The impact of dampening demand variability in a production/inventory system with multiple retailers

B. Van Houdt
Department of Mathematics and Computer Science
University of Antwerp - IBBT
Antwerpen, Belgium
benny.vanhoudt@ua.ac.be

J.F. Pérez
Department of Electrical and Electronics Engineering
Universidad de los Andes
Bogotá, Colombia
jf.perez33@uniandes.edu.co

ABSTRACT

We study a supply chain consisting of a single manufacturer and two retailers. The manufacturer produces goods on a make-to-order basis, while both retailers maintain an inventory and use a periodic replenishment rule. As opposed to the traditional (r, S) policy, where a retailer at the end of each period orders the demand seen during the previous period, we assume that the retailers dampen their demand variability by smoothing the order size. More specifically, the order placed at the end of a period is equal to β times the demand seen during the last period plus $(1 - \beta)$ times the previous order size, with $\beta \in (0, 1]$ the smoothing parameter.

We develop a GI/M/1-type Markov chain with only two nonzero blocks A_0 and A_d to analyze this supply chain. The dimension of these blocks prohibits us from computing its rate matrix R in order to obtain the steady state probabilities. Instead we rely on fast numerical methods that exploit the structure of the matrices A_0 and A_d, i.e., the power method, the Gauss-Seidel iteration and GMRES, to approximate the steady state probabilities.

Finally, we provide various numerical examples that indicate that the smoothing parameters can be set in such a manner that all the involved parties benefit from smoothing. We consider both homogeneous and heterogeneous settings for the smoothing parameters.

1. INTRODUCTION

Consider a two-echelon supply chain consisting of a single retailer and a single manufacturer, where the retailer places an order for a batch of items with the manufacturer at regular time instants, i.e., the time between two orders is fixed and denoted as r. The manufacturer may be regarded as a single server queue that produces these items and delivers them to the retailer as soon as a complete order is finished. The retailer sells these items and maintains an inventory on hand to meet customer demands. When the customer demand exceeds the current inventory on hand, only part of the demand is immediately fulfilled and the remaining items are delivered as soon as new items become available at the retailer. Hence, items are backlogged instead of being lost (i.e., there are no *lost sales*). We assume that the manufacturer does not maintain an inventory, but simply produces items whenever an order arrives, i.e., it operates on a *make-to-order* basis.

A key performance measure in such a system is the *fill-rate*, which is a measure for the proportion of customer demands that can be met without any delay. In order to guarantee a certain fill-rate it is important to determine the size of the orders placed at the regular time instants. This size will depend on the current *inventory position*, defined as the inventory on hand plus the number of items on order minus the number of backlogged items. The rule that determines the order size is termed the *replenishment* rule. A well-studied replenishment rule exists in ordering an amount such that the inventory position is raised after each order to some fixed position S, called the *base-stock level*. This basically means that at the regular time instants, you simply order the amount of items sold since the last order instant. As a result, the order policy of the retailer is called an (r, S) policy.

A common approach in the analysis of such a policy is to assume an *exogenous* lead time, which means that the time required to *deliver* an order is independent of the size of the current order and independent of the lead time of previous orders. In [4] the (R, S) policy was studied with *endogenous* lead times, meaning the lead times depend on the order size and consecutive lead times are correlated. The results in [4] indicate that exogenous lead times result in a severe underestimation of the required inventory on hand, as expected.

When the lead times are endogenous, it is clear that a high variability in the order sizes comes at a cost, as this increases the variability of the arrival process at the manufacturer and therefore increases the lead times. As a result, replenishment rules that smooth the order pattern at the retailer were studied in [3] and it was shown that the retailer can reduce the upstream demand variability without having to increase his safety stock (much) to maintain customer service at the same target level. Moreover, on many occasions the retailer can even decrease his safety stock somewhat when he smooths his orders. This is clearly advantageous for both the retailer and the manufac-

turer. The manufacturer receives a less variable order pattern and the retailer can decrease his safety stock while maintaining the same fill rate, so that a cooperative surplus is realized.

In this paper we analyze the same set of replenishment rules as in [3], but now we look at a two-echelon supply chain consisting of one manufacturer and two retailers, where either both, one or neither of the retailers uses a smoothing rule. The main question that we wish to address therefore exists in studying whether all parties can still benefit when the orders are smoothed and moreover who benefits most.

As in [3], one of the key steps in the analysis of this supply chain system will exist in setting up a GI/M/1-type Markov chain [7], that has only two non-zero blocks, denoted as A_0 and A_d. However, as opposed to [3], the size of these blocks often prohibits us from storing them into our main (or secondary) memory. This implies that iteratively computing the dense R matrix, used to express the matrix geometric steady state vector of the GI/M/1-type Markov chain, by one of the existing methods such as the functional iteration or cyclic reduction [1], is no longer possible/efficient. Instead, we will rely on the specific structure of the matrices A_0 and A_d and will make use of numerical methods typically used to solve large finite Markov chains, such as the shuffling algorithm [5], Kronecker products, the power method, the Gauss-Seidel iteration and GMRES [9].

2. MODEL DESCRIPTION

We consider a two-echelon supply chain with two retailers and a single manufacturer, where both retailers maintain their own inventory. Every period, both retailers observe their customer demand. If there is enough on-hand inventory available at a retailer, the demand is immediately satisfied. If not, the shortage is backlogged. To maintain an appropriate amount of inventory on hand, both retailers place a replenishment order with the manufacturer at the end of every period. The manufacturer does not hold a finished goods inventory but produces the orders on a make-to-order basis. The manufacturers production system is characterized by a single server queueing model that sequentially processes the orders, which require stochastic processing times. Once the complete replenishment order of both retailers is produced, the manufacturer replenishes both inventories. Hence, the order in which the two orders are produced is irrelevant, as shipping only occurs when both orders are ready.

