SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853-3801

TECHNICAL REPORT NO. 962

February 1991
Revised May 1993

A COMPARISON OF ALTERNATIVE KANBAN CONTROL MECHANISMS: PART 1 BACKGROUND AND STRUCTURAL RESULTS[1]

by

John A. Muckstadt and
Sridhar R. Tayur

[1]This work was supported in part by National Science Foundation Grant DDM-8819542.

A Comparison of Alternative Kanban Control Mechanisms: Part 1 Background and Structural Results

John A. Muckstadt
School of Operations Research and Industrial Engineering
Cornell University, Ithaca, NY 14850

Sridhar R. Tayur
Graduate School of Industrial Administration
Carnegie Mellon University, Pittsburgh, PA 15213

May 1993

Abstract

The design of inventory control policies for serial systems is a topic currently being explored by a number of researchers. Our goal – in two papers – is to synthesize and extend some of these efforts. We consider, simultaneously, four sources of variability in production lines – processing time variability, machine breakdowns, rework and yield loss – and show some similarities and differences in their effects on the performance of the line. In this paper, we introduce an enhanced model that accomodates raw material and demand uncertainity and demonstrate that many of the sample path results obtained previously can be extended.

The main objectives of this paper are: (1) to demonstrate that Constant Work-in-Process (CONWIP) and the traditional *kanban* control are just two extremes in a *finite* family of implementable *pull* controls; (2) to show that while machine breakdowns, rework and random processing times have a *similar* effect in terms of the *optimal* decisions, yield losses in the line may have to be managed differently.

1 Introduction

During the past decade much attention has been given by both academics and practitioners to alternative approaches for controlling the flow of material in production systems. A cursory perusal of some of recent research gives an impression that various new and different approaches are being developed, and that some of them are in conflict with others. However, upon closer examination, we observe that many are only special cases of a family of controls. Furthermore, different papers emphasize specific instances of sources of variability that are present in general production systems; however, these differences are not always real. Consider the following two specific examples. First, there has been considerable discussion about CONWIP (Woodruff et al.(1990)) and its comparison with the traditional *kanban* mechanism (Schonberger(1982), Karmarkar and Kekre(1990)). Second, papers have have been written that study the effects of variability in processing times (Karmarkar and

Kekre(1990), Conway et. al (1988)), and the study of machine unreliability and demand variability (Deleersnyder et al.(1989)) on system performance. That these and other sources of variation can conceptually be considered equivalent seems to have been overlooked. Using this observation, structural results independent of processing time distributions were developed in Tayur(1992), which we extend in this paper. For an exhaustive list of references on kanban research, see Buzacott and Shanthikumar (1992).

We have two main objectives for this paper. First, we demonstrate that CONWIP and the conventional kanban control are just two extremes in a finite family of implementable *pull* control systems. Our analysis extends the structural results of Tayur(1992) by considering stochastic demand and raw material processes, and so demonstrates that many sources of variabilty can be considered to be equivalent in a more general setting.

Second, we demonstrate that there is a difference between yield losses and the sources of variability considered in Tayur(1992)–processing time variability, rework and machine breakdown. Studying the system via the structural results helps greatly in managing systems with greater complexity as it provides the intuition required to set the design parameters.

We first present a model that accomodates raw material and demand uncertainites in our framework which extends the model introduced by Mitra and Mitrani(1990).

2 The Model

We study a serial manufacturing system that uses a general *kanban* control mechanism. Processing times are variable, machine breakdowns are possible, rework may be required and yield is not perfect (the yield at any processing step is random). Further, the demand process as well as the raw material arrival process are stochastic. The undesirable effects of randomness include reducing throughput capacity, missing delivery dates and limiting the effectiveness of planning and scheduling activities. By buffering a production line (by *safety time* or *safety stock*) most of the undesirable effects of uncertainty can be mitigated. The greater the buffer capacities, the greater the protection against uncertainties; but, this protection is not without expense. Apart from the dollar value of inventory, other costs include the inability to respond quickly to changes in demand (due to long lead times) and to identify poor quality of products (as it takes time to identify the problem that caused defects). It is also well known that by locating inventory in different places in a line– different buffering strategies– the system performance can be altered considerably. Consequently, it is important to identify the best buffer capacities in a line. Thus, the trade-off lies in balancing the benefits of buffering with the costs of inventory. A kanban or pull system has two attractive qualities for line management: (1) there is a *clear control* of the amount of inventory at each location, and (2) the kanban mechanism reacts *dynamically* and *immediately* to a yield loss and other sources of variability. Because of these attributes, many variants of the mechanism discussed here have already been successfully implemented around the world.

The serial production line we will study consists of M machines arranged in a series (or in tandem). These M machines are partitioned into N *cells*. Each cell consists of a set of machines grouped together such that the total number of kanbans for this group is fixed. Thus, a cell is simply a kanban loop. A *cell partition* is a collection of non-overlapping and collectively exhaustive groups of consecutive machines. If all the M machines are in the same *cell*, we have a CONWIP (CONstant-Work-In-Process) type control system; if, on the other hand, there are a total of M cells, each cell containing exactly one machine, we have a *traditional kanban control system* (TKCS). To formalize our ideas and to make our exposition precise, we introduce the following (mathematical) description of a control system.

We will use N/$(M_1, \ldots, M_N)/(C_1, \ldots, C_N)$/(B) to denote a serial production line with N cells, M_i machines in cell i, C_i white kanbans in cell i, $i = 1 \ldots$ N and B colored cards that circulate throughout the shop floor. By *allocation* we mean the vector $(C_1, \ldots, C_N)$, and by *partition* we mean $(M_1, \ldots, M_N)$. The set $\{$ N/$(M_1, \ldots, M_N)/(C_1, \ldots, C_N)$: $\sum_{i=1}^{N} M_i = M, M_i \geq 1, \sum_{i=1}^{N} C_i = C, C_i \geq 1, N \leq M\}$ contains all possible *configurations* for a line with M machines and C white kanbans. Using this notation, we see that CONWIP is 1/(M)/(C)/(.) system, and TKCS is a M/$(1, \ldots, 1)/(C_1, \ldots, C_M)$/(.) system. All other configurations give rise to other possible designs within this *family* of controls. Henceforth, we will refer to the general control scheme as *kanban* control.

We briefly describe the essentials of a single-product kanban controlled system. The description is in two stages. First, we show how material is moved within a cell and from one cell to another (Figure 1), and second, we show how a signal of a satisfied demand pulls raw material into a cell (Figure 2).

