A PERFORMANCE COMPARISON OF PULL TYPE CONTROL MECHANISMS FOR MULTI-STAGE MANUFACTURING

Fikri Karaesmen

and

Yves Dallery

Laboratoire d'Informatique de Paris 6 (LIP6-CNRS)

Université Pierre et Marie Curie

4, place Jussieu

75252 Paris Cedex 05, France

Fikri.Karaesmen@lip6.fr, Yves.Dallery@lip6.fr

A preliminary version of this paper was presented at the Tenth International Working Seminar in Production Economics, Igls, Austria, Feb. 16-20, 1998.

A Performance Comparison of Pull Type Control Mechanisms for Multi-Stage Manufacturing Systems

Fikri Karaesmen
and
Yves Dallery
Laboratoire d'Informatique de Paris 6 (LIP6)
CNRS - Université Pierre et Marie Curie
4, place Jussieu
FR-75252 Paris Cedex 05
France
Fax.: +331/44.27.62.86
E-mail: Fikri.Karaesmen@lip6.fr, Yves.Dallery@lip6.fr

April 1, 1998

Abstract

With the emergence of Just-in-Time manufacturing, production control mechanisms that react rapidly to actual occurrences of demand are gaining importance. Several *pull* type control mechanisms have been proposed to date, but it is usually difficult to quantify how good these mechanisms are, as well as understanding the structural properties that make them desirable. By using a two stage model and an optimal control framework, we study some of these issues here. Our framework permits quantifying the performance of classical mechanisms such as base stock and kanban and more complex mechanisms such as generalized and extended kanban. We also analyze the tradeoffs between single versus multiple control points and service level constraints on the back-orders.

1 Introduction

The emergence of Just-in-Time manufacturing approach has underlined the importance of production control and coordination mechanisms that react to actual occurrences of demand rather than future demand forecasts. This issue is especially important for manufacturing systems consisting of multiple stages where there is also the additional complexity of coordinating the different stages of production in addition to the effort to follow the realizations of demand. Production control mechanisms that use the actual occurrences of demand rather than future demand forecasts to control the flow of material are known as *pull* type control mechanisms. Several control mechanisms have been proposed for *pull* type manufacturing. However due to the complexity of the problem, it is difficult to quantify, in terms of cost, the advantages and disadvantages of these existing mechanisms, as well as understanding, in general, the properties of good control mechanisms. In this paper, we attempt to clarify some of these issues using a simple two stage model that admits an exact analysis.

Two of the better known pull control mechanisms are base stock and kanban (see Buzacott and Shanthikumar [3] for example). These mechanisms resolve the tradeoff between unsatisfied demand and holding costs in different ways. The base stock system was originally proposed for production/inventory systems with infinite production capacity and uses the idea of a safety stock for finished good inventory as well as safety buffers between stages for coordination. Kanban mechanism, on the other hand, has its emphasis on coordinating production by using a finite number of production authorization cards that transmit demand requests. Both systems are fairly simple to implement requiring the definition of a single parameter per each stage which corresponds to safety stocks and production authorization cards respectively for base stock and kanban.

Since the base stock mechanism offers the feature of rapid reaction to demand and the kanban mechanism achieves better coordination and controlled work in process inventories, intuitively, combining the respective merits of base stock and kanban control mechanisms would entail many potential benefits. Buzacott [2] and Zipkin [10] initiate the first implementation of this approach. The resulting mechanism, called the *generalized kanban*, borrows the idea of safety stocks from the base stock system and production authorization cards from the kanban system. As a relative drawback however, this hybrid system is defined by two parameters per stage, one defining the safety stocks the other defining the number of production authorization cards.

Recently, Dallery and Liberopoulos [4] have introduced a new pull type control mechanism called *extended kanban* which is also a mixture of base stock and kanban. This mechanism is also defined by two parameters per stage but is conceptually clearer than generalized kanban and is potentially easier to implement. The generalized kanban as well as the extended kanban include both the base stock and kanban systems as special cases (see [4]).

Although the two parameter per stage mechanisms such as generalized or extended kanban offer potential improvements over single parameter mechanisms, it is not obvious how these improvements translate into savings in cost and whether or not it is worth investing in a more complex mechanism. Our aim in this paper is to explicitly quantify these trade-offs albeit in a rather simplified framework.

The model we study is the simplest system that captures the key issues in pull type control in a multi stage production environment: we consider two single machines in tandem with a work in process inventory in between the two stages and a finished goods inventory after the second stage. Demands that arrive to the system are satisfied from the finished goods inventory whenever possible and are backordered otherwise. This system allows us to analyze the important tradeoff between backorders and the cost of holding finished goods and work in process inventory that help reduce backorders. To quantify this tradeoff, we consider two different cases. In the first case, linear holding and backorder costs are incurred for the items that are held in stock and those that are backordered respectively. In the second case, a certain service level with respect to backordered items is required. When processing times of both machines are exponentially distributed and demands occur according to a Poisson process, these production control problems can be set as optimal control problems. The first case (i.e. with linear backorder costs) has been studied previously by Veatch and Wein [9] who also give some numerical examples on the performance of some of the pull type control mechanisms considered here. We use the same framework to compare the performance of many alternative pull mechanisms ranging from simpler mechanisms to more complicated ones than those considered in [9].

One interesting issue that we can analyze through our framework is single versus multiple control points in the system. When a system consisting of multiple machines in tandem is viewed as a single stage system, control mechanisms that control the system only at the point where the raw parts enter the system can be defined. CONWIP (Spearman et al., [7]) is such a mechanism where the shipment of a finished part to the customer causes a raw part to enter the system. We give theoretical and numerical results that explain some of the tradeoffs in single stage versus multiple stage decompositions of the system.

It is frequently argued that in many cases linear backorder costs are not appropriate and a service level approach is more desirable. We discuss the extension of the basic model to a case with a constraint on the proportion of unfilled demand and give numerical examples.

By quantifying the tradeoffs between single and two parameter policies per stage and single versus two stage control, our comparisons shed further light into the desirable properties and shortcomings of a given pull control mechanism.

The outline of the paper is as follows: in section 2, we introduce the model and the corresponding control problem. We also describe the pull control mechanisms to be analyzed and provide a qualitative comparison based on the control space descriptions of the mechanisms. Single stage control mechanisms differ from multi-stage mechanisms, we describe them briefly and give a structural result. In section 3 we give numerical results on the performance of various pull control mechanisms and discuss some of the tradeoffs involved. Section 4 studies the extension of the basic model to the case with service level constraints. Our conclusions are given in section 5.

