrives.*

A proof by contradiction is provided in Appendix C.

The following corollary shows that next to being of maximum size, orders should always arrive just in time. A formal proof is provided in Appendix C, but the logic is as follows. An order that arrives early (when the inventory level is still positive) can be postponed, which leads to an alternative policy with an order of non-maximum size. As the alternative policy cannot be optimal (using Theorem 5.1), neither can the original policy. We remark that Anily (1991) proves the same, but along different lines.

**Corollary 5.1.** *Under an optimal policy, a product arrives exactly when its inventory level reaches zero.*

Now, we derive optimal order quantities as follows. Since the product 1 order is of maximum size, by Theorem 5.1, and arrives when the product 1 stock level is zero, by Corollary 5.1, the stock level of product 2 at that time is  $(1 - Q_1)$ . So, product 2 order 1 will be placed  $\frac{1-Q_1}{d_2}$  time units later and, by Theorem 5.1, its quantity is equal to the total demand over this period, i.e.

$$Q_{2,1} = \frac{d_1 + d_2}{d_2} (1 - Q_1). \quad (5.3)$$

In the same way, we find that

$$Q_{2,j} = \frac{d_1 + d_2}{d_2} Q_{2,j-1}, \text{ for } j = 2, \dots, m. \quad (5.4)$$

Combining (5.3) and (5.4) gives

$$Q_{2,j} = \left( \frac{d_1 + d_2}{d_2} \right)^j (1 - Q_1), \text{ for } j = 1, \dots, m. \quad (5.5)$$

Using the same reasoning, but now applied to the relation between product 2 order  $m$  and the next product 1 order that follows, we get

$$Q_1 = \frac{d_1 + d_2}{d_1} (1 - Q_{2,m}). \quad (5.6)$$

Combining (5.5), applied to product 2 order  $m$ , and (5.6) yields a system of two linear equations with two unknowns. After some algebra this results in

$$Q_1 = 1 - \frac{1}{\left(\frac{d_1+d_2}{d_2}\right)^{m+1} - \frac{d_1}{d_2}}. \quad (5.7)$$

Inserting (5.7) into (5.5) gives

$$Q_{2,j} = \frac{\left(\frac{d_1+d_2}{d_2}\right)^j}{\left(\frac{d_1+d_2}{d_2}\right)^{m+1} - \frac{d_1}{d_2}}, \text{ for } j = 1, \dots, m. \quad (5.8)$$

Letting  $C_1(m)$  denote the cost per time unit in case product 1 is the base product and product 2 is ordered  $m$  times, combining (5.1), (5.2) and (5.7) gives

$$C_1(m) = \frac{d_1 (A_1 + mA_2)}{1 - \frac{1}{\left(\frac{d_1+d_2}{d_2}\right)^{m+1} - \frac{d_1}{d_2}}}, \text{ for } m \in \mathbb{N}. \quad (5.9)$$

As mentioned before, the analysis of the case where product 2 is the base product is completely similar. Letting  $C_2(n)$  denote the cost per time unit in case product 2 is the base product and product 1 is ordered  $n$  times, we get

$$C_2(n) = \frac{d_2 (nA_1 + A_2)}{1 - \frac{1}{\left(\frac{d_1+d_2}{d_1}\right)^{n+1} - \frac{d_2}{d_1}}}, \text{ for } n \in \mathbb{N}. \quad (5.10)$$

### 5.3.2 Selecting the base product and optimizing $m$ or $n$

We can obtain the optimal value of  $m$  or  $n$ , given as  $m^* = \arg \min \{C_1(m) | m \in \mathbb{N}\}$  and  $n^* = \arg \min \{C_2(n) | n \in \mathbb{N}\}$ , respectively, by a straightforward numerical search procedure. Furthermore, we next develop a graphical aid for determining the optimal base product and the number of order per cycle for the other product.

It is straightforward to rewrite  $C_1(m+1) \leq C_1(m)$  and  $C_2(n+1) \leq C_2(n)$  for general  $m$  and  $n$ , respectively, in terms of  $\frac{A_1}{A_2}$  and  $\frac{d_1}{d_2}$  rather than the individual cost parameters. Therefore, when determining  $m^*$  and  $n^*$ , only the two cost dimensions  $\frac{A_1}{A_2}$  and  $\frac{d_1}{d_2}$  matter. So we can display  $m^*$  and  $n^*$  graphically in Figure 5.1.



