Optimal Stock Allocation for a Capacitated Supply System

Francis de Véricourt • Fikri Karaesmen • Yves Dallery
Fuqua School of Business, Duke University, Durham, North Carolina 27708
Department of Industrial Engineering, Koç University, 80910 Sariyer-Istanbul, Turkey
Laboratoire Génie Industriel, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Châtenay-Malabry Cedex, France
fdv1@mail.duke.edu • fkaraesmen@ku.edu.tr • dallery@lgi.ecp.fr

We consider a capacitated supply system that produces a single item that is demanded by several classes of customers. Each customer class may have a different backorder cost, so stock allocation arises as a key decision problem. We model the supply system as a multicustomer make-to-stock queue. Using dynamic programming, we show that the optimal allocation policy has a simple and intuitive structure. In addition, we present an efficient algorithm to compute the parameters of this optimal allocation policy. Finally, for a typical supply chain design problem, we illustrate that ignoring the stock allocation dimension—a frequently encountered simplifying assumption—can lead to incorrect managerial decisions.
(*Inventory/Production: Stock Allocation; Stochastic: Multi-Class; Queues: Make-to-Stock System*)

1. Introduction

Increasing demand variety and product proliferation at all stages of supply chains result in highly complicated structures. To deal with this complexity, a number of design strategies have emerged in recent years. Delayed product differentiation, centralization of stocks, or elaborate contracts between supply chain partners are well-known examples of such approaches. In many cases, the underlying design problem involves a stock allocation aspect. This paper investigates efficient stock allocation strategies and illustrates their impacts on a representative redesign decision.

The allocation problem typically appears when a supplier maintains a common stock in order to satisfy different customers. The stock, as well as the production capacity, are limited resources, therefore they must be rationed between the customers, possibly according to their relative economic importance for the supplier.

Delayed product differentiation is an attractive strategy in dealing with end-product variety. If differentiation can be postponed to a late stage in the supply process, a common stock of standard items is held and customer demands are responded through a rapid differentiation operation. When the customers have different economic values for the supplier, an allocation (or rationing) problem arises for the common stock. Centralization of inventories is another attractive design strategy to face geographical variety. In multiretailer systems, items are differentiated geographically due to multiple locations. Centralizing the retailer inventories to a common location raises again a stock allocation problem between the retail locations.

Both delayed product differentiation and centralization of inventories require significant changes in the structure of supply chain processes. A typical question that managers face is whether or not the redesign effort is compensated by the potential benefits of these strategies. In terms of inventory centralization, Eppen (1979) and Schwarz (1989) are examples that explore the benefits of pooling inventories. Recently, Alfaro and Corbett (1999) and Benjaafar et al. (2001) have presented detailed investigations for uncapacitated and capacitated systems respectively. As for

0025-1909/02/4811/1486$5.00
1526-5501 electronic ISSN

delayed differentiation, recent papers by Lee (1996) and Lee and Tang (1997) investigate inventory-cost-related benefits.

Another important class of design problems concerns supply chain contracts and their specifications. When each customer has a specific contract, delivery performance requirements may differ from one customer to another. This exerts pressure on the supplier to differentiate stock allocation priorities.

Despite the recent emphasis in analyzing such design issues, stock allocation strategies are rarely taken into account in these investigations. There seem to be two main reasons for this. First, as explained by Tsay et al. (1999) stock allocation problems are extremely difficult and generally considered intractable. Second, the allocation aspect is often viewed as an operational decision whereas the design problem is considered as a strategic one. This leads to the implicit assumption that stock allocation efficiency has little effect on the global outcome and hence can be disregarded. In this paper, we, not only show that the optimal allocation policy can be explicitly described, but also illustrate that it plays a critical role in the design decision.

In our model there is a supplier that produces the standard items and places them in a buffer in a make-to-stock manner. There are multiple classes of demand that can either be satisfied from stock (whenever available) or can be backordered. Since different demand classes have different backorder costs, some demands can be backordered in consideration of future (and more expensive) demand arrivals even though there may be items available in stock. The supplier has finite capacity and processes items one by one. The dynamic decision problem of interest is to find an optimal stock allocation policy that minimizes average inventory holding and backorder costs.

The basic stock allocation (or stock rationing) problem has been studied in various contexts in inventory control. Topkis (1968) formulates and solves an optimal dynamic rationing problem for an uncapacitated discrete time system facing random demand and shows that the optimal policy has a particular threshold structure that reserves items in stock for future (uncertain) demands of more valuable customers. Nahmias and Demmy (1981) consider a two-class uncapacitated inventory system and employ simple ordering and rationing policies to analyze the cost improvements due to stock rationing. Cohen et al. (1988) study a two-class inventory system that employs an (s, S)-type ordering policy and a strict priority rule for stock rationing. Finally, Frank et al. (1999) study the rationing problem for two classes of customers where, due to the supply contract, the demands of the first class must be completely satisfied but the demands of the second class can be partially satisfied. It is shown that the optimal rationing policy does not have a simple structure but that effective heuristics can be developed.

All of the above articles raise and analyze interesting issues in the context of inventory control. However, they do not model an important characteristic of the problem: the limited production capacity of the underlying supply system. An alternative modeling approach in this context is to employ queueing-based systems to explicitly model the limited production capacity and the associated randomness in material processing. We follow this approach and model the production stage by a single server. Since the system operates in a make-to-stock mode, the underlying basic model is the make-to-stock queue (see Buzacott and Shanthikumar (1993)). More precisely, our model follows the single-server, single-product, make-to-stock queue with multiple demand classes introduced in Ha (1997b, c). Ha (1997b) studies the stock rationing problem in a multiclient system with lost sales and characterizes the optimal policy. Carr and Duenyas (2000) investigate the structure of the optimal policy for a related two-class admission control/sequencing problem where demands from one of the classes can be rejected.

In a later article, Ha (1997c) studies a two demand class rationing problem for the make-to-stock queue where unsatisfied demands are backordered. In this case, the analysis is considerably more difficult than for the lost sales case due to the two-dimensionality introduced by backordered demands. Nevertheless, Ha shows that the optimal production policy is of base-stock type and that the optimal stock allocation policy has a monotone switching curve structure. Unfortunately, these properties do not lead to a complete characterization and, in general, to tractable policies. In this paper—as a special case of our more general result—we completely characterize the optimal stock allocation policy for the two-class problem. This optimal policy turns out to be very simple to understand and facilitates the formulation of managerial insights.

Moreover, there was apparently little hope of obtaining a full characterization for beyond two-customer classes. In fact, as stated in the conclusion of Ha(1997c), "… as the number of customer classes increases the optimal policy will be difficult to compute because of the curse of dimensionality and will be even more difficult to implement." We present in this paper a complete solution of the optimal stock allocation problem for any number of customer classes, generalizing the previous simple structure of the two-customer class cases.

More precisely, we consider the multiple-demand class extension of the two-demand class make-to-stock model by Ha (1997c). We investigate the structure of optimal stock allocation policies through dynamic programming. By exploiting the nested structure between an n-class problem and a related $n-1$ class problem, we obtain an exact characterization of the optimal stock allocation policy. Moreover, the characterization of the optimal policy is surprisingly simple: There are thresholds for each product such that it is optimal to satisfy the arriving demand from a customer from the on-hand stock if the stock level is above the threshold for that customer, and to backorder the demand otherwise. These thresholds also determine production priorities for backordered products in a simple way. Finally, we present an efficient algorithm to compute the optimal threshold levels thereby obtaining a complete characterization of the optimal policy.

We then focus on the investigation of the relative value of efficient stock allocation with regards to design decisions concerning delayed product differentiation or inventory centralization. In particular, we compare the potential inventory related benefits of such design decisions for two cases: when stock allocation is disregarded (by using a plausible but suboptimal allocation policy) and when stocks are allocated efficiently (using the optimal policy). The analysis shows that disregarding the stock allocation aspect can easily lead to wrong conclusions on system design.

In §2, we introduce the model and formulate the stock allocation problem as a dynamic programming problem. In §3, we study the structural properties of the optimal policy. In §4 we present the computation of the optimal policy for the problem of average cost minimization. In §5, we investigate the impact of stock allocation on inventory pooling through a numerical study. Our conclusions can be found in §6.

2. The Model

Consider a supplier who produces a single item at a single facility for n different classes of clients. The finished items are placed in a common inventory. When this inventory is empty, demands are backordered. When it is not, an arriving demand can be either satisfied by the on-hand inventory, or can be backordered. The items held in stock have unit holding costs of h (per unit time). Customers of class-i have unit backorder costs of b_i (per unit time). We denote by $\mathbf{b}$, the n-dimensional vector of backorder costs whose ith dimension is b_i. Suppose without loss of generality that the backorder costs are ordered such that $b_1 > \cdots > b_n$; that is, customer classes are ordered from the most valuable to the least valuable one. Note that the strict ordering is without loss of generality; if two distinct classes have identical backorder costs then they can be lumped into a single aggregate class. The demands of class-i customers arrive according to a Poisson process with rate λ_i and we note $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$. The supplier's facility is modeled by a single server whose processing time is exponentially distributed with mean $1/\mu$. Finally, in order to ensure stability, we assume that $\sum_{i=1}^{n} \lambda_i < \mu$.

At any time, one can decide to produce and allocate the finished product to satisfy the backorders of a class, to produce and allocate the product to the stock or not to produce at all. When a demand arrives, one can decide either to satisfy it from the on-hand inventory, or to backorder it. With linear backorder costs, an arriving demand of class 1, which is the most expensive class, is always satisfied from the inventory if possible (see Ha 1997c). For class $i > 1$, backordering the arriving demand and rationing the inventory can be needed to protect the classes $1\ldots i-1$ from being backordered in the future. We can not have both inventory and backorders of class 1, but we can have inventory with class i backorders for $i>1$. The state variable of our system can be described by $\mathbf{x}(t)=(x_1(t),\ldots,x_n(t))$, with $x_1\in Z$ and $x_i\in Z^-$ for $1<i\leq n$. $x_1^+(t)=\max(0,x_1(t))$ is the on-hand inventory at time t. $x_1^-(t)=-\min(0,x_1(t))$ is the number of backorders of class 1 at time t. For $i>1$, $-x_i(t)$ are the number of backorders of class i at time t.

