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

TECHNICAL REPORT NO. 954

January 1991

KANBANS, GENERALIZED SEMI-MARKOV PROCESSES AND ANTI-MATROIDS

by

Sridhar R. Tayur

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

Kanbans, Generalized Semi-Markov Processes and Anti-Matroids

Sridhar R. Tayur

School of Operations Research and Industrial Engineering

Cornell University, Ithaca, NY 14853

Abstract

In this paper we show some structural results of a kanban system by connecting GSMP's with anti-matroids. The main results from this connection are: Dominance, Partition, Consistency and Convexity. Dominance deals with the allocation of kanbans to cells and Partition with the formation of cells. Consistency deals with the derivatives of performance measures with respect to the system parameters, and convexity with the dependence of throughput on processing times. Furthermore, we show concavity of throughput with respect to number of kanbans in a special setting using coupling methods. Although some of the results developed here were obtained previously by sample path arguements, the proofs here are *slick* and generalize the earlier results. The second-order properties for kanban lines are new. These form a basis for results in multi-product lines where proofs by sample paths tend to become a notational annoyance.

Key Words: Optimal Design, Production Lines, Stochastic Comparisons, Coupling Methods.

1 Introduction

Recently, much research has concentrated in studying structural properties of complex systems and has moved away from tractable (for unrealistically simple settings) markovian mathematics; for example, see [3], [4], [8]. We are interested in the analysis of a kanban controlled serial production line in a general setting– general, in the sense that we do not assume a particular distribution of the processing times on the machines, or restrict the total number of machines or kanbans. Consequently, we do not expect an exact analytic solution. Rather, we are interested in studying the *structure* of the line to develop *theoretical* results that help in understanding the effects of variability on the optimal design of serial lines, in stochastic optimization, in simulation and in developing heuristics for particular cases (see [8]).

We show the following: one, a large number of candidate kanban allocations need not be considered at all in determining the optimal design of serial lines; two, the optimal partition of the line is insensitive to the nature of the variabilities if the objective is to minimize the number of kanbans in the system (while meeting the desired throughput);three, the Infinitisimal Perturbation Analysis yields a consistent estimate of the derivative of the throughput with respect to the processing times on the machines; four, the throughput is convex in processing times and five, the throughput is concave in number of kanbans. Thus, second order properties are proved which coupled with derivatives provide us with a tool for stochastic optimization that is theoretically correct. The above results emerge by considering the Generalized

Semi-Markov Process that defines the kanban system and a resulting anti-matroid (with repetition). The structural results also form the basis of a *heuristic* to analyze serial lines; this heuristic (see [8] for details) is easy to code, is very quick, can handle large problems and is found to be very accurate.

To make the exposition self-contained, we begin with an introduction to a kanban system, and a brief description of GSMP's and anti-matroids.

2 The Kanban 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). 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 larger the buffers, the greater the protection against uncertainties; but, this protection is not without expense. Apart from the dollar value of inventory that is tied up due to large buffers, 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 con-

siderably. Consequently, it is important to identify the best places at which to buffer 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 $\mathrm{N}/(M_1,\ldots,M_N)/(C_1,\ldots,C_N)$ to denote a serial production line with N cells, M_i machines in cell i, and C_i kanbans in cell i, $i = 1 \ldots \mathrm{N}$. By *allocation* we mean the vector $(C_1,\ldots,C_N)$, and by *partition* we mean $(M_1 \ldots M_N)$. The set $\{\ \mathrm{N}/(M_1,\ldots,M_N)/(C_1,\ldots,C_N)\colon\ \sum_{i=1}^{N} M_i = M, M_i \geq 1, \sum_{i=1}^{N} C_i = C, C_i \geq 1, N \leq$

$M\}$ contains all possible *configurations* for a line with M machines and C kanbans. Using this notation, we see that CONWIP is 1/(M)/(C) system, and TKCS is a $\mathrm{M}/(1,\ldots,1)/(C_1,\ldots,C_M)$ 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. This is an extension of the model by [6]. As shown in Figure 1, a cell consists of

1. machines in tandem; the processing times on the machines are 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.