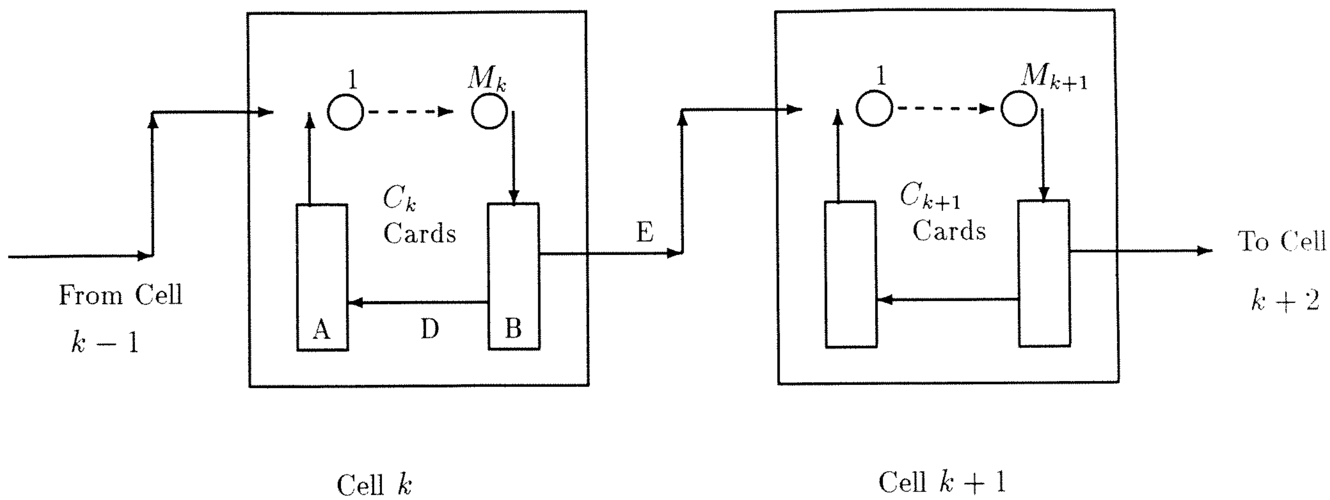
As shown in Figure 1, a cell consists of

1. machines in tandem – the processing times on the machines may be stochastic, and all parts go through each machine exactly once.

2. an output hopper – in which batches of material that have completed all operations in the cell (and have not suffered a complete loss) wait for withdrawal by the successor cell.

3. a bulletin board – where requests are posted for material from the predecessor cell, in the form of kanbans. (We assume that these kanbans are white in color.)

The product moves through the line in batches, which can be of size one. The service discipline is first-come, first-served, and each machine can process only one part at a time. No preemptions are allowed. The parts completed in cell $k-1$ become the input material for cell k, for k=2,$\ldots, N$. A batch must acquire one of these white cards in order to enter the cell, and must continue to hold it throughout its stay in that cell. After a batch has been completed in cell k, it is placed in the output hopper with its white kanban, awaiting admission into the next cell. If there is a *complete* yield loss at a particular machine in a cell (say in cell k, *all* items in the batch are scrapped), then the batch is thrown away and the white kanban that was attached to this (rejected) batch is placed on the bulletin board of cell k, signalling a need for replenishment. This *immediate* pull response to a yield loss is an attractive quality of this mechanism. (If at the end of a processing step a batch contains at least one good item, then it is sent to the next processing stage. The determination of the



A: Bulletin Board of Cell k

B: Output Hopper of Cell k

D: Card traveling from Output Hopper to Bulletin Board

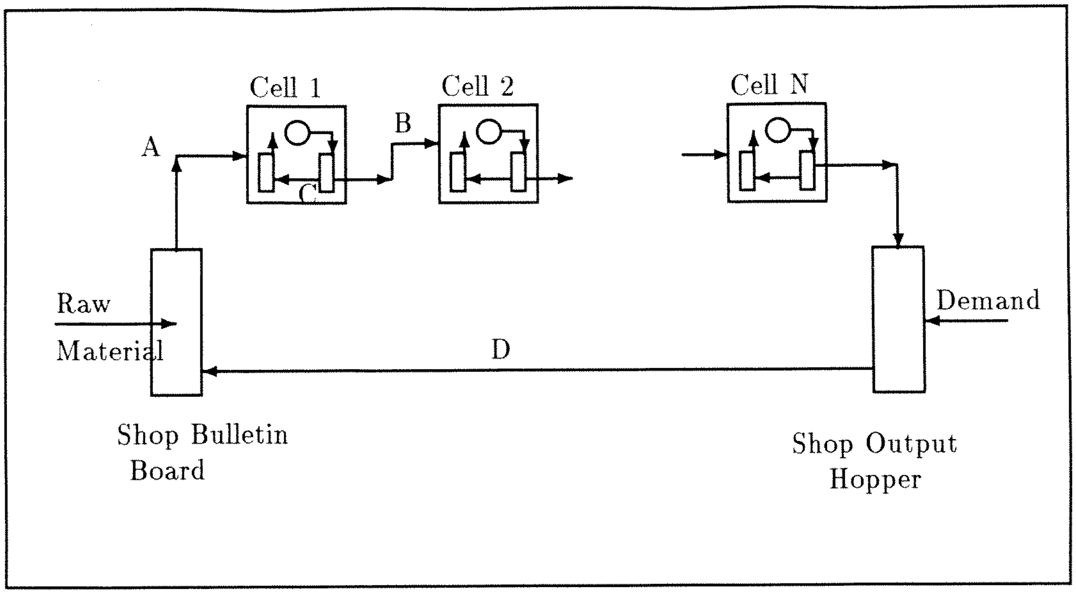
E: Part transferred to Cell $k+1$ from Cell k

Figure 1: Cell Description.

number of *non-defective* items in a batch is made at the end of the processing of the batch.) Both rework and machine breakdowns are accommodated by a suitable change to the form of the processing time distribution at a machine.

Note that the mechanism is *pull* between cells, and *push* within a cell. Also note that it is not possible for both the output hopper of cell k and the bulletin board of cell k+1 to be simultaneously non-empty. If a white kanban is present on the bulletin board of cell $k+1$, and a batch is available in the output hopper of cell k, the batch would be moved to the queue in front of the first machine in cell $k+1$ along with the white kanban from the bulletin board of cell $k+1$. Thus, the maximum inventory possible in cell k is C_k batches, and no inventory can sit between adjacent cells. This is how white kanbans control inventory in the cells. When a completed part is withdrawn to the next cell (cell k+1) the white kanban of cell k stays within the cell, and is posted on the bulletin board of cell k. This is a signal to the preceding cell, cell k-1, that cell k needs a part. Thus, white kanbans also serve as an information system that controls material transfer between successive cells.