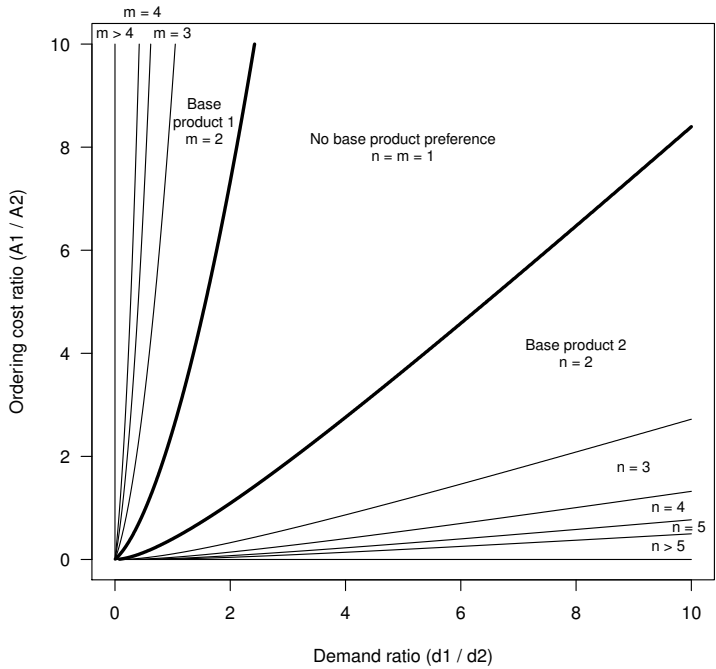


Figure 5.1: Optimal choices of the base product and of  $m$  or  $n$  depending on  $\frac{A_1}{A_2}$  and  $\frac{d_1}{d_2}$

## 5.4 Comparison to other methods

In this section, the simple cycle optimization procedure of the previous section will be compared to methods that are used in the literature. We restrict to solution methods generating cyclic ordering policies, as these are comparable (in complexity) to ours.

### 5.4.1 Capacity partitioning

Although in general an optimal partitioning cannot easily be obtained, as several authors show, a capacity partitioning variant of the model considered in this paper can be solved to optimality algebraically. This solution will be derived here in order to compare it with the solution of Section 5.3.

Consider the model as introduced in Section 5.2. Introduce a capacity parti-

tioning  $P$ , where  $0 < P < 1$ . Assume that product 1 uses at most  $P$  capacity units and product 2 at most  $1 - P$  capacity units. In this model, the cost per time unit, denoted by  $C^{CP}$ , becomes,

$$C^{CP} = \frac{A_1}{T_1} + \frac{A_2}{T_2}, \quad (5.11)$$

where  $T_1$  and  $T_2$  denote the cycle lengths of product 1 and 2, respectively. Obviously, given  $P$ , the optimal cycle lengths are  $T_1 = \frac{P}{d_1}$  and  $T_2 = \frac{1-P}{d_2}$ . Omitting the straightforward algebra, the optimal partitioning  $P^*$  and corresponding costs  $C^{CP^*}$  are

$$P^* = \begin{cases} \frac{d_1 A_1 - \sqrt{d_1 A_1 d_2 A_2}}{d_1 A_1 - d_2 A_2} & \text{if } d_1 A_1 \neq d_2 A_2, \\ \frac{1}{2} & \text{if } d_1 A_1 = d_2 A_2, \end{cases} \quad (5.12)$$

$$C^{CP^*} = d_1 A_1 + d_2 A_2 + 2\sqrt{d_1 A_1 d_2 A_2}. \quad (5.13)$$

The latter expression can be used to compare the costs with the optimal costs in case of simple cycles. In our numerical tests, we observed the largest relative difference for cases with  $d_1 = d_2$  and  $A_1 = A_2$ . Using the results of Section 5.3, it follows that an optimal simple cycle policy for this case with product 1 as the base product,  $m = 1$  and  $Q_1 = Q_{2,1} = T = \frac{2}{3}$  with cost  $C = 3d_1 A_1$  per time unit. Using (5.12) and (5.13), we find that the best partitioning is  $P = \frac{1}{2}$  with cost  $C^{CP} = 4d_1 A_1$  per time unit. Hence, capacity partitioning can lead to 33% extra cost compared to simple cycle policies.

## 5.4.2 Other approaches

The other basic approach, besides capacity partitioning, is the fixed cycle approach. Anily (1991) has shown that costs resulting from this approach may be infinitely many times larger than her lower bound. Our numerical results indicate that this result carries over to policies with varying order sizes, i.e. that fixed cycles can be arbitrarily worse than simple cycles.

Güder et al. (1995) provide a myopic procedure that determines an ordering policy that is always at least as good as the optimal capacity partitioning policy. However, if the products have the same cycle time in the optimal capacity partitioning solution, which is clearly the case if  $P = \frac{1}{2}$  and  $d_1 = d_2$ , the procedure of Güder et al. (1995) yields exactly the same policy. So ordering policies generated by the procedure of Güder et al. (1995) can also lead to 33% extra cost compared

to simple cycle policies.