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,..., N. In this study we assume that raw material needed by cell 1 is immediately available, and that demand for the output of cell N is infinite (these assumptions are not required for the structural results to hold, but

largely simplify the details). In cell k there are C_k kanbans (cards) and M_k machines. A batch must acquire one of these 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 waits in the output hopper with its 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 destroyed), then the batch is thrown away and the 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, if a batch contains at least one good item, then it is sent to the next processing stage. The determination of the 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 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 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 kanbans control inventory. When a completed part is withdrawn to the next cell (cell

k+1) the 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 kanbans also serve as an information system that controls material transfer between successive cells.

This paper is a part of a stream of research ([8], [9], [10], [11]) with the following objectives:(a) to provide understanding into the working of these lines, in particular to the interactions between various design parameters (such as the effect card count in one cell has on its neighbors), (b) to provide theoretical results that provide computational relief in simulation experiments, (c) to develop heuristics that are efficient to solve *large* problems, (d) to provide theoretical support for stochastic optimization, and (e) to understand the effects of various designs on the overall performance of the line (*cycle time*, average inventory, and throughput). The primary purpose of this paper is to use the state-of-the-art mathematical results to provide the theoretical support required for analysis of kanban lines.

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.

Non-negative random variables X, Y are ordered *stochastically*, represented by $X \leq_{st} Y$, if $P(X \geq a) \leq P(Y \geq a)\ \forall a \geq 0$.

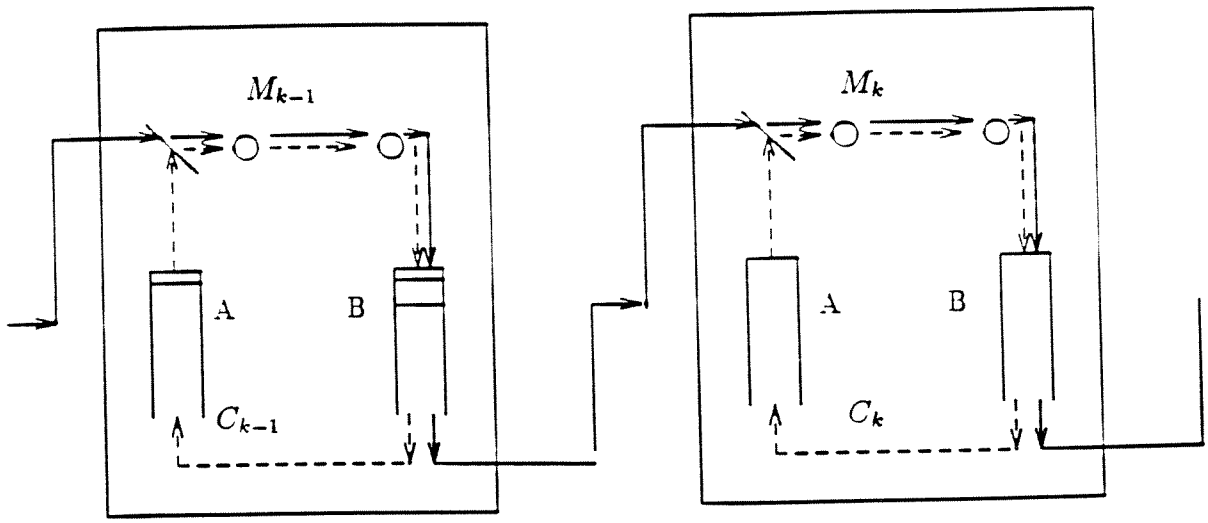

Figure 1: Cell Description.

The decisions to be made while designing a serial line can be divided into the following categories:

1. Given a *sequence* of machines, is there any benefit of grouping some machines together from an operational point of view? This is the *partitioning* issue.

2. Given a *partition* of the line and a target throughput, how many kanbans are required, and how many should be placed in each cell? This is the *allocation* issue.

We will first consider the allocation issue, and then the partition issue. Before we prove our results we provide a brief overview of GSMP's and anti-matroids.