As shown in Figure 2, in addition to the cells described above, there are colored cards that circulate through the shop, a shop bulletin board and a shop output hopper. To illustrate how the system functions, at time zero start with the shop output hopper full–thus all the white cards are in their respective bulletin boards. Any raw material is waiting before the shop bulletin board. If a demand occurs, the part leaves the shop output hopper, and the colored card is placed in the shop bulletin board. If there is raw material, and there is a white



A: Colored Card and Raw Material

B: Colored card and semi-finished product

C: White Card

D: Colored Card

Figure 2: Model of the shop floor.

card in the bulletin board of cell 1 (for the first demand there always will be), then the raw material and the colored card enter cell 1. From Cell 1 to Cell N, the part and the colored card never separate, and the transfer between cells is the same as described earlier. After processing in cell N, the part and the colored card are placed in the shop output hopper. Thus, Cell N's output hopper is never used. If a batch is scrapped entirely, the colored card is placed on the shop bulletin board. Note that a white card in the bulletin board of cell 1 implies either there is no raw material or there is no colored card. The throughput of the system depends on the mean of the demand process. Note that backorders of demand can become arbitrarily large, and as the raw material process is assumed exogenous, the raw materials can pile up too, if not co-ordinated well. Increasing the number of colored cards decreases the waiting time of the demands, while increasing the white cards improves the maximum achievable throughput of the line.

In our context, then, the problem of buffering a M machine serial production line is equivalent to partitioning the line into N cells, allocating a certain number of white kanbans to each cell and allowing a certain number of colored cards in the system. There is no reason, *apriori*, to expect any one control from the above *family* to be superior to all others in all possible scenarios. In particular, neither CONWIP nor the traditional kanban control can claim superiority over another in all situations. However, some controls will be superior to others for particular objectives. We will demonstrate this fact subsequently and in the sequel to this paper (Muckstadt and Tayur 1993).

The remainder of this paper is organized as follows. In section 3, using sample path methods, we (1) provide structural results for allocation of white cards to cells, (2) demonstrate that reversibility is lost in the presence of yield losses and (3) show that the optimal partition is CONWIP when there is a constraint on the total number of white cards. We also provide a partial characterization of the sequence of machines in the presence of yield losses. The sequel to this paper (Muckstadt and Tayur(1993)) discusses the allocation and partition decisions when the constraint is on average inventory, where we show that for balanced lines, TKCS may be the best strategy. Section 4 concludes this paper. all proofs are provided in the appendix.

Remark. This model can be extended to a multi-product setting, by having different sets of colored cards for different products. There is now an additional set of decisions to be made. We must determine the sequence in which jobs at machines should be processed when there is more than one type of job awaiting processing. Note that the white cards will limit the number of parts in the cell, as before. Two loading rules–static and dynamic– are being analyzed in a parallel work along with the interaction between white and colored cards. The effect of setup times on batch sizes is also relevant in multi-product lines. These issues will not be discussed here.

3 Structural Results

The structural results of Tayur(1992) are extended and the effects of yield losses are studied in the subsections that follow. The sample path description of the line is provided in Appendix A; the proofs of the theorems are in appendices B-E. As will become clear by the sample path analysis provided in the appendix, the results of this section are independent of the demand variations.

3.1 Dominance

In this section we show that certain allocations of white kanbans are superior to others in any setting in the sense that we can conclude that a particular allocation is preferred (the average waiting time of a demand will be lower) to another without any knowledge of the processing time distributions of the machines or on the yield probabilities. This result is called *dominance*, and helps reduce the computational work required in simulation experiments by a large factor. This result indicates how white kanbans in a given cell affect other cells. The proof uses stochastic ordering ideas. For a description of stochastic ordering ($\leq_{st}$), see Stoyan(1983). In the theorem below, W_n and $\hat{W}_n$ stand for the waiting time of nth demand in a $N/(1,\ldots,1)/(.)$ line with allocation $(C_1,\ldots,C_N)$ and $(C'_1,\ldots,C'_N)$ respectively. As the proof is similar to that in Tayur(1992), we omit the details here.

Theorem 1 *Given a $N/(1,\ldots,1)/(.)$ line, allocations $(C_1,\ldots,C_N)$ and $(C_1',\ldots,C_N')$, define, for all $k \leq N$ and all k-$2 \leq j \leq k$-1,*

$C(j;k) = \sum_{i=j}^{k} C_i$ *and*

$C(j;k)' = \sum_{i=j}^{k} C_i'$.

Then, if

$C(j;k) \geq C(j;k)' \quad \forall\, j,k$

we have

$W_n \leq_{st} \hat{W}_n, \quad \forall n \geq 1.$

If the hypothesis of Theorem 1 holds, we say that the allocation $(C_1',\ldots,C_N')$ is dominated by the allocation $(C_1,\ldots,C_N)$.

Example 1 *Consider a five cell line with a total of 11 cards to be allocated. Consider the following five feasible allocations:*

(3,2,2,2,2), (2,3,2,2,2), (2,3,2,3,1), (1,4,2,2,2) and (1,4,2,3,1).

It is easily verified that (1,4,2,3,1) dominates the rest in the sense of theorem 1. This implies that whatever be the five machines in the line, allocation (1,4,2,3,1) yields the lowest average waiting time among the above five candidates.

This is analogous to the dominance result obtained in Tayur(1992) where there were no yield losses. The interesting feature of the result is the following: the structure of the optimal allocation to a line is *not affected by presence of yield losses*, but is determined by the overall variability on the machines. Thus, all the sources of variation can be aggregated. Of course, the optimal allocation may change depending on the yield probabilities, but that is not a loss in structure. The intuitive explanation of the *dominance* result is therefore as in Tayur(1992). Briefly, unlike the *traditional* buffered tandem lines, the kanbans in cell k are capable of providing buffering on both upstream and downstream of the same machine. This *dynamic* buffering by the cards is the reason why we need to consider two and three consecutive cells at a time.
As in Tayur(1992), the following corollaries are immediate.

Corollary 1 *Increasing the number of kanbans in any cell decreases the waiting time.*

Corollary 2 *A uniform allocation of white kanbans to cells is not optimal unless $N = C$ or $N = 2$.*

Corollary 3 *In systems with three or more cells, optimal allocations have exactly 1 white card in each of the two end cells.*

Corollary 4 *In a two cell system, every allocation of a fixed number of white cards yields the same throughput.*

Corollary 5 *In a three cell system the optimal allocation for C white cards is 1,C-2,1.*

As in Tayur(1992), we need to allocate exactly one card to the end cells if $N \geq 3$. Thus, the number of feasible allocations that cannot be dominated is $\binom{C-3}{N-3}$. This is because we first allocate N cards, one to each cell, and then we are free to allocate the remaining $C - N$ cards to $N - 2$ cells in any manner. In contrast, by a similar argument, the total number of feasible allocations is $\binom{C-1}{N-1}$.