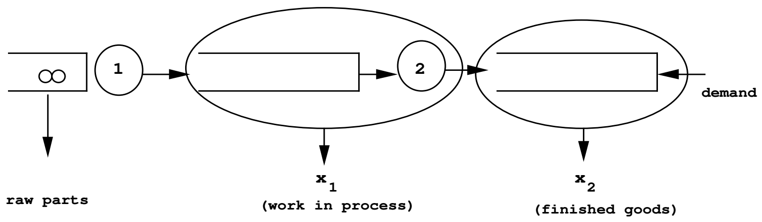

Figure 1: The Two Stage Production System

2 Definitions and Qualitative Results

We consider two single machines in tandem which are connected by an intermediate buffer. Whenever a part is finished in the first machine, it is placed in an intermediate buffer and whenever a part is finished in the second machine, it is placed in the finished goods inventory. The input buffer of the first machine consists of raw material which is always available, so the first machine is never starved. The demand that arrives to the system is satisfied from the finished goods inventory whenever possible and is backlogged otherwise. This system is displayed in Figure 1. Holding costs are incurred for the parts held in the intermediate buffer and the finished goods inventory. Furthermore, whenever a demand is backordered, backorder costs are incurred. We are interested in controlling the release of parts from a buffer to the downstream machine so that the sum of the long run average holding and backorder costs are minimized.

To give a precise description of the model, consider the case where demands arrive to the system according to a Poisson process with rate λ and the machine in stage i has exponentially distributed service times with rate μ_i $(i = 1, 2)$. Let $X_1(t)$ denote the number of parts in the intermediate buffer plus the part that is currently in production in the second machine at time t and let $X_2(t)$ be the number of parts in the finished goods inventory at time t. Linear holding costs of h_1 proportional to $X_1(t)$ is incurred in the first stage. As for the second stage, holding costs are incurred at rate h_2^+ whenever the finished goods inventory is non-negative and backorder costs are incurred at rate b whenever there are backorders. To simplify the notation, we can define the piecewise linear cost function h_2, such that:

$$h_2(x) = \begin{cases} h_2^+ x & \text{if } x \geq 0 \\ bx & \text{if } x < 0 \end{cases} \tag{1}$$

A part release control policy, π determines dynamically whether the machines should be authorized to work or not. Our objective is to find a part release policy π such that the long run average cost per unit time:

$$\lim_{T\to\infty} \sup \frac{E\left[\int_0^T h_1 X_1(t) + h_2(X_2(t))\, dt\right]}{T} \tag{2}$$

is minimized.

By standard results in Markov decision processes, an optimal stationary policy π^* exists for the above problem and can be obtained through the solution of the optimality equation:

$$\begin{aligned} V(x_1, x_2) + g^* &= h_1 x_1 + h_2(x_2) + \lambda V(x_1, x_2 - 1) + \mu_1 \min\{V(x_1, x_2),\, V(x_1 + 1, x_2)\} \\ &\quad + \mu_2 \min\{V(x_1, x_2),\, V(x_1 - 1, x_2 + 1)\} \end{aligned} \tag{3}$$

where $V(x_1, x_2)$ is the relative value function and g^* is the optimal cost per unit time. Note that we have set $\lambda + \mu_1 + \mu_2 = 1$ without loss of generality as well as using the convention that $\min\{V(x_1, x_2),\, V(x_1 - 1, x_2 + 1)\} = V(x_1, x_2)$ when $x_1 = 0$.

2.1 Control Mechanisms

Below, we introduce the details of the pull type control mechanisms that will be analyzed in the sequel. The development here follows closely that of Liberopoulos and Dallery [6] where more details can be found.

For ease of exposition, we represent all control mechanisms by queueing networks with synchronization stations. All the mechanisms that follow can be represented using at most five different type of queues: one corresponding to finished parts in stage i (denoted by

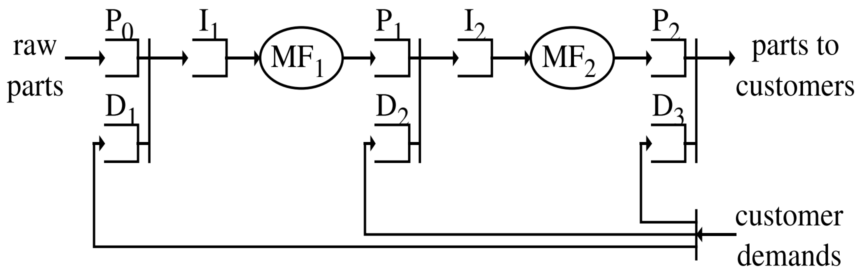

Figure 2: The base stock mechanism

P_i), one corresponding to demands for production of new parts in stage i (D_i), one that corresponds to production authorizations in stage i (A_i), one corresponding to pairs of finished parts and production authorizations in stage i (PA_i) and the final one corresponding to pairs of demands for production and production authorizations in stage i (DA_i). Note that, P_0 corresponds to the raw parts buffer which is assumed to be always non-empty. One can also define the queue of parts waiting to be processed in stage i, I_i, for a complete description although this queue is not critical for our purpose here.

Base Stock Mechanism

The base stock mechanism is displayed in Figure 2. In the figure, D_3 corresponds to customer demands. The base stock mechanism is completely described by two parameters S_1 and S_2 corresponding to the base stock levels in stages 1 and 2 respectively. Initially, there are S_1 (S_2) parts in queue P_1 (P_2) while all other queues are empty. Whenever a customer demand arrives, it joins the queue D_3 and requests the release of a finished part from P_2. At the same time, this demand is also transmitted to D_2 and D_1 thereby requesting a release of parts from P_0 to I_1 and P_1 to I_2. Hereon, we use the notation TSBS(S_1, S_2) to denote the two stage base stock policy with parameters S_1 and S_2.