## 5.5 General cycles

In this section, the model of Section 5.3 will be extended to general cycles and we present an example where a general cycle policy outperforms the best simple cycle policy. The cost saving is small though, a finding that was replicated for many other settings that we considered. A cost gap of 1.74% was the largest that we found in our numerical testing.

If both  $n$  (the number of product 1 orders in a cycle) and  $m$  (the number of product 2 orders in the same cycle) are allowed to be larger than 1, then (5.1) and (5.2) become, respectively,

$$T = \frac{\sum_{i=1}^n Q_{1,i}}{d_1} = \frac{\sum_{i=1}^m Q_{2,i}}{d_2}, \quad (5.14)$$

$$C = \frac{nA_1 + mA_2}{T}. \quad (5.15)$$

Consider the following example. Let  $d_1 = d_2 = A_2 = 1$  and  $A_1 = 3$ . Using the optimization procedure of Section 5.3, the following optimal simple cycle is determined. Product 1 is the base product,  $m = 2$ ,  $Q_1 = \frac{6}{7}$ ,  $Q_{2,1} = \frac{2}{7}$  and  $Q_{2,2} = \frac{4}{7}$ . The cost per time unit of this solution is  $C = \frac{35}{6}$ .

Now, consider a general cycle with  $n = 2$  and  $m = 3$ . Note that these values do not allow an equivalent simple cycle policy. Assume that product 2 is ordered twice between product 1 order 1 and product 1 order 2, and once between product 1 order 2 and the beginning of the next cycle. Using similar reasoning as in Subsection 5.3.1, we get  $Q_{2,1} = 2(1 - Q_{1,1})$ ,  $Q_{2,2} = 2Q_{2,1} = 4(1 - Q_{1,1})$ ,  $Q_{1,2} = 2(1 - Q_{2,2})$ ,  $Q_{2,3} = 2(1 - Q_{1,2})$ ,  $Q_{1,1} = 2(1 - Q_{2,3})$ .

This gives  $Q_{1,1} = \frac{26}{31}$ ,  $Q_{2,1} = \frac{10}{31}$ ,  $Q_{2,2} = \frac{20}{31}$ ,  $Q_{1,2} = \frac{22}{31}$  and  $Q_{2,3} = \frac{18}{31}$ . Using (5.14) and (5.15), it turns out that  $T = \frac{48}{31}$  and  $C = \frac{93}{16} < \frac{35}{6}$ . Hence the costs are smaller than the costs of the optimal simple cycle, though only by 0.4%.

## 5.6 Conclusion and further research

We proposed a new class of so-called simple cycle policies for two-product capacitated inventory systems with constant demand. These policies differ from those proposed in the literature by allowing varying order sizes for one of the products, where the other product serves as the base product with a constant order quantity. Insightful structural properties on the optimal policy of this type were derived for situations where holding costs are negligible, i.e. when order quantities are capped because of the limited storage capacity available rather than to limit storage costs. These properties state that orders should only arrive if stock drops to zero, and should always be of maximum size, i.e. use up all available spare capacity upon arrival. Using these properties, we were able to derive closed form expressions for determining the optimal simple cycle policy. We showed that simple cycle policies can lead to considerable savings compared to existing approaches with fixed order quantities. Capacity partitioning can lead to 33% extra cost. The same holds for the approach proposed by Güder et al. (1995).

An important extension of our research would be to consider non-negligible holding costs. In fact, our cost and order quantity expressions can easily be generalized for such situations. However, it is important to realize that it may no longer be optimal to always maximize the size of orders when holding costs are positive. Intuitively, this remains logical when storage availability and not storage costs drive ordering decisions, i.e. in situations with relatively small holding costs and low storage availability (as for the examples that we considered). In opposite situations with high holding costs and ample storage capacity, as usually assumed in the inventory literature, products can always be ordered in constant economic order quantities. Future research could address when to best switch from one type of policy to another, and also develop and evaluate hybrid policies.

Another important extension is to consider more than two products. Our analysis of deriving optimal order quantities can be extended to such situations for given order frequencies and sequences, but the number of possible order frequencies and sequences grows exponentially with the number of products. With respect to order frequencies, one possibility is to only allow power-of-2 policies that have been shown to perform very well for other multi-product problems such as the joint replenishment problem and the economic lot scheduling problem (Axsäter, 2007).

Situations with varying demand rates also deserve attention. The traditional approach (for single product models) of using a fixed safety stock per product

will no longer be effective if this uses up a large fraction of the limited available storage capacity (in periods with peak demand). Policies should be developed that dynamically optimize order quantities, safety stocks and storage buffer space for future orders simultaneously.

Finally, it would be interesting to consider pooling of stocks and storage space between multiple inventory locations of a company to deal with storage restrictions. It is well known from the literature that pooling reduces total stocks needed. Sharing storage space may help avoid the need for renting additional space.

**Acknowledgment.** The authors thank the editor and reviewers for their continued critical but constructive remarks.