The time from the moment an order is placed to the moment that it replenishes the retailers inventory, is the *replenishment* lead time T_r. The queueing process at the manufacturer clearly implies that the retailers replenishment lead times are stochastic and correlated with the order quantity. The sequence of events in a period is as follows. The retailer first receives goods from the manufacturer, then he observes and satisfies customer demand and finally, he places a replenishment order with the manufacturer. The following additional assumptions are made:

1. Customer demand during a period for retailer i is independently and identically distributed (i.i.d.) over time according to an arbitrary, finite, discrete distribution $D^{(i)}$ with a maximum of $m_D^{(i)}$, for $i = 1$ and 2. The demand at the retailers is also assumed to be independent of each other. For further use, denote $m_D = m_D^{(1)} + m_D^{(2)}$.

2. The order quantity $O_t^{(i)}$ of retailer i during period t is determined by the retailers replenishment rule and influences the variability in the orders placed on the manufacturer. Possible replenishment rules are discussed in the next section.

3. The replenishment orders are processed by a single FIFO server. This excludes the possibility of order crossovers. When the server is busy, new orders join a queue of unprocessed orders.

4. The orders placed during period t are delivered when both orders have been produced.

5. Orders consist of multiple items and the production time of a single item is i.i.d. according to a discrete-time phase type (PH) distribution with representation (α, U). For further use, we define $u^* = e - Ue$, with e a column vector of ones.

The PH distribution is determined using the matching procedure presented in [3], that matches the first two moments of the production time using an order 2 representation, even if the squared coefficient of variation is small by exploiting the scaling factor as in [2]. This implies that the length of a time slot is chosen as half of the mean production time of an item. In other words, the mean production time of an item is two time slots, while the length of a period is denoted as d time slots, where d is assumed to be an integer.

The time from the moment the order arrives at the production queue to the point that the production of the entire batch is finished, is the *production* lead time or response time, denoted by T_p. Note that the production lead time is not necessarily an integer number of periods. Since in our inventory model events occur on a discrete time basis with a time unit equal to one period, the replenishment lead time T_r is expressed in terms of an integer number of periods. For instance, suppose that the retailer places an order at the *end* of period t, and it turns out that the production lead time is 1.4 periods. This order quantity will be added to the inventory in period $t + 2$, and due to our sequence of events, can be used to satisfy demand in period $t + 2$. As such, we state that the replenishment lead time T_r is $\lfloor T_p \rfloor$ periods, i.e., 1 period in our example.

3. REPLENISHMENT RULES

The retailers considered in this paper apply an (r, S) policy with or without smoothing, meaning amongst others they place an order at the end of

each period. Without smoothing, the order size is such that the inventory position IP, defined as the on-hand inventory plus the number of items on order minus the backlogged items, equals some fixed S after the order is placed. In other words, the size of the order O_t at the end of period t simply equals the demand D_t observed during period t.

If smoothing is applied with parameter $0 < \beta < 1$, we do not order the difference between S and IP, but instead only order β times $S - IP$. As will become clear below, this does not imply that fewer items are ordered in the long run, it simply means that some items will be ordered at a later time. As shown in [3], this rule is equivalent to stating that the size of the order at the end of period t, denoted O_t, is given by

$$O_t = (1 - \beta)O_{t-1} + \beta D_t,$$

where D_t is the demand observed by a retailer in period t. Hence, setting $\beta = 1$ implies that we do not smooth. This equation also shows that the mean order size is still equal to the mean demand size $E[D]$. It is also easy to show [3] that the variance of the order size $Var[O]$ equals

$$\frac{\beta}{(2 - \beta)} Var[D],$$

meaning the variance decreases to zero as β approaches zero, where $Var[D]$ is the variance in the demand. It is also possible to consider β values between 1 and 2, but this would amplify the variability instead of dampening it.

The key question that our analytical model will answer is how to select the base-stock level S such that the fill-rate, a measure for the proportion of demands that can be immediately delivered from the inventory on hand, defined as

$$1 - \frac{\text{expected number of backlogged items}}{\text{expected demand}},$$

is sufficiently high. The level S is typically expressed using the *safety stock* SS, defined as the average net stock just before a replenishment arrives (where the net stock equals the inventory on hand minus the number of backlogged items). For a retailer that smooths with parameter β, S and SS are related as follows [3]

$$S = SS + (T_r + 1)E[D] + \frac{1 - \beta}{\beta} E[D],$$

where T_r is the mean replenishment lead time. Thus, a good policy will result in a smaller safety stock SS, which implies a lower average storage cost for the retailer.

4. THE MARKOV CHAIN

Both Markov chains developed in this section are a generalization of the Markov chain introduced in [3], for the system with a single retailer. The numerical method to attain their stationary probability vector, discussed in Section 5, is however very different.

From now on we will express all our variables in time slots, where the length of a single slot equals half of the mean production time, i.e., $\alpha(I-U)^{-1}e/2$, and orders are placed by both retailers every d time slots. Hence, the order size of retailer i at the end of period t is now written as $O_{td}^{(i)}$ and

$$O_{td}^{(i)} = (1 - \beta_i)O_{(t-1)d}^{(i)} + \beta_i D^{(i)},$$

where β_i is the smoothing parameter of retailer i, for $i = 1, 2$. As the order size must be an integer, the integer amount ordered $O_{td}^{(i*)}$ will equal $\lceil O_{td}^{(i)} \rceil$ with probability $O_{td}^{(i)} - \lfloor O_{td}^{(i)} \rfloor$ and $\lfloor O_{td}^{(i)} \rfloor$ with probability $\lceil O_{td}^{(i)} \rceil - O_{td}^{(i)}$ in case $O_{td}^{(i)}$ is not an integer. This guarantees that $E[O_{td}^{(i*)}] = E[O_{td}^{(i)}] = E[D^{(i)}]$.

The joint order O_{td}^{*} of both retailers placed at time td equals $O_{td}^{(1*)} + O_{td}^{(2*)}$. Recall, both these orders are only delivered by the manufacturer when the joint order has been produced. Next, define the following random variables:

- t_n: the time of the n-th observation point, which we define as the n-th time slot during which the server is busy,
- $a(n)$: the arrival time of the joint order in service at time t_n,
- B_n: the age of the joint order in service at time t_n, expressed in time slots, i.e., $B_n = t_n - a(n)$,
- C_n: the number of items part of the joint order in service that still need to start or complete service at time t_n,
- S_n: the service phase at time t_n.