Kanban Mechanism

The kanban mechanism can be seen in Figure 3. Initially, the queue PA_1 (PA_2) contains

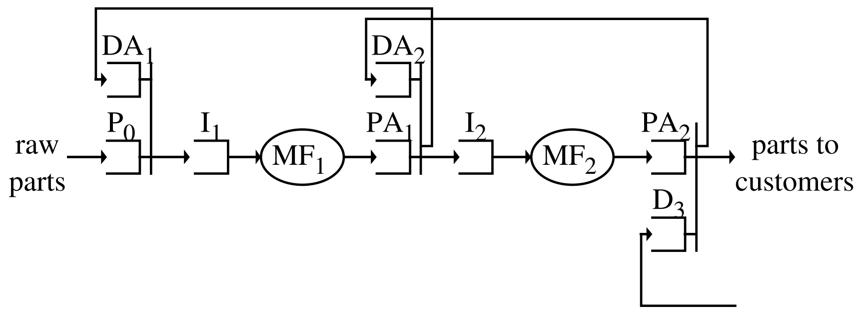

Figure 3: The kanban mechanism

K_1 (K_2) finished parts each part with a kanban card attached on it while all the other queues are empty. Whenever a customer demand arrives to the system, it joins queue D_3 and requests the release of a finished part from queue PA_2. If a part is available in PA_2, it is released to the customer after having detached the kanban card attached to it. The freed kanban card then joins the queue DA_2 and requests the release of a finished part from PA_1 to I_2. If PA_1 is not empty, this release will be performed with the kanban detached from the part transferred to DA_1 where it will cause the release of a raw part into I_1. This way, customer demands are transmitted upstream in the system using the kanban cards. The control is exerted through the availability of a card in a given stage (if the card is not available at the time of request, demand will not be transmitted upstream until a card becomes available). Once again, the mechanism will be completely described by the initial number of kanban cards at each stage, K_1 and K_2. We will denote this system by TSK(K_1, K_2).

Generalized Kanban Mechanism

The generalized kanban mechanism (Buzacott [2], Zipkin [10]) is displayed in Figure 4. Initially, the queue P_1 (P_2) contains S_1 (S_2) parts and the queue A_1 (A_2) contains K_1 (K_2) production authorizations while all other queues are empty. The evolution of generalized kanban is very similar to that of the kanban mechanism except for the effects

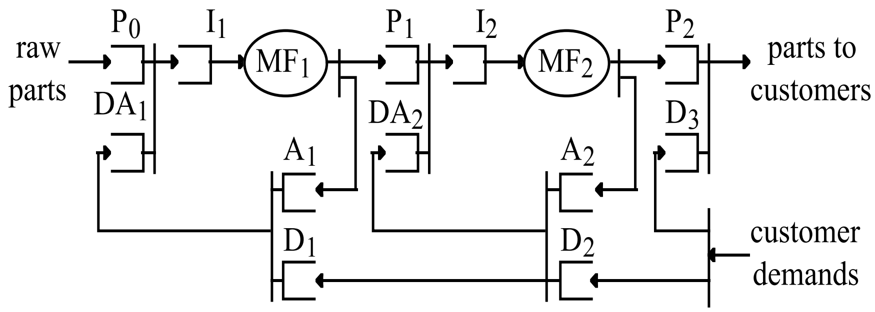

Figure 4: The generalized kanban mechanism

of the additional (initially) free kanban cards. In the kanban mechanism whenever there are no finished parts in a certain stage, demand requests cannot be transferred upstream. The additional kanbans in the generalized kanban system serve the purpose of relaxing this constraint. In this case, demand can be transferred upstream from a stage even in the absence of finished parts as long as a kanban card is available. We denote the two stage generalized kanban system with parameters S_1, S_2 and K_1, K_2 by TSGK(S_1, K_1, S_2, K_2). Note that, TSGKS(K_1, K_1, K_2, K_2) is equivalent to TSK(K_1, K_2) and TSGK(S_1, ∞, S_2, ∞) is equivalent to TSBS(S_1, S_2) (see [3] and [4]) .

Extended Kanban Mechanism

The extended kanban mechanism (Dallery and Liberopoulos [4]) is displayed in Figure 5. In this mechanism, there are initially S_1 (S_2) parts with kanbans attached to each of them in queue PA_1 (PA_2) and $K_1 - S_1$ ($K_2 - S_2$) free kanbans in queue A_1 (A_2) while all other queues are empty. Note that, this mechanism has the condition that $K_i \geq S_i$ ($i = 1, 2$). When a demand arrives to the system, it joins the queue D_3 as well as the queues D_2 and D_1 (as in the base stock mechanism). The demand that joins the queue D_3 requests a finished part from queue PA_2, if there is a part available in PA_2, it is released to the customer and the detached kanban is transferred to the queue of free kanbans A_2. Concurrently, the demand that has joined D_2 requests the release of a finished from PA_1. The release is now dependent on the availability of finished parts in PA_1 as before but also

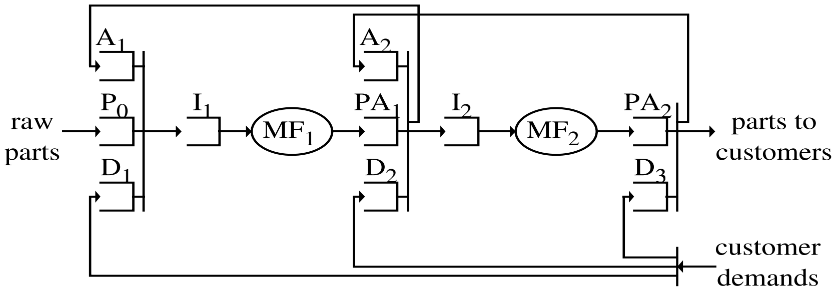

Figure 5: The extended kanban mechanism

on the availability of free kanbans in the queue A_2. If PA_1 and A_2 are both non-empty, the release takes place, the part from moves PA_1 to I_2 while a kanban card is transferred from A_2 to A_1. A similar type of synchronization is required for the release of raw parts from P_0 to A_1. We denote the extended kanban system with parameters, S_1, S_2 and K_1, K_2 by TSEK(S_1, K_1, S_2, K_2). As in the generalized kanban system, setting the parameters K_1 and K_2 to infinity in a TSEK results in an equivalence to TSBS(S_1, S_2). On the other hand, setting $K_i = S_i$ $(i = 1, 2)$ in TSEK leads to an equivalence to TSK(K_1, K_2) (see [3] and [4]).

2.2 A Qualitative Comparison: State Space Representations

It has been shown by Veatch and Wein [8] that optimal control policies have certain monotonicity properties. In particular, the authors show that for both machines the "produce/do not produce" regions are separated by monotone switching curves. Figure 6 displays typical switching curves and the control regions. As can be seen in the figure, machine 1 is authorized to produce when, x_1, the level of work in process is below a decreasing (in x_2) switching curve. Similarly, machine 2 is authorized to produce when, x_2, the level of finished goods inventory is below a second (increasing in x_1) switching curve. In fact, note that the regions where only machine 1 or machine 2 is authorized to produce are transient.