A control policy states the action to take at any time given the current state $\mathbf{x}(t)$, and we restrict the analysis to Markovian policies since the optimal policy belongs to this class (Bertsekas 1995). Let $C^\pi(\mathbf{x})=(C_0^\pi(\mathbf{x}),\ldots,C_n^\pi(\mathbf{x}))$ the control associated with a policy π. C_0^π corresponds to the control action pertaining to the production of items and to their allocation. C_k^π is the control action upon arrival of a class k demand. More formally, we define:

$$C_0^\pi(\mathbf{x})=\begin{cases}0 & \text{not to produce}\\ 1 & \text{to produce and to allocate the produced item:}\\ & \quad\text{to the on-hand inventory when } x_1\geq 0\\ & \quad\text{or to satisfy a backorder of class 1 when } x_1<0\\ k & 1<k\leq n,\text{ to produce and to allocate the item to a backorder of class } k,\end{cases}$$

$C_1^\pi(\mathbf{x})=1$ to satisfy an arriving class 1 demand from the inventory, or to backorder it if the inventory is empty

$$C_k^\pi(\mathbf{x})=\begin{cases}1 & \text{to satisfy an arriving class } k \text{ demand from the inventory}\\ k & \text{to backorder an arriving class } k \text{ demand.}\end{cases},\quad k>1.$$

C_1^π is constant and is introduced for notational consistency. It states to always satisfy the demands of the most expensive class, if possible. C_k^π for $k>1$ corresponds to the rationing of class k.

In state $\mathbf{x}$, the system incurs a cost rate $c(\mathbf{x})$ which is equal to

$$c(\mathbf{x})=hx_1^+ + b_1x_1^- - \sum_{i=2}^n b_ix_i.$$

If we denote by α the discount rate, the objective is then to find a control policy which minimizes the expected discounted inventory costs over an infinite horizon. We define the n-class problem $\mathbf{P}_n(\mu,\boldsymbol{\lambda},h,\mathbf{b},\alpha)$ by:

$$\min_\pi E_{\mathbf{x}_0}^\pi\left[\int_0^\infty e^{-\alpha t}c(\mathbf{x}(t))\,dt\right] \tag{1}$$

where $\mathbf{x}_0=\mathbf{x}(0)$.

In most of the rest of the paper we will concentrate on this discounted cost optimization. However, we will also be interested in the closely related average cost case given by:

$$\min_\pi \lim_{T\to\infty}\frac{E_{\mathbf{x}_0}^\pi\left[\int_0^T c(\mathbf{x}(t))\,dt\right]}{T}. \tag{2}$$

Let S_n be the state space with $S_n=Z\times(Z^-)^{n-1}$ and $\mathbf{x}=(x_1,\ldots,x_n)$ an element of S_n.

Without loss of generality, we can take $\alpha+\sum_{i=1}^n\lambda_i+\mu=1$. The optimal value function v^* for $\mathbf{P}_n(\mu,\boldsymbol{\lambda},h,\mathbf{b},\alpha)$ can be shown to satisfy the following optimality equations:

$$v^*(\mathbf{x})=c(\mathbf{x})+\mu T_0v^*(\mathbf{x})+\sum_{k=1}^n\lambda_kT_kv^*(\mathbf{x}) \tag{3}$$

where the operators T_k are,

$$T_0v(\mathbf{x})=\min_{1<i\leq n}\left[v(\mathbf{x}),\,v(\mathbf{x}+\mathbf{e}_1),\,v(\mathbf{x}+\mathbf{e}_i\ \mathbf{I}_{\{x_i<0\}})\right]$$

where $\mathbf{I}$ is the indicator function

$$T_1v(\mathbf{x})=v(\mathbf{x}-\mathbf{e}_1)$$

$$T_kv(\mathbf{x})=\begin{cases}\min[v(\mathbf{x}-\mathbf{e}_k),\,v(\mathbf{x}-\mathbf{e}_1)] & \text{if } x_1>0\\ v(\mathbf{x}-\mathbf{e}_k) & \text{if } x_1\leq 0\end{cases}$$

for k such that $1<k\leq n$

We also define T the operator such that $Tv(\mathbf{x})=c(\mathbf{x})+\mu T_0v(\mathbf{x})+\sum_{k=1}^n\lambda_kT_kv(\mathbf{x})$. T_0 is the operator associated with the optimal control of production C_0^π, and T_k for $1\leq k\leq n$ associated with the optimal control of rationing C_k^π. Once again, the operators T and T_k

depend on n, but we suppress this dependence for simplicity.

It is convenient to define the operators Δ_{ij} of the function v defined over S_n such that: $\Delta_{ij}v(\mathbf{x}) = v(\mathbf{x}+\mathbf{e}_i) - v(\mathbf{x}+\mathbf{e}_j)$, with $\mathbf{e}_i$ the unit vector of dimension i. We also define the first difference operators Δ_i such that $\Delta_i v(\mathbf{x}) = v(\mathbf{x}+\mathbf{e}_i) - v(\mathbf{x})$. In the sequel, we will consider at the same time problems with different numbers of classes of customers. Note that $\Delta_{ij}v$, $\Delta_i v$, as well as the vector $\mathbf{e}_i$, should depend on the number of classes, but for the sake of simplicity, the same notation will be used for all n. Also, we take $\Delta_{ij}v(\mathbf{x}) = \Delta_i v(\mathbf{x})$ if j is strictly larger than the dimension of $\mathbf{x}$ (for instance, with n classes of customers $\Delta_{i(n+1)}v = \Delta_i v$). Furthermore, for the sake of clarity, we will implicitly assume that, for $i \neq 1$ and $j \neq 1$, $x_i < 0$ and $x_j < 0$ when we consider $\Delta_{ij}v(\mathbf{x})$ or $\Delta_i v(\mathbf{x})$ (otherwise these quantities are not defined). Finally, $\mathbf{x}^i = (x_1^i, \ldots, x_i^i)$ denotes the i-dimensional vector constructed on a n-dimensional vector $\mathbf{x}$, such that for $1 \leq k \leq i$ $x_k^i = x_k$. It follows from the previous definitions that $\Delta_{in}v(\mathbf{x}^{n-1}) = \Delta_i v(\mathbf{x}^{n-1})$.

Finally, in the rest of the paper, we will frequently refer to the most expensive class of customers which has backordered demands. This class is given by the following function m:

$$\forall \mathbf{x} \in S_n,\, m(\mathbf{x}) = \begin{cases} \min_{i:x_i<0}(i) & \text{if } \exists i,\, x_i < 0 \\ n+1 & \text{otherwise} \end{cases}.$$

3. Exploration of the Optimal Policy

3.1. Structural Results

In this subsection, we study the structure of the optimal policy for n classes of customers. Ha (1997c) presents an investigation of the structure for two demand classes. Some of the important preliminary properties therein can be generalized to the multiclass case. Lemma 1 provides this generalization. It establishes four properties of the optimal policy (where the last two are consequences of the first two).

The first two properties are fairly intuitive. Assume that there are backordered demands of a class. The first property states that it is better to satisfy these demands than to do nothing. In other words, the facility has to produce when there are unsatisfied demands. The second property states that if there are backordered demands of two different classes, satisfying the more expensive one saves a larger cost than satisfying the other. This is reminiscent of the $c\mu$ rule for the control of multiclass queues, or of the $b\mu$ rule of the corresponding two product make-to-stock system (Ha 1997a). The first consequence of this result is that, if the policy states to increase the inventory when there are backordered demands of class i, it still states to produce for inventory rather than to satisfy backordered demands of the classes less expensive than i. The second consequence is symmetrical to the first one. It says that if the policy states to satisfy an arriving demand of class j with the on-hand inventory, it also states to satisfy the arriving demands of the classes more expensive than j. These results can be obtained directly through induction on the time horizon by value iteration. Hence, we consider a set of value functions verifying the conditions which correspond to the four properties described above. (Remember that we implicitly take $\mathbf{x}$ with $x_i < 0$ and $x_j < 0$ when we consider $\Delta_{ij}v(\mathbf{x})$ or $\Delta_i v(\mathbf{x})$.)

We define a set $\mathcal{U}_n$ of functions on S_n such that if $v \in \mathcal{U}_n$ then,

(1) $\Delta_i v(\mathbf{x}) < 0$
(2) $\Delta_{ij}v(\mathbf{x}) < 0$ when $1 \leq i < j$
(3) $\Delta_{1j}v(\mathbf{x}) < \Delta_{1i}v(\mathbf{x})$ when $1 < i < j$
(4) $\Delta_{1j}v(\mathbf{x}-\mathbf{e}_j) < \Delta_{1i}v(\mathbf{x}-\mathbf{e}_i)$ when $1 < i < j$

The following lemma states that the dynamic programming operator T preserves $\mathcal{U}_n$. A direct application of value iteration implies that the optimal value function belongs to $\mathcal{U}_n$.

Lemma 1. *If $v \in \mathcal{U}_n$ then $Tv \in \mathcal{U}_n$.*

Proof. See Appendix A.

A useful consequence of Lemma 1, is that for $v \in \mathcal{U}_n$, the operators T_k satisfy,

$$T_0 v(\mathbf{x}) = v(\mathbf{x}+\mathbf{e}_k) + \min(0, \Delta_{1k}v(\mathbf{x})) \quad \text{with } k = m(\mathbf{x}) \tag{4}$$

$$T_k v(\mathbf{x}) = v(\mathbf{x}-\mathbf{e}_1) + \min(0, \Delta_{1k}v(\mathbf{x}-\mathbf{e}_k-\mathbf{e}_1)). \tag{5}$$

with the notation $\Delta_{1(n+1)}v = \Delta_1 v$ when $m(\mathbf{x}) = n+1$. Note then that from (4) and (5) C^π is entirely determined by the sign of $\Delta_{1i}v$, for $1 \leq i \leq n+1$.

It is interesting to note that the parallels with the approach of Ha (1997c) ends with Lemma 1. From here on, we exploit these basic properties using a different approach.

3.2. Multilevel Rationing Policies

Consider a particular class of policies entirely described by $n+1$ parameters, one corresponding to each type of demand, and the last one corresponding to a base stock level in the following way. If we denote by $\mathbf{z}$ the $n+1$ dimensional vector of these parameters, z_k is the rationing level of demand k, that is, all arriving demands of this type are backordered when the on-hand inventory is below (or equal to) z_k. z_{n+1} is the base-stock level of the system for production. Moreover, when a part is produced it is allocated to a backordered demand of class k, only if the on-hand inventory is larger than or equal to z_k. It is allocated to the stock otherwise. Note that if some of these parameters are equal, the resource is allocated to the most expensive customer class (that is to the class $m(\mathbf{x})$ in state $\mathbf{x}$). This class of policies will be referred to as *Multilevel Rationing* (ML) policies. These policies are reminiscent of the multiple threshold policy characterized by Ha (1997c) in the lost sales case.

Definition 1 gives a formal description of ML policies based on Controls C_k^π, which is required by the technical arguments that follow. An alternative and somewhat simpler definition can be found in de Véricourt et al. (2001).