All events, such as arrivals, transfers from the waiting line to the server, and service completions are assumed to occur at instants immediately after the discrete time epochs. This implies that the age of an order in service at some time epoch t_n is at least 1. We start by introducing the Markov chain for the case where both retailers smooth.

4.1 Both retailers smooth

It is clear that the stochastic process $(B_n, C_n, O_{a(n)}^{(1)}, O_{a(n)}^{(2)}, S_n)_{n \geq 0}$ forms a discrete time Markov process on the state space $\mathbb{N}_0 \times \{(c, x_1, x_2) | c \in \{1, \ldots, m_D\}, 1 \leq x_i \leq m_D^{(i)}, i \in \{1, 2\}\} \times \{1, 2\}$, as the PH service requires only two phases. Note that the process makes use of the order quantities $O_{a(n)}^{(i)}$ instead of the integer values $O_{a(n)}^{(i*)}$. Since this order quantity is a real number, the Markov process $(B_n, C_n, O_{a(n)}^{(1)}, O_{a(n)}^{(2)}, S_n)_{n \geq 0}$ has a continuous state space which makes it very hard to find its steady state vector.

Therefore, instead of keeping track of $O_{a(n)}^{(i)}$ in an exact manner, we will round it in a probabilistic way to the nearest multiple of $1/g$, where $g \geq 1$ is an integer termed the *granularity* of the system. Clearly, the larger g, the better the approximation. Hence, we approximate the Markov process above by the Markov chain $(B_n, C_n, O_{a(n)}^{g,(1)}, O_{a(n)}^{g,(2)}, S_n)_{n \geq 0}$ on the discrete

state space $\mathbb{N}_0 \times \{(c, x_1, x_2)|c \in \{1, \ldots, m_D\}, x_i \in \mathbb{S}_g^{(i)}, i \in \{1, 2\}\} \times \{1, 2\}$, where $\mathbb{S}_g^{(i)} = \{1, 1 + 1/g, 1 + 2/g, \ldots, m_D^{(i)}\}$ and the quantity $O_{td}^{g,(i)}$ evolves as follows. Let

$$x = (1 - \beta_i)O_{(t-1)d}^{g,(i)} + \beta_i D^{(i)},$$

then $O_{td}^{g,(i)} = x$ if $x \in \mathbb{S}^{(i)}$, otherwise it equals $\lceil x \rceil_g$ with probability $g(x - \lfloor x \rfloor_g)$, or $\lfloor x \rfloor_g$ with probability $g(\lceil x \rceil_g - x)$, where $\lceil x \rceil_g$ ($\lfloor x \rfloor_g$) rounds up (down) to the nearest element in $\mathbb{S}_g^{(i)}$. Notice, by induction, we have $E[O_{td}^{g,(i)}] = E[D^{(i)}]$. Using this probabilistic rounding, we can easily compute the conditional probabilities $P[O_{td}^{g,(i)} = q'|O_{(t-1)d}^{g,(i)} = q]$, which we denote as $p_g^{(i)}(q, q')$, from $D^{(i)}$.

The transition matrix P_g of the Markov chain $(B_n, C_n, O_{a(n)}^{(1)}, O_{a(n)}^{(2)}, S_n)_{n \geq 0}$ is a GI/M/1-type Markov chain [7] with the following structure

$$P_g = \begin{bmatrix} A_d & A_0 & & & & \\ \vdots & & \ddots & & & \\ A_d & & & A_0 & & \\ & A_d & & & A_0 & \\ & & \ddots & & & \ddots \end{bmatrix},$$

as B_n either increases by one if the same joint order remains in service, or decreases by d if a joint order is completed. The size m of the square matrices A_0 and A_d is $2m_D m_g$, with $m_g = \prod_{i=1}^{2}(m_D^{(i)} g - g + 1)$, which is typically such that we cannot store the matrices A_0 and A_d in memory. Although we can eliminate close to 50% of the states by removing the transient states with $C_n > \lceil O_{a(n)}^{(1)} \rceil + \lceil O_{a(n)}^{(2)} \rceil$, the size m remains problematic and this would slow down the numerical solution method presented in Section 5.

4.2 One retailer smooths

Assume without loss of generality that retailer one smooths, while retailer two does not, i.e., $\beta_1 < 1$ and $\beta_2 = 1$. In this case we can also rely on the Markov chain defined above, but now there is no longer a need to keep track of $O_{a(n)}^{g,(2)}$, as the orders of retailer two are distributed according to $D^{(2)}$. This not only simplifies the transition probabilities, but also considerably reduces the time and memory requirements of the numerical solution method introduced in Section 5. Although storing the matrices A_0 and A_d in memory may no longer be problematic, a numerical approach as presented in the next section still outperforms the more traditional approach that relies on computing the rate matrix R [7].

5. NUMERICAL SOLUTION

The objective of this section is to introduce a numerical method to compute the steady state distribution of the Markov chain introduced in Section 4.1 by avoiding the need to store the matrices A_0 and A_d.

5.1 Fast multiplication

In order to multiply the vector $x = (x_0, x_1, \ldots)$ with P_g, where x_i is a length m vector, without storing the matrices A_0 or A_d, we will write P_g as the sum of $P_g^{(0)} + P_g^{(d)} =$

$$\begin{bmatrix} A_0 & & & \\ & \ddots & & \\ & & A_0 & \\ & & & \ddots \end{bmatrix} + \begin{bmatrix} A_d & & \\ \vdots & & \\ A_d & & \\ & \ddots & \end{bmatrix},$$

and compute xP_g as $xP_g^{(0)} + xP_g^{(d)}$. To express the time complexity of these multiplications, assume $x_i = 0$ for $i \geq n$ for some n (as will be the case in the next subsection).

