Journal of the Operations Research
Society of Japan
Vol. 38, No. 2, June 1995

THE OPTIMAL (r,S) POLICY IN A SINGLE ITEM PRODUCTION/INVENTORY SYSTEM WITH SETUP TIMES

Hyo-Seong Lee
Kyung Hee University

(Received December 7, 1992; Revised April 11, 1994)

Abstract In this paper, the (r,S) policy is considered for production/inventory systems where a single type items are produced on an item by item basis by a single production facility. The (r,S) policy considered in this paper is a pull type threshold policy which is very useful in situations where the time and cost required to setup the production facility are relatively high. The demand for the item is assumed to arrive according to a Poisson process. The processing time required to produce an item is assumed to follow an arbitrary distribution. When an item is demanded, one of the items in the inventory is delivered to the customer and the kanban on the item is removed. The removed kanban is immediately transmitted to the production facility. At the instant r kanbans are accumulated at the production facility, the operator turns the production facility on, which takes a random amount of time. In the production period, items are produced one by one and whenever each item is produced, a kanban is attached to it. When there are no kanbans at the production facility, the machine is shut off and a non-production period begins, which lasts until the number of kanbans accumulated at the production facility is raised back to r. In this paper, assuming a linear cost structure, an efficient search procedure is developed to find the optimal threshold value r as well as the optimal number of kanbans S, which minimizes the expected cost incurred per unit time.

1. Introduction

Production/Inventory systems can be classified into two types, viz., push and pull types. In push type systems, the amount of demands for an item is forecasted in advance and the production schedule is set based on this forecast value. In pull type systems, on the other hand, production schedule is not set beforehand, but is decided by the evolution of the demand process of the item. This pull type system is actually operated by means of the *kanban*, a sort of card or tag. Since the successful applications of the pull type systems were reported in Japanese industry, there has been a considerable work in modeling of production/inventory systems within the framework of pull type systems.

It is known that the pull type system works particularly well under the conditions of smoothed demands and reduced setup times. However, even when the setup costs/times are high, the pull type system could be applied successfully in some manufacturing environments as will be considered in this paper. We consider in this paper a pull type production/inventory system in which a single production facility produces items of a given type. Items are produced one by one by a single production facility with completed items going directly into output store. The processing time for producing (replenishing) one item is assumed to be an independent, identically distributed, random variable which follows an arbitrary distribution. Each time production is initiated, a random amount of time to turn the facility on is required. We assume that whenever each item is produced, a kanban is attached to it. Thus, to every item in the output store, a kanban is attached. The demand for the item is assumed to arrive according to a Poisson process with rate λ. If an item is demanded, one of the items in the output store is delivered to the customer if the inventory is available. At this time, the kanban on the item is removed. When this removed kanban is transmitted to the production facility, it becomes a production authorization card, that is, the presence of one kanban at the production facility authorizes the production of one item. If the inventory is not available at the instant of a demand arrival, the demand is backordered. We assume that there are unlimited raw materials available in front of the production facility at all times.

If the time and cost required to setup the production facility are negligible, we can use the following very simple policy[1]: produce whenever kanbans are available. However, when the setup cost/time is high, such a policy could be very costly. Thus in this case, to avoid the frequent setup of the machine, the following threshold policy can be adopted:

When an item is demanded, one of the items in the output store is delivered to the customer and the kanban on the item is removed. The removed kanban, which is now activated as a production authorization card, is immediately transmitted to the production facility. When the number of kanbans accumulated at the production facility reaches r while the machine is shut off, the non-production period terminates and the operator takes a random amount of time to turn the production facility on, which initiates the production period. As soon as the setup is completed, items are produced one by one. During the production period as during the non-production period, removed kanbans are transmitted to the production facility whenever the items are delivered to the customers. When there are no kanbans at the production facility, the machine is shut off and a non-production period begins. The non-production period lasts until the number of kanbans accumulated at the production facility is raised back to r, at which moment the non-production period ends and the next production period begins, initiating another cycle.

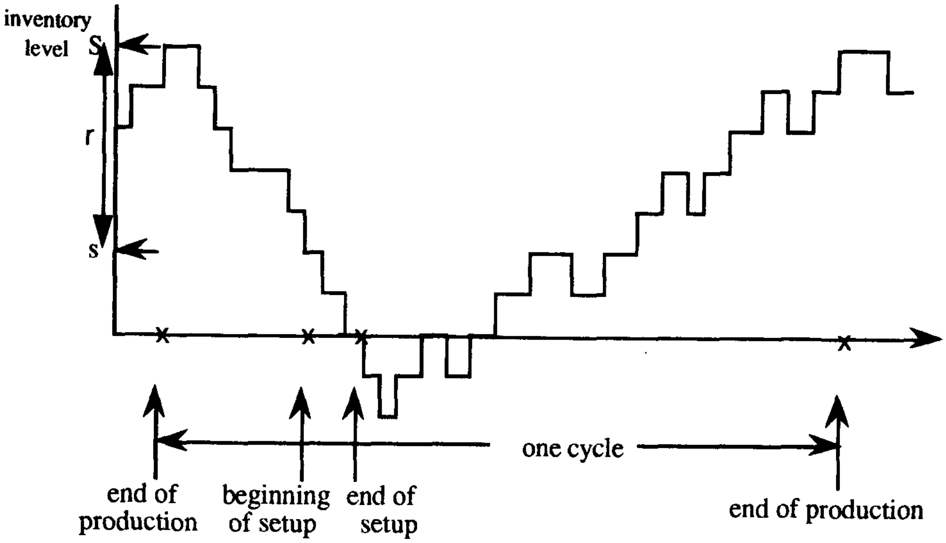

Figure 1. the (r,S) policy for a production/inventory system with setup times

Suppose that the total number of kanbans, i.e., the maximum inventory level in the output store, is S. Since our policy is specified by values of the two control variables, namely, the total number of kanbans, S, and the threshold value, r, we will call this policy the (r, S) policy in this paper. Notice that the inventory level in the output store at the beginning of the production period is always $S - r$ and the inventory level at the end of the production period is always S. From this fact it is easily noticed that if we set $s = S - r$, the (r, S) policy in this paper is just the (s, S) policy for the production/inventory system in the literature (see figure 1)

Associated with this (r, S) production/inventory system we impose the following classical cost structure: i) inventory holding costs are incurred linearly over time with respect to the inventory level ii) backorder costs are incurred linearly over time with respect to the backorder level and iii) a set-up cost is incurred each time the production facility is turned on. The objective of this study is to find an optimal (r, S) policy which minimizes the expected cost per unit time in the long run.

The (r, S) policy in this paper is particularly useful in the situations where the production facility is used for some secondary work during a non-production period. For instance, when the production period ends, the production facility is switched to a secondary mode of operation and is used to produce items of another type. The production facility could be used to repair failed items or to rework defective items during the non-production period. In some cases, the regular maintenance work is performed during the non-production period. Since significant startup costs are usually incurred to switch the machine from one operation mode to another, the (r, S) policy would be very effectively used in all these situations. Other applications of the (r, S) policy could be found, for example, in multiproduct scheduling problems or production schedule in the chemical industry (see Gavish and Graves[5]).