Definition 1. An ML policy π, is a policy characterized by an $(n+1)$ dimensional parameter $\mathbf{z}$ where $z_1 = 0 \le z_2 \le \cdots \le z_{n+1}$, such that

$$C_0^\pi(\mathbf{x}) = \begin{cases} 0 & \text{if} \quad x_1 \ge z_{n+1} \quad \text{and} \quad m(\mathbf{x}) = n+1 \\ k & \text{if} \quad x_1 \ge z_k \quad \text{and} \quad m(\mathbf{x}) = k < n+1 \\ 1 & \text{if} \quad x_1 < z_k \quad \text{and} \quad m(\mathbf{x}) = k \end{cases}$$

$$C_k^\pi(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad x_1 > z_k \quad \text{and} \quad m(\mathbf{x}) \ge k \\ k & \text{if} \quad x_1 \le z_k \end{cases}$$

$\mathbf{z}$ is called the rationing level vector.

To illustrate the behaviour of an ML policy, consider the following example with 3 classes of customers. Let us assume that $z_4 > z_3 = z_2 > 0$. When the stock level x_1 is between z_2+1 and z_4, all arriving demands are satisfied. When $x_1 \le z_2 = z_3$, demands of classes 2 and 3 are backordered such that the on-hand stock is reserved for class 1. As for the production, when $x_1 < z_2 = z_3$, the policy states to replenish the stock until $x_1 = z_2 = z_3$. At this point, backlogged demands of class 2 are satisfied, followed by those of class 3. Finally, production takes place to bring the stock level to the base-stock level z_4.

Remarks. Control C_k^π is not specified when $x_1 > z_k$ and $m(\mathbf{x}) < k$, so that the policy does not necessarily state to satisfy an arriving demand of type k. But states such that the most expensive backordered demands are more expensive than k when $x_1 > z_k$ are transient for ML policies. Furthermore, the policy can state to produce even when $x_1 = z_{n+1}$. For recurrent states, it is only when $x_1 = z_{n+1}$ and $m(\mathbf{x}) = n+1$, that the policy states not to produce. In this case, we say that $\mathbf{x}$ is at the base-stock level of the system.

We claim that the optimal policy is an ML policy. We will argue inductively on the number of customer types. The construction of the proof is based on the following key property: The optimal value function of an n-dimensional problem is closely related to the optimal value function of an $n-1$ dimensional problem, in the region of the state space where $x_1 \le z_n$. In particular, it will be shown that for this region, the corresponding controls do not depend on the nth dimension. The transformation which relates an n dimensional problem $\mathbf{P}_n(\mu, \boldsymbol{\lambda}, h, \mathbf{b}, \alpha)$ to an $n-1$ dimensional problem is based on a decomposition of the cost function such that $c(\mathbf{x}) = \tilde{c}(\mathbf{x}^{n-1}) - b_n \sum_{i=1}^n x_i$. The operator $\tilde{(\cdot)}$ maps the set of linear cost functions defined on S_n on to the set of linear cost functions defined on S_{n-1} and is such that:

$$\tilde{h} = h + b_n \quad \text{and} \quad \tilde{\mathbf{b}} = \mathbf{b}^{n-1} - b_n \mathbf{1}_{n-1}, \tag{6}$$

where $\mathbf{1}_{n-1} = \sum_{i=1}^{n-1} \mathbf{e}_i$.

The cost $\tilde{c}(\mathbf{x}^{n-1})$ corresponds to the cost of an $n-1$ dimensional problem, which does not depend on x_n. More specifically, this subproblem is:

$$\mathbf{P}^{n-1}\left(\frac{\mu}{1-\lambda_n}, \frac{\boldsymbol{\lambda}^{n-1}}{1-\lambda_n}, \frac{\tilde{h}}{1-\lambda_n}, \frac{\tilde{\mathbf{b}}}{1-\lambda_n}, \frac{\alpha}{1-\lambda_n}\right) \tag{7}$$

(where the factor $1-\lambda_n$ is a consequence of the uniformization with discounting, which allows to discard the events due to the arrival process of class n).

To start the induction, assume that the optimal policy of any $n-1$ dimensional problem is an ML policy. In particular, the optimal policy $\boldsymbol{\pi}^*$ of the $n-1$ dimensional subproblem (7) is an ML policy. Denote by $\mathbf{z}^* = (z_1^*, \ldots, z_n^*)$ its rationing level vector. Note also that the optimality equations associated to $\boldsymbol{\pi}^*$ can be expressed as:

$$v^*(\mathbf{x}) = \tilde{c}(\mathbf{x}) + \sum_{k=1}^{n-1} \lambda_k T_k v^*(\mathbf{x}) + \lambda_n v^*(\mathbf{x}) + \mu T_0 v^*(\mathbf{x}) \quad (8)$$

where v^* is the optimal value function, $\tilde{c}$ is the linear cost function of the problem and $\mathbf{x} \in S_{n-1}$.

Based on policy $\boldsymbol{\pi}^*$ and its associated value function v^*, we introduce $\mathcal{V}_n \subset \mathcal{U}_n$, a structured set of value functions. Value iteration will be used to show that the optimal value function of the n-dimensional original problem belongs to this particular set. $\mathcal{V}_n$ is defined by the following conditions, with $[\mathbf{x}+\mathbf{e}_i]_1$ equal to x_1+1 if $i=1$ and to x_1 otherwise:

If $v \in \mathcal{V}_n \subset \mathcal{U}_n$, then

Condition C.1. $\Delta_{ij} v(\mathbf{x}) = \Delta_{ij} v^*(\mathbf{x}^{n-1})$, $1 \le i < j \le n$, when $[\mathbf{x}+\mathbf{e}_i]_1 \le z_n^*$;

Condition C.2. $\Delta_{1i} v(\mathbf{x}) \ge 0$, $i = m(\mathbf{x}) < n+1$, when $x_1 \ge z_n^*$;

Condition C.3. For $\mathbf{x}$ such that $m(\mathbf{x}) \ge n$ and $x_1 \ge z_n^*$

(a) $\Delta_{1n} v(\mathbf{x})$ is increasing in x_1 and decreasing in x_n,

(b) $\Delta_1 v(\mathbf{x})$ is increasing in x_n or equivalently $\Delta_n v(\mathbf{x})$ increasing in x_1,

(c) $\Delta_1 v(\mathbf{x})$ is increasing in x_1 and $\Delta_n v(\mathbf{x})$ increasing in x_n,

Condition C.4. $\Delta_1 v(\mathbf{x}) \le 0$, when $x_1 < z_{n+1}$, where $z_{n+1} = \min_x(\Delta_1 v(x, 0, \ldots, 0) > 0)$ where z_{n+1} of Condition C.4 is well defined from Condition C.3.c.

Note that $\Delta_{ij} v^*(\mathbf{x}^{n-1})$ of Condition C.1 is well defined when $j=n$ from §2 ($\Delta_{in} v^*(\mathbf{x}^{n-1}) = \Delta_i v^*(\mathbf{x}^{n-1})$).

$\mathcal{V}_1$ is only characterized by Conditions C.3.c where $z_1^* = 0$ and C.4. Clearly, the optimal value function is convex in x_1 when $n=1$. The optimal policy is then a base-stock policy. In other words, when the system consists of a single class of customers, the optimal value function belongs to $\mathcal{V}_1$ and the optimal policy is an ML policy. The following statement is the generalization of this result that we are going to show inductively:

Definition 2. We say that $P(n)$ is true, if for all k dimensional problems, $k \le n$,

(1) the optimal policy is an ML policy,

(2) the optimal value function belongs to $\mathcal{V}_k$.

The previous discussion states that $P(1)$ is true. If we assume that $P(n-1)$ is true, then $\mathcal{V}_n$ is well defined and not empty since $v^*(\mathbf{x}^{n-1}) \in \mathcal{V}_n$. Moreover, under this assumption, the next property establishes the first part of $P(n)$.

Property 1. If $P(n-1)$ is true, then policy $\boldsymbol{\pi}$, the associated policy to $v \in \mathcal{V}_n$, is an ML policy with the rationing level vector $\mathbf{z} = [\mathbf{z}^*, z_{n+1}]$ where z_{n+1} is defined by Condition C.4.

Proof. See Appendix A.

In order to establish the second part of $P(n)$, the following lemma states that the operator T preserves $\mathcal{V}_n$.

Lemma 2. *If $P(n-1)$ is true and if $v \in \mathcal{V}_n$ then $Tv \in \mathcal{V}_n$.*

Proof. See Appendix A.

Lemma 2 shows that the ML policies associated with $\mathcal{V}_n$ are preserved under the assumption that $P(k)$ is true for $k < n$. In other words, the n-dimensional policy associated to a value function of $\mathcal{V}_n$, is not only constructed on any $n-1$-dimensional ML policy but more precisely, it is constructed on the *optimal* $n-1$-dimensional ML policy. Theorem 1 states this result more precisely.

Theorem 1. *For all n-dimensional problems, the optimal policy is an ML policy with the rationing level vector $\mathbf{z}^n$. In addition $\mathbf{z}^n$ is such that for $k < n$ its projection $\mathbf{z}^k$ is the optimal rationing level vector of the k dimensional subproblem*:

$$\mathbf{P}^k(\mu, \boldsymbol{\lambda}^k, h + b_{k+1}, \mathbf{b}^k - b_{k+1}\mathbf{1}_k, \alpha). \quad (9)$$

Proof. See Appendix A.

For the two-class problem, the optimal policy has a simple interpretation in terms of the switching curve separating the state space in two regions (see Ha 1997c for more details). This curve is, in fact, a vertical line defined by $x_1 = z_2$. This phenomenon was also identified in other related two-dimensional problems involving multiclass resource sharing (see de Véricourt et al. (2000)).

In order to gain insight into why the optimal control policy depends only on x_1 and $m(\mathbf{x})$, let us focus on the two-class case. The instantaneous cost function can, in this case, be expressed as follows: $c(\mathbf{x}) = (h+b_1)x_1^+ - (b_2-b_1)x_1^- - b_2(x_1+x_2)$.

The key observation then is that (x_1+x_2) does not depend on the allocation policy, while $(h+b_1)x_1^+ - (b_2-b_1)x_1^-$ is the instantaneous cost of a single-class system (which corresponds to subproblem (9)). It should be noted that this type of transformation exploits the linearity of the cost function, suggesting that Theorem 1 does not directly extend to other (nonlinear) cost structures.