3.2 Reversibility

In the previous section we showed that the *structure* of white kanban allocation to a fixed sequence of machines was not affected by the presence of yield losses. Unfortunately, the same cannot be said about the *structure* regarding the optimal *sequence* of the machines. In particular, we show here that the line operated by the kanban scheme is not reversible in general when there are yield losses. This leads naturally to the issue of the optimal sequencing of a set of machines when the objective is to minimize the waiting time, when the total number of cards to be allocated is fixed. We prove a characterization of this issue.

As we commented at the end of Appendix A, as long as the demand and the raw material processes are exogenous to the line, we can assume them to be infinite. For this sub-section, we will continue to assume that this is the case. We need the following definitions.

The *capacity* of the line is defined as the expected departure rate from the last station when there is an infinite supply of raw material and infinite demand. Thus, the capacity of the line is the maximum possible expected throughput rate given a fixed configuration.

A reversed system is one which has its cells in the reverse order of the original system.

Definition The line is said to be C-reversible if the original system has the same capacity as its reversed system.

Definition The line is said to be D-reversible if the distribution of the nth departure epochs (out of the line) in both lines are identical for every n, both systems starting empty at time zero.

Note that D-reversibility implies C-reversibility.

Theorem 2 *The line $N/(1, \ldots, 1)/(C_1, \ldots, C_N)$ is not D-reversible in general if there are yield losses.*

The proof is provided in Appendix B. That C-reversibility is not preserved either can be seen by the following example.

Example 2 *Consider a two machine line with machine A feeding machine B, represented by (A,B). Let machine A have a yield loss with probability p at any service completion, and a* *processing time that is deterministic with a value d_A, and let machine B have deterministic processing time of d_B. Let $d_B > d_A$. Then, it can be seen that sequence (A, B) has an average a higher throughput rate than the sequence (B, A). Notice, however, that if $d_A = d_B$, we have reversibility.*

The natural question at this point is to determine whether given a set of machines, it is possible to characterize the *sequence* that maximizes throughput for a fixed allocation of cards. The proof of Theorem 2 (appendix A) gives a hint that such a characterization is possible: in a two machine system, the machine (if any) that does not have yield losses, should go in second in the sequence. This is independent of the processing times of the two machines. This naturally implies the following.

Corollary 6 *In a line with N machines with exactly one machine that has yield losses, this machine will be first in an optimal sequence.*

Yield losses, however, can occur in more than one machine. In this case, the best sequence is answered partially by the next theorem. We are interested in stochastic ordering ($\leq_{st}$) of the departure epochs of the *nth* good part out of the lines (each line is a different sequence of a given set of machines). Note that altering the sequence may not always be possible due to engineering reasons; this result should be used to identify the order in which improvements in processing time and yield losses to a line will provide maximum benefits. See Stoyan(1983) for the definition of $\leq_{st}$ and $\leq_{lr}$ orderings. The proof is provided in Appendix C; we need some notation here.

Define

$$Y_n^{(j)} = \begin{cases} 1, & \text{if there is a complete yield loss at the nth service at machine j,} \\ 0, & \text{otherwise,} \end{cases}$$

and (with $S_0^{(j)} = 0$)

$$\begin{aligned} r_n^{(j)} &= \min\{k \geq 1 : Y^{(j)}_{k+S^{(j)}_{n-1}} = 0\} \\ S_n^{(j)} &= S_{n-1}^{(j)} + r_n^{(j)}, \end{aligned}$$

that is, $S_n^{(j)}$ represents the total number of service completions on machine j required to complete the *nth* good batch, and $r_n^{(j)}$ counts the total number of service completions on machine j, after the $(n-1)st$ good batch has been produced, to obtain the *nth* good batch at this machine. Let X_n^j denote the service time of the nth part at machine j.

Theorem 3 *If a sequence has* $\sum_{k=1}^{S_1^{(m+1)}-1} \sum_{j=1}^{r_k^{(m)}} X^m_{S^m_{(k-1)}+j} \leq_{lr} \sum_{j=1}^{S_1^{(m)}-1} \sum_{j=1}^{r_k^{(m+1)}} X^{m+1}_{S^{(m+1)}_{k-1}+j}$, *for all consecutive pairs* $(m, m+1)$, *where* $m = 1 \ldots N-1$, *then it is optimal.*

That the conditions imposed by the above theorem are not stringent is shown by the next result. The proof of Lemma 1 is provided in Appendix D.

Lemma 1 *Let α and β be two independent geometrically distributed random variables with parameters p_1 and p_2, respectively, and let α_i and β_j be independent and identical copies of α and β respectively. Then, if $p_1 > p_2$, we have the following relation between the random sums of random variables:*

$$\sum_{i=1}^{\alpha-1} \beta_i \quad \leq_{lr} \quad \sum_{i=1}^{\beta-1} \alpha_i.$$

Combining the above two results, we obtain the following theorem. The proof is in Appendix E.

Theorem 4 *In a line with* **N** *machines having identical processing time distributions, and yield losses that are bernoulli for machine* **m** *(at every service completion), the optimal sequence of the machines has the following property: if machine* **i** *precedes machine* **j***, then $p_i \leq p_j$, where p_m is the probability of a good part at machine m.*

The intuitive explanation of the results of this sub-section can be summarized as follows. Given a set of machines with yield losses, one should sequence these machines so that the machine with *highest* yield loss and *smallest* processing time is *first*, the machine with the *second* highest yield loss and *second smallest* processing time is *second*, and so on. Thus, the *last* machine in the sequence has the *largest* processing time and the *smallest* yield loss. This result is intuitive because one wishes to minimize the *wasting* of capacity due to a yield loss (all the processing in previous stages is capacity used up for a batch that will be discarded), and also prefers to minimize the *time to replenish* a part. This would imply that one should not have high yield losses later on in the sequence, and that the processing times should be smaller in earlier stages. Note that in reality this inverse relation between yield loss and processing times cannot be expected to hold; however, this provides intuition as to where to expend appropriate effort to improve the performance of the line.

We can now explain the relative importances of *dominance* and *reversibility* results. Kanban *allocation* is important when the processing times are variable, and machine *sequencing* is important when there are yield losses. Thus, in a line with no yield losses, the kanban *allocation* is critical. Conversely, in a line with no processing time variability, the *sequence* of machines is critical. In general, both issues must be considered simultaneously.