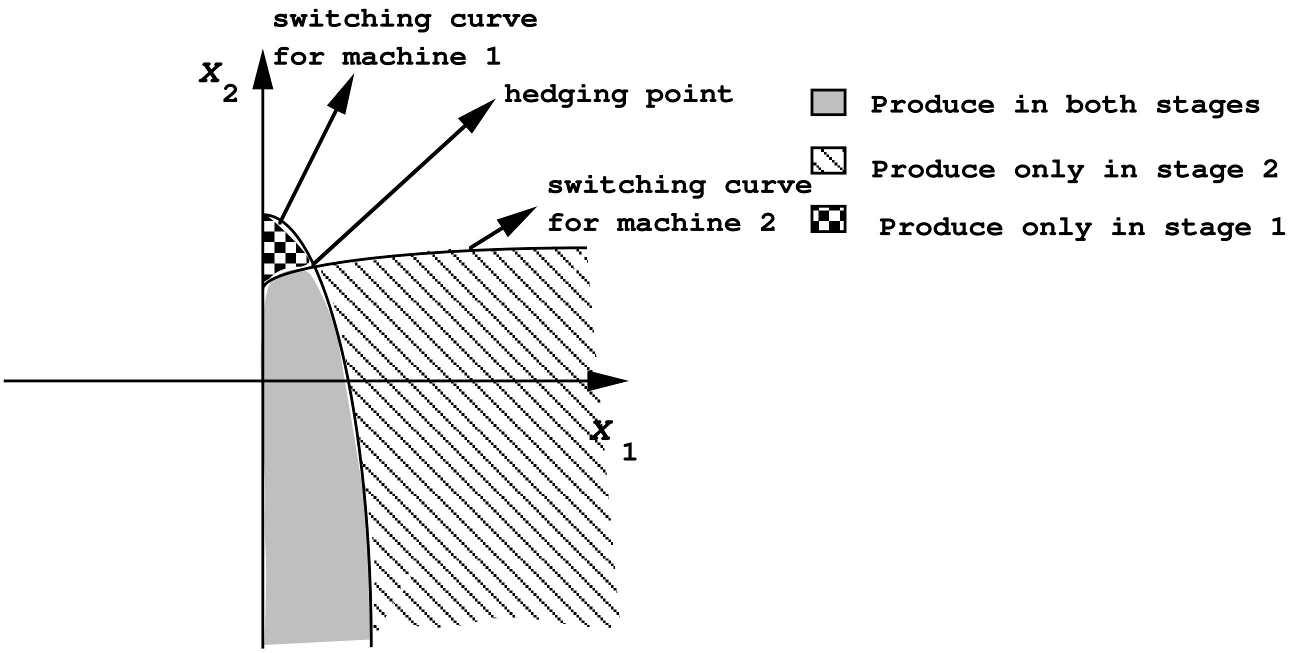

Figure 6: Optimal Switching Curves

Monotone control policies can be completely characterized by a pair of switching curves which define the region where both machines are authorized to produce. Also note that, the point where the two switching curves intersect is the *hedging point* of the system, the point which the control policy drives the system towards. The monotonicity properties provide interesting qualitative insights into the structure of good control policies. The implications of the monotone structure is quite intuitive; good policies must constrain the work in process levels in addition to the level of finished goods inventory. At the same time, the work-in-process levels should change depending on the level of the finished items buffer with higher levels of backlog (or lower levels of finished items) requiring higher (or equivalent) levels of work-in-process. We can qualitatively analyze the performance of the pull control mechanisms described earlier keeping these concerns in mind.

Figure 7 displays the state space representations of the base stock and kanban policies. Note that, the second stage parameters S_2 and K_2 play identical roles with respect to the

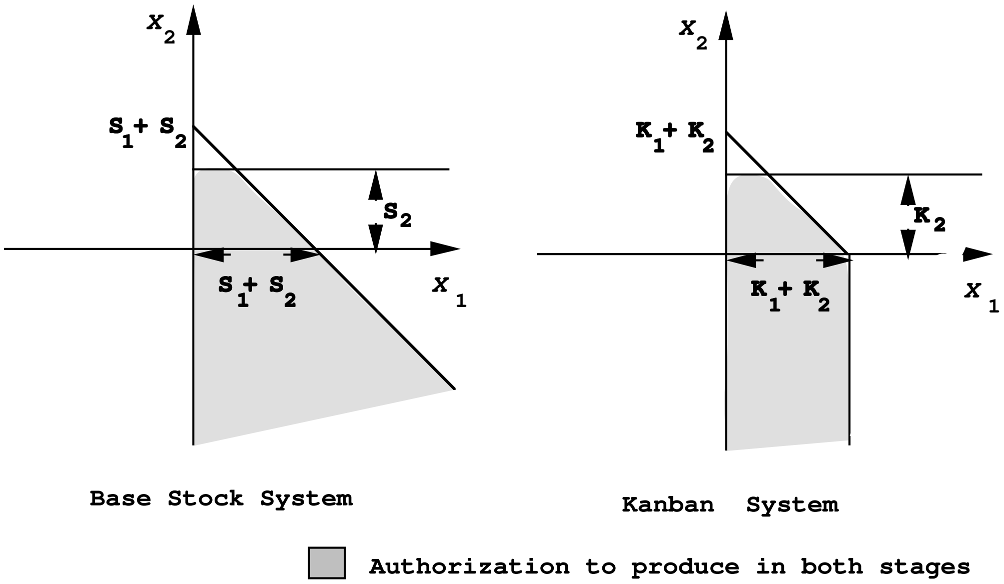

Figure 7: Base Stock and Kanban Policies

control of the second machine. On the other hand, in the base stock mechanism the first machine works as long as $x_1 + x_2 < S_1 + S_2$ whereas in the kanban mechanism the first machine will work as long as $x_1 + x_2 < K_1 + K_2$ and $x_1 < K_1 + K_2$. As a result, while the work in process inventory x_1 may grow unboundedly in the base stock mechanism, it is bounded by the total number of production authorization cards in the kanban mechanism. In fact, Veatch and Wein [9] prove that the base stock mechanism can never be exactly optimal due to this drawback.