3 Generalized Semi-Markov Processes

We borrow heavily from [3]. A *generalized semi-markov scheme* is a mathematical description of a system that evolves due to discrete events at random points in time. We first describe a scheme with *deterministic* routing, and this corresponds to a *kanban* model with no yield losses. The scheme is described by $(S, A, \epsilon, p, P, r)$, where S is the state-space, A is the set of possible events, $\epsilon : S \rightarrow 2^A$ is a mapping that yields the set of active events in a state (thus, $\epsilon(s)$ for $s \in S$ is the event list for s). ($A = \{\alpha_1, \ldots \alpha_m\}$ is the finite set of event types. Note that $\epsilon(s) \subseteq A$.) For a generic $s \in S$, and $\alpha \in \epsilon(s)$, $\phi(s, \alpha)$ is the state to which the system moves from state s

due to event α. The input to the scheme is $\{\omega_\alpha(n) : \alpha \in A, n = 1, 2 \ldots\}$; $\omega_\alpha(n)$ is the nth clock sample for α. This input drives the system, and gives rise to outputs $T = \{T_\alpha(n) : \alpha \in A, n = 1, 2 \ldots\}$, and $D = \{D_\alpha(t) : \alpha \in A, t \geq 0\}$, where $T_\alpha(n)$ is the epoch of the nth occurence of α and $D_\alpha(t)$ the number of occurences of α in [0,t]. If the initial state is s_0, then the system evolves through states $s_1, \ldots$, where $s_k = \phi(s_{k-1}, \beta^{k-1})$, with $\beta^{k-1} \in \epsilon(s_{k-1})$ being the event that occured.

Example 1 *In a* $2/(1,1)/(C_1, C_2)$ *system, we have* $A = \{\alpha_1, \alpha_2\}$, *where* α_i *corresponds to the end of service on machine* i. *The states are* $\{C_1, \ldots, -C_2\}$. *The states are the difference between the contents of the output hopper of cell 1 and the bulletin board of cell 2. Also,* $\epsilon(C_1) = \{\alpha_2\}, \epsilon(-C_2) = \{\alpha_1\}$, *and* $\epsilon(x) = \{\alpha_1, \alpha_2\}$, *for* $x \in \{C_1 - 1, \ldots, -C_2 + 1\}$. *Finally, for example,* $\phi(C_1, \alpha_2) = C_1 - 1$.

At time $t = 0$, clocks are set for all events in $\epsilon(s_0)$; if $\alpha \in \epsilon(s_0)$, the clock for α is $\omega_\alpha(1)$. The first *event* happens at $t_1 = \min\{\omega_\alpha(1) : \alpha \in \epsilon(s_0)\} = \omega_{\beta^0}(1)$. The other clocks keep running (*non-interruptive* GSMP), and new clocks may need to be set (at time t_1) for state s_1 $(=\phi(s_0, \beta^0))$.

For a *probabilistic* routing, we have for $\alpha \in \epsilon(s)$, $p(.; s, \alpha)$ a probability mass function on S. To be clear, $p(s'; s, \alpha)$ is the probability of being in state s' due to the occurence of event α, given that you are now in state s. P is a probability law that governs the sequence of new clocks for each event. Finally, r is the set of clock speeds.

Example 2 *If there were yield losses at each machine with probability q_1 at machine 1, and q_2 at machine 2, then $p(-C_2; -C_2, \alpha_1) = q_1$, and $p(-C_2+1; -C_2, \alpha_1) = 1-q_1$. Note that q_2 would not affect the state changes, only whether or not a good part was produced on machine 2.*

4 Anti-matroids, with repetition

We (again) borrow from [3] [4]. By viewing A as a set of symbols, every feasible outcome of a GSMP is a *string*, and the collection of all such strings can be thought of as a *language* ($\mathcal{L}$) over A. $\mathcal{L}$ satisfies some basic properties, and this makes it an *anti-matroid with repetition.* It so happens that many of the properties of the GSMP can be obtained by studying the properties of $\mathcal{L}$ (see [3] , a fascinating paper). Thus, in this work we demonstrate that the GSMP that describes a kanban line gives rise to a language $\mathcal{L}$ that has the following properties: (M), (CX), (C), (PM), and (PC). These properties are stated below. First, define for $\sigma \in \mathcal{L}$, $N(\sigma) = (N_{\alpha_1}(\sigma), \ldots, N_{\alpha_m}(\sigma))$, the score function that keeps track of the number of times an event(α_i) has occured in σ.