4. Optimizing the Parameters of the ML Policy

4.1. Extension to the Average Cost Problem

In the previous sections, we concentrated on the discounted cost problem where the slightly simpler structure of the optimality equation facilitates induction type arguments. Here, we argue that all structural results that were obtained in the discounted cost case can be extended to the average cost case denoted by $\mathbf{P}^n(\mu, \boldsymbol{\lambda}, h, \mathbf{b})$. In the average cost case, the optimal (relative) value function v^* and the optimal cost g per unit time satisfy an optimality equation of the following form:

$$v^*(\mathbf{x}) + g = c(\mathbf{x}) + \mu T_0 v^*(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i T_i v^*(\mathbf{x}) \qquad (10)$$

To argue that the structural properties are retained for the average cost case, we use the conditions of Weber and Stidham (1987), under which the average cost problem can be obtained as the limit of discounted cost problems as the discounting factor vanishes. Theorem 1 can then be directly adapted to the average cost problem, as it is stated in the following corollary:

Corollary 1. *For all n-dimensional problems, the optimal policy of the average cost problem is an ML policy with the rationing level vector* $\mathbf{z}^n$. *In addition* $\mathbf{z}^n$ *is such that for* $k < n$ *its projection* $\mathbf{z}^k$ *is the optimal rationing level vector of the k dimensional subproblem*:

$$\mathbf{P}^k(\mu, \boldsymbol{\lambda}^k, h+b_{k+1}, \mathbf{b}^k - b_{k+1}\mathbf{1}_k). \qquad (11)$$

4.2. An Exact Algorithm to Compute the Optimal Parameters

Theorem 1 gives a precise characterization of optimal allocation policies but does not address a significant issue: how to determine the parameters of the optimal ML policy. For average cost minimization, the nested structure of ML policies can be exploited in a systematic manner in order to construct an efficient algorithm to compute the rationing levels as well as the cost of the optimal policy. As shown in §3, the successive rationing levels of an ML policy are computed using successive subproblems of the same type. Theorem 2 presents the resulting optimization algorithm. Note that even though the algorithm is presented for the optimal policy below, it can also be applied to compute the average cost for any ML policy with a given rationing level vector. (Throughout $\lfloor y \rfloor$ denotes the largest integer that is less than or equal to y).

Theorem 2. *Consider an n-dimensional average cost problem. Construct the sequences* z_k, g_k, *and* ρ_k *as follows*:
Initialize $z_1 = g_1 = \rho_0 = b_{n+1} = 0$.
For $k = 1, \ldots, n$ *do*,

$$\rho_k = \rho_{k-1} + \frac{\lambda_k}{\mu},$$

$$z_{k+1} = z_k + \left\lfloor \frac{\ln \frac{\rho_k(h+b_{k+1})}{\rho_k(h+b_k)+(1-\rho_k)(g_k-(h+b_k)z_k)}}{\ln \rho_k} \right\rfloor$$

$$g_{k+1} = \left(z_{k+1} - \frac{\rho_k}{1-\rho_k}\right)(h+b_{k+1}) + \left(g_k - \left(z_k - \frac{\rho_k}{1-\rho_k}\right)(h+b_k)\right)\rho_k^{z_{k+1}-z_k}.$$

The optimal rationing level vector $\mathbf{z}$ *and the optimal cost* g^*, *are then equal to*: $\mathbf{z} = (z_1, \ldots, z_{n+1})$ *and* $g^* = g_{n+1}$.

Proof. See Appendix A.

Remark that the algorithm provides us closed-form analytical expressions for the optimal rationing levels as well as the optimal cost, for instance, in the case of two customer classes. We exploit these expressions in the next section for certain special cases. For the general case, the algorithm in Theorem 2 has a complexity of $O(n)$. The computation of the optimal rationing levels as well as the optimal cost is thus very efficient.

As a final remark, note that the algorithm presented above can also be directly applied to compute the cost of any given ML policy (i.e. where the z_k are given). In this case, the step of the algorithm where the rationing levels are computed is skipped and the rest of the computations remains unchanged.

5. Inventory Pooling and Stock Allocation: A Numerical Investigation

Both delayed differentiation and centralization of stocks achieve "statistical economies of scale" through redesign. The benefits of both approaches are based on reducing the number of items held in stock. The resulting reduction in demand variability is, in general, beneficial in terms of managing inventories and most of the existing research focuses on quantifying these benefits and comparing them with the cost of the redesign investment.

To illustrate the impact of stock allocation on design decisions, we consider a basic problem related to inventory pooling. Let us consider two distinct supply chains that share a common supplier with limited production capacity. Initially, the supplier stocks two components dedicated to each downstream supply chain. Because of revenue considerations or long-term strategic reasons, the supply chains are not equally important for the supplier. We assume that the supplier could redesign its products and production process such that a single item can satisfy both supply chains. One would be inclined to think that the new structure should lead to a reduction in inventory related costs (which should then be weighed against the investment required) for the supplier. Below, we investigate this design problem in further detail.

To simplify our study, we assume that the orders sent by the downstream supply chains to the supplier are two Poisson processes with demand rates λ_1 and λ_2. In the initial situation, the processing times of both items are identical and are exponentially distributed with rate μ. We assume that the holding costs are identical and equal to h. We also consider that the supplier is linked to the two supply chains by contracts specifying backorder costs b_1 and b_2 ($b_1 \geq b_2$), which reflect their relative importance.

The above model corresponds to the multiclass make-to-stock queue investigated in Veatch and Wein (1996), Ha (1997a), Pena-Perez and Zipkin (1997), and de Véricourt et al. (2000). The optimal scheduling policy is rather complicated and even robust heuristic solutions are subject of on-going research. We, therefore, restrict our attention to a base-stock policy with FCFS scheduling rule for production orders. The expected inventory and backorder cost per unit time of this system, g_0, for optimal base-stock levels is given by (see Buzacott and Shanthikumar 1993):

$$g_0 = \left(z_1 + z_2 - \frac{\rho}{1-\rho}\right)h + \frac{\rho_1}{1-\rho}(h+b_1)\left(\frac{\rho_1}{1-\rho_2}\right)^{z_1} + \frac{\rho_2}{1-\rho}(h+b_2)\left(\frac{\rho_1}{1-\rho_2}\right)^{z_2}$$

where

$$z_1 = \left\lfloor \frac{\ln(\frac{h}{h+b_1})}{\ln \frac{\rho_1}{1-\rho_2}} \right\rfloor, \quad z_2 = \left\lfloor \frac{\ln(\frac{h}{h+b_2})}{\ln \frac{\rho_2}{1-\rho_1}} \right\rfloor,$$

$$\rho_1 = \frac{\lambda_1}{\mu} \quad \text{and} \quad \rho_2 = \frac{\lambda_2}{\mu}.$$

The redesign of the structure leads to a single-item system with two classes of customers (a special case of the model in §2). To simplify the comparison, we assume that the parameters of the system (h, b_i, λ_i, μ) stay unchanged after the redesign process. Figure 1 depicts the system before and after the redesign process.

For this system, we investigate two different cases. The first case corresponds to the situation where the allocation issue is not fully taken into account in the design decision. We model this case by a base-stock policy using a FCFS stock allocation rule for demands. This system generates an expected inventory and backorder cost per unit time of g_1, for an optimal base-stock level (de Véricourt et al. 2001):

$$g_1 = \hat{b}\frac{\rho^{\hat{z}+1}}{1-\rho} + h\left[\hat{z} - \frac{\rho}{1-\rho}(1-\rho^{\hat{z}})\right]$$

where

$$\hat{z} = \left\lfloor \frac{\ln \frac{h}{\hat{b}+h}}{\ln \rho} \right\rfloor$$

and $\hat{b} = \lambda_1/(\lambda_1+\lambda_2)b_1 + \lambda_2/(\lambda_1+\lambda_2)b_2$.

Figure 1 The Initial System and the System After Inventory Pooling

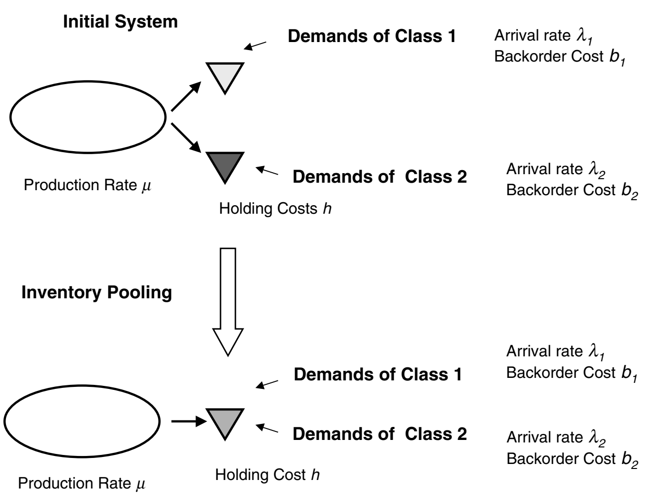
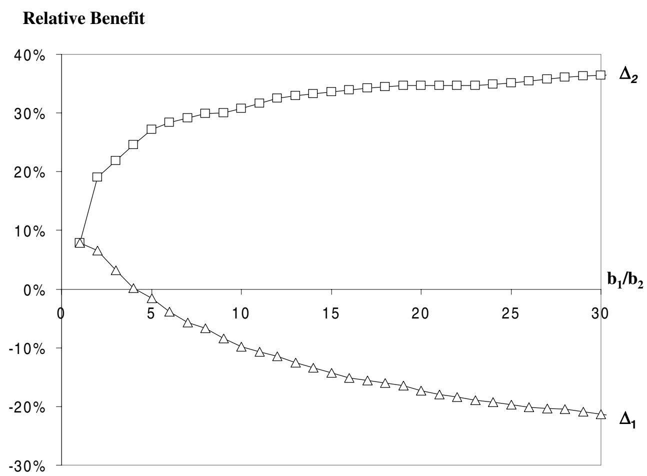


Figure 2 The Relative Benefits of Redesign as a Function of b_1/b_2



In the second case, the allocation issue is integrated in the design decision by using the ML policy. Under this policy, the average cost, g_2, is given by the algorithm of §4.2.

This setting allows us to carry out two comparisons. The relative difference $\Delta_1 = (g_0 - g_1)/g_0$ is the benefit due to inventory pooling without allocation considerations and the relative difference $\Delta_2 = (g_0 - g_2)/g_0$ is the benefit of inventory pooling when the optimal stock allocation policy is employed.