3.3 Some Results for the $N/(M_1, \ldots, M_N)/(C_1, \ldots, C_N)$ system.

In the previous part of this section we restricted attention to the case of N cells and one machine per cell. We now state some results for other, more general cases. As the proof is essentially by induction on the recursions that characterize the system's dynamics, we omit the details.

Theorem 5 *(a) For a 2/(1, M) / (C_1, C_2) system with the total number of kanbans equal to C, the mean waiting time is minimized when C_1 = 1 , and C_2 = C - 1.*

(b) For an $N/(1, M_2, \ldots, M_{N-1}, 1)\,/(C_1, \ldots, C_N)$ system with the total number of kanbans $= C$, the average waiting time is minimized when $C_1 = C_N = 1$.

(c) In an $N/(M_1, \ldots, M_N)/(C_1 \ldots C_N)$ line, increasing C_i in any cell i decreases mean waiting time.

We end this section with another structural result. We compare different partioning strategies for a given line and a fixed total number of kanbans. The proof is provided in Appendix E.

Theorem 6 *Let $L = \{ N/(M_1, \ldots, M_N)/(C_1, \ldots, C_N)/(.): N \leq M, \sum_{i=1}^{N} C_i = C, C_i \geq 1, \sum_{i=1}^{N} M_i = M, M_i \geq 1 \}$ be the set of possible configurations for a given serial line with M machines and C kanbans, and let L^* be the configuration that yields the stochastically smallest waiting time (for every nth demand). Then, $L^* = 1/(M)/(C)/(.)$.*

There is a catch to the comparison made above, namely, that the average inventory is not the same in all of the configurations. When an attempt is made to minimize mean waiting time among all configurations of equal average inventory, then 1/(M)/(C) may not be the best strategy. This is discussed in detail in the sequel to this paper. Thus, in this paper, we have only resolved the issue of minimizing the waiting time under a constraint on the maximum allowable inventory in the cells, but not for a constraint on the average inventory.

4 Summary

Having described a number of general results, we briefly summarize the results of our paper.

1. If I have complete freedom to choose the number of cells and the number of machines I can place in them, then should I put them all in one cell or should I put exactly one machine in every cell and thus have many cells?

   If your objective is to minimize the number of cards, then have only one cell.

2. If I can partially alter the sequence in which the operations (machines) are performed, then what is a good sequence to select?

   If there are yield losses, use the intuition developed in section 3.

3. I have some machines that breakdown, while others have wide variance in processing times. How is the control strategy affected by these two different sources of variability?

   It isn't. These are different manifestations of variance that have the similar effect on structural properties (and differ only in degree).

4. Some processes require rework, while others cause scrap. How do I design the system to account for these?

   Rework is equivalent to processing time variation. Use the same strategy as above. Yield loss is different; sequence the machines appropriately as discussed earlier. Observe that the kanban mechanism will react dynamically to these variabilities.

In this paper, we have considered four main sources of variability: processing time variation, rework requirement, machine breakdowns and yield losses. Using an extension of available models, we have shown that many structural results carry over to more general and realistic settings. We have also demonstrated the similarities and differences between the sources of variabilty. The sequel to this paper (Muckstadt and Tayur (1993)) studies several different objectives via simulation and complements the results obtained here.

Appendix

A Sample Path Description

We begin with a sample-path description of the system dynamics for a line with N cells and exactly one machine per cell. The recursions developed here are used in the proofs of the results of this section. For the time being, assume that every part that completes production in cell N is shipped immediately (infinite demand rate) and that raw material is always available. At the end of the *nth* service at machine j, it is determined whether or not the batch just produced is good or bad. If it is good (at least one item in this batch is good), then it is placed in the outout hopper of cell j with its *kanban*; otherwise, the batch is discarded, and the *kanban* that was attached to this batch is placed in the bulletin board of cell j. A service process is on a *batch*.

Let $(C_1, .., C_N)$ be a kanban allocation.

Start the system at time 0 with C_j cards on the bulletin board of cell $j, j \geq 2$. Since raw materials are assumed to be available, all C_1 cards are initially in front of machine 1 in cell 1.

Define

$A_n^{(j)}$ = time when a white card arrives to the bulletin board in front of cell j for the nth time

and

$D_n^{(j)}$ = time nth service process at machine j is completed.

$A^{(j)}$ and $D^{(j)}$ with negative subscripts are to be assumed as zero. Note that

$$D_n^{(j)} \leq D_n^{(j+1)}, \quad \text{and}$$
$$D_n^{(j)} \leq D_{n+1}^{(j)},$$

by the definitions above.

Because of yield losses, the nth service at a machine need not correspond to the nth *good* (non-rejected) batch out of that machine. To account for this, define

$\tilde{A}_{n+C_j}^{(j)}$ = time when nth batch arrives into the queue in front of machine $j+1$

and

$\tilde{D}_n^{(j)}$ = departure time of *nth good* batch after receiving service from machine j.

Note that the *nth* good part is completed at machine j before the *nth* service completion at machine $j+1$ and the $(n+1)st$ good part at machine j can be completed only after the *nth* good part has been completed on machine j. Thus, we have

$$\tilde{D}_n^{(j)} \leq D_n^{(j+1)}, \qquad \text{and}$$
$$\tilde{D}_n^{(j)} \leq \tilde{D}_{n+1}^{(j)},$$

by the definitions above. Further, we have

$$A_1^{(1)} = . . = A_{C_1}^{(1)} = 0,$$
(because all white cards at time 0 are on the bulletin board)
$$A_{C_1+1}^{(1)} = D_1^{(1)},$$
(because there are C_2 cards on bulletin board of cell 2 at time 0)
$$\vdots$$
$$A_{C_1+C_2}^{(1)} = D_{C_2}^{(1)}, \qquad \text{and}$$
$$A_{C_1+C_2+1}^{(1)} = \max(\ D_{C_2+1}^{(1)},\ A_{C_2+1}^{(2)}).$$
$$\vdots$$

The explanation for the last two equations are as follows. Because there are C_2 cards on the bulletin board of cell 2 at time 0, the arrival of any of the first $C_1 + C_2$ white cards to bulletin board of cell 1 is not affected by service completion at cell 2. In general, we have $A_1^{(j)} = \ldots = A_{C_j}^{(j)} = 0$ for all $j = 1, \ldots, N$, because at time zero all white cards are at their respective bulletin boards. Further,

$$A_n^{(j)} \quad = max(D_{n-C_j}^{(j)}, A_{n-C_j}^{(j+1)}), \qquad j = 1,..N-2, \tag{1}$$
$$A_n^{(N-1)} = max(D_{n-C_{N-1}}^{(N-1)}, D_{n-C_{N-1}-C_N}^{(N)}). \tag{2}$$