The matrix A_0 corresponds to the case where the same joint order remains in service, meaning C_n either remains the same or decreases by one. Due to the order of the random variables, the matrix A_0 is a bi-diagonal block Toeplitz matrix, with blocks of size $2m_g$. The block appearing on the main diagonal equals $I \otimes U$, as the production of the same item continues in this case. The block below the main diagonal is $I \otimes u^* \alpha$, as the item is finished, but at least one item of the joint order needs to be produced. Hence, as the PH representation is of order 2 (even in case of low variability), we can multiply x with $P_g^{(0)}$ in $O(mn)$ time.

When multiplying with A_d, we first note that only its first $2m_g$ rows are non-zero, as C_n must equal one and a service completion must occur. Hence, each of the vectors x_i is reduced to a length m_g vector in $O(nm_g)$ time. These vectors must be multiplied with $W_1 \otimes W_2$, where the (q, q')-th entry of W_i equals $p_g^{(i)}(q, q') = P[O_{td}^{g,(i)} = q'|O_{(t-1)d}^{g,(i)} = q]$, for $i = 1, 2$. The multiplication with $W_1 \otimes W_2$ is done in two steps. First we multiply with $(I \otimes W_2)$, which can be trivially done in $O((m_D^{(2)} g)^2 m_D^{(1)} g) = O(m_g m_D^{(2)} g))$ for each vector, followed by the multiplication with $(W_1 \otimes I)$. This latter multiplication can be rewritten as a multiplication with $(I \otimes W_1)$ using the shuffle algorithm[5]. Hence, it can also be done in $O(m_g m_D^{(1)} g)$.

Performing the multiplication with $W_1 \otimes W_2$ corresponds to determining the new order sizes for each retailer. To complete the transition we need to determine the joint order size and the initial service phase of the first item part of the joint order. The first corresponds to distributing the outcome of the abovementioned multiplications in the proper entries, while the service phase is found using α. In conclusion, the time required to multiply x with $P_g^{(d)}$ can be written as $O(nm_g(m_D^{(1)} + m_D^{(2)})g) = O(nmg)$ and the time needed to multiply x with P_g is therefore also $O(nmg)$. In practice, for g small, the multiplication with $P_g^{(0)}$ is more time demanding than the multiplication with $P_g^{(d)}$ and a considerable percentage of the time is also spend on allocating memory.

5.2 The power method, the Gauss-Seidel iteration and GMRES

To determine the steady state probability vector of the transition matrix P_g we rely on the fast matrix multiplication between a vector x and P_g introduced above.

When combined with the power method, we basically start with some initial vector $x(0)$ and define $x(k+1) = x(k)P_g$ until the infinity norm of $x(k+1) - x(k)$ is smaller than some predefined ϵ_1 (e.g., $\epsilon_1 = 10^{-8}$). If we start from an empty system, $x(0)$ has only one nonzero component $x_0(0)$ of length m and $x(k)$ has $k+1$ nonzero components $x_0(k)$ to $x_k(k)$. Whenever some of the last components are smaller than some predefined ϵ_2, we reduce the length of $x(k)$ (by adding these components to the last component larger than ϵ_2). When solving several related systems (e.g., when investigating the impact of β_i), a considerable amount of time can also be saved by using the steady state of one system as starting value $x(0)$ for the next system.

When applying the *forward* Gauss-Seidel iteration [8], we compute $x(k+1)$ from $x(k)$ by solving the linear system

$$x(k+1)(I - P_g^{(0)}) = x(k)P_g^{(d)},$$

which can be done efficiently using forward substitution as $(I - P_g^{(0)})$ is upper triangular. If x is an arbitrary stochastic vector, we initialize $x(0)$ such that it solves $x(0)(I - P_g^{(0)}) = x$. As indicated in [8], this Gauss-Seidel iteration is equivalent to a preconditioned power method if we use $(I - P_g^{(0)})$ as the preconditioning matrix M. Notice, we can benefit from the fast multiplications discussed in the previous section when computing $x(k)P_g^{(d)}$ as well as during the forward substitution phase.

The GMRES method [9] computes an approximate solution of the linear system $(I - P_g')x = 0$, by finding a vector $x(1)$ that minimizes $\left\|(I - P_g')x\right\|_2$ over the set $x(0) + \mathcal{K}(I - P_g', r_0, n)$. Here r_0 is the residual of an initial solution $x(0)$: $r_0 = -(I - P_g')x(0)$; $\mathcal{K}(I - P_g', r_0, n)$ is the Krylov subspace, i.e., the subspace spanned by the vectors $\{r_0, (I - P_g')r_0, \ldots, (I - P_g')^{n-1}r_0\}$; and n is the dimension of the Krylov subspace [6]. To do this GMRES relies on the Arnoldi iteration to find an orthonormal basis V_n for the Krylov subspace, such that $V_n'(I - P_g')V_n = H_n$, where H_n is an upper Hessenberg matrix of size n. Once V_n and H_n have been obtained, a vector y_n is found such that $J(y) = \left\|\beta e_1 - \tilde{H}_n y\right\|_2$ is minimized. Here β is the 2-norm of r_0, e_1 is the first column of the identity matrix, and $\tilde{H}_n$ is an $(n+1) \times n$ matrix whose first n rows are identical to H_n, and its last row has one nonzero element that also results from the Arnoldi iteration. A new approximate solution $x(1)$ is computed as $x(1) = x(0) + V_n y_n$. The process is then repeated with $x(1)$ as $x(0)$ until the difference between two consecutive solutions is less than some predefined ϵ. Although this algorithm is defined to solve linear systems of the type $Ax = b$, with A nonsingular, it can also be used to solve homogeneous systems with A singular, as is the case with Markov chains [10].

The GMRES algorithm also benefits from the fast multiplication discussed in the previous section. To find the residual r_0 at each iteration we need to compute the product $(I - P_g')x(0) = x(0) - P_g'x(0)$. Also, for the Arnoldi process we need to determine the vectors $v_j = (I - P_g')^{j-1}r_0$, which are computed iteratively, and require $n-1$ products of the type $(I - P_g')v_{j-1} = v_{j-1} - P_g'v_{j-1}$. As with the power method, when analyzing several scenarios we can use the final approximate solution of one scenario as the starting solution for the next one to speed up convergence.