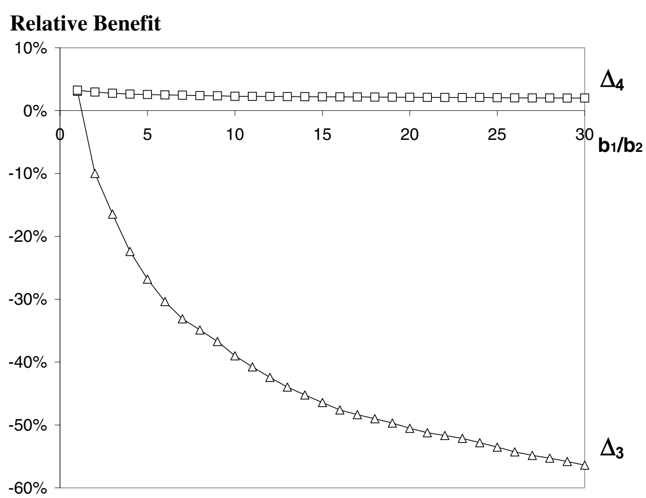
We focus on the value of system redesign as a function of the relative importance of the two products expressed by the ratio of the backorder costs b_1/b_2. Figure 2 depicts an example of this comparison for parameter values $h = 0.5$, $b_2 = 1$, $\lambda_1 = \lambda_2 = 0.4$, $\mu = 1$. The first striking result is that Δ_1 can be negative. In other words, if one does not pay attention to how stocks are allocated, redesigning the system can increase the costs. This result can appear counterintuitive at first sight and seems to contradict most of the previous research results on the value inventory pooling. Indeed, most of these previous studies assume symmetrical demands (in terms of backorder costs) and it is clear then that inventory-related costs decrease by pooling. On the other hand, when the asymmetry between the respective backorder costs increase and if stock allocation is not carefully taken into consideration, the benefits of redesign decrease—and can be negative—as evidenced by the value of Δ_1 in Figure 2. The second observation that can be made from the figure is that, if the ML policy is employed, Δ_2 is always positive regardless of the values of the parameters.

In managerial terms, efficient stock allocation among the customers guarantees a reduction in terms of inventory-related costs. For instance, if in the initial system b_1 is 10 times larger than b_2, redesigning the system without a careful stock allocation consideration increases the costs by 10% while an efficient allocation strategy would decrease costs by 35%. In that sense, redesigning the system brings potential benefits but these benefits can only be realized if stocks are allocated in an efficient manner. In fact, careless stock allocation may result in increased costs despite the pooling effect. For the initial system, even when a FCFS policy is employed, some inventory is reserved for the important demands due to the physically separated stocks by specifying appropriate base-stock levels. Inventory pooling destroys this reservation structure. The control policy needs then to address the issue how the inventory should be reserved to different classes. A FCFS policy does not address these requirements, which explains why it can perform worse after pooling.

An important point about the ML policy is that it is robust in the sense that it guarantees an inventory cost reduction after inventory pooling. For the system before pooling, the FCFS policy, which was used as the first benchmark, is not optimal. Note, however, that for any scheduling policy, including the optimal, the ML policy would enable a cost reduction after pooling. In fact, this point can be formalized by a straightforward sample-path argument but an intuitive explanation should suffice. Intuitively, the pooled system delays allocation decisions and hence uses more accurate information about demands with respect to the multiproduct system. Consequently, the average cost under its optimal policy is inferior to the average cost of the multiproduct system under the respective optimal policy. Figure 3 presents the same comparison as in Figure 2 where the initial system is controlled using the optimal scheduling policy. Under the optimal policy, the optimal cost g_3 was computed numerically (see for instance de Véricourt et al. 2000). This time, the relative differences $\Delta_3 = (g_3 - g_1)/g_3$ and $\Delta_4 = (g_3 - g_2)/g_3$ represent the benefits due to inventory pooling without and with allocation considerations, respectively. The benefits of inventory pooling when the initial system is controlled by the optimal policy are not as significant. It should be noted however that, the optimal scheduling policy is extremely difficult to implement. On the other hand, if the common stock is not properly allocated, the losses can easily reach 50%.

A final remark is noteworthy, as the above comparison indicates another, more subtle, benefit of inventory pooling. A capacitated multiproduct system is

Figure 3 **The Relative Benefits of Redesign as a Function of b_1/b_2 when the Optimal Scheduling Policy is Used Before Pooling**



difficult to manage "optimally." In fact, for the multiproduct case, there seems to be little hope that the exact optimal policy can be parametrized in a simple way. The optimal policy for the single-product system, on the other hand, can be expressed in a few parameters. This is clearly an additional advantage of product standardization.

6. Conclusions and Future Research

We have provided a characterization of stock allocation policies for a multiclient make-to-stock system. This characterization is one of the few known complete characterizations in dynamic allocation problems of multiclass make-to-stock queues. In addition, in this case, the optimal stock allocation is intuitive, easy to communicate and to implement. The approach that was employed for the characterization of the optimal policy exploits a nested structure by relating an n-class problem to a corresponding $(n-1)$-class problem. A similar approach had enabled a partial characterization of the optimal policy in a related multiproduct system (see de Véricourt et al. 2000). An interesting question for future research is investigating the general class of problems where such an approach can be used to characterize optimal policies.

A second significant contribution of the paper is an efficient algorithm to identify the optimal parameters of a given ML policy. As a consequence, optimal stock allocation problems with a large number of classes can be solved instantenously. This makes our methods very promising as benchmark performance measures.

One related important issue in regards to the contribution of the paper is the value of "optimal stock allocation" with respect to other plausible—but suboptimal—allocation policies. We do not directly address this point here but it is known that optimal stock allocation results in significant benefits as demonstrated under a variety of settings by Nahmias and Demmy (1981), Ha (1997b), and de Véricourt et al. (2001). While these previous papers shed light onto the benefits of optimal stock allocation, none of them investigate these benefits in the framework of higher-level supply chain design decisions that create allocation problems. We add new insights on the value of stock allocation by considering its impact in a supply chain design problem concerning pooling of inventories. Our results show that ignoring the stock allocation dimension in a typical example of such a design problem can lead to severely inaccurate evaluations and to incorrect design decisions.

As a final remark, in order to obtain a tractable formulation, we modeled processing times by an exponential distribution. While the relaxation of this assumption leads to less tractable models, there are reasons to expect that ML-type policies should be optimal or at least near optimal. In particular, Ha (2000) has recently shown the optimality of ML-type policies with Erlang production times for a corresponding lost sales model. This suggests that ML policies are highly promising under more general assumptions. Our optimization algorithm should then provide a natural starting point of investigation for parameter optimization.

Acknowledgments
The authors thank two anonymous reviewers and the associate editor for helpful comments and suggestions.

Appendix

A. Proofs

A.1. Proof of Lemma 1. (a) Let us show that Tv verifies the first condition of the definition. We take $\mathbf{x}$ such that $x_i < 0$. We will prove that the operators T_k conserve the first condition of $\mathcal{U}_n$. For $k > 0$, suppose that $T_k(\mathbf{x}) = v(\mathbf{x}-\mathbf{e}_p)$ with $p = 1$ or $p = k$. From the definition of T_k and the fact that $v \in \mathcal{U}_n$ we have:

$$\Delta_i T_k v(\mathbf{x}) \le v(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_p) - v(\mathbf{x}-\mathbf{e}_p) = \Delta_i v(\mathbf{x}-\mathbf{e}_p) \le 0.$$

For $k = 0$, note that from the first part of the definition of $\mathcal{U}_n$, $v(\mathbf{x}+\mathbf{e}_i) \le v(\mathbf{x})$ and we can assume that $T_0 v(\mathbf{x}) = v(\mathbf{x}+\mathbf{e}_p)$.

When $p = i$, $\Delta_i T_0 v(\mathbf{x}) \le v(\mathbf{x}+\mathbf{e}_i) - v(\mathbf{x}+\mathbf{e}_p) = 0$.

When $p \ne i$, $\Delta_i T_0 v(\mathbf{x}) \le v(\mathbf{x}+\mathbf{e}_i+\mathbf{e}_p) - v(\mathbf{x}+\mathbf{e}_p) \le 0$.

Thus $\Delta_i T_k$ for $k \ge 0$ is negative or null. Furthermore $\Delta_i c(\mathbf{x}) = -b_i < 0$, and $\Delta_i T v(\mathbf{x}) < 0$.

(b) Let us show that Tv verifies the second condition. We take $\mathbf{x}$ such that $x_i < 0$ and $x_j < 0$ with $1 \le i < j$. For $k > 1$, if $T_k v(\mathbf{x}+\mathbf{e}_j) = v(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_p)$ with $p = 1$ or $p = k$, then $\Delta_{ij} T_k v(\mathbf{x}) \le \Delta_{ij} v(\mathbf{x}-\mathbf{e}_p) \le 0$.

For $k = 0$, assume that $T_0 v(\mathbf{x}+\mathbf{e}_j) = v(\mathbf{x}+\mathbf{e}_j+\mathbf{e}_p)$ (note that from the first condition of $\mathcal{U}_n$, $T_0 v(\mathbf{x}+\mathbf{e}_j) \le v(\mathbf{x}+\mathbf{e}_j)$).

When $p = i$, $\Delta_{ij} T_0 v(\mathbf{x}) \le v(\mathbf{x}+\mathbf{e}_i+\mathbf{e}_j) - v(\mathbf{x}+\mathbf{e}_i+\mathbf{e}_p) = 0$.

When $p \ne i$, $\Delta_{ij} T_0 v(\mathbf{x}) \le \Delta_{ij} v(\mathbf{x}+\mathbf{e}_p) \le 0$.

Finally $\Delta_{ij} c(\mathbf{x}) = b_j - b_i < 0$, thus $\Delta_{ij} T v(\mathbf{x}) < 0$.

(c) Conditions 3 and 4 of $\mathcal{U}_n$ are direct consequences of Condition 2, applied respectively in $\mathbf{x}$ and in $\mathbf{x}+\mathbf{e}_1-\mathbf{e}_i-\mathbf{e}_j$. □

A.2. Proof of Property 1. Take $v \in \mathcal{V}_n$ and consider its associated policy π. We first need to check that $z_{n+1} \ge z_n^*$. From the definition of z_{n+1}, it is sufficient to show that for states $\mathbf{x}$ where $m(\mathbf{x}) = n+1$ and $x_1 < z_n^*$, $\Delta_1 v(\mathbf{x})$ is strictly negative:

$$\begin{aligned}\Delta_1 v(\mathbf{x}) &= \Delta_n v(\mathbf{x}+\mathbf{e}_1-\mathbf{e}_n) + \Delta_{1n} v(\mathbf{x}-\mathbf{e}_n)\\ &= \Delta_n v(\mathbf{x}+\mathbf{e}_1-\mathbf{e}_n) + \Delta_1 v^*(\mathbf{x}^{n-1}).\end{aligned} \tag{12}$$

(12) comes from Condition C.1. The first term of this expression is strictly negative from the first condition of $\mathcal{U}_n$, while the second one is negative from Condition C.4 of $\mathcal{V}_{n-1}$.