Figure 8 displays the extended and generalized kanban control mechanisms. The roles of the parameters S_1, S_2, K_1 and K_2 in defining the respective switching curves can be seen on this figure. The basic difference between these two mechanisms and the kanban mechanism is apparent from the figure. In particular, generalized and extended kanban handle the work in process constraints in a different way than standard kanban. While the first switching curve of the kanban mechanism changes its slope from -1 to ∞ when $x_2 = 0$, the other two mechanisms have further flexibility in selecting the point where this change occurs. While extended kanban permits changing the slope at levels $x_2 \leq 0$, generalized kanban permits changing the slope at both positive and negative levels of x_2. In fact, in the particular case of the model considered in this paper, extended kanban can be viewed as a special case of the generalized kanban mechanism. This is an interesting feature of the particular model, since in general both mechanisms have distinctly different behavior and properties as elaborated by Dallery and Liberopoulos [4]. The equivalence of the two mechanisms for this model can be explained as follows: although in the TSGK the parameter K_1 does not seem to play a role (see Figure 8), the definition of the mechanism enforces setting $K_1 \geq 1$ as otherwise, the first machine would never have the authorization produce. Alternatively, in Figure 8, initially K_1 seems to be a crucial parameter but a closer investigation reveals that the selection of K_1 does not really matter in itself (as long as $K_1 \geq S_1$), since the switching curve (and thus the behaviour) is defined by the sum $K_1 + K_2$ which can always be adjusted by the choice of K_2.

2.3 Single Stage Control

In the previous sections, we discussed in detail the coordination mechanisms which control the release of material both to the first and the second machine. An alternative approach is to view the system as consisting of a single stage which has two machines in tandem and control the release of material only to the first machine. In this case, while the first machine is directly controlled as before, the second machine is not directly controlled and produces whenever it can (i.e. whenever there are items completed in the first stage and waiting to be produced). The single stage kanban system, also known as the CONWIP system (see Spearman et al. [7]) has received particular attention. However, single stage basestock, kanban, generalized kanban mechanisms can also be defined analogous to their previously described two stage versions. It turns out that in this case generalized and extended kanban policies with identical parameters are equivalent (Liberopoulos and Dallery [5]). Hence, it

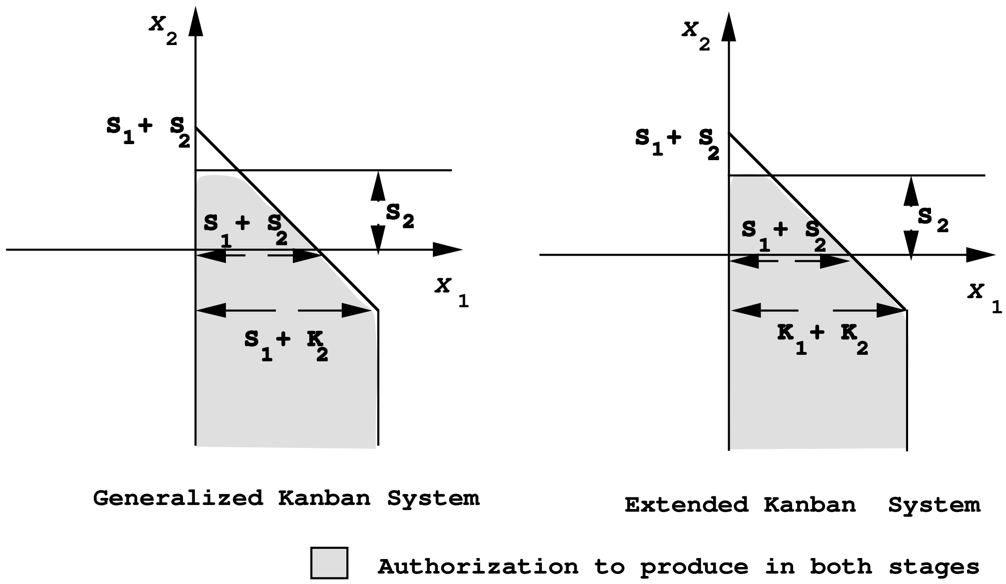

Figure 8: Extended and Generalized Kanban Policies

will suffice to consider SSGK from the point of view of performance. We use the shorthands SSBS(S), SSK(K) and SSGK(S, K) to denote these mechanisms having parameters S and K. Our framework enables us to quantify the single stage versus two-stage control tradeoffs through the optimal control framework, but first we elaborate on some qualitative issues.

Intuitively, the necessity to control the entry of material at multiple stages seems to stem from the fact that as material moves downstream in the production system some value is added to the part in process and as a result the holding costs at upstream stages can be considerably smaller than those at downstream stages. Hence, the difference in upstream and downstream holding costs motivates keeping inventories upstream whenever possible which implies that it would be necessary to control the release of material in some intermediate stages. On the other hand, when holding costs do not change significantly between different stages of the system, it is plausible that intermediate control points are unnecessary, since in this case what matters is the total number of parts in the system regardless of their particular positions (upstream or downstream). Within our framework, we can concretize this last point by the following proposition which states that whenever holding costs are identical, the optimal policy is to always authorize the machine in the second stage to produce.

Proposition 1 When $h_1 = h_2^+$, the optimal control policy in the second stage is to produce whenever possible (i.e. when $x_1 \geq 0$).

Proof: See Appendix.

Remark: Proposition 1 can alternatively be stated as follows: one should never hold any intermediate inventory between stages 1 and 2. It should be noted that some pull control mechanisms violate this proposition by definition of their behavior. This is the case, for instance, of the TSK, for which any positive value of K_1, parts will be held in queue PA_1 (see Figure 3) at certain times. On the other hand, in TSBS for example, setting S_1 to zero in TSBS(S_1, S_2) results in an equivalence with SSBS(S_2). The same equivalence also holds true between TSGK and SSGK, as well as between TSEK and SSEK.

3 Performance Analysis

To analyze the performance of control mechanisms, we use the following setup. We set the demand rate λ to 1 without loss of generality and vary μ_1 and μ_2 as well as the cost parameters. The 18 different sets of data used in the following numerical experiments are displayed in Table 1. The first three data sets have also been used by Veatch and Wein [9].

Using the parameter sets in Table 1, we perform the following experiment: for each control mechanism of interest, i.e. single stage base stock (SSBS), single stage kanban(SSK), single stage generalized kanban(SSGK), two stage base stock(TSBS), two stage kanban(TSK) and two stage generalized kanban (TSGK) (we omit extended kanban, since it is a special case of the generalized kanban for this problem), we find the values of the parameters that give the minimum cost by performing a search in the state space combined with the value

Set No.	μ_1	μ_2	h_1	h_2^+	b
1	1.2	1.2	1	2	4
2	2	1.2	1	2	4
3	1.2	2	1	2	4
4	1.2	1.2	1	2	2
5	2	1.2	1	2	2
6	1.2	2	1	2	2
7	1.2	1.2	1	2	8
8	2	1.2	1	2	8
9	1.2	2	1	2	8
10	1.2	1.2	1	1	2
11	2	1.2	1	1	2
12	1.2	2	1	1	2
13	1.2	1.2	1	1	4
14	2	1.2	1	1	4
15	1.2	2	1	1	4
16	1.2	1.2	1	1	8
17	2	1.2	1	1	8
18	1.2	2	1	1	8

Table 1: Sets of parameters used

iteration algorithm (see Bertsekas, [1] for example). We also compute the optimal policy for the given parameters by using value iteration in a truncated state space (state spaces of dimension up to 50 by 100 have been used). The comparisons are hence between the best performances that can be obtained from a given mechanism. In Table 2 below, we report the cost achieved by the optimal policy (denoted by "Opt." in the table) and the percentage suboptimality of the minimum cost achieved by each mechanism as well as the parameters of each mechanism yielding the minimum cost (given in parenthesis after the suboptimality value). The parameters are given in the order defined in the previous sections. In displaying the parameters, we set K_1 to 1 in TSGK, since it does not play any role in our problem.