6. THE SAFETY STOCK

The required safety stock SS_i for each retailer to guarantee a certain fill rate is one of the main performance measures of this supply chain problem. As indicated in Section 3, computing SS_i is equivalent to determining the base-stock S_i provided that we know the mean replenishment lead time T_r (which equals the floor of the production lead time T_p). The production lead time distribution T_p is easy to obtain from the steady state probability vector π of P_g as follows. First define the length $2m_g$ vectors $\pi_b(c)$ as the steady state probabilities of being in a state with $B_n = b$ and $C_n = c$. Then, the probability of having a production lead time of b slots equals

$$P[T_p = b] = \rho\pi_b(1)(e \otimes u^*)/(1/d)$$

for $b > 0$, where $\rho = 2(E[D^{(1)}] + E[D^{(2)}])/d$ is the load at the manufacturer and $1/d$ is the arrival rate of the joint orders.

The fill rate is defined as $1 - E[(-NS)^+]/E[D]$, where NS is the net stock (i.e., inventory on hand minus backlog) and $x^+ = \max\{0, x\}$. Hence, $E[(-NS)^+]$ is the expected number of backlogged items. Similar to [3, Section 5.1], we can show that

$$NS_i = S_i + \sum_{j=1}^{k} D^{(i)} + O_k^{(i)}/\beta,$$

where k is the age, expressed in periods, of the joint order in production at the manufacturer at the end of a period and this joint order contains $O_k^{(i)}$ items for retailer i, for $i = 1, 2$. If $k = 0$, meaning the last order left the queue before the end of the period, $O_k^{(i)}$ is the number of items ordered by retailer i in the next joint order. Thus, the key step in determining the required base-stock value S_i, exists in computing the joint probabilities $p_{k,q}^{(i)}$ of having an order of age kd in service when a period ends and the order in service contains q items for retailer i, for $i = 1, 2$, $k \geq 0$ and $q \in \{1, \ldots, m_D^{(i)}\}$.

These joint probabilities can be readily obtained from the steady state of the Markov chain introduced in Section 4.1 as

$$p_{k,q}^{(i)} = \rho d \pi_{kd}^{(i)}(q) e,$$

for $k > 0$, where $\pi_b^{(i)}(q)$ is the steady state vector for the states with $B_n = b$ and $O_{a(n)}^{g,(i)} = q$. For $k = 0$, we note that an order finds the queue empty upon arrival

if the previous order had a lead time of at most $d-1$, yielding

$$p_{0,q}^{(i)} = \rho d \sum_{b=1}^{d-1} \sum_{q_1,q_2,s} \pi_b(1,q_1,q_2,s) u_s^* p_g(q_i,q),$$

where $\pi_b(c,q_1,q_2,s)$ is the steady state probability of state (b,c,q_1,q_2,s).

If we wish to compute the joint probabilities $p_{k,q}^{(2)}$ from the Markov chain $(B_n, C_n, O_{a(n)}^{g,(1)}, S_n)_{n\geq 0}$ in case only the first retailer smooths, things are somewhat more involved when $k > 0$. For $k = 0$, we clearly have

$$p_{0,q}^{(2)} = P[T_p < d]P[D^{(2)} = q].$$

For $k > 0$, we start by computing $p_w(q_1, x)$, the probability that an order consisting of q_1 items for retailer 1 has a waiting time of $x > 0$ slots. As the waiting time x of an order with $x > 0$ equals the lead time of the previous order minus the inter-arrival time d, we find

$$p_w(q_1,x) = \frac{\rho d}{\pi(q)} \sum_{q,s} \pi_{x+d}(1,q,s) u_s^* p_g(q,q_1),$$

where $\pi_b(c,q,s)$ is the steady state probability of state (b,c,q,s) and $\pi(q)$ is the probability that an arbitrary order contains q items for retailer 1.

Next, we determine the probabilities $p_o(q_1,q_2,y)$ that an arbitrary joint order consists of q_i items for retailer i and its production time equals y time slots. These probabilities are readily obtained from $p_g(q,q')$ and (α, U). Then,

$$p_a(q_1,q_2,x) = \sum_{y\geq x} \frac{p_o(q_1,q_2,y)}{2(E[D^{(1)}] + E[D^{(2)}])},$$

is the probability that we find a joint order consisting of q_i items for retailer i in service at an arbitrary moment when the server is busy, while the service of this joint order started x time slots ago. Taking the convolution over x between $p_w(q_1,x)$ and $p_a(q_1,q_2,x)$ and summing over q_1, gives us the probability that the order in service has an age of x time slots and consists of q_2 items for retailer 2, given that we observe the system when the server is busy. From these probabilities the joint probabilities $p_{k,q}^{(2)}$ are readily found.

We can also compute the probabilities $p_{k,q}^{(2)}$ from the Markov chain in Section 4.1 by setting $\beta_2 = 1$, but this approach requires considerably more time and memory. As required, the numerical experiments indicated a perfect agreement between both approaches.

7. NUMERICAL EXAMPLES

In this section we illustrate the effect of smoothing on the performance of the production/inventory system. We focus on the mean replenishment lead time and the safety stock as the main measures of performances, and consider various scenarios for the demand distribution, the load and the smoothing parameters β_1 and β_2. The required safety stock in all the numerical examples guarantees a fill rate of 0.98.