Another type of policy which could be used to avoid the frequent setup of the machine is the (s, Q) policy[1]. In the (s, Q) policy, the total number of kanbans is $s + Q$ and the kanbans removed from the items at the instant of demand arrivals are kept in the output store until the number of kanbans reaches Q. Whenever Q kanbans are accumulated, they are sent to the production facility immediately. At the production facility, production is performed so long as it has kanbans. When there are no kanbans at the production facility, the machine is shut off and a non-production period begins. The non-production period continues until Q kanbans are transmitted from the output store to the production facility. As soon as the production facility receives Q kanbans, it is setup and the production period begins. Note that the inventory level at the end of each non-production period is always s in the (s, Q) policy.

Therefore, in the (s, Q) policy, if more than Q demands arrive during the production of Q items, the production period does not terminate as soon as the production of these Q items are completed. In this case, since the production facility receives another Q kanbans while producing the first Q items, it produces at least another batch of Q items. Hence the number of items produced in one production period is always a multiple of Q. It should be noticed that the (s, Q) policy is just the traditional production/inventory policy with fixed reproduction point s and production quantity Q.

It appears that the (s, Q) policy is a more planned policy compared to the (r, S) policy in that it has a predetermined production batch size Q. However, when the demand rate is very high, it is probable that more than one batches are produced in one production period. This fact indicates that the length of the production period is strongly dependent on the demand process in the (s, Q) policy. In addition, in the (s, Q) policy, since the inventory level at the end of the production period is not fixed, the variation of the length of the non-production period is very high compared to the (r, S) policy. Therefore, the (s, Q) policy is not suitable for the situations where the non-production period is used for some secondary work. For this reason we analyze in this study only the (r, S) policy.

The operating policy considered in this paper is a kind of the (s, S) policy. The (s, S) policies are known to be optimal in a variety of inventory situations where as many inventories as needed can be replenished all at once([8], [11]). However, the (r, S) policy considered in this study is different from these (s, S) policies in that the replenishment of the inventory can be done only on an item-by-item basis. Since the replenishment is made on an item-by-item basis, it is easy to note that our model has an analogy with an $M/G/1$ queueing system. In fact, when the upper control value S is set to zero, our (r, S) policy becomes the Heyman's N-policy[6]. Recently for this special case where the item is not produced to inventory as in job-shop systems, Federgruen and So proved that the $(r, 0)$ policy is optimal among all policies under very general assumptions[2].

While much work has been done on (s, S) inventory policies, only limited work has been done on (r, S) policies in which inventory is replenished on an item-by-item basis. The first model for this type of production/inventory system was introduced by Tijms [10]. He considers a system with unit Poisson demands, general processing times and general setup times. He finds the optimal control values by using a denumerable state semi-Markov decision process. Gavish and Graves[4] consider a system with Poisson demands and deterministic processing times. In a subsequent paper, Gavish and Graves[5] extend the analysis to consider general processing times. Srinivasan and Lee[9] consider a more general model with compound Poisson demands, general processing times and with random inspection intervals. They develop an efficient search procedure to find the optimal policy. Federgruen and Zheng[3] consider a similar model with a more general cost structure. In their study, they derive an efficient algorithm for determination of the optimal policy.

The production/inventory model considered in this paper is rather simple compared with Srinivasan and Lee's model or Federgruen and Zheng's model. In their models, however, setup times to turn the production facility on are not considered. Thus properties of the cost functions developed in their studies are not guaranteed to hold for the systems with setup times. Tijms[10] and Gavish and Graves[5] consider models with setup times. However, a semi-Markov decision process approach used by Tijms is reported (Gavish and Graves[4]) to take too much computational time compared to an intelligent search technique. Gavish and Graves use an efficient search procedure to find an optimal policy when setup times do not exist. In this search procedure some properties for cost functions are effectively exploited. But, when setup times exist, such an intelligent search procedure is not presented since no special properties for cost functions are proved in this case.

In this paper, we analyze the same (r, S) inventory system as the one considered by Tijms or Gavish and Graves. However, for the system with setup times, we characterize the cost functions by deriving a set of properties these cost functions possess. Then by exploiting these properties effectively, we develop an extremely efficient search procedure to find an optimal policy.

2. Analysis

Let

$V(t)$ = distribution of the setup time,

$\bar{v}, v^{(2)}$ = mean and the second moment of the setup time,

b_j = probability that number of demands which arrive during a setup time is $j, j = 0, 1, 2, \cdots,$

$U(t)$ = distribution of the processing time to produce a unit item,

$\bar{u}, u^{(2)}$ = mean and the second moment of the processing time,

q_j = probability that number of demands which arrive during one processing time is $j, j = 0, 1, 2, \cdots$,
K = setup cost
C_h = holding cost/item/unit time,
C_b = backorder cost/item/unit time,
$C(r, S)$ = sum of expected holding and backorder costs during a cycle when r and S are used as control values,
$L(r)$ = expected length of a cycle when r is used as a control value,
$TC(r, S)$ = expected cost per unit time when r and S are used as control values.

Note that we use $L(r)$ instead of $L(r, S)$ to denote the expected cycle time when r and S are used as control values. This is because the expected cycle time is a function of r alone. For stability of the system, we assume that the production rate is greater than the demand rate, that is, $\rho = \lambda \bar{u} < 1$.

Our first objective in this paper is to obtain an expression for $TC(r, S)$, the expected cost per unit time for given control values r and S. The approach used to obtain an expression for $TC(r, S)$ is based on that used in Srinivasan and Lee [9]. Let the time interval from the beginning epoch of a non-production period to the beginning epoch of the subsequent non-production period be defined as a cycle. Then from the renewal reward theorem [7], the expected cost per unit time when r and S are used as control values, is obtained by

$$TC(r, S) = \frac{K + C(r, S)}{L(r)}. \tag{2.1}$$

To obtain $C(r, S)$ and $L(r)$, we divide a cycle into the following three sub-periods:

(i) Period 1 begins when the facility is turned off and lasts until the inventory level drops to $S - r$ for the first time.

(ii) Period 2 begins at the end of period 1 (beginning epoch of a setup time) and lasts until the inventory level is raised back to $S - r$ for the first time.

(iii) Period 3 begins at the end of period 2 and lasts until the inventory level reaches S for the first time.

Let $C_1(r, S)$ denote the expected cost incurred during period 1 and $C_3(r, S)$ denote the expected cost incurred during period 3. Note that the expected cost incurred during period 2 is completely determined by the inventory level at the beginning epoch of period 2. Let $\tilde{C}_k$ denote the expected cost incurred during period 2 which is initiated with k items in inventory. Then, since the inventory level at the beginning epoch of period 2 when r and S are used as control values is $S - r$, the expected cost incurred during period 2 is just $\tilde{C}_{S-r}$. Therefore, the sum of the expected holding and backorder costs during a cycle is expressed as

$$C(r, S) = C_1(r, S) + C_3(r, S) + \tilde{C}_{S-r}. \tag{2.2}$$

We now show how the terms $C_1(r, S), C_3(r, S)$ and $\tilde{C}_{S-r}$ are obtained.