We derive now the different controls of Policy π. From Conditions C.3.c and C.4, if $m(\mathbf{x}) = n+1$ the production control states to produce if and only if $x_1 < z_{n+1}$ which is also the case for an ML policy. Otherwise, these controls are given by the sign of $\Delta_{1i} v(\mathbf{x})$ where $i \le n$ (see (4) and (5)). More precisely, the production control of π is given by the difference $\Delta_{1i} v(\mathbf{x})$ where $i = m(\mathbf{x}) < n+1$. This quantity is positive when $x_1 \ge z_n^*$ from Condition C.2 meaning that $C_0^\pi(\mathbf{x}) = m(\mathbf{x})$. When $x_1 < z_n^*$, Condition C.1 implies that the sign of $\Delta_{1i} v(\mathbf{x})$ is given by that of $\Delta_{1i} v^*(\mathbf{x}^{n-1})$, which in turns also determines $C_0^{\pi^*}(\mathbf{x})$. As a result, $C_0^\pi(\mathbf{x})$ and $C_0^{\pi^*}(\mathbf{x})$ are equal when $x_1 < z_n^*$, and

$$C_0^\pi(\mathbf{x}) = \begin{cases} 0 & \text{if } x_1 \ge z_{n+1} \quad \text{and } m(\mathbf{x}) = n+1\\ 1 & \text{if } z_n^* \le x_1 < z_{n+1} \quad \text{and } m(\mathbf{x}) = n+1\\ m(\mathbf{x}) & \text{if } x_1 \ge z_n^* \quad \text{and } m(\mathbf{x}) \le n\\ C_0^{\pi^*}(\mathbf{x}^{n-1}) & \text{if } x_1 < z_n^* \quad \text{and } m(\mathbf{x}) \le n. \end{cases} \tag{13}$$

The kth rationing control is given by the sign of $\Delta_{1k} v(\mathbf{x}-\mathbf{e}_1-\mathbf{e}_k)$. This quantity is positive when $x_1 > z_n^*$ and $m(\mathbf{x}) \ge k$ from Condition C.2, so that $C_k^\pi(\mathbf{x}) = 1$. When $x_1 \le z_n^*$, the same approach we followed for the production control leads to $C_k^\pi(\mathbf{x}) = C_k^{\pi^*}(\mathbf{x})$ with $k < n$. For $k = n$, the rationing control is given by the sign of $\Delta_1 v^*(\mathbf{x}^{n-1})$, which is negative from Condition C.4 of $\mathcal{V}_{n-1}$, leading to $C_n^\pi(\mathbf{x}) = n$. So we can write

$$k < n,\quad C_k^\pi(\mathbf{x}) = \begin{cases} 1 & \text{if } x_1 > z_n^* \quad \text{and} \quad m(\mathbf{x}) \ge k\\ C_k^{\pi^*}(\mathbf{x}^{n-1}) & \text{if } x_1 \le z_n^* \end{cases} \tag{14}$$

$$k = n,\quad C_n^\pi(\mathbf{x}) = \begin{cases} 1 & \text{if } x_1 > z_n^* \quad \text{and} \quad m(\mathbf{x}) \ge n\\ n & \text{if } x_1 \le z_n^* \end{cases}. \tag{15}$$

But π^* is an ML policy from $P(n-1)$, and $C_k^{\pi^*}$ can be replaced in (13) and (15) by ML policy controls with the rationing level vector $\mathbf{z}^*$. The controls C_k^π can hence in turns be rewritten as ML policy controls with the rationing vector $[\mathbf{z}^*, z_{n+1}]$. □

A.3. Proof of Lemma 2. Consider $v \in \mathcal{V}_n$ and its associated policy π. Form Property 1, π is an ML policy with the rationing vector $\mathbf{z} = [\mathbf{z}^*, z_{n+1}]$, and whose associated controls are described in Definition 1.

(a) Let us prove that Tv verifies Condition C.1. We consider in this part states $\mathbf{x}$ such that $[\mathbf{x}+\mathbf{e}_i]_1 \le z_n^*$. Since the first $n-1$ rationing levels of policies π are equal to those of policy π^*, their first $n-1$ rationing controls are equal as well. Taking $C_k^\pi(\mathbf{x}+\mathbf{e}_i)=p$ and $C_k^\pi(\mathbf{x}+\mathbf{e}_j)=q$ where p and q are equal to 1 or k for $k<n$, we have

$$\begin{aligned}\Delta_{ij}T_k v(\mathbf{x}) &= v(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_p)-v(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_q)\\ &= \Delta_{ij}v(\mathbf{x}-\mathbf{e}_p)+\Delta_{qp}v(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_p-\mathbf{e}_q)\\ &= \Delta_{ij}v^*([\mathbf{x}-\mathbf{e}_p]^{n-1})+\Delta_{qp}v^*([\mathbf{x}+\mathbf{e}_j-\mathbf{e}_p-\mathbf{e}_q]^{n-1}) \qquad (16)\\ &= v^*([\mathbf{x}+\mathbf{e}_i-\mathbf{e}_p]^{n-1})-v^*([\mathbf{x}+\mathbf{e}_j-\mathbf{e}_q]^{n-1})\\ &= \Delta_{ij}T_k v^*(\mathbf{x}^{n-1}), \qquad (17)\end{aligned}$$

where (16) comes from Condition C.1.

For the nth rationing operator T_n, Policy π states to backorder arriving demands of class n and applying C.1 once again, we can derive

$$\Delta_{ij}T_n v(\mathbf{x}-\mathbf{e}_n)=\Delta_{ij}v(\mathbf{x}-\mathbf{e}_n)=\Delta_{ij}v^*(\mathbf{x}^{n-1}). \qquad (18)$$

As for the production operator T_0, if $\mathbf{x}^{n-1}+\mathbf{e}_1$ is not at the base-stock level of π^*, that is to say if $m(\mathbf{x})<n$ or $x_1+1<z_n^*$, C_0^π is strictly positive and strictly less than n. A similar approach to the one that was used previously for C_k^π with $1\le k<n$ leads to

$$\Delta_{ij}T_0 v(\mathbf{x}) = \Delta_{ij}T_0 v^*(\mathbf{x}^{n-1}). \qquad (19)$$

If $\mathbf{x}^{n-1}+\mathbf{e}_1$ is at the base-stock level of policy π^* then it states not to produce while policy π indicates to produce to satisfy waiting demand of class n, and

$$\Delta_{1n}T_0 v(\mathbf{x})=0=\Delta_1 T_0 v^*(\mathbf{x}^{n-1})=\Delta_{1n}T_0 v^*(\mathbf{x}^{n-1}) \qquad (20)$$

(with $(\mathbf{x}+\mathbf{e}_n)^{n-1}=\mathbf{x}^{n-1}$).

We are now able to compute the differences in T, noting that $\Delta_{ij}c(\mathbf{x})=\Delta_{ij}\tilde{c}(\mathbf{x}^{n-1})$:

$$\begin{aligned}\Delta_{ij}Tv(\mathbf{x}) &= \Delta_{ij}\tilde{c}(\mathbf{x}^{n-1})+\sum_{k=1}^{n-1}\lambda_k\Delta_{ij}T_k v^*(\mathbf{x}^{n-1})\\ &\quad +\lambda_n\Delta_{ij}v^*(\mathbf{x}^{n-1})+\mu\Delta_{ij}T_0 v^*(\mathbf{x}^{n-1}) \qquad (21)\\ &= \Delta_{ij}v^*(\mathbf{x}^{n-1}). \qquad (22)\end{aligned}$$

(21) follows from (17), (18), (19), and (20). (22) is true since v^* verifies the optimality equations (8).

(b) Let us prove that Tv verifies Condition C.2. We consider in this part states $\mathbf{x}$ such that $x_1\ge z_n^*$. We take also $m(\mathbf{x})=i<n+1$. For this region of the state space, $x_1\ge z_n^*\ge z_k^*$ and the production control of policy π specifies to produce for waiting demands of class $m(\mathbf{x})$ when there are backorders in the system. When there are no backorders, the production control is given by the base-stock level z_{n+1} or equivalently by the sign of $\Delta_1 v$:

$$\Delta_{1i}T_0 v(\mathbf{x}) = \begin{cases}\Delta_{1i}v(\mathbf{x}+\mathbf{e}_i) & \text{if } m(\mathbf{x}+\mathbf{e}_i)=i\\ \Delta_{1k}v(\mathbf{x}+\mathbf{e}_i) & \text{if } m(\mathbf{x}+\mathbf{e}_i)=k<n+1\\ 0 & \text{if } m(\mathbf{x}+\mathbf{e}_i)=n+1, \quad \Delta_1 v(\mathbf{x}+\mathbf{e}_i)<0\\ \Delta_1 v(\mathbf{x}+\mathbf{e}_i) & \text{if } m(\mathbf{x}+\mathbf{e}_i)=n+1, \quad \Delta_1 v(\mathbf{x}+\mathbf{e}_i)\ge 0\end{cases} \qquad (23)$$

Using Condition C.2, one can easily check that (23) is positive so that $T_0 v$ also verifies Condition C.2.

For the rationing operators, take $C_k^\pi(\mathbf{x}+\mathbf{e}_1)=p$ and $C_k^\pi(\mathbf{x}+\mathbf{e}_i)=q$ with p and q equal to 1 or k.

For $k\ge i$, the rationing control C_k^π are then not characterized by the rationing level vector and we need to study the sign of $\Delta_{1i}T_k v$ for all possible values of p and q:

If $p=q=k$,

$\Delta_{1i}T_k v(\mathbf{x})$ is equal to $\Delta_{1i}v(\mathbf{x}-\mathbf{e}_k)$ which is positive from Condition C.2.

If $p=q=1$,

$\Delta_{1i}T_k v(\mathbf{x})$ is equal to $\Delta_{1i}v(\mathbf{x}-\mathbf{e}_1)$. But since $q=1$, $\Delta_{1k}v(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_1-\mathbf{e}_k)$ is positive from (5), which also implies that $\Delta_{1i}v(\mathbf{x}-\mathbf{e}_1)$ is positive from the last condition of $\mathcal{U}_n$.

If $p=1$ and $q=k$,

$\Delta_{1i}T_k v(\mathbf{x})$ is equal to $\Delta_{ki}v(\mathbf{x}-\mathbf{e}_k)$, which is positive from the second condition of $\mathcal{U}_n$.