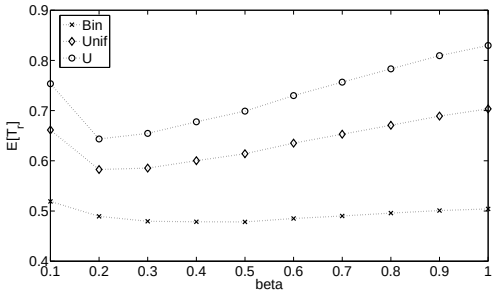


Figure 1: Mean Lead Time vs. β - $\rho = 0.85$

For the demand we consider three different distributions, let us call the three associated random variables X, Y and Z, respectively. X is defined as $X = 1 + \hat{X}$, where $\hat{X}$ is a Binomial distribution with parameters $N-1$ and $p = 1/2$. Thus, X takes vales on the set $\{1,\ldots,N\}$. The expected value and variance of X are $E[X] = (N+1)/2$ and $\mathrm{Var}(X) = (N-1)/4$. The second random variable Y is uniformly distributed between 1 and N, and its expected value and variance are $E[Y] = (N+1)/2$ and $\mathrm{Var}(Y) = (N^2-1)/12$. The last random variable is defined as $P(Z = k) = (1+\alpha)P(Y = k) - \alpha P(X = k)$, for $k = 1,\ldots,N$. As a result Z has a U-shaped probability mass function, with $E[Z] = (N+1)/2$ and $\mathrm{Var}(Z) = (N^2 - 1 + \alpha(N^2 - 3N + 2))/12$. Clearly, for Z to be a proper random variable, the value of α has to be such that $P(Z = k) \geq 0$ for all k. In our experiments we set $N = 10$, for which α can take values up to roughly 0.68. We choose 0.6 to make Z highly variable. With this setup, $\mathrm{Var}(X) = 2.25$, $\mathrm{Var}(Y) = 8.25$ and $\mathrm{Var}(Z) = 8.25 + 6\alpha = 11.85$. Also, setting the maximum demand size to $N = 10$, the size of the square blocks A_0 and A_D is 4000 (for $g = 1$).

As mentioned before, the mean production time is set equal to 2, and for the experiments in this section the standard deviation is also set to 2. The load is set by adjusting d, the number of slots between two orders placed by the retailers. In our setup we choose d from the set $\{40,\ 34,\ 29,\ 26\}$, which generate loads of roughly $\{0.55,\ 0.65,\ 0.76,\ 0.85\}$, respectively. In the next section we start by looking at the case where both retailers use the same value of the smoothing parameters β_1 and β_2. Afterward we consider the case where these parameters may differ.

7.1 Homogeneous smoothing

We start by looking at a system facing a load of $\rho = 0.85$, and we consider values of $\beta = \beta_1 = \beta_2$ in the set $\{0.1, 0.2, \ldots, 1\}$, and the three different demands described above. The results are included in Figure 1, where we observe that the mean replenishment lead time is minimized at a β value different from 1, meaning that both retailers benefit from smoothing with respect to the replenishment time. As expected, the effect of the smoothing increases with the variability of the demand distribution. This confirms the ability of smoothing as a means to dampen

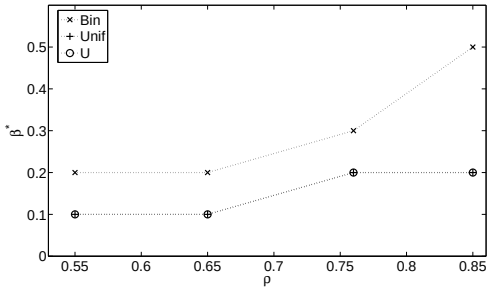


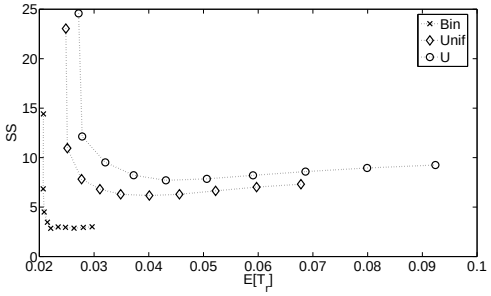
Figure 2: Optimal β vs. ρ

variability. Although the behavior of the mean replenishment lead time as a function of β is similar for different loads, the load does affect the value of β for which the mean lead time is minimized, as illustrated in Figure 2. There we see that the optimal value of β within the set $\{0.1, \ldots, 0.9\}$, with respect to the lead time, increases with the load, and this value is smaller under uniform and U-shaped demand distributions than under Binomial demand. This means that smoothing is less beneficial under high loads. As the load increases, the lead times are mostly affected by the orders' queueing time at the manufacturer, reducing the effect of the order variability, and therefore, of applying a smoothing rule.

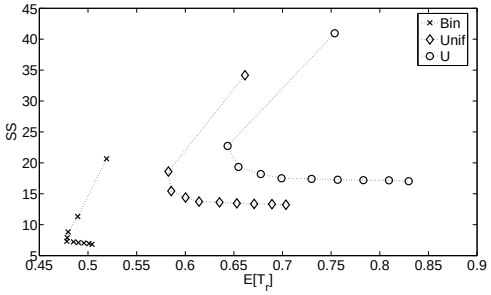
Next, we look at the behavior of the safety stock together with the lead time. In Figure 3 we depict the mean replenishment lead time $E[T_r]$ vs. the safety stock of one of the retailers. Since both retailers use the same value for the smoothing parameter, β, the safety stock is the same for both. As mentioned above, the values of β considered are $\{0.1,\ 0.2,\ \ldots,\ 1.0\}$, and the safety stock is largest when $\beta = 0.1$. In the case of medium loads, $\beta = 0.1$ also produces a (almost) minimal replenishment lead time. For higher loads, however, setting β too small results in both a higher SS and an increase in the mean lead time. Further, at high loads, decreasing the value of β increases the SS consistently, but we observe a rather large set of β values for which the SS increases only slightly, while the mean lead time varies more significantly. For instance, for a load of 0.85 and under Binomial demand, decreasing the value of β below 0.5 implies a large increase in both SS and mean lead time. However, values of β above 0.5 have a comparatively small effect on the SS, but have a significant influence on the mean lead time.