Computing the term $C_1(r, S)$

The expected total cost during period 1 is easy to calculate. Let $g_{k,k-1}$ denote the expected cost incurred from the epoch at which the inventory level becomes k to the epoch when the inventory level drops to $k - 1$ during a non-production period. If $k \geq 0$, the cost for carrying k items is incurred until the next demand arrives and if $k < 0$, the cost for backlogging $-k$ items is incurred until the next demand arrives. Thus $g_{k,k-1}$ is given by

$$(2.3a) \qquad g_{k,k-1} = \frac{C_h}{\lambda}k, \quad \text{if } k \geq 0,$$

$$(2.3b) \qquad -\frac{C_b}{\lambda}k, \quad \text{if } k < 0.$$

The expected cost during period 1 when r and S are used as control values is then expressed by

$$(2.4) \qquad C_1(r,S) = \sum_{k=S-r+1}^{S} g_{k,k-1}.$$

Computing the term $C_3(r,S)$

During the production period, the production completion epochs are the times at which the inventory is replenished. To compute $C_3(r,S)$, we restrict our attention only to these epochs. Let $f_{i,j}$ denote the expected cost from the epoch at which the inventory level reaches i to the epoch at which the inventory level is raised to $j (j \geq i)$ for the first time with $f_{i,i} = 0$ for any integer i. Then the expected total cost incurred during period 3, $C_3(r,S)$, is just $f_{S-r,S}$ which, in turn, is expressed as

$$(2.5) \qquad C_3(r,S) = f_{S-r,S} = \sum_{k=S-r}^{S-1} f_{k,k+1}.$$

From equation (2.5), we see that the term $C_3(r,S)$ is obtained if we can calculate each value of $f_{k,k+1}$.
If we set

$$\Delta f_k = f_{k,k+1} - f_{k-1,k}, \quad \text{for all } k,$$

$f_{k,k+1}$ is computed recursively using the following lemma. The proof of lemma 2.1 is given in the appendix.

Lemma 2.1

The term $f_{k,k+1}$ is expressed recursively as

(2.6.a)

$$f_{k,k+1} = f_{k-1,k} - \frac{\bar{u}}{(1-\rho)}C_b \quad , \quad \text{for } k \leq 0,$$

$$(2.6.b) \quad f_{k-1,k} + \frac{1}{q_o}\{\Delta f_{k-1} + \frac{1}{\lambda}\sum_{j=k}^{\infty} q_j(C_h + C_b) - \sum_{j=1}^{k} q_j \Delta f_{k-j} + \frac{\bar{u}}{1-\rho}C_b \sum_{j=k+1}^{\infty} q_j\}, k > 0,$$

with an initial value

$$f_{-1,0} = \frac{C_b}{(1-\rho)}\{\frac{\lambda u^{(2)}}{2(1-\rho)} + \bar{u}\},$$

where $\rho = \lambda \bar{u}$.

Computing the term $\tilde{C}_{S-r}$

Let

$$\tau_k = g_{k+1,k} + f_{k,k+1}, \quad \text{for all } k. \tag{2.7}$$

Then, using equations (2.2), (2.4), (2.5) and (2.7), $C(r,S)$ is expressed as

$$C(r,S) = \sum_{k=S-r}^{S-1} \tau_k + \tilde{C}_{S-r}. \tag{2.8}$$

To compute the term $\tilde{C}_{S-r}$, we need the following lemma.

Lemma 2.2

The term $\tilde{C}_k$ is expressed recursively as

$$\tilde{C}_k = \tilde{C}_{k-1} + \sum_{j=0}^{\infty} b_j(\tau_{k-1} - \tau_{k-1-j}), \quad \text{for all } k. \tag{2.9}$$

Proof Let ξ_κ denote the expected cost incurred during a setup time that is initiated with k items in inventory. Then $\tilde{C}_k$ is expressed as

$$\tilde{C}_k = \xi_k + \sum_{j=1}^{\infty} b_j f_{k-j,k}. \tag{2.10}$$

From equation (2.10), $\tilde{C}_k - \tilde{C}_{k-1}$ is expressed as

$$\begin{aligned}\tilde{C}_k - \tilde{C}_{k-1} &= \xi_k - \xi_{k-1} + \sum_{j=1}^{\infty} b_j(f_{k-j,k} - f_{k-1-j,k-1}) \\ &= \xi_k - \xi_{k-1} + \sum_{j=1}^{\infty} b_j(f_{k-1,k} - f_{k-1-j,k-j}) \\ &= \xi_k - \xi_{k-1} + \sum_{j=1}^{\infty} b_j(\tau_{k-1} - \tau_{k-1-j} + g_{k-j,k-1-j} - g_{k,k-1}).\end{aligned} \tag{2.11}$$

The term ξ_k in equation (2.11) is obtained as follows. Let Ψ_n denote the expected cost incurred per unit time when the inventory level is n. Suppose the length of the setup time is x and n items are demanded during x. Given n demand arrivals during x, the joint distribution of these arrival epochs have the same distribution as the order statistics of n independent random variables uniformly distributed on $[0,x]$ (see, for example, Ross 1970). From this fact, the expected cost incurred during x is expressed as $\sum_{j=0}^{n} \frac{x}{n+1}\Psi_{k-j}$. By unconditioning number of demand arrivals and length of the setup time, ξ_k is expressed as

$$\begin{aligned}\xi_k &= \int_0^\infty \sum_{n=0}^{\infty} \frac{e^{-\lambda x}(\lambda x)^n}{n!} \sum_{j=0}^{n} \frac{x}{n+1}\Psi_{k-j} dV(x) \\ &= \sum_{n=0}^{\infty} \int_0^\infty \frac{e^{-\lambda x}(\lambda x)^{n+1}}{(n+1)!} dV(x) \sum_{j=0}^{n} \frac{\Psi_{k-j}}{\lambda} \\ &= \sum_{n=0}^{\infty} b_{n+1} \sum_{j=0}^{n} g_{k-j,k-j-1}.\end{aligned}$$

Hence, $\xi_k - \xi_{k-1}$ is expressed as

$$\begin{aligned}\xi_k - \xi_{k-1} &= \sum_{n=0}^{\infty} b_{n+1} \sum_{j=0}^{n} (g_{k-j,k-j-1} - g_{k-j-1,k-j-2}) \\ &= \sum_{j=1}^{\infty} b_j (g_{k,k-1} - g_{k-j,k-j-1}). \end{aligned} \tag{2.12}$$

Substituting equation (2.12) into equation (2.11) yields lemma 2.2. □

Lemma 2.2 cannot be used for a computational purpose since there is an infinite number of terms in equation (2.9). However, as will be seen later, lemma 2.2 will play an important role to characterize the cost functions. To compute $\tilde{C}_k$, we use the following alternative equation which is stated as lemma 2.3. The proof of lemma 2.3 is given in the appendix.

Lemma 2.3

The term $\tilde{C}_k$ is obtained from

$$\tilde{C}_k = \tilde{C}_{k-1} - C_b \frac{\bar{v}}{(1-\rho)}, \quad k \leq 0. \tag{2.13.a}$$

$$\tilde{C}_{k-1} + \tau_{k-1} - \sum_{j=0}^{k-1} b_j \tau_{k-1-j} - \tau_{-1}(1 - \sum_{j=0}^{k-1} b_j) - \frac{C_b}{(1-\rho)\lambda}\{\lambda\bar{v} - k + \sum_{j=0}^{k-1}(k-j)b_j\}, k > 0, \tag{2.13.b}$$

with an initial boundary value,

$$\tilde{C}_0 = \frac{\lambda v^{(2)}}{2} C_b + \frac{\rho\lambda v^{(2)} + 2\rho\bar{v}}{2(1-\rho)} C_b + \frac{\lambda^2 \bar{v} u^{(2)}}{2(1-\rho)^2} C_b. \tag{2.13.c}$$