Consider the columns of Table 2, that correspond to single stage control mechanisms. We observe in general that in most cases, either SSBS or SSK performs well. A more careful observation reveals that for the cases where both machines have equal production rates SSBS performs better than SSK and for the cases where the second machine is slower than the first machine SSK performs better. In either case, SSGK is the clear winner, with a maximum error of 4.2%. One would be tempted to state that single state space policies are extremely efficient if it were not for the less than satisfactory results obtained in data

Set	Opt.	SSBS	SSK	SSGK	TSBS	TSK	TSGK
1	22.10	1.79 (11)	6.04 (13)	1.55 (11,22)	0.44 (4,8)	3.73 (6,8)	0.37 (4,1,8,16)
2	15.75	10.08 (7)	1.50 (7)	1.46 (7,6)	10.08 (0,7)	2.92 (1,6)	1.46 (0,1,7,6)
3	11.77	13.67 (7)	15.70 (7)	13.67 (7,30)	0.11 (7,2)	0.84 (7,2)	0.11 (7,1,2,13)
4	15.97	2.42 (8)	9.42 (11)	1.57 (8,18)	1.81 (2,6)	6.45 (6,6)	0.66 (3,1,6,14)
5	11.02	16.63 (4)	3.02 (5)	2.56 (5,6)	18.56 (1,4)	3.78 (1,4)	2.56 (0,1,5,6)
6	8.23	8.33 (4)	15.8 (6)	8.33 (4,30)	0.04 (5,1)	3.13 (6,1)	0.04 (5,1,1,15)
7	29.08	2.01 (15)	4.40 (17)	2.01 (15,30)	0.22 (5,11)	1.96 (6,12)	0.81 (5,1,11,15)
8	21.45	7.49 (9)	2.69 (9)	2.11 (10,7)	7.49 (0,9)	3.68 (1,8)	2.11 (0,1,10,7)
9	16.02	19.09 (9)	19.69 (10)	19.09 (9,30)	0.31 (9,3)	1.96 (10,3)	0.31 (9,1,3,11)
10	13.40	2.47 (11)	3.05 (12)	1.16 (11,17)	2.47 (0,11)	3.15 (1,11)	1.16 (0,1,11,17)
11	9.12	22.22 (7)	4.96 (6)	4.20 (7,5)	22.22 (0,7)	7.69 (1,5)	4.20 (0,1,7,5)
12	7.19	0.05 (7)	1.57 (7)	0.05 (7,30)	0.05 (0,7)	1.59 (1,6)	0.05 (0,1,7,30)
13	17.16	1.15 (15)	1.98 (16)	1.00 (15,23)	1.15 (0,15)	2.05 (1,15)	1.00 (0,1,15,23)
14	12.08	16.04 (9)	7.05 (9)	3.97 (10,6)	16.04 (0,9)	9.28 (1,8)	3.97 (0,1,10,6)
15	10.03	0.13 (9)	0.59 (10)	0.13 (9,30)	0.13 (0,9)	0.60 (1,9)	0.13 (0,1,9,30)
16	21.20	0.64 (20)	1.02 (20)	0.63 (20,27)	0.64 (0,20)	1.07 (1,19)	0.63 (0,1,20,27)
17	15.47	12.24 (13)	8.37 (12)	3.80 (13,7)	12.24 (0,13)	10.43 (1,11)	3.80 (0,1,13,7)
18	13.35	0.32 (13)	0.35 (13)	0.32 (13,30)	0.32 (0,13)	0.35 (1,12)	0.32 (0,1,13,30)
Average Suboptimality		7.60	5.96	3.76	5.24	3.59	1.32
Worst Case Suboptimality		22.22	19.69	19.09	22.22	10.43	4.2

Table 2: The cost of the optimal policy and the percentage suboptimality of the best policy within each pull control mechanism

sets 3, 6 and 9. The common characteristics of these sets are a faster production rate in the second machine and higher holding costs in stage 2 than stage 1. The imbalance between the machines necessitates a considerable amount of safety stock in between but since the holding finished goods inventory is expensive, this safety stock should not be converted into finished goods until necessary and this can only be achieved by controlling the machine in the second stage.

The columns of Table 2 corresponding to two stage policies reveal other interesting properties. Firstly, TSBS performs better than TSK in all cases except those where the first machine is faster than the second machine. The problem, however, is that when TSBS performs worse that TSK, it performs poorly whereas TSK seems more robust. In any case, when robustness is the issue TSGK is considerably more reliable than either TSK or TSBS. Once again, the interesting observation here is that usually either one of TSK or TSBS performs well while TSGK always performs well since it can imitate the better system by an appropriate choice of the parameters.

Finally, note that in rows corresponding to parameter sets 10-18 of Table 2 SSKS performs better than TSKS while SSBS and SSGK perform as well as TSBS and TSGK respectively. This is not surprising in light of our previous results since this part of the data set corresponds to the the cases where the holding costs are identical in both stages of the system.

4 Service Level Constraints

It is frequently argued that although backordering demand is an important concern, backlog costs are difficult to quantify. An alternative approach to analyze the tradeoff due to unfilled demand is through service level constraints. A frequently used service level is *fill rate* defined as the proportion of demands that can be satisfied from on hand inventory upon arrival. In this section, we extend the previous discussion on qualitative properties of good control policies to systems with fill rate constraints.