7.2 Heterogeneous smoothing

We now consider a system where each retailer chooses the value of its smoothing parameter independently. In Figure 4 we show the mean lead time for different values of β_1 and β_2. In this case, where the load is 0.85 and the demand follows a Binomial distribution, we observe that the best value for β_1 is 0.5, as selecting any other value of β_1, among those considered here, implies a larger mean lead time, independently of the value of β_2. Similarly, choosing



(a) $\rho = 0.55$



(b) $\rho = 0.85$

Figure 3: Safety Stock vs. Mean Lead Time

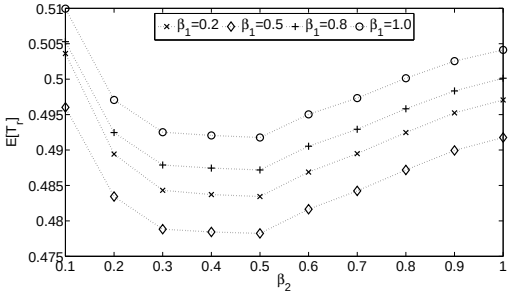


Figure 4: Mean Lead Time vs. β_1 - $\rho = 0.85$ - Binomial$(1/2)$

$\beta_2 = 0.5$ is optimal for every value of β_1. Thus, there exists an optimal choice of (β_1, β_2) that the retailers and the manufacturer might agree upon, as they all (especially the manufacturer) benefit from a less variable order pattern and shorter lead times (the retailers benefit either directly as their SS might decrease or indirectly as the manufacturer will reward them for the more regular order pattern).

We now introduce Figure 5, which conveys similar information as Figure 3 in the previous section. In this case we consider the total safety stock, i.e., the sum of the safety stock of both retailers, against the mean lead time. The demand is assumed to follow a Binomial distribution and we consider different values for β_1 and β_2. As before, due to the high load, increasing any of the smoothing parameters decreases the total SS. Whenever one of the smoothing parameters

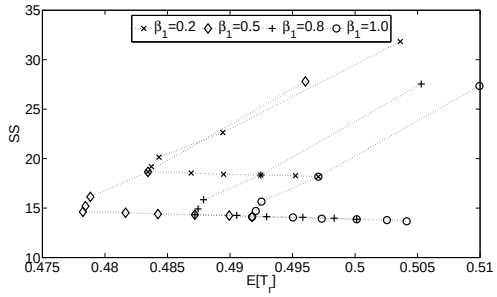


Figure 5: Safety Stock vs. Mean Lead Time - $\rho = 0.85$ - Binomial$(1/2)$

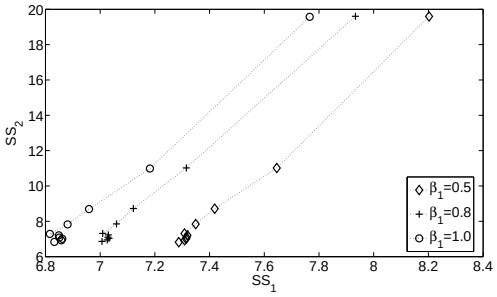


Figure 6: Safety Stock per retailer - $\rho = 0.85$ - Binomial$(1/2)$

becomes too small, there is a large increase in both the lead time and the SS. For larger β_i values, decreasing β_i causes a marginal increase in safety stock and a decrease in mean lead time. Hence, the lead time can be minimized without increasing the total SS much. In this case we observe that $\beta_1 = \beta_2 = 0.5$ results in a minimum mean lead time.

In Figure 6 we consider the same scenario as in the previous two figures, but we now look at the behavior of the safety stock for each retailer. Recall that here we left β_1 fixed, and change the value of β_2. The highest value of SS_2 corresponds to $\beta_2 = 0.1$, and it decreases as β_2 increases. We observe a very significant decrease in SS_2 when β_2 increases from 0.1 to 0.4. Further increasing β_2 has very little effect on the safety stock of any of the retailers. This means that the large decrease in the total SS observed before, comes mostly from a decrease in the SS of retailer 2, which is the one modifying its smoothing parameter. However, and more importantly, the *other* retailer also benefits from this smoothing since its SS also decreases. As we showed before, setting $\beta_2 = 0.5$ is optimal for the mean replenishment time, which is most beneficial for the manufacturer. Therefore, all the participants benefit from an adequate choice of β_2, creating an incentive for reaching agreements between them. It is interesting to note that these results also hold for $\beta_1 = 1$, i.e., when the first retailer does not smooth. Thus, even if one of the retailers decides not to smooth, the manufacturer will benefit from a smoothing agreement with the other retailer.

For heterogeneous smoothing, we have so far considered the case where the demand follows a Binomial distribution. We now look at the case where the demand has a U-shaped distribution. The results are shown in Figure 7(b), where we depict the total safety stock, the mean lead time, and the safety stock for each retailer, for different values of β_1 and β_2. As in the previous scenarios, the case with $\beta_2 = 0.1$ is the point with the largest total SS in Figure 7(a). As the value of β_2 increases, the total SS decreases. In this scenario we observe that the combination $(\beta_1, \beta_2) = (0.2, 0.2)$ generates the smallest mean replenishment lead time. Increasing the value of any of the smoothing parameters beyond this value increases the mean lead time, but decreases the total SS. Here we also observe that if $\beta_1 = 0.2$, the minimal total SS that can be achieved is significantly, more that 10%, higher than the one achieved for higher values of β_1. Although this also occurs under Binomial demand, in this case the value of β_1 that minimizes the mean lead time (0.2) also forces the retailers to keep a significantly larger SS than the one that could be achieved with less or no smoothing. Therefore, it is harder in this case for the manufacturer to agree with the retailers on a smoothing pattern that minimizes the mean lead time.

Another difference that arises from the higher demand variability can be observed in Figure 7(b). As opposed to the Binomial demand, the safety stocks of the retailers do not decrease monotonically as β_1 or β_2 increases. Keeping β_1 fixed, at any of the values considered, increasing β_2 always decreases the SS of retailer 2, but the SS of retailer 1 can also increase. In other words, the SS of the first retailer increases if the second retailer decides to stop smoothing, as long as the smoothing parameter of the second retailer was not too far from one. However, this increase is comparatively small and might not be enough to force the retailers to agree upon a smoothing pattern. Also, over-smoothing has a very negative effect on the SS of the retailer that adopts that mechanism, especially under highly variable demand.