Computing the term $L(r)$

If setup times do not exist, the expected length of a cycle for (r, S) system is given by $\frac{r}{(1-\rho)\lambda}$ [9]. When setup times exist, the expected length of period 2 should be added to this term. The expected length of period 2 is obtained easily using the analogy between the production/inventory system and the queueing system as follows. Suppose j items are demanded during a setup time. Then, the inventory level at the end of the setup time is $S - r - j$. Note that from this epoch until the instant the inventory level is first raised back to $S - r$ is just the convolution of j independent busy periods in an $M/G/1$ queueing system. Therefore, from the busy period analysis of an $M/G/1$ system, the expected length from the end of the setup time until the end of period 2 is $\frac{j\bar{u}}{1-\rho}$. Since the probability that j items are demanded during a setup time is b_j, the expected length of period 2, L_2, is given by

$$\begin{aligned} L_2 &= \bar{v} + \sum_{j=0}^{\infty} b_j j \frac{\bar{u}}{(1-\rho)} \\ &= \frac{\bar{v}}{1-\rho}. \end{aligned} \tag{2.14}$$

The expected length of a cycle is, therefore, expressed as

$$\begin{aligned} L(r) &= \frac{r}{(1-\rho)\lambda} + \frac{\bar{v}}{1-\rho} \\ &= \frac{r+\lambda\bar{v}}{(1-\rho)\lambda}. \end{aligned} \tag{2.15}$$

Hence, from equations (2.1), (2.8) and (2.15), we have

$$TC(r,S) = (1-\rho)\lambda \frac{K + \sum_{k=S-r}^{S-1} \tau_k + \tilde{C}_{S-r}}{r+\lambda\bar{v}}. \tag{2.16}$$

3. The Optimal Control Values

To find the optimal policy $(r^*, S^*), TC(r, s)$ must be minimized over the two-dimensional integer parameter space. This search could be performed quite efficiently if we exploit some properties of the cost functions. Let us denote by $S^*(r)$ the optimal S value for a given r. We now demonstrate some properties that are possessed by the cost functions.

Property 1

τ_κ is convex with respect to k.

Proof The proof is given in Srinivasan and Lee [9]. □

Property 2

$\tilde{C}_k$ is convex with respect to k.

Proof Define $\Delta\tilde{C}_k = \tilde{C}_k - \tilde{C}_{k-1}$ and $\Delta\tau_k = \tau_k - \tau_{k-1}$. To show the convexity of $\tilde{C}_k$, it is sufficient to show that $\Delta\tilde{C}_k - \Delta\tilde{C}_{k-1} \geq 0$ for all k, which can be proved using lemma 2.1 as well as the convexity of τ_κ as follows:

$$\Delta\tilde{C}_k - \Delta\tilde{C}_{k-1} = \sum_{j=0}^{\infty} b_j(\Delta\tau_{k-1} - \Delta\tau_{k-1-j}) \geq 0, \quad \text{for all } k.$$

□

Property 3

For a given value of $r, C(r, S)$ is convex with respect to S.

Proof Since $C(r,S) = \sum_{k=S-r}^{S-1} \tau_k + \tilde{C}_{S-r}, C(r,S)$ is convex if $\sum_{k=S-r}^{S-1} \tau_k$ and $\tilde{C}_{S-r}$ are both convex. From properties 1 and 2, both of these terms can be shown to be convex with respect to S for a given value of r. □

Property 4

If $S^*(r) = k$, then $S^*(r+1) \leq k+1$.

Proof From the given condition $S^*(r) = k$, we have

$$C(r, K+1) - C(r, K) = \tilde{C}_{k-r+1} - \tilde{C}_{k-r} + \tau_k - \tau_{k-r} \geq 0. \tag{3.1}$$

Using lemma 2.1, inequality (3.1) can be rewritten as

$$\tau_k \geq \sum_{j=0}^{\infty} b_j \tau_{k-r-j}. \tag{3.2}$$

Since τ_κ is convex, inequality (3.2) implies that

$$\tau_\kappa \leq \tau_{\kappa+1} \leq \tau_{\kappa+2} \leq \cdots. \tag{3.3}$$

To show $S^*(r+1) \leq k+1$, it is enough to show that $C(r+1, k+2) \geq C(r+1, k+1)$, which can be proved from inequality (3.2) as well as the fact that $\tau_{\kappa+1} \geq \tau_\kappa$ as follows:

$$C(r+1, k+2) - C(r+1, k+1) = \tau_{k-1} - \sum_{j=0}^{\infty} b_j \tau_{k-r-j} \geq 0.$$

□

Property 5

If $S^*(n+1) > S^*(n)$, then $S^*(r+1)$ is either $S^*(r)$ or $S^*(r)+1$ for $r > n$.

Proof

Suppose $S^*(n) = k$ and $S^*(n+1) = k+1$. Then, from $C(n, k+1) \geq C(n, k)$, we have

$$\sum_{j=0}^{\infty} b_j \tau_{k-n-j} \leq \tau_k. \tag{3.4}$$

Similarly, from $C(n+1, k+1) \leq C(n+1, k)$, we have

$$\sum_{j=0}^{\infty} b_j \tau_{k-1-n-j} \geq \tau_k. \tag{3.5}$$

From inequalities (3.4) and (3.5), the following relationship is obtained:

$$\sum_{j=0}^{\infty} b_j \tau_{k-1-n-j} \geq \sum_{j=0}^{\infty} b_j \tau_{k-n-j}. \tag{3.6}$$

Inequality (3.6) and the convexity of $\sum_{j=0}^{\infty} b_j \tau_{k-j}$ imply that

$$\sum_{j=0}^{\infty} b_j \tau_{k-i-1-n-j} \geq \sum_{j=0}^{\infty} b_j \tau_{k-i-n-j} \text{ for } i \geq 0. \tag{3.7}$$

Suppose $S^*(r) = m$ for $r \geq n+2$. Then, from property 4, the inequality $k - n \geq m - r$ should be satisfied. Also from $C(r, m) \leq C(r, m-1)$, we have

$$\sum_{j=0}^{\infty} b_j \tau_{m-1-r-j} \geq \tau_{m-1}. \tag{3.8}$$

To prove property 5, we need to show that $C(r+1,m) \leq C(r+1,m-1)$, which can be written as

$$\sum_{j=0}^{\infty} b_j \tau_{m-2-r-j} \geq \tau_{m-1}. \tag{3.9}$$

From inequality (3.7), $k - n \geq m - r$. Thus, we have

$$\sum_{j=0}^{\infty} b_j \tau_{m-2-r-j} \geq \sum_{j=0}^{\infty} b_j \tau_{m-1-r-j}. \tag{3.10}$$

Inequality (3.10) together with inequality (3.8) implies that inequality (3.9) is true. □

Property 6

If $S^*(n+1) > S^*(n)$, then $TC(r, S^*(r))$ is unimodal in r for $r \geq n$.