Define $X_n^{(j)}$ as the service time of the nth batch at the jth machine. The nth departure can take place only after a service is completed for the nth time, and the service for the nth time can only begin after both the nth part has arrived at a machine and the $(n-1)$st service at that machine has been completed. Thus:

$$D_n^{(j)} \quad = max(\tilde{A}_{n+C_{j-1}}^{(j-1)}, D_{n-1}^{(j)}) + X_n^{(j)}, \quad j \geq 2, \tag{3}$$
$$D_n^{(1)} \quad = max(A_n^{(1)}, D_{n-1}^{(1)}) + X_n^{(1)}. \tag{4}$$

Similarly, a part arrives at queue $j+1$ for the $(n-C_j)th$ time only after the $(n-C_j)$th good part has been produced on machine j and a white card is available on bulletin board of cell $j+1$. Thus:

$$\tilde{A}_n^{(j)} \quad = max(\tilde{D}_{n-C_j}^{(j)}, A_{n-C_j}^{(j+1)}), \qquad j = 1,..N-2, \tag{5}$$
$$\tilde{A}_n^{(N-1)} = max(\tilde{D}_{n-C_{N-1}}^{(N-1)}, D_{n-C_N-C_{N-1}}^{(N)}). \tag{6}$$

We need to connect $\tilde{A}_n^{(j)}$ and $A_n^{(j)}$ with $\tilde{D}_n^{(j)}$ and $D_n^{(j)}$. To do that, define

$$Y_n^{(j)} = \begin{cases} 1 & \text{if complete yield loss at the nth service at machine j} \\ 0 & \text{otherwise} \end{cases}$$

and (with $S_0^{(j)} = 0$)

$$\begin{aligned} r_n^{(j)} = & \min\{k \geq 1 : Y^{(j)}_{k+S^{(j)}_{n-1}} = 0\} \\ S_n^{(j)} = & S^{(j)}_{n-1} + r_n^{(j)}, \end{aligned}$$

that is, $S_n^{(j)}$ represents the total number of service completions on machine j required to complete the *nth* good batch, and $r_n^{(j)}$ counts the total number of service completions on machine j, after the $(n-1)st$ good batch has been produced, to obtain the *nth* good batch at this machine. We have, then

$$\tilde{D}_n^{(j)} = D^{(j)}_{S_n^{(j)}}.$$

If we had a batch size of D at the input of cell 1, then at the end of a *batch* service at every cell j, we determine the *current* batch size (as a few might be lost at each processing step). As long as there is at least *one* good item left in the batch, the batch will be transferred to the output hopper of cell j with its *kanban*. If *all* items are bad, then the batch is discarded and its *kanban* is placed in the bulletin board of cell j. This is a protocol known as *no lot-splitting.* Under this protocol, the results developed in the next few sections will hold. We also need the following notation. Let $B_n^{(0)} = D$ (the input batch size at cell 1), $Z_n^{(j)}$ the fraction lost in processing the *nth* batch at stage j, and (with $\lfloor\gamma\rfloor$ to denote the largest *integer* less than or equal to γ)

$$B_n^{(j)} = \lfloor B^{(j-1)}_{S_n^{(j-1)}} Z_n^{(j)} \rfloor.$$

Then, at the *nth good* (not complete loss) service completion at machine j, the total number of parts that have successfully completed the first j processing steps is

$$T_n^{(j)} = \textstyle\sum_{k=1}^n B^{(j)}_{S_k^{(j)}}.$$

It can be verified at once that it is sufficient to keep track of the departure epochs of good batches and it is not necessary to keep track of individual items in a batch. In what follows we will, therefore, assume that $D = 1$.

We now include the stochastic raw material and demand processes. Let B be the number of colored cards. Let $\{E_n, n \geq 1\}$ and $\{R_n, n \geq 1\}$ be the epochs of demand and raw material arrivals. Let, at time 0, all B colored cards be on the shop output hopper, with the finished goods. Consequently, all white cards are at their respective bulletin boards. Thus, W_n, the waiting time of the n^{th} customer, is given by $(D^M_{n-B} - E_n)_+$ for a line with no yield loss at the last machine, where $(a)_+$ is $\max(a, 0)$. To understand how the stochastic raw material and demand processes enter the recursions, consider a $2/(1,1)/(C_1, C_2)$ line with no yield losses:

$$D^1_n = \max(D^1_{n-1}, E_n, R_n, A^1_n) + X^1_n \tag{7}$$
$$D^2_n = \max(D^2_{n-1}, D^1_n) + X^2_n \tag{8}$$
$$A^1_{n+C_1} = \max(D^1_n, D^2_{n-C_2}). \tag{9}$$

It now is apparent that to prove results about waiting times in the general setting, we need only to prove the results about departure times for the case with infinite raw material and demand. In what follows, therefore, we will concentrate on the departure epochs of the n^{th} good part out of the N^{th} cell. Furthermore, we will suppress the dependence on B for the rest of the paper.

B Proof of Theorem 2

Proof. It suffices to consider a two cell system with one machine having yield losses (machine A), and the other without yield losses (machine B), and one kanban in each cell. We prove the result using a sample path argument. We show that the first time there is a yield loss, there is a difference in the distribution of the output if the sequence of the machines was reversed. We know from Tayur(1992) that this line would have been reversible if there were no yield losses. Thus, without loss of generality we can assume that the first item processed on machine A suffered a yield loss.

Let the first sequence be (A, B) and the second sequence be (B, A). Then, we have for the sequence (A, B),

$$\tilde{D}^{(A)}_1 = (\sum_{i=1}^{S^A_1} X^A_i)$$
$$\tilde{D}^{(B)}_1 = (\sum_{i=1}^{S^A_1} X^A_i) + X^B_1$$

and for sequence (B, A),

$$\hat{\tilde{D}}^{(B)}_1 = \hat{X}^B_1$$
$$\hat{\tilde{D}}^{(B)}_2 = \hat{X}^B_1 + \hat{X}^B_2$$
$$\hat{\tilde{D}}^{(B)}_n = \hat{X}^B_1 + \sum_{i=1}^{n-1} max(\hat{\tilde{D}}^{B}_{n-1}, \hat{D}^A_{n-2})$$
$$n = 3, \ldots, S^A_1$$

$$\hat{\hat{D}}_1^{(A)} = \sum_{i=1}^{S_1^A-1} max(\hat{X}_{i+1}^B, \hat{X}_i^A) + \hat{X}_1^B + \hat{X}_{S_1^A}^A$$

Note that X_n^A and $\hat{X}_{S_1^A-n+1}^A$ for $n = 1 \dots S_1^A$ are identically distributed, and X_1^B has the same distribution as $\hat{X}_1^B$. Now, it is can be verified that $\hat{\hat{D}}_1^{(A)}$ is *larger* than $\tilde{D}_1^{(B)}$ in *distribution*, which proves the theorem.