(M) If $\sigma_1, \sigma_2 \in \mathcal{L}$, and $N(\sigma_1) \geq N(\sigma_2)$ then

$$\epsilon(\phi(s, \sigma_1)) - A_{\sigma_1\sigma_2} \subseteq \epsilon(\phi(s, \sigma_2)),$$

where $A_{\sigma_1\sigma_2} = \{\alpha : N_\alpha(\sigma_1) > N_\alpha(\sigma_2)\}$.

(CX) If $\sigma_1, \sigma_2, \sigma_3 \in \mathcal{L}$, then

$$N(\sigma_3) \geq N(\sigma_1) \bigwedge N(\sigma_2) \Rightarrow [\epsilon(\phi(s,\sigma_1)) \bigcap \epsilon(\phi(s,\sigma_2))] - A \subseteq \epsilon(\phi(s,\sigma_3)),$$

where $A = \{\alpha : N_\alpha(\sigma_3) > N_\alpha(\sigma_1) \bigwedge N(\sigma_2)\}$.

(C) If $\{\alpha, \beta\} \in \epsilon(s)$ then $\phi(s, \alpha\beta) = \phi(s, \beta\alpha)$.

The conditions (PM) and (PC) are probabilistic versions of the conditions (M) and (C) respectively (these help in the case when there are yield losses). We have, from [4] [3], (C) $\Rightarrow$ (M), and (PC) $\Rightarrow$ (PM).

5 Allocation

Our first result is on the *allocation* of kanbans to cells in a N cell line with one machine in each cell. Our objective is to *maximize* the *capacity* of the line among all feasible allocations with a fixed total number of kanbans. This is called the *dominance* theorem. The processing times on each machine is stochastic, the machines are subject to breakdowns, rework may be required, and there may be yield losses at any step (The $\prime$s are used to denote the second system in the comparisons).

Theorem 1 *Given* $(C_1, ..C_N)$ *and* $(C'_1, ..C'_N)$, *for all* $2 \leq k \leq N$ *and all* $j \leq k-1$, *let*

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

$T(n) \leq_{st} T'(n), \quad \forall n \geq 1.$

Proof We need to show that $\mathcal{L}$ satisfies (PM). We show (in theorem 4) that $\mathcal{L}$ satisfies (PC) which implies that it satisfies (PM). The key is to observe that $S' \subseteq S$, and $p(.;s,\alpha) = p'(.;s,\alpha) \forall \alpha \in \epsilon(s)$ and $\forall s \in S \bigcap S'$.

Q.E.D.

If the hypothesis of theorem 1 holds, we say that the allocation $(C_1', \ldots C_N')$ is dominated by the allocation $(C_1, \ldots C_N)$. Note that $T(n) \leq_{st} T'(n)$ implies that the expected throughput rate due to allocation $(C_1, \ldots C_N)$ is greater than or equal to the expected throughput rate due to allocation $(C_1', \ldots C_N')$.

Example 3 *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 highest mean throughput rate among the above five candidates.

The intuitive explanation of the above result is as follows. Unlike the traditional buffered serial line, a kanban can buffer both upstream and downstream of a given machine. For this reason, we need to consider two and three consecutive machines at a time.

Six corolaries are immediate.

Corollary 1 *Increasing the number of kanbans in any cell increases capacity.*

Corollary 2 *Increasing the total number of cards to be allocated increases optimal capacity.*

Corollary 3 *A uniform allocation of kanbans to cells is dominated.*

Corollary 4 *In systems with three or more cells, there exists an optimal allocation that has exactly 1 card in each of the two end cells.*

Corollary 5 *In a two cell system every allocation with a total of C cards yields the same capacity.*

Corollary 6 *In a three cell system the only non-dominated allocation for C cards is 1,C-2,1.*

The first two corollaries provide a short term solution to meeting target throughput. A successful long term strategy is to improve the performance of the line by reducing the variance of processing times, losses in yield and machine downtimes so

that the target throughput can be achieved with fewer cards. This is because the throughput of the line is monotone with respect to the above parameters (see [2]). The Japanese method has been to first remove a card from a line that is currently meeting the target and improve the processes until the target is met again. Another card is then removed, and the improvement is again achieved. Continuing in this manner, the ideal situation in which every cell has exactly one card is eventually attained.