Proof Suppose $S^*(r) = m$ for some $r > n$. To prove property 6, we first need to show that

$$C(r+2,S^*(r+2))-C(r+1,S^*(r+1)) \geq C(r+1,S^*(r+1))-C(r,S^*(r)), \text{ for } r \geq n. \tag{3.11}$$

From property 5, only the following 4 cases can happen. Inequality (3.11) can be proved for each case as follows:

i) case 1 : $S^*(r+1) = m+1,\ S^*(r+2) = m+2$
$\{C(r+2,m+2) - C(r+1,m+1)\} - \{C(r+1,m+1) - C(r,m)\} = \tau_{m+1} - \tau_m \geq 0$ since, as shown in property 4, if $S^*(r) = m$ then $\tau_{m+1} \geq \tau_m$.

ii) case 2 : $S^*(r+1) = m+1,\ S^*(r+2) = m+1$
$\{C(r+2,m+1) - C(r+1,m)\} - \{C(r+1,m) - C(r,m)\} = C(r+1,m) - C(r+1,m+1) \geq 0$.

iii) case 3 : $S^*(r+1) = m,\ S^*(r+2) = m+1$
$\{C(r+2,m+1) - C(r+1,m)\} - \{C(r+1,m) - C(r,m)\} = C(r+1,m+1) - C(r+1,m) \geq 0$.

iv) case 4 : $S^*(r+1) = m,\ S^*(r+2) = m$
$\{C(r+2,m) - C(r+1,m)\} - \{C(r+1,m) - C(r,m)\} = \sum_{j=0}^{\infty} b_j(\tau_{m-r-2-j} - \tau_{m-r-1-j}) \geq 0$ since, as shown in inequality (3.10), $\sum_{j=0}^{\infty} b_j(\tau_{m-r-2-j} - \tau_{m-r-1-j}) \geq 0$.

We are now ready to prove the unimodality of $TC(r, S^*(r))$. To this end we only need to show that

$$\text{if } \frac{K + C(r,S^*(r))}{L(r)} \leq \frac{K + C(r+1,S^*(r+1))}{L(r+1)}, \text{ then}$$
$$\frac{K + C(r+1,S^*(r+1))}{L(r+1)} \leq \frac{K + C(r+2,S^*(r+2))}{L(r+2)}. \tag{3.12}$$

Since $L(r+1)-L(r)=\dfrac{1}{(1-\rho)\lambda}$ for all r, from inequality (3.11), we have

$$\frac{C(r+2,S^*(r+2))-C(r+1,S^*(r+1))}{L(r+2)-L(r+1)}\geq\frac{C(r+1,S^*(r+1))-C(r,S^*(r))}{L(r+1)-L(r)}. \tag{3.13}$$

For positive values of a, b, c and d with $\frac{a}{b}\geq\frac{c}{d}$, it is true that if $a > c$ and $b > d$, then $\frac{a-c}{b-d}\geq\frac{c}{d}$. If we apply this algebraic fact to the given condition $\frac{K+C(r+1,S^*(r+1))}{L(r+1)}\geq\frac{K+C(r,S^*(r))}{L(r)}$, we obtain

$$\frac{C(r+1,S^*(r+1))-C(r,S^*(r))}{L(r+1)-L(r)}\geq\frac{K+C(r,S^*(r))}{L(r)}. \tag{3.14}$$

Collecting (3,13) and (3.14) yields

$$\begin{aligned}\frac{C(r+2,S^*(r+2))-C(r+1,S^*(r+1))}{L(r+2)-L(r+1)}&\geq\frac{C(r+1,S^*(r+1))-C(r,S^*(r))}{L(r+1)-L(r)}\\&\geq\frac{K+C(r,S^*(r))}{L(r)}.\end{aligned} \tag{3.15}$$

We now use another algebraic fact that if a, b, c, d, e, f are all positive, then $\frac{a+c+e}{b+d+f}\geq\frac{c+e}{d+f}$ holds if $\frac{a}{b}\geq\frac{c}{d}\geq\frac{e}{f}$. Applying this algebraic fact to inequality (3.15), we finally obtain

$$\frac{K+C(r+2,S^*(r+2))}{L(r+2)}\geq\frac{K+C(r+1,S^*(r+1))}{L(r+1)}.$$

□

Property 7

Let us consider an associated production/inventory system with zero setup times, which differs from our system only by the fact that setup times are zero. Let an optimal upper control value for a given r for this system be denoted by $S_0^*(r)$. Then $S^*(r)\geq S_0^*(r)\geq 0$ for all r.

Proof

Note that for a given value of r,

$$\operatorname*{Min}_{s} C_0(r,S)=C_0(r,S_0^*(r))=\operatorname*{Min}_{n}\sum_{k=n}^{n+r-1}\tau_k.$$

Let $C_0(r,S)$ be minimized at $n=m$, that is, $S^*(r)=m+r$. To prove property 7, it is sufficient to show that

$$C(r,m+r)=\sum_{j=m}^{m+r-1}\tau_j+\tilde{C}_m\leq\sum_{j=m-1}^{m+r-2}\tau_j+\tilde{C}_{m-1}=C(r,m+r-1). \tag{3.16}$$

From $S_0^*(r) = m + r$, we know $\sum_{j=m}^{m+r-1} \tau_j \leq \sum_{j=m-1}^{m+r-2} \tau_j$. Thus, if $\tilde{C}_m \leq \tilde{C}_{m-1}$ inequality (3.16) holds. From the convexity of τ_k as well as the fact that $S_0^*(r) = m + r$, we have $\tau_m \leq \tau_{m-1} \leq \tau_{m-2} \leq \cdots$. Hence,

$$(3.17) \qquad \tilde{C}_{m-1} - \tilde{C}_m = \sum_{j=0}^{\infty} b_j(\tau_{m-1-j} - \tau_{m-1}) \geq 0.$$

□

These properties enable us to devise an efficient search procedure. Property 3 can be used in searching $S^*(r)$ for a given value of r. Once $S^*(r)$ is obtained for some $r, S^*(r)$ for a different value of r can be found readily using properties 4 and 5. Properties 5, 6 and 7 restrict considerably the range where the search should be performed. Property 5 states that once the function $S^*(r)$ increases, it never decreases. Property 6 states that in the range where $S^*(r)$ is non-decreasing, $TC(r, S^*(r))$ is unimodal with respect to r. Usually, the function $S^*(r)$ begins to increase when r is very small, namely, $r = 1, 2$ or 3. Therefore, properties 5 and 6 not only reduce the search time significantly but also guarantee that the solution found by the search procedure is a global optimum. In particular, if a setup time is exponentially distributed, we can develop more strong properties for the cost functions.

Property 8

Suppose a setup time V follows an exponential distribution, then

(a) $S^*(r+1)$ is either $S^*(r)$ or $S^*(r)+1$, for $r \geq 1$.

(b) $TC(r, S^*(r))$ is unimodal in r, for $r \geq 1$.

Proof

(a) Suppose $S^*(r) = k$. Then from the condition $S^*(r, k-1) \geq C(r, k)$, we have $\sum_{j=0}^{\infty} b_j \tau_{k-1-r-j} \geq \tau_{k-1}$. To prove property 8.a, it is sufficient to show that $C(r+1, k) \leq C(r+1, k-1)$, which can be rewritten as $\sum_{j=0}^{\infty} b_j \tau_{k-2-r-j} \geq \tau_{k-1}$. Therefore, property 8.a is true if we can prove that $\sum_{j=0}^{\infty} b_j \tau_{k-2-r-j} \geq \sum_{j=0}^{\infty} b_j \tau_{k-1-r-j}$, which is shown below. If V follows an exponential distribution, it can easily be shown that $b_j = \frac{\{\lambda \bar{v}\}^j}{\{1 + \lambda \bar{v}\}^{j+1}}$. Thus b_j is expressed recursively as $b_j = b_{j-1}(1 - b_0)$, for $j \geq 1$. Using this fact, we can express

$\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} - \sum_{j=0}^{\infty} b_j\tau_{k-1-r-j}$ as

$$\begin{aligned}\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} - \sum_{j=0}^{\infty} b_j\tau_{k-1-r-j} &= \sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} - \sum_{j=1}^{\infty} b_j\tau_{k-1-r-j} - b_0\tau_{k-r-1}\\ &= \sum_{j=0}^{\infty}(b_j - b_{j+1})\tau_{k-2-r-j} - b_0\tau_{k-r-1}\\ &= \sum_{j=0}^{\infty} b_0 b_j\tau_{k-2-r-j} - b_0\tau_{k-r-1}\\ &= b_0(\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} - \tau_{k-r-1}).\end{aligned}$$

To show $\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} \geq \tau_{k-r-1}$ we consider the following two cases: First, if $\tau_{k-2-r-j} \geq \tau_{k-r-1}$, then from the convexity of τ_k, it is obvious that $\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} \geq \tau_{k-r-1}$. On the other hand, if $\tau_{k-2-r} < \tau_{k-r-1}$, this implies that $\tau_{k-r-1} \leq \tau_{k-2}$. But, in this case, from the given condition $C(r,k-2) \geq C(r,k-1)$, we have $\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} \geq \tau_{k-2}$. Hence, the inequality $\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} \geq \tau_{k-r-1}$ is proved.