7.3 Computation times and accuracy

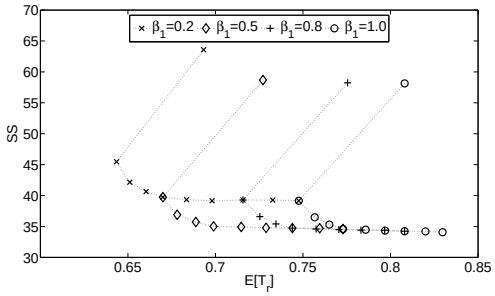
We end the paper with a few results regarding the accuracy and computation times required to obtain the results in the paper with the power, Gauss-Seidel and GMRES methods. Table 1 shows the accuracy of both the power and Gauss-Seidel methods (for $g = 1$), compared against a solution obtained with a precision of 10^{-10}, as well as the computation times and the required number of iterations. Table 2 provides the same info for the GMRES method, where the size of the Krylov subspace was set equal to 1, 3 and 5, These results corresponds to the example where the demand follows a Binomial distribution, the load $\rho = 0.85$, and both retailers smooth with $\beta_1 = \beta_2 = 0.8$. All the experiments were run on a PC with 4 cores at 2.93GHz and 4GB of RAM. We observe that, for the

Table 1: Accuracy and computation times of the power and Gauss-Seidel method

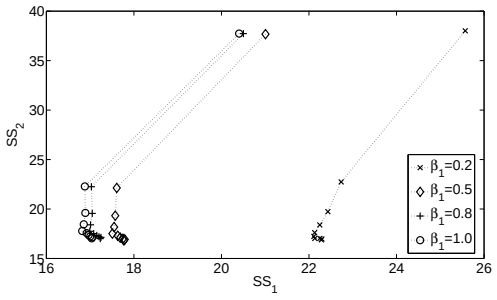
	Power			Gauss-Seidel		
	10^{-6}	10^{-7}	10^{-8}	10^{-6}	10^{-7}	10^{-8}
SS	0.3%	0.1%	0.0%	0.6%	0.1%	0.0%
E[T]	0.2%	0.0%	0.0%	0.6%	0.1%	0.0%
time (sec)	31	54	79	1.7	3.0	4.4
iter	804	1207	1636	21	34	49

Table 2: Accuracy and computation times of GMRES

	GMRES - n=1			GMRES - n=3			GMRES - n=5		
	10^{-6}	10^{-7}	10^{-8}	10^{-6}	10^{-7}	10^{-8}	10^{-6}	10^{-7}	10^{-8}
SS	15.4%	8.0%	0.9%	9.7%	2.0%	0.3%	6.8%	1.2%	0.2%
E[T]	10.6%	4.7%	0.5%	6.9%	1.1%	0.1%	4.4%	0.6%	0.1%
time (sec)	15	34	105	21	64	290	38	170	446
iter	186	261	797	89	341	301	61	120	190



(a) Safety Stock vs. Mean Lead Time



(b) Safety Stock per retailer

Figure 7: $\rho = 0.85$ - U-shaped demand

same precision level, the Gauss-Seidel method is far superior to both the power method and GMRES, as it requires far less time and has a similar accuracy than the power method. This can be explained by the fact that the Markov chain characterized by P_g typically makes many consecutive upward transitions according to A_0 followed by an occasional downward jump using A_d.

The accuracy of GMRES is quite poor when the required precision is low and is far worse than the power or Gauss-Seidel method. As the precision increases the difference in accuracy between GMRES and the other methods becomes smaller (and eventually negligible). GMRES is faster than the power method for a precision of 10^{-6} and when n is one or three, but the accuracy of GMRES is far worse in these cases. As the precision is tightened and n is increased, GMRES becomes slower than the power method.

As stated in Section 5.2 the Gauss-Seidel method may be regarded as a preconditioned power method where the preconditioning matrix M is equal to $(I - P_g^{(0)})$. In principle we can use the same preconditioning for GMRES, which should improve the performance of GMRES significantly. However, as GMRES is typically inferior to the power method, it seems unlikely that we can do better than the Gauss-Seidel method using $(I - P_g^{(0)})$ as a preconditioning matrix. We are also planning to explore other iterative methods and preconditioning matrices to see whether the computation times can be further improved.

Finally, we should mention that a significant amount of the computation time is devoted to allocating memory (due to the large sizes of the vectors, e.g., for a precision of 10^{-8}, the final vector x has a length of 732000 using Gauss-Seidel). Since GMRES computes n large vectors, it is more significantly affected by the memory allocation delay. Also, the computation times of all the methods are highly influenced by the system parameters, especially by the load ρ and the variance of the demand and processing times. Larger values for these parameters imply longer computation times and larger memory requirements.

8. REFERENCES

[1] D. A. Bini, B. Meini, S. Steffé, and B. Van Houdt. Structured Markov chains solver: algorithms. In *SMCtools Workshop*, Pisa, Italy, 2006. ACM Press.

[2] A. Bobbio, A. Horváth, and M. Telek. The scale factor: a new degree of freedom in phase type approximation. *Performance Evaluation*, 56:121–144, 2004.

[3] R. N. Boute, S. M. Disney, M. R. Lambrecht, and B. Van Houdt. An integrated production and inventory model to dampen upstream demand variability in the supply chain. *European Journal of Operational Research*, 178:121–142, 2007.

[4] R.N. Boute, M.R. Lambrecht, and B. Van Houdt. Performance evaluation of a production/inventory system with periodic review and endogeneous lead times. *Naval Research Logistics*, 54:462–473, 2007.

[5] P. Fernandes, B. Plateau, and W.J. Stewart. Efficient descriptor-vector multiplications in stochastic automata networks. *J. ACM*, 45:381–414, 1998.

[6] G. H. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[7] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. The John Hopkins University Press, Baltimore, 1981.

[8] B. Philippe, Y. Saad, and W. J. Stewart. Numerical methods in Markov chain modeling. *Operations Research*, 40:1156–1179, November 1992.

[9] Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[10] W.J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.