If $p=k$ and $q=1$,

it can be easily shown that $\Delta_{1i}T_k v(\mathbf{x})=\Delta_{1i}v(\mathbf{x}-\mathbf{e}_k)+\Delta_{1k}v(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_1-\mathbf{e}_k)$, whose first term is positive from Condition C.2. Furthermore $q=1$ implies from (5) that the second term is positive as well.

Consider now that $k<i$. p is then equal to one since $m(\mathbf{x}+\mathbf{e}_1)>k$ and $x_1+1>z_n^*\ge z_k^*$. q is also equal to one if $x_1\ge z_n^*$ when $z_n^*>z_i^*$, or if $x_1>z_n^*$ when $z_n^*=z_i^*$ for similar reasons. In these cases, $\Delta_{1i}T_k v(\mathbf{x})$ is equal to $\Delta_{1i}v(\mathbf{x}-\mathbf{e}_1)$, which is positive for $x_1>z_n^*$ from Condition C.2, and for $x_1=z_n^*$ from Condition C.1 of $\mathcal{V}_n$ and Condition C.2 of $\mathcal{V}_{n-1}$. Hence, $T_k v$ verifies Condition C.2 and since $\Delta_{1i}c(\mathbf{x})$ and $\Delta_{1i}T_0 v(\mathbf{x})$ are also positive, Tv verifies the condition as well. On the other hand, q can be equal to k if $x_1=z_n^*=z_i^*$, which implies that $\Delta_{1i}T_k v$ can then be negative.

So it remains to show that $Tv(\mathbf{x})$ is positive if $x_1=z_n^*=z_i^*$ and $k<i$. p is still equal to one and

$$\Delta_{1i}T_k v(\mathbf{x})=\Delta_{qk}v(\mathbf{x}-\mathbf{e}_q). \qquad (24)$$

Also, since $P(n-1)$ is true with $z_l^*=z_n^*$ for $i\le l<n$, we can reiterate $n-i$ times the application of Condition C.1 to the optimal value function of the successive rationing subproblems. (24) becomes then

$$\Delta_{1i}T_k v(\mathbf{x})=\Delta_q v_{i-1}^*(\mathbf{x}^{i-1}-\mathbf{e}_q) \qquad (25)$$

where v_{i-1}^* is the optimal value of the $i-1$ dimensional rationing problem recursively defined from (7). The instantaneous cost $\tilde{c}_{i-1}$ of this problem is defined by the holding cost $\tilde{h}_{i-1}=h+b_i$ and the backorder cost vector $\tilde{\mathbf{b}}_{i-1}=\mathbf{b}^{i-1}-b_i\mathbf{1}_{i-1}$. Note also that $\Delta_1\tilde{c}_{i-1}$ is equal to $\Delta_{1i}c$.

Moreover, the first $i-1$ rationing controls of the optimal policy associated to v_{i-1}^* are the same as those of policy π. (25) becomes then

$$\Delta_{1i}T_k v(\mathbf{x})=\Delta_1 T_k v_{i-1}^*(\mathbf{x}^{i-1}-\mathbf{e}_q). \qquad (26)$$

For $k=0$ or $k\geq i$, we have already proven that $\Delta_{1i}T_k v(\mathbf{x})$ is positive, and using the definition of operator T along with (26) we can derive the following inequality

$$\Delta_{1i}Tv(\mathbf{x}) \geq \Delta_1\tilde{c}_{i-1}(\mathbf{x}^{i-1})+\sum_{k=1}^{i-1}\lambda_k\Delta_1 T_k v^*_{i-1}(\mathbf{x}^{i-1}). \tag{27}$$

But $\mathbf{x}^{i-1}$ is then at the base-stock level of the ML policy associated to v^*_{i-1}. As a consequence, $\Delta_1 T_0 v^*_{i-1}(\mathbf{x}^{i-1})$ is equal to $\Delta_1 v^*_{i-1}(\mathbf{x}^{i-1})$ which is positive from the definition of z^*_n given by Condition C.4 of $\mathcal{V}_{n-1}$. The optimality equations satisfied by v^*_{i-1} can hence be written as:

$$\Delta_1 v^*_{i-1}(\mathbf{x}^{i-1}) = \Delta_1\tilde{c}_{i-1}(\mathbf{x}^{i-1})+\sum_{k=1}^{i-1}\lambda_k\Delta_1 T_k v^*_{i-1}(\mathbf{x}^{i-1}) + \sum_{k=i}^{n}\lambda_k\Delta_1 v^*_{i-1}(\mathbf{x}^{i-1})+\mu\Delta_1 v^*_{i-1}(\mathbf{x}^{i-1}). \tag{28}$$

Using (28) in (27) we obtain then

$$\Delta_{1i}Tv(\mathbf{x}) \geq \left(1-\sum_{k=i}^{n}\lambda_k-\mu\right)\Delta_1 v^*_{i-1}(\mathbf{x}^{i-1}), \tag{29}$$

but as we discussed it previously, $\Delta_1 v^*_{i-1}(\mathbf{x}^{i-1})$ is positive and Tv verifies Condition C.2.

(c) Let us show that Tv verifies Condition C.3. We consider in this part states $\mathbf{x}$ such that $x_1\geq z^*_n$ and $m(\mathbf{x})\geq n$. In this region of the state space π is fully characterized by z^* and z_{n+1}. We can then directly compute $\Delta_{1n}T_k v(\mathbf{x})$ and $\Delta_1 T_k v(\mathbf{x})$, using Condition C.1 when it applies:

$$\Delta_{1n}T_0 v(\mathbf{x}) = \begin{cases}\Delta_1 v(\mathbf{x}+\mathbf{e}_n) & \text{if } m(\mathbf{x}+\mathbf{e}_n)=n+1, \quad x_1\geq z_{n+1}\\ 0 & \text{if } m(\mathbf{x}+\mathbf{e}_i)=n+1, \quad x_1< z_{n+1}\\ \Delta_{1n}v(\mathbf{x}+\mathbf{e}_n) & \text{if } m(\mathbf{x}+\mathbf{e}_i)=k<n+1\end{cases} \tag{30}$$

$$\Delta_{1n}T_k v(\mathbf{x}) = \begin{cases}\Delta_{1n}v(\mathbf{x}-\mathbf{e}_1) & \text{if } x_1>z^*_n\\ \Delta_1 v^*(\mathbf{x}^{n-1}-\mathbf{e}_1) & \text{if } x_1=z^*_n, \quad z^*_k<z^*_n\\ \Delta_k v([\mathbf{x}-\mathbf{e}_k]^{n-1}) & \text{if } x_1=z^*_n, \quad z^*_k=z^*_n\end{cases} \tag{31}$$

$$\Delta_1 T_0 v(\mathbf{x}) = \begin{cases}\Delta_1 v(\mathbf{x}) & \text{if } m(\mathbf{x}+\mathbf{e}_n)=n+1, \quad x_1\geq z_{n+1}\\ 0 & \text{if } m(\mathbf{x})=n+1, \quad x_1+1=z_{n+1}\\ \Delta_1 v(\mathbf{x}+\mathbf{e}_1) & \text{if } m(\mathbf{x})=n+1, \quad x_1+1<z_{n+1}\\ \Delta_1 v(\mathbf{x}+\mathbf{e}_n) & \text{if } m(\mathbf{x})=n\end{cases} \tag{32}$$

$$\Delta_1 T_k v(\mathbf{x}) = \begin{cases}\Delta_1 v(\mathbf{x}-\mathbf{e}_1) & \text{if } x_1>z^*_n\\ \Delta_1 v(\mathbf{x}-\mathbf{e}_1) & \text{if } x_1=z^*_n, \quad z^*_k<z^*_n\\ \Delta_k v([\mathbf{x}-\mathbf{e}_k]^{n-1}) & \text{if } x_1=z^*_n, \quad z^*_k=z^*_n\end{cases} \tag{33}$$

Let us study $\Delta_{1n}T_0 v$ using (30). $\Delta_{1n}T_0 v$ increases in x_1 from Conditions C.3, and the definition of z_{n+1}. When $m(\mathbf{x}+\mathbf{e}_n)<n+1$, $\Delta_{1n}T_k v$ decreases in x_n from Condition C.3.a, and is positive from Condition C.2, while it is null when $m(\mathbf{x}+\mathbf{e}_n)=n+1$ and $x_1<z_n+1$. Finally when $m(\mathbf{x}+\mathbf{e}_n)=n+1$ and $x_1\geq z_{n+1}$, we have

$$\Delta_{1n}T_k v(\mathbf{x}) = \Delta_1 v(\mathbf{x}+\mathbf{e}_n) \tag{34}$$

$$= \Delta_n v(\mathbf{x}+\mathbf{e}_1)+\Delta_{1n}v(\mathbf{x}) \tag{35}$$

$$= \Delta_n v(\mathbf{x}+\mathbf{e}_1)+\Delta_{1n}T_k v(\mathbf{x}-\mathbf{e}_n), \tag{36}$$

and, since $\Delta_n v(\mathbf{x}+\mathbf{e}_1-\mathbf{e}_n)$ is negative from the first condition of $\mathcal{U}_n$, $\Delta_{1n}T_k v$ decreases in x_n. It follows that $T_k v$ verifies condition C.3.a.

Let us study $\Delta_{1n}T_k v$ using (30). When $x_1=z^*_n$, $\Delta_{1n}T_k v(\mathbf{x})$ is negative from Condition C.4 of $\mathcal{V}_{n-1}$ and the first condition of $\mathcal{U}_n$, and does not depend on x_n. When $x_1>z^*_n$, $\Delta_{1n}T_k v(\mathbf{x})$ is positive from Condition C.2, increases in x_1 and decreases in x_n from Condition C.3.a. As a result $T_k v$ verifies Condition C.3.a as well.

Following a similar approach one can easily show that $\Delta_1 T_k v(\mathbf{x})$ increases for $0\leq k\leq n$. (For $k>0$ and $x_1=z^*_n=z^*_k$, $\Delta_1 T_k v(\mathbf{x})$ can be written as the sum $\Delta_n v(\mathbf{x}-\mathbf{e}_n)+\Delta_1 v^*(\mathbf{x}^n-\mathbf{e}_1)$, where the first term decreases in x_n from Condition C.3.c, and where the second term does not depend on n.) As for the last condition, it is well-known that Conditions C.3.a and C.3.b together imply Condition C.3.c.