(b) Using the relationship $\sum_{j=0}^{\infty} b_j\tau_{k-2-r-j} \geq \sum_{j=0}^{\infty} b_j\tau_{k-1-r-j}$, which is shown in the proof of property 8.a, it can be easily verified that $C(r+2, S^*(r+2)) - C(r+1, S^*(r+1)) \geq C(r+1, S^*(r+1)) - C(r, S^*(r))$ for all r. Now the unimodality proof can be done in the same way as in the proof of property 6 □

While the property $S^*(r+1) = S^*(r)$ or $S^*(r)+1$ as well as the unimodality of $TC(r, S^*(r))$ does hold for all r for the case of exponential setup times, for general cases these properties are not proved to hold until a point is encountered where $S^*(r)$ is first increased. However, our extensive computational experience suggests that these properties hold even before the first increase of $S^*(r)$ occurs although we have not been able to prove these. Based on these observations as well as properties 1 through 8 we can devise an extremely efficient search procedure to find the optimal policy as follows. Our algorithm starts with $r = 1$ and find $C(1, S^*(1))$ first. To compute $C(1, S^*(1))$, we simply compute $C(1,k)$ from $k = 0$ up to the point $k = S^*(1)+1$, at which $C(1,k)$ is first increased. Then, due to the convexity of $C(1,k)$, the minimum value is found at $C(1, S^*(1))$. After we find $C(1, S^*(1))$, we compute $TC(1, S^*(1))$ using equation (2.16). Note that the search starts from $k = 0$ since by property 7, $C(1,k)$ for $k < 0$ cannot be a minimum. Besides, since the initial boundary values for the recursive computation of $f_{k,k+1}$ and $\tilde{C}_k$ are $f_{-1,0}$ and $\tilde{C}_0$ respectively, $C(1,0)$ is computed directly from these initial values. In the process of finding $C(1, S^*(1))$, values of τ_k and $\tilde{C}_k$ for $-1 \leq k \leq S^*(1)$ are computed. We store all these values because these are used repeatedly to find $S^*(r)$ for $r > 1$.

Once $C(1, S^*(1))$ is obtained, we find $C(r, S^*(r))$ (and therefore, $TC(r, S^*(r))$ sequentially in the order of $r = 2, 3, 4, \cdots$ using properties 4 and 5. Note that from property 4, we do not have to consider $C(r+1,k)$ for $k > S^*(r)+1$. Furthermore, due to property 5, once we encounter r where $S^*(r)$ is increased, from that point onwards, we have only to consider $S^*(r)$ and $S^*(r)+1$ to find $S^*(r+1)$. Thus, very few evaluations of $C(r,k)$ are, in fact, needed in order to obtain $C(r,S^*(r))$ for each r. Our algorithm repeats this process (increase r by 1 and compute $C(r,S^*(r))$ and $TC(r,S^*(r))$ for new r) until $TC(r,S^*(r))$ is first increased. Then, by property 6 as well as the unimodality observation, the local minimum point encountered can be considered as a global minimum and the optimal control values r^* and S^* are obtained. Note that, however, the solution found by this algorithm cannot be proved to be a global optimum except for the case of exponential setup times. But if we want to obtain a solution which is guaranteed to be a global optimum, it can be done simply as follows: Suppose $S^*(r)$ is increased first at $r = k$. If r^* obtained by the algorithm is equal to or greater than $k, (r^*, S^*)$ should be a global optimum by property 6. On the other hand, if r^* is less than k, a global optimum is guaranteed to be obtained simply by computing $C(r,S^*(r))$ and $TC(r,S^*(r))$ until r becomes k. In all problems we have tested, the first increase of $S^*(r)$ is observed to occur before r reaches 5. Hence, a global optimum can always be guaranteed to be obtained with very little additional computation.

Algorithm to find the Optimal Control Values (r^*, S^*)

```
0. Set $r = 1, k = 0$.
   Compute $C(1,k)$
1. Set $k = k+1$
   Compute $C(1,k)$
   If $C(1,k) > C(1,k-1)$, then
       $S^*(1) = k-1$. Compute $TC(1,S^*(1))$
       go to step 2
   else
       go step 1
   endif
2. Set $r = r+1, k = S^*(r-1)+1$
   Compute $C(r,k)$
3. Set $k = k-1$, compute $C(r,k)$
       If $C(r,k) > C(r,k+1)$, then
           $S^*(r) = k+1$. Compute $TC(r,S^*(r))$
           If $TC(r,S^*(r)) > TC(r-1,S^*(r-1))$ then
               Optimal control values have been found ; $(r-1, S^*(r-1))$.
               stop
           else
               If $S^*(r) > S^*(r-1)$, then
                   go to step 4
               else
                   go to step 2
               endif
           endif
       else
           go to step 3
       endif
4. Set $r = r+1$,
   Compute $C(r,S^*(r-1))$ and $C(r,S^*(r-1)+1)$
   If $C(r,S^*(r-1)) < C(r,S^*(r-1)+1)$, then
```

```
        $S^*(r) = S^*(r-1)$
else
        $S^*(r) = S^*(r-1) + 1$
endif
Compute $TC(r, S^*(r))$
If $TC(r, S^*(r)) > TC(r-1, S^*(r-1))$ then
    Optimal Control Values have been found; $(r-1, S^*(r-1))$
    stop
else
    go to step 4
endif
```

4. Numerical Examples

In order to verify the efficiency of our algorithm and to check whether $TC(r, S^*(r))$ is unimodal for all $r \geq 1$, we made extensive numerical tests. Among these, we present two examples below. Usually the processing time for a typical manufacturing system has a very small coefficient of variation: it is almost deterministic. However, if the facility fails during a processing time, then the time to repair the facility could be accounted for in the processing time. In the first example, we consider such a distribution. Thus, in example 1, the processing time is a sum of the actual processing time of an item, plus the repair time of the failure that might occur during the processing of the item. In example 1, we assume that the actual processing time has a constant value, 3. The probability that the machine fails during a processing of an item is 0.05. The repair time to fix the machine follows an exponential distribution with mean 10. The setup time is assumed to have a deterministic value 20. Other parameter values are given as $\lambda = 0.1$, $K = 500, C_h = 1$ and $C_b = 10$. In example 2, the setup time is assumed to follow an exponential distribution with mean 20 and the processing time is assumed to follow a uniform distribution in the range [8,10]. Other parameter values are given as $\lambda = 0.1$, $K = 500$, $C_h = 1$ and $C_b = 30$. Note that example 2, compared to example 1, represents the situation where the utilization of the production facility is very high and backorder cost is high. The results of the policy comparisons for these two test examples are presented in tables 1 and 2. In each table, to show the behavior of the cost functions, the values of $r, s^*(r), S^*(r)$ and $TC(r, S^*(r))$ are given for each value of r. The optimal policy found by the algorithm for example 1 is $r^* = 7$ and $S^* = 9$. This policy is a global optimum because $S^*(r)$ increases at $r = 1$, while r^* is 7. The policy, $r^* = 5$ and $S^* = 21$, obtained for example 2 is also a global optimum because the setup time follows an exponential distribution. Although only two distributions for the setup time are demonstrated in the examples, many other distributions can be implemented easily because b_j can be expressed in closed form for many distributions of practical interest.