□

C Proof of Theorem 3

Proof. The crux of the proof lies in an *adjacent pairwise interchange* argument on the sample paths. It is again sufficient to consider a two machine system with one *kanban* in each cell. The definitions of $\leq_{lr}$ and $\leq_{st}$ ordering, and some of their elementary properties can be found in Stoyan(1978), and Shanthikumar and Yao(1990).

Label the machines A and B, and we will compare the sequence (A, B) with the sequence (B, A). Again, we first consider the departure times of the first *good* output from both sequences. The following can be derived in a straightforward manner ($\hat{D}$ corresponds to sequence (B,A)).

$$\tilde{D}_1^{(B)} = \sum_{i=1}^{S_1^A} X_i^A + \sum_{j=1}^{S_1^B-1} max(\sum_{i=1}^{r_{j+1}^A} X_{i+S_j^A}^A, X_j^B) + X_{S_1^B}^B$$

$$\hat{\hat{D}}_1^{(A)} = \sum_{i=1}^{\hat{S}_1^B} \hat{X}_i^B + \sum_{j=1}^{\hat{S}_1^A-1} max(\sum_{i=1}^{\hat{r}_{j+1}^B} \hat{X}_{i+\hat{S}_j^B}^B, \hat{X}_j^A) + \hat{X}_{\hat{S}_1^A}^A$$

Observe that $X_{S_1^A}^A$ and $\hat{X}_{\hat{S}_1^A}^A$ have the same *distribution* and so do $X_{S_1^B}^B$ and $\hat{X}_{\hat{S}_1^B}^B$. In general, X_k^A and $\hat{X}_l^A$ have the same distribution for $k = 1 \dots S_{S_i^B}^A$, and $l = 1 \dots \hat{S}_1^A$. Similarly, X_k^B and $\hat{X}_l^B$ have the same distribution for $k = 1 \dots S_i^B$, and $l = 1 \dots \hat{S}_{\hat{S}_1^A}^B$. Also, note that $r_j^A, \hat{S}_1^A, \hat{r}_k^B$, and S_1^B are all ≥ 1 for $j = 1 \dots S_1^B$, and $k = 1 \dots \hat{S}_1^A$.

We have from Shanthikumar and Yao(1990) that if for two sequences of random variables $\{P_i\}$ and $\{Q_i\}$ that are ordered by $\leq_{lr}$ for every i, and for two other (discrete) random variables α, and β(independent of each other and of the above sequences) that satisfy $\alpha \leq_{lr} \beta$, we have $\sum_{i=1}^{\alpha} P_i \leq_{lr} \sum_{i=1}^{\beta} Q_i$. Combining the above with the fact that $\leq_{lr} \Rightarrow \leq_{st}$, and the hypothesis of the theorem, we obtain by straightforward comparison of the terms the ordering of the departure epochs of the first *good* output out of the lines.

Similar analysis on the departure epochs of the *nth* good output from the two lines gives the desired result. These departure epochs can be also be derived in a straightforward manner. We have, for example for the sequence (A, B),

$$
\begin{aligned}
\tilde{D}_n^{(B)} &= \sum_{p=0}^{n-1}(max(X^B_{S^B_p}, \sum_{i=1}^{r^A_{S^A_{S^B_p}+1}} X^A_{i+S^A_{S^B_p}}) \\
&+ \sum_{j=1}^{r^B_{p+1}-1} max(\sum_{i=1}^{r^A_{S^A_{S^B_p}+j+1}} X^A_{i+j+S^A_{S^B_p}}, X^B_{j+S_p})) + X^B_{S_n}
\end{aligned}
$$

where all terms with subscript 0 are zero (such as $S^B_0 = 0$, and $X^B_{S^B_0} = 0$).

□

D Proof of Lemma 1

Proof. We have, for $n \geq 1$,

$$
\begin{aligned}
\text{Prob}(\beta_i = n) &= v_n = & p_2(1-p_2)^{n-1} \\
\text{Prob}(\alpha_i = n) &= u_n = & p_1(1-p_1)^{n-1}
\end{aligned}
$$

and consequently for $n \geq 1$ (noting that $\alpha = 1$, and $\beta = 1$ do not contribute to the sums),

$$
\begin{aligned}
\text{Prob}(\sum_{i=1}^{\alpha-1} \beta_i = n) &= \sum_{k=1}^{n} u_{k+1}(\sum_{l_1,\ldots l_k \geq 1, \sum_{i=1}^{k} l_i = n} \frac{n!}{l_1!\ldots l_k!} v_{l_1}\ldots v_{l_k}) \\
\text{Prob}(\sum_{i=1}^{\beta-1} \alpha_i = n) &= \sum_{k=1}^{n} v_{k+1}(\sum_{l_1,\ldots l_k \geq 1, \sum_{i=1}^{k} l_i = n} \frac{n!}{l_1!\ldots l_k!} u_{l_1}\ldots u_{l_k})
\end{aligned}
$$

Observe that $(\sum_{l_1,\ldots l_k \geq 1, \sum_{i=1}^{k} l_i = n} \frac{n!}{l_1!\ldots l_k!} v_{l_1}\ldots v_{l_k})$ equals $(p_2)^k(1-p_2)^{n-k}\begin{pmatrix} n-1 \\ k-1 \end{pmatrix}$ and $(\sum_{l_1,\ldots l_k \geq 1, \sum_{i=1}^{k} l_i = n} \frac{n!}{l_1!\ldots l_k!} u_{l_1}\ldots u_{l_k})$ equals $(p_1)^k(1-p_1)^{n-k}\begin{pmatrix} n-1 \\ k-1 \end{pmatrix}$. This implies that, by the binomial theorem,

$$
\begin{aligned}
\text{Prob}(\sum_{i=1}^{\alpha-1} \beta_i = n) &= P(n) = & p_1 p_2(1-p_1)(1-p_1 p_2)^{n-1} \\
\text{Prob}(\sum_{i=1}^{\beta-1} \alpha_i = n) &= Q(n) = & p_1 p_2(1-p_2)(1-p_1 p_2)^{n-1}
\end{aligned}
$$

which implies that $\frac{P(n)}{Q(n)} = \frac{1-p_1}{1-p_2} \quad \forall n \geq 1$. Finally, notice that $\frac{P(0)}{Q(0)} = \frac{p_1}{p_2}$, which shows that $\sum_{i=1}^{\alpha-1} \beta_i \leq_{lr} \sum_{i=1}^{\beta-1} \alpha_i$.