Consider a fill rate constraint of the following type: the probability of fulfilling an order from on hand inventory upon arrival must be at least $(1-\alpha)$. To make this definition more precise, let t_n be the time corresponding to the nth event (arrival of demand or service completion in either stage) in the system. Let $I_A()$ be the indicator function that corresponds to the demand arrival event, i.e. $I_A(t_n) = 1$ if the nth event is an arrival and $I_A(t_n) = 0$ otherwise. Furthermore, we can define a second indicator function, $I_b()$ that marks demand arrivals that are not satisfied from on-hand inventory. Hence:

$$I_b(t_n) = \begin{cases} 1 & \text{if } I_A(t_n) = 1 \text{ and } X_2(t_n^-) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

The fill rate constraint is then:

$$\lim_{n \to \infty} \frac{E\left[\sum_0^n I_b(t_n)\right]}{E\left[\sum_0^n I_A(t_n)\right]} \leq \alpha \tag{4}$$

In other words, by setting the backorder cost $b = 0$, we obtain the identical objective function as in (2), however this time the minimization is subject to the constraint (4)

If we consider truncated state spaces, we can solve the above problem exactly using a linear programming formulation. However this approach will not provide a lot of insight, since the optimal policy will be randomized and will not be easy to implement. To obtain a close to optimal non randomized solution of the above problem, we use a Lagrangian relaxation by adding the constraint to the objective function with a penalty of r. The resulting problem is referred to as the *problem with unfill penalties.*

To analyze the problem with unfill penalties, note that the Lagrangian leads to the following optimality equations:

$$\begin{aligned}
V(x_1,x_2) &= h_1x_1 + h_2^+x_2 + \lambda V(x_1,x_2-1) + \mu_1 \min\{V(x_1-1,x_2+1),V(x_1,x_2)\} \\
&\quad +\mu_2 \min\{V(x_1,x_2+1),V(x_1,x_2)\} \text{ if } x_2 > 0 \\
V(x_1,x_2) &= h_1x_1 + \lambda(r + V(x_1,x_2-1)) + \mu_1 \min\{V(x_1-1,x_2+1),V(x_1,x_2)\} \\
&\quad +\mu_2 \min\{V(x_1,x_2+1),V(x_1,x_2)\} \text{ if } x_2 \leq 0
\end{aligned} \tag{5}$$

once again with the convention that $\min\{V(x_1-1,x_2+1),V(x_1,x_2)\}$ automatically equals $V(x_1,x_2)$ whenever $x_1 = 0$.

Examining the above optimality equations, we note that the unfill penalties can be converted into equivalent backorder costs. The equivalent backorder cost function is given by:

$$h_2(x) = \begin{cases} h_2^+x & \text{if } x > 0 \\ \lambda r & \text{if } x \leq 0 \end{cases} \tag{6}$$

The only difference between the backorder cost and unfill penalty problems is the backorder cost function. This prompts the question as to whether the monotonicity properties are retained for this problem as well. Unfortunately, the new holding cost function does not satisfy the directional submodularity conditions used by Veatch and Wein [8] to prove monotonicity which rules out an inductive proof. Furthermore, there are numerical examples in which optimal switching curves are not monotone. Nevertheless, in most numerical examples the optimal switching curves seem to be monotone.

To relate the problem with unfill penalties to the one with service level constraints, note that each unfill penalty r induces an associated fill rate $\alpha(r)$. One can then vary r until

Set	10 % fill rate				20 % fill rate			
	Opt	SSBS	SSKS	SSGKCS	Opt	SSBS	SSKS	SSGKCS
1	27.46	15.48(23)	14.86(23)	14.86(23,23)	19.16	9.81 (17)	6.58 (17)	5.85 (17,16)
2	20.66	15.00 (15)	13.02(15)	3.29 (15,6)	13.50	23.40 (11)	17.40 (11)	6.81 (11,6)
3	14.95	19.60(14)	19.60 (14)	19.26 (14,9)	10.20	7.94(10)	7.75 (10)	7.25(10,8)
10	16.73	9.56(23)	8.37 (23)	8.37(23,23)	12.01	8.91 (17)	4.08 (17)	3.08(17,16)
11	11.72	23.12(15)	19.54(15)	3.92(15,6)	8.13	33.21 (11)	23.25 (11)	7.38(11,6)
12	9.36	0.85(14)	0.85(14)	0.53(14,9)	6.08	-1.32 (10)	-1.32 (10)	-1.97(10,8)
Average Suboptimality		13.94	12.71	8.37		13.66	9.62	4.73
Worst Case Suboptimality		23.12	19.60	19.26		33.21	23.25	7.38

Table 3: Service Level Constraint Results

$\alpha(r)$ is sufficiently close to the desired service level α. The optimal holding cost under this policy can then be computed by fixing the policy and recomputing the cost by setting $r = 0$.

Table 3 reports the optimal performance of single stage base stock, kanban and generalized kanban policies for different parameter sets under two different fill rate constraints, 10% and 20 %. The "Opt." column reports the results of the minimum cost found by the Lagrangian heuristic (which is not necessarily the minimum cost that can be obtained by a stationary policy, in fact in the last row of the table the all three control mechanisms perform better than the Lagrangean heuristic.).

Table 3 is consistent with the preceding numerical results on the case with linear backorder costs. The SSGK performs significantly better than SSBS and SSK on the average. Furthermore, the difference in the average performance is sharpened due to the existence of cases where single parameter policies can perform quite poorly (such as the case in the rows correspoding to parameter sets 2 and 11).

5 Conclusion and Future Research

Using a two stage model and an optimal control approach, we presented performance comparisons between various control mechanisms. It turns out that simple mechanisms such as kanban, base stock and even their single stage variants are very effective for the model considered. On the other hand, these simple mechanisms have a major drawback in that under certain conditions they can perform poorly. This highlights the significant advantage of more complicated mechanisms such as generalized or extended kanban. These mechanisms do not necessarily perform significantly better than simpler ones for a given case but they are guaranteed to perform well under all circumstances.

Many interesting research issues remain unaddressed. An important problem is the optimization of the parameters of a given control mechanism. This is especially important for generalized and extended kanban mechanisms which require more parameters than the others. One of our results that could be useful from the design point of view is that good generalized (or extended) kanban policies in general tend to imitate the better of base stock and kanban policies. It seems plausible then to consider an approach where a good base stock or kanban policy is improved upon by iteratively adjusting the additional parameters to obtain a good generalized or extended kanban policy.

Another interesting and relevant extension is to consider multiple part types. This brings in the additional difficulty of sharing manufacturing resources between different part types in addition to the decisions of whether or not to produce that were considered for the single part type case. The design of simple but effective multi stage pull mechanisms for multiple part type systems remains as a challenging issue for future research.

References

[1] Bertsekas, D.P., *Dynamic Programming and Optimal Control, Volume 2*, Athena Scientific, 1995.