5. Conclusions

We have introduced the (r, S) control policy for production/inventory systems where items are produced on an item by item basis with completed items going directly into inventory. The (r, S) policy considered in this paper is a pull type policy which is very useful in situations where the production facility is used for some secondary work during a non-production period. In this study, we assume that processing time to produce an item follows a general distribution and each time production is initiated a random amount of setup time is taken. For this system, under a linear cost structure, we obtained an expression for the expected cost per unit time for given control values and developed an extremely simple, yet efficient search procedure to find the optimal control values. In devising search procedures, unimodality property as well as the recursive nature of the cost functions was effectively exploited. Although proofs of optimality were presented only for the system with exponential setup times, a solution guaranteed to be a global optimum can be always obtained with very little additional computation.

Table 1.
Result of Example 1

r	s*(r)	S*(r)	TC(r,S*(r))
1	3	4	14.303
2	3	5	11.872
3	2	5	10.595
4	2	6	9.751
5	2	7	9.288
6	2	8	9.063
*7	2	9	9.000
8	1	9	9.043
9	1	10	9.084
10	1	11	9.200
11	0	11	9.736

(* indicates the optimal policy)

Table 2.
Result of Example 2

r	s*(r)	S*(r)	TC(r,S*(r))
1	19	20	19.301
2	18	20	18.897
3	17	20	18.711
4	17	21	18.604
*5	16	21	18.596
6	16	22	18.608
7	16	23	18.694

(* indicates the optimal policy)

Acknowledgment

The author is grateful to the anonymous referees for their helpful and constructive comments. This study was partially supported by grants from the Korea Research Foundation in 1991.

Appendix: proof of lemmas 2.1 and 2.3

Lemma 2.1

The term $f_{k,k+1}$ is expressed recursively as

$$f_{k,k+1} = f_{k-1,k} - \frac{\bar{u}}{(1-\rho)}C_b \quad , \quad \text{for } k \leq 0,$$

$$f_{k-1,k} + \frac{1}{q_0}\{\Delta f_{k-1} + \frac{1}{\lambda}\sum_{j=k}^{\infty} q_j(C_h + C_b) - \sum_{j=1}^{k} q_j \Delta f_{k-j} + \frac{\bar{u}}{1-\rho}C_b \sum_{j=k+1}^{\infty} q_j\}, k > 0,$$

with an initial value

$$f_{-1,0} = \frac{C_b}{(1-\rho)}\{\frac{\lambda u^{(2)}}{2(1-\rho)} + \bar{u}\}, \text{ where } \rho = \lambda\bar{u}.$$

Proof

let E_k denote the expected cost incurred ***during a processing time*** that is initiated with k items in inventory. Then $f_{k,k+1}$ is expressed as

$$\text{(A.1)} \qquad f_{k,k+1} = E_k + \sum_{j=1}^{\infty} q_j f_{k+1-j,k+1}.$$

The term E_k in equation (A.1) is obtained by the following equation (The proof can be found in Srinivasan and Lee[9]).

If $k \geq 0$,

$$\text{(A.2.a)} \qquad E_{k+1} = E_k + h_{k+1}(c_h + c_b) - \bar{u}c_b,$$

where

$$h_{k+1} = h_k + \frac{1}{\lambda}(1 - \sum_{j=0}^{k} q_j), \text{ with } h_k = 0 \text{ for } k \leq 0,$$

if $k < 0$,

$$\text{(A.2.b)} \qquad E_k = E_{k+1} + \bar{u}c_b,$$

with

$$E_0 = \frac{\lambda u^{(2)}}{2} c_b.$$

Let us define $\Delta E_k = E_k - E_{k-1}$. Then, from equation (A.1), we obtain

$$\text{(A.3)} \qquad \Delta f_k = \Delta E_k + \sum_{j=1}^{\infty} q_j(f_{k+1-j,k+1} + f_{k-j,k}).$$

After a little algebra using equation (A.3), $\Delta f_k - \Delta f_{k-1}$ is expressed as

$$\Delta f_k - \Delta f_{k-1} = \Delta E_k - \Delta E_{k-1} + \sum_{j=1}^{\infty} q_j\{(f_{k+1-j,k+1} - f_{k-j,k}) - (f_{k-j,k} - f_{k-1-j,k-1})\}$$

$$\text{(A.4)} \qquad = \Delta E_k - \Delta E_{k-1} + \sum_{j=1}^{\infty} q_j(\Delta f_k - \Delta f_{k-j}).$$

Note that the last term in equation (A.4) consists of infinite terms. However, $\Delta f_k - \Delta f_{k-1}$ can be simplified without these infinite terms as described below.

Let D_k denote the time period from the epoch when the inventory level reaches k to the epoch when the inventory level is raised to $k+1$ for the first time. Note that the length of D_k is equivalent to one busy period in an $M/G/1$ queueing system, hence, from the well known busy period analysis, the expected length of D_k is $\bar{u}/(1-\rho)$ where $\rho = \lambda\bar{u}$. If we compare the inventory level during D_k and the inventory level during D_{k+1}, the inventory level during D_k has the same stochastic path as the inventory level during D_{k+1} if one item of inventory is added to the inventory level during D_k throughout this period. Consequently, if $k < 0$, the inventory level during D_k has only one more shortage than the inventory level during D_{k+1} on the average. Thus, for $k < 0$,

$$\text{(A.5)} \qquad \Delta f_k = f_{k,k+1} - f_{k-1,k} = -\frac{\bar{u}}{(1-\rho)}c_b.$$

If we solve equation (A.4) in terms of Δf_k and substitute $\Delta f_k = -\frac{\bar{u}}{(1-\rho)}c_b$ for $k < 0$, we obtain the following recursive equation:

$$\text{(A.6)} \qquad \Delta f_k = \frac{1}{q_0}\{\Delta f_{k-1} + \Delta E_k - \Delta E_{k-1} - \sum_{j=1}^{k} q_j \Delta f_{k-j} - (1 - \sum_{j=1}^{k} q_j)\frac{\bar{u}}{(1-\rho)}c_b\},$$

where from equation (A.2),

$$\text{(A.7)} \qquad \Delta E_k - \Delta E_{k-1} = (h_k - h_{k-1})(c_b - c_h) = \frac{1}{\lambda}(1 - \sum_{j=0}^{k-1} q_j)(c_h + c_b).$$

Now, from equation (A.6), the term Δf_k for $k \geq 0$, can be computed recursively using the initial value $\Delta f_{-1} = -\frac{\bar{u}}{(1-\rho)}c_b$, and hence, Δf_k for any k can be computed. From this, $f_{k,k+1}$ is obtained recursively as

$$\begin{aligned} f_{k,k+1} = f_{k-1,k} + \frac{1}{q_0}\{\Delta f_{k-1} + \frac{1}{\lambda}(1 - \sum_{j=0}^{k-1} q_j)(c_h + c_b) - \sum_{j=1}^{k} q_j \Delta f_{k-j} \\ - (1 - \sum_{j=1}^{k} q_j)\frac{\bar{u}}{(1-\rho)}c_b\}. \end{aligned}$$

The initial value $f_{-1,0}$ can be obtained directly from the fact that $f_{-1,0}$ is just the expected total cost incurred during the busy period in an associated $M/G/1$ queueing system where the unit waiting cost per customer is c_b.