□

E Proof of Theorem 4

Proof. The distribution of r_n^m is i.i.d for every $n \geq 1$, for a fixed m, and is geometrically distributed with parameter p_m. As the processing times are i.i.d for *all* machines m, the comparison in Theorem 3 is equivalent to the comparison of $\sum_{i=1}^{S_1^m - 1} r_i^{m+1}$ with $\sum_{i=1}^{\tilde{S}_1^{m+1}-1} \tilde{r}_i^m$.

□

F Proof of Theorem 6

Proof. It is sufficient to compare the single cell configuration with an arbitrary two-cell configuration. Specifically, we compare the 1/M/C configuration with a $2/(M_1, M_2)/(C_1, C_2)$ configuration ($C_1 + C_2 = C$, and $M_1 + M_2 = M$). Label the machines 1, …, M from upstream to downstream. We use the same notation as before, i.e., D stands for departure with appropriate subscripts and superscripts. We use $\hat{D}$ for the two cell configuration. The proof is by induction. We provide the induction step. We have for the one cell configuration,

$$
\begin{aligned}
D_n^M &= max(\tilde{D}_n^{M-1}, D_{n-1}^M) + X_n^M \\
&= max(D^{M-1}_{S_n^{M-1}}, D^M_{n-1}) + X_n^M \\
&= max(max(\tilde{D}^{M-2}_{S_n^{M-1}}, D^{M-1}_{S_n^{M-1}-1}) + X^{M-1}_{S_n^{M-1}}, D^M_{n-1}) + X_n^M \\
&\;\;\vdots \\
&= max(max(\ldots max(\tilde{D}^1_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}}, D^2_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}-1}) + X^2_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}})\ldots) \\
&\quad + X^{M-1}_{S_n^{M-1}}), D^M_{n-1}) + X_n^M,
\end{aligned}
$$

and similarly, for the two cell configuration,

$$
\begin{aligned}
\hat{D}_n^M &= max(\tilde{\hat{D}}_n^{M-1}, \hat{D}^M_{n-1}) + X_n^M \\
&= max(D^{M-1}_{S_n^{M-1}}, D^M_{n-1}) + X_n^M \\
&= max(max(\tilde{D}^{M-2}_{S_n^{M-1}}, D^{M-1}_{S_n^{M-1}-1}) + X^{M-1}_{S_n^{M-1}}, D^M_{n-1}) + X_n^M \\
&\;\;\vdots \\
&= max(max(\ldots max(\tilde{\hat{D}}^{M_1}_{S^{M_1+1}_{S^{M_1+2}_{\ddots_{S_n^{M-1}}}}}, \hat{D}^{M_1+1}_{S^{M_1+1}_{S^{M_1+2}_{\ddots_{S_n^{M-1}}}}-1}, \tilde{\hat{D}}^{M}_{S^{M_1+1}_{S^{M_1+2}_{\ddots_{S_n^{M-1}}}}-C_2}) + X^{M_1+1}_{S^{M_1+1}_{S^{M_1+2}_{\ddots_{S_n^{M-1}}}}})\ldots) \\
&\quad + X^{M-1}_{S_n^{M-1}}), \hat{D}^M_{n-1}) + X_n^M \\
&\geq max(max(\ldots max(\tilde{\hat{D}}^{M_1}_{S^{M_1+1}_{S^{M_1+2}_{\ddots_{S_n^{M-1}}}}}, D^{M_1+1}_{S^{M_1+1}_{S^{M_1+2}_{\ddots_{S_n^{M-1}}}}-1}) + X^{M_1+1}_{S^{M_1+1}_{S^{M_1+2}_{\ddots_{S_n^{M-1}}}}})\ldots)
\end{aligned}
$$

$$
\begin{aligned}
& +X^{M-1}_{S_n^{M-1}}), \hat{D}^M_{n-1}) + X^M_n \\
& \vdots \\
= \; & max(max(\dots max(\tilde{\hat{D}}^1_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}}, \hat{D}^2_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}-1}) + X^2_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}})\dots) \\
& +X^{M-1}_{S_n^{M-1}}), \hat{D}^M_{n-1}) + X^M_n \\
\geq \; & max(max(\dots max(\tilde{D}^1_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}}, D^2_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}-1}) + X^2_{S^2_{S^3_{\ddots_{S_n^{M-1}}}}})\dots) \\
& +X^{M-1}_{S_n^{M-1}}), D^M_{n-1}) + X^M_n.
\end{aligned}
$$

□

References

[1] Buzacott, J. and J. G. Shanthikumar. *A General Approach for Coordinating Production in Multiple-Cell Manufacturing Systems*, POMS, vol. 1, no. 1, pp. 34-52, 1992.

[2] Conway, R., W. Maxwell, J. McClain, and L. J. Thomas. *The Role of Work-in-process Inventory in Serial Production Systems*, Oper. Res., vol. 36, no. 2, pp. 229-241, 1988.

[3] Deleersnyder, Jean-Luc, Thom J. Hodgson, Henri Muller, and Peter O' Grady. *Pull Systems: An Analytic Approach*, Mgmt. Sci., vol. 35, no. 9, , pp. 1079-1091, 1989.

[4] Karmarkar, U. and S. Kekre. *Batching Policy in Kanban Systems*, J. Manuf. Sys., vol. 8, no. 4, pp.317-328, 1990.

[5] Mitra, D., and I. Mitrani. *Analysis of a Kanban Discipline for Cell Coordination in Production Lines, 1.*, Mgmt. Sci., vol.36, no. 12, pp. 1548-1566, 1990.

[6] Muckstadt, J. and S. Tayur. *A Comparison of Alternative Kanban Control Mechanisms: Experimental Results*, GSIA Report, 1993.

[7] Muth, E. J. *The Reversibility Property of Production Lines*, Mgmt. Sci., vol. 25, no. 2, pp 152-158, 1979.

[8] Schonberger, R. J. *Japanese Manufacturing Techniques*, The Free Press, Macmillan, New York, 1982.

[9] Stoyan, D. *Comparison Methods for Queues and Other Stochastic Models*, John Wiley & Sons, 1983.

[10] Tayur, S. *Analysis of a Kanban Controlled Serial Manufacturing System.* Ph.D Thesis, Cornell University, 1990.

[11] Tayur, S. *Structural Results and a Heuristic for Kanban Controlled Serial Lines*, Technical Report No. 934, SORIE, Cornell University, (revised) May 1992.

[12] Woodruff, D. L., M. L. Spearman and W. J. Hopp. *CONWIP: A Pull Alternative to Kanban*, Int. J. Prod. Res., vol. 28, no. 5, pp.879-894, 1990.