[2] Buzacott, J.A., "Queueing Models of Kanban and MRP Controlled Manufacturing Systems", *Engineering Cost and Production Economics*, 1989.

[3] Buzacott, J.A. and J.G. Shantikumar, *Stochastic Models of Manufacturing Systems*, Prentice-Hall, 1993.

[4] Dallery, Y. and G. Liberopoulos, "Extended Kanban Control System: A New Kanban Type Pull Control Mechanism for Multi-Stage Manufacturing Systems", *Working Paper*, 1997.

[5] Liberopoulos G. and Y. Dallery, "On the Optimization of Single Stage Kanban-Type Control Systems in Manufacturing", *Working Paper*, 1995.

[6] Liberopoulos G. and Y. Dallery, "A Unified Framework for Pull Control Mechanisms in Multi Stage Manufacturing Systems", *Working Paper*, 1997.

[7] Spearman M.L., D.L. Woodruff and W.J. Hopp, "CONWIP: A Pull Alternative to Kanban", *International Journal of Production Research*, Vol. 28, pp. 879-894, 1990.

[8] Veatch M.H. and L.M. Wein, "Monotone Control of Queueing Networks", *Queueing Systems*, Vol. 12, pp.391-408, 1992.

[9] Veatch M.H. and L.M. Wein, "Optimal Control of a Make-to-Stock Production System", *Operations Research*, Vol. 42 pp. 337-350, 1994.

[10] Zipkin, P., "A Kanban Like Production Control System: Analysis of Simple Models", *Research Working Paper No. 89-1, Graduate School of Business*, Columbia University, New York, 1989.

Appendix

Proof of Proposition 1: Consider the case of minimizing the total discounted costs over an infinite horizon with discount factor α, i.e we would like to find the policy that minimizes:

$$\lim_{T\to\infty} \sup E\left[\int_0^T e^{-\alpha t}(h_1 X_1(t) + h_2(X_2(t)))\, dt\right] \tag{7}$$

Let $\lambda+\mu_1+\mu_2+\alpha = 1$ without loss of generality, the corresponding optimality equations are as follows:

$$V(x_1,x_2) = h_1(x_1)+h_2(x_2)+\lambda V(x_1,x_2-1)+\mu_1\min\{V(x_1+1,x_2),V(x_1,x_2)\} \\ +\mu_2\min\{V(x_1-1,x_2+1),V(x_1,x_2)\} \tag{8}$$

We would like to argue through value iteration by using the fact that the optimal infinite horizon cost $V(x_1,x_2)$ can be obtained as the limit of corresponding k-horizon cost functions as the horizon k tends to infinity. To this end let $V^k(x_1,x_2)$ denote the the minimum total cost incurred over k stages starting from state (x_1,x_2). Furthermore, let $V^0(x_1,x_2)=0$ for all x_1 and x_2.

To obtain the necessary result, we need to show that:

$$\min\{V(x_1-1,x_2+1),V(x_1,x_2)\}=V(x_1-1,x_2+1) \tag{9}$$

or equivalently $V(x_1-1),x_2+1)\leq V(x_1,x_2)$, whenever $x_1>0$.

The above property holds trivially for $V^0(x_1,x_2)$, now we assume that it also holds true for $V^k(x_1,x_2)$, to complete the proof we need to show that $V^{k+1}(x_1-1),x_2+1)\leq V^{k+1}(x_1,x_2)$. Note that:

$$V^{k+1}(x_1,x_2) = h_1x_1+h_2(x_2)+\lambda V^k(x_1,x_2-1)+\mu_1\min\{V^k(x_1+1,x_2),V^k(x_1,x_2)\} \\ +\mu_2\min\{V^k(x_1-1,x_2+1),V^k(x_1,x_2)\} \tag{10}$$

and

$$V^{k+1}(x_1-1,x_2+1) = h_1.(x_1-1)+h_2(x_2+1)+\lambda V^k(x_1-1,x_2) \\ +\mu_1\min\{V^k(x_1,x_2+1),V^k(x_1-1,x_2+1)\} \\ +\mu_2\min\{V^k(x_1-2,x_2+2),V^k(x_1-1,x_2+1)\} \tag{11}$$

Now we will perform a term by term comparison: firstly, since $h_1=h_2^+$, $h_1.(x_1-1)+h_2(x_2+1)\leq h_1x_1+h_2(x_2)$. Secondly, $\lambda V^k(x_1-1,x_2)\leq\lambda V^k(x_1,x_2-1)$ by the induction assumption. By the same assumption: $\mu_2V^k(x_1-2,x_2+2)\leq\mu_2V^k(x_1-1,x_2+1)$. Hence, we are left with the terms corresponding to production in the first stage. We concentrate on these terms by considering all 4 possible combinations of control actions:

Case 1: Optimal kth stage actions are to produce in stage 1 in both (x_1, x_2) and $(x_1 - 1, x_2 + 1)$. In this case, the resulting term on the right hand side of (10) is: $V^k(x_1 + 1, x_2)$ and the term on the right hand side of (11) is: $V^k(x_1, x_2 + 1)$. By the induction assumption, we have: $V^k(x_1, x_2 + 1) \le V^k(x_1 + 1, x_2)$.

Case 2: Optimal kth stage actions are not to produce in machine 1 in both (x_1, x_2) and $(x_1 - 1, x_2 + 1)$. The desired inequality is obtained exactly as in the previous case by the induction assumption.

Case 3: Optimal kth stage actions for machine 1 are to produce in state (x_1, x_2) and not produce in state $(x_1 - 1, x_2 + 1)$. This case can not happen since it contradicts the monotonicity property proved in Veatch and Wein [8] which states that if it is optimal to produce in machine 1 in state (x_1, x_2), it is also optimal to produce in machine 1 is state $(x_1 - 1, x_2 + 1)$.

Case 4: Optimal kth stage actions for machine 1 are not to produce in state (x_1, x_2) and produce in state $(x_1 - 1, x_2 + 1)$. Since the optimal action in state $(x_1 - 1, x_2 + 1)$ is to produce $V^k(x_1, x_2 + 1) \le V^k(x_1 - 1, x_2 + 1)$. However, $V^k(x_1 - 1), x_2 + 1) \le V^k(x_1, x_2)$ by the induction assumption giving the desired inequality.

We have proved that the desired property propagates through value iteration. To complete the proof, we note that the infinite horizon problem will also inherit the desired property by letting $k \to \infty$. Furthermore, under standard assumptions, limits can be taken as the discounting factor α approaches 1 to show that average cost per unit time problem also has the identical property.