Lemma 2.3

The term $\tilde{C}_k$ is obtained from

$$\tilde{C}_k = \begin{cases} \tilde{C}_{k-1} - C_b\frac{\bar{v}}{(1-\rho)}, & k \leq 0, \\ \tilde{C}_{k-1} + \tau_{k-1} - \sum_{j=0}^{k-1} b_j \tau_{k-1-j} - \tau_{-1}(1 - \sum_{j=0}^{k-1} b_j) - \frac{C_b}{(1-\rho)\lambda}\{\lambda\bar{v} - k + \sum_{j=0}^{k-1}(k-j)b_j\}, & k > 0, \end{cases}$$

with an initial value,

$$\tilde{C}_0 = \frac{\lambda v^{(2)}}{2}C_b + \frac{\rho\lambda v^{(2)} + 2\rho\bar{v}}{2(1-\rho)}C_b + \frac{\lambda^2\bar{v}u^{(2)}}{2(1-\rho)^2}C_b.$$

Proof Case $k \leq 0$:

Note that the expected length of period 2 initiated with k items in inventory is $\frac{\bar{v}}{1-\rho}$ (refer to equation (2.14)). We use the same argument as we used in the proof of lemma 2.1. Then by comparing the stochastic path of the inventory level during period 2 initiated with k inventories with that of the inventory level during period 2 initiated with $k-1$ inventories, $\tilde{C}_k$ is expressed as $\tilde{C}_k = \tilde{C}_{k-1} - C_b\frac{\bar{v}}{(1-\rho)}$.

Case $k > 0$:

$$\begin{aligned} \tilde{C}_k &= \tilde{C}_{k-1} + \tau_{k-1} - \sum_{j=0}^{\infty} b_j\tau_{k-1-j} \\ &= \tilde{C}_{k-1} + \tau_{k-1} - \sum_{j=0}^{k-1} b_j\tau_{k-1-j} - \sum_{j=k}^{\infty} b_j\tau_{k-1-j}. \end{aligned} \tag{A.8}$$

The last term in equation (A.8) consists of infinite terms, which can be handled as follows. Let T_k denote the time period during which the expected cost of τ_k is incurred. Then, there is a relationship that the length of T_k is equivalent to one cycle in an associated $M/G/1$ queueing system. Hence, from the known result of an $M/G/1$ system, the expected length of T_k is $\frac{1}{(1-\rho)\lambda}$. Using this fact, we can express τ_k for $k < 0$ as a function of τ_{-1} as follows:

$$\tau_k = \tau_{-1} - (k+1)\frac{1}{(1-\rho)\lambda}C_b, \text{ for } k < 0. \tag{A.9}$$

Now the infinite terms in equation (A.9) can be expressed as

$$\begin{aligned} \sum_{j=k}^{\infty} b_j\tau_{k-1-j} &= \sum_{j=k}^{\infty} b_j\{\tau_{-1} + (j-k)\frac{c_b}{(1-\rho)\lambda}\} \\ &= \tau_{-1}\sum_{j=k}^{\infty} b_j + \{\lambda\bar{v} - k + \sum_{j=0}^{k-1}(k-j)b_j\}\frac{c_b}{(1-\rho)\lambda}. \end{aligned} \tag{A,10}$$

By substituting (A.10) into (A.8), we obtain the desired result. The term $\tilde{C}_0$ which is used as an initial value can be computed in a similar way. From Fubini's theorem, ξ_0 is obtained as $\frac{\lambda v^{(2)}}{2}C_b$. Note also that $f_{k,k+1} = f_{-1,0} - (1+k)\frac{\bar{u}}{1-\rho}C_b$, for $k < 0$. Using these facts $\tilde{C}_0$ is expressed as

$$\begin{aligned} \tilde{C}_0 &= \xi_0 + \sum_{j=1}^{\infty} b_j f_{-j,0} \\ &= \xi_0 + \sum_{j=1}^{\infty} b_j \sum_{k=0}^{j-1}\{f_{-1,0} + (j-1-k)\frac{\bar{u}}{1-\rho}C_b\} \\ &= \xi_0 + f_{-1,0}\lambda\bar{v} + \frac{\bar{u}}{1-\rho}C_b\sum_{j=1}^{\infty} b_j\frac{j(j-1)}{2} \\ &= \frac{\lambda v^{(2)}}{2}C_b + \frac{\rho\lambda v^{(2)} + 2\rho\bar{v}}{2(1-\rho)}C_b + \frac{\lambda^2\bar{v}u^{(2)}}{2(1-\rho)^2}C_b. \end{aligned}$$

□

References

[1] Buzacott, J.A. and Shanthikumar, J. : Stochastic Models of Manufacturing Systems. Prentice-Hall, New Jersey 1993.

[2] Federgruen, A. and So, K. : Optimality of Threshold Policies in Single Server Queueing Systems with Server Vacations. *Adv. Appl. Prob.* Vol. 23 (1991), 388-405.

[3] Federgruen, A. and Zheng, Y. S. : Characterization and Efficient Computation of Optimal Policies for General Inventory Systems Endogenously Supplied by a Single Server Production Facility. Technical Report, Graduate School of Business, Columbia University, USA 1991.

[4] Gavish, B. and Graves, S. C. : A One Product Production/Inventory Problem Under Continuous Review Policy. *Opns. Res.* Vol. 28 (1980), 1228-1236.

[5] Gavish, B. and Graves, S. C. : Production/Inventory Systems with a Stochastic Production Rate under a Continuous Review Policy. *Comp. & Opns. Res.* Vol. 8 (1981), 169-183.

[6] Heyman, D. P. : Optimal Operating Policies for $M/G/1$ Queueing Systems. *Opns. Res.* Vol. 16(1968), 362-382.

[7] Ross, S. M. : Applied Probability Models with Optimization Applications. Holden-Day, Sanfrancisco 1970.

[8] Scarf, H. : The Optimality of (s, S) Policies in the Dynamic Inventory Problem, in K. Arrow, S. Karlin and P. Suppes (Eds.), Mathematical Methods in the Social Sciences. Stanford University, Stanford 1960.

[9] Srinivasan, M.M. and Lee, H.S. : Random Review Production/Inventory Systems with Compound Poisson Demands and Arbitrary Processing Times. *Management Science* Vol. 37(1991), 813-833.

[10] Tijms, H.C. : An Algorithm for Denumerable State Semi-Markov Decision Problems with Application to Controlled Production and Queueing Systems, in Recent Developments in Markov Decision Theory, ed. R. Hartley et al. Academic Press, New York 1980.

[11] Veinott, A. : On the Optimality of (s, S) Inventory Policies: New Condition and a Proof. *J. SIAM Appl. Math.* Vol. 14(1966), 1067-1083.

Hyo-Seong Lee
Department of Industrial Engineering
Kyung Hee University
Kiheung, Yongin-goon, Kyunggi-do
449-900, Korea
email: hslee.nms.kyunghee.ac.kr