(d) Finally, let us show that Tv verifies Condition C.4. Here again we can directly compute $\Delta_1 T_0 v$ for $x_1<z_{n+1}$:

$$\Delta_1 T_0 v(\mathbf{x}) = \begin{cases}0 & \text{if } m(\mathbf{x})=n+1, \quad x_1+1=z_{n+1}\\ \Delta_k v(\mathbf{x}+\mathbf{e}_1) & \text{if } m(\mathbf{x})=k<n+1, \quad x_1+1=z_k\\ \Delta_1 v(\mathbf{x}+\mathbf{e}_1) & \text{if } m(\mathbf{x})=k\leq n+1, \quad x_1+1<z_k,\end{cases} \tag{37}$$

which can easily be checked to be negative from Condition C.4 and the first condition of $\mathcal{U}_n$.

As for $\Delta_1 T_k v$, taking $C^\pi_k(\mathbf{x}+\mathbf{e}_1)=p$ and $C^\pi_k(\mathbf{x})=q$ with p and q equal to 1 or k, we have:

$$\Delta_1 T_k v(\mathbf{x}) = \Delta_{1p}v(\mathbf{x}-\mathbf{e}_p)+\Delta_q v(\mathbf{x}-\mathbf{e}_q). \tag{38}$$

The first term is null if $p=1$, and (5) implies that it is negative or null otherwise. The second term is negative from Condition C.4 if $q=1$ and from the first condition of $\mathcal{U}_n$ if $q=k$. □

A.4. Proof of Theorem 1. As we mentioned it earlier, $P(1)$ is true. Suppose that $P(n-1)$ is true. Property 1 states the first part of $P(n)$. From Lemma 2, using value iteration and the fact that the optimal infinite-horizon policy can be obtained as the limit of finite-horizon optimal policies, the second part of $P(n)$ is also true. As a result, for all n-dimensional problems, the optimal policy is an ML policy.

To prove the second part of the theorem, we will recursively apply Lemma 1. However the dynamics of the subproblem (7) of $(n-1)$ dimensions is equivalent to that of the original n-dimensional problem where an arrival of type n does not change the current state. An alternative interpretation is that the time scale is changed by the factor $1-\lambda_n$. More formally, the value function $(1-\lambda_n)v^*$ is the solution of the subproblem:

$$\mathbf{P}^{n-1}(\mu, \boldsymbol{\lambda}^{n-1}, \tilde{h}, \tilde{\mathbf{b}}, \alpha). \tag{39}$$

The optimal control depends only on the sign of Δv^* (see (4) and (5)), which is equal to the sign of $(1-\lambda_n)\Delta v^*$. Therefore, if $\mathbf{z}^*$ is the optimal rationing level vector of Subproblem (39), it is optimal for Subproblem (7) as well.

To compute $\mathbf{z}^k$, we must then consider $n-k$ subproblems of the type (39). A cost function $c(\cdot)$ defined on S_n is characterized by a vector $(h, \mathbf{b})$. We then denote by $\tilde{c}_k(\cdot)$, the cost function defined on

S_k, characterized by the vector $((h+b_{k+1}), \mathbf{b}^k - b_{k+1}\mathbf{1}_k)$ with $k < n$. It follows that for $p < q$, $\tilde{c}_p(\tilde{c}_q(\cdot)) = \tilde{c}_p(\cdot)$.

Now applying Property 1 recursively $n-k$ times, we obtain that the projection $\mathbf{z}^k$ is the optimal rationing-level vector of a k dimensional subproblem where the cost is given by $\tilde{c}_k \circ \cdots \circ \tilde{c}_{n-1} = \tilde{c}_k$. □

A.5. Proof of Theorem 2. We prove, by induction, on the dimension k, that g_{k+1} and $(z_1, \ldots, z_{k+1})$ are respectively the optimal cost and optimal rationing level vector of the subproblem $\mathbf{P}^k(\mu, \boldsymbol{\lambda}^k, h+b_{k+1}, \mathbf{b}^k - b_{k+1}\mathbf{1})$ for $0 \le k \le n$. For $k=0$, $g_1 = z_1 = 0$ and the property is trivially true. For $k=1$, we recognize g_2 and z_2 the optimal cost and the optimal base stock of the well-known single-part-type problem (see Veatch and Wein 1996). Assume that this property is true for $k-1$. From Theorem 1, the $(k+1)$st optimal rationing level is the base-stock level of

$$\mathbf{P}^k\big(\mu, \boldsymbol{\lambda}^k, h+b_{k+1}, \mathbf{b}^k - b_{k+1}\mathbf{1}_k\big). \tag{40}$$

A direct computation leads to

$$\tilde{c}_k(\mathbf{x}^k) = \tilde{c}_{k-1}(\mathbf{x}^{k-1}) - (b_k - b_{k+1})\sum_{i=1}^{k} x_i. \tag{41}$$

Let us compute the optimal rationing-level vector and the corresponding average cost. Consider the ML policy whose rationing-level vector is equal to $(\mathbf{z}^k, z)$ and let $\mathbf{X}$ represent the random variable corresponding to $\mathbf{x}$. The corresponding average cost $g(z)$, is equal to:

$$\begin{aligned} g(z) &= E\big[\tilde{c}_{k-1}(\mathbf{X}^{k-1})\big] - (b_k - b_{k+1})E\left[\sum_{i=1}^{k} X_i\right] \qquad (42)\\ &= P\{X_1 \le z_k\}E[\tilde{c}_{k-1}(\mathbf{X}^{k-1}) \mid X_1 \le z_k]\\ &\quad + \sum_{s=z_k+1}^{z}(h+b_k)sP\{X_1 = s\} - (b_k - b_{k+1})E\left[\sum_{i=1}^{k} X_i\right]\\ &= g_k\rho_k^{z-z_k} + (h+b_k)\\ &\quad \times \sum_{s=z_k+1}^{z} s(1-\rho_k)\rho_k^{z-s} + (b_k - b_{k+1})\left(\frac{\rho_k}{1-\rho_k} - z\right). \qquad (43)\end{aligned}$$

(42) follows from (41). (43) comes from the fact that $z - \sum_{i=1}^{k} X_i$ is an $M/M/1$ queue-length process with an utilization equal to ρ_k where $z - \sum_{i=1}^{k} X_i = z - X_1$ for $X_1 > z_k$. $E[\sum_{i=1}^{k} X_i]$ is hence equal to $z - \rho_k/(1-\rho_k)$, and $P\{X_1 = s\}$ is equal to $(1-\rho_k)\rho_k^{z-s}$ for $s > z_k$.

We can then evaluate the difference $\Delta g(z) = g(z+1) - g(z)$

$$\Delta g(z) = -\rho_k^{z-z_k}((1-\rho_k)(g_k - (h+b_k)z_k) + \rho_k(h+b_k) + h + b_{k+1} \tag{44}$$

which is nondecreasing in z. $g(z)$ is convex, and its minimum is attained at $\min_z\{\Delta g(z) > 0\}$, that is at z where,

$$z = z_k + \left\lfloor \frac{\ln \frac{\rho_k(\tilde{h}+\tilde{b}_k)}{\rho_k\tilde{h} + (1-\rho_k)(g_{k-1} - \tilde{h}z_k)}}{\ln \rho_k} \right\rfloor = z_{k+1} \tag{45}$$

Using the value of z_{k+1} and (43), a direct computation leads to the expression of g_{k+1} of the theorem. □

References

Alfaro, J. A., C. J. Corbett. 1999. The value of SKU rationalization in practice. Working paper, University of California, Los Angeles, CA.

Benjaafar, S., W. L. Cooper, J.-S. Kim. 2001. On the benefits of inventory pooling in production-inventory systems. Working paper, University of Minnesota, Minneapolis, MN.

Bertsekas, D. P. 1995. *Dynamic Programming and Optimal Control*, Volume 2. Athena Scientific Publishing, Belmont, MA.

Buzacott, J. A. and J. G. Shanthikumar. 1993. *Stochastic Models of Manufacturing Systems.* Prentice Hall, NJ.

Carr, S. and I. Duenyas. 2000. Optimal admission control and sequencing in a make-to-stock/make-to-order production system. *Oper. Res.* **48** 709–720.

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

Eppen, G. 1979. Effects of centralization on expected costs in a multilocation newsboy problem. *Management Sci.* **25** 498–501.

Frank, K. C., R. C. Zhang, I. Duenyas. 1999. Optimal inventory policies for inventory systems with priority demand classes. Working paper, University of Michigan, Ann Arbor, MI.

Gross, D., C. M. Harris. 1985. *Fundamentals of Queueing Theory*. Wiley, New York.

Ha, A. 1997a. Optimal dynamic scheduling policy for a make-to-stock production system. *Oper. Res.* **45** 42–53.

——. 1997b. Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Management Sci.* **43** 1093–1103.

——. 1997c. Stock-rationing policy for a make-to-stock production system with two priority classes and backordering. *Naval Res. Logist.* **44** 458–472.

——. 2000. Stock-rationing in an $M/E_k/1$ make-to-stock queue. *Management Sci.* **46** 77–87.

Lee, H. L. 1996. Effective inventory and service management through design for localization. *Oper. Res.* **44** 151–159.

——, C. S. Tang. 1997. Modeling the costs and benefits of delayed product differentiation. *Management Sci.* **43** 40–53.

Nahmias, S., W. S. Demmy. 1981. Operating characteristics of an inventory system with rationing. *Management Sci.* **27** 1236–1245.

Peña-Perez, A., P. Zipkin. 1997. Dynamic scheduling rules for a multiproduct make-to-stock queue. *Oper. Res.* **45** 919–930.

Schwarz, L. B. 1989. A model for assessing the value of warehouse risk pooling over outside-supplier leadtimes. *Management Sci.* **35** 828–842.

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

Tsay, A. A., S. Nahmias, N. Agrawal. 1999. Modeling supply chain contracts: A review. S. Tayur, R. Ganeshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management*. Kluwer's International Series, Boston, MA.

Veatch, M., L. M. Wein. 1996. Scheduling a make-to-stock queue: Index policies and hedging points. *Oper. Res.* **44** 634–647.

de Véricourt, F. 2000. Modèles stochastiques de décision pour le pilotage des systèmes de production multi-produits. Ph.D. thesis, Université Pierre et Marie Curie, Paris 6, France.

——, F. Karaesmen, Y. Dallery. 2000. Dynamic scheduling in a make-to-stock system: A partial characterization of optimal policies. *Oper. Res.* **48** 811–819.

——, ——, ——. 2001. Assessing the benefits of different stock allocation policies for a make-to-stock production system. *Manufacturing & Service Oper. Management* **3** 105–121.

Weber, R. R., S. Stidham. 1987. Optimal control of service rates in networks of queues. *Adv. Appl. Probab.* **19** 202–218.

Wein, L. M. 1992. Dynamic scheduling of a multiclass make-to-stock queue. *Oper. Res.* **40** 724–735.

Accepted by Wallace J. Hopp; received September 28, 2001. This paper was with the authors 4 months for 1 revision.