Corollaries 3-6 can be counter-intuitive at first glance. Consider the case when the first machine has a highly variable processing time, or large downtimes, or high yield losses. Is it still best to give only one card to the first cell? The answer is an emphatic YES. If additional cards are needed in the short run, they should be allocated to the second cell. The reason for this is that a card in the second cell can buffer between first and second cells, or between second and third cells as required in a dynamic manner. A card allocated to the first cell can only buffer between first and second cells.

The main strength of the above theorem is the following.

An important fact: The above theorem and its corollories hold for all distributions of processing times and yield losses. The processing times on the different machines for a given job need not be independent, nor need the consecutive processing times on the same machine be independent. In fact, the processing times on a given machine need not arise from a fixed distribution, but can vary from job to job in any

way. Thus many realistic cases of dependence between processing times and between yield losses are handled by the dominance theorem. It is this strength of the result that helps us group different sources of variability together. The optimal allocation always has one card in the end cells (if the number of cells are greater than 2), and the only parameter to determine is C, the total number of cards, which is determined by the variability and other parameters of the system.

A useful by-product of the above theorem is the computational saving as only certain allocations need to be evaluated, namely only those with exactly one card in each of he end cells.

Similar results hold for $N/(M_1, \ldots, M_N)/()$ lines, namely

Theorem 2 *(a) For a 2/(1, M) / (C_1, C_2) system with the total number of kanbans = C, T(n) is stochastically 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, T(n) is stochastically 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 T(n) stochastically.

6 Partition

Given a sequence of machines, we are interested in grouping consecutive machines to achieve the *efficient* frontier. If our objective is to minimize the the *maximum* inventory in the system (equivalently minimize the total number of kanbans), it turns

out that all the machines should be grouped together in one cell. This is a CONWIP type configuration. This is stated in the following theorem.

Theorem 3 *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 consecutive partitions and kanban allocations for a given serial line with* M *machines and* C *kanbans, and let* L^* *be the configuration that yields the stochastically smallest departure epoch (for every* n*th good output) out of the system. Then,* $L^* = 1/(M)/(C)$.

Proof Here, it suffices to compare 1/M/C with $2/(M_1, M_2)/(C_1, C_2)$ with $M_1 + M_2 = M$ and $C_1 + C_2 = C$. We can map the states of $2/(M_1, M_2)/(C_1, C_2)$ to a subgraph of the states of 1/M/C. Next, it is easily verified that $\epsilon'(s) \subseteq \epsilon(s) \forall s \in S' \bigcap S$. Combining the above with the fact that $\mathcal{L}$ satisfies (PM), the desired result is obtained.

Q.E.D.

Observe that Theorem 3, like Theorems 1 and 2, holds in good generality. The structure of the optimal partition for the objective of minimizing the total number of kanbans is insensitive to the particular set of parameters or sources of variability. In fact, the throughput of any $N/(M_1,\ldots,M_N)/(C_1,\ldots,C_N)$ line with a total of C kanbans and M machines, is bounded above by the throughput of a 1/(M)/(C) configuration. (A lower bound in the same generality is developed in [12]).

There is a catch to the comparison made in Theorem 3, namely, that the average inventory is not the same in all of the configurations. When an attempt is made to

maximize mean throughput among all configurations of equal average inventory, then 1/(M)/(C) may not be the best strategy. Extensive simulation shows that the configuration that provides the best capacity versus *average* inventory is $N/(1,\ldots,1)/()$. This is shown in [7], from which Figure 2 is reproduced here as an example.

Graph Set 2: *We demonstrate the* efficient frontiers *for all the four partitions in a three-machine line in Figure 2. Note the* uniform *superiority of TKCS over CONWIP when looking at* average *inventory as the objective. This is in strong contrast to Theorem 3, where TKCS is* uniformly *inferior to CONWIP when the objective is to minimize the* maximum *inventory. Also note that the other two partitions are sandwiched between CONWIP and TKCS. The observations made here are* not *limited to a three-machine line.*

Discussions related to the selection of the objectives, and the optimal partitions corresponding to different objectives can be found in [7].

7 Second Order Properties

We now present results that justify gradient search procedures in stochastic optimization.

Theorem 4 *The Infinitisimal Perturbation Analysis (IPA) estimate for the gradient of throughput with respect to processing rates is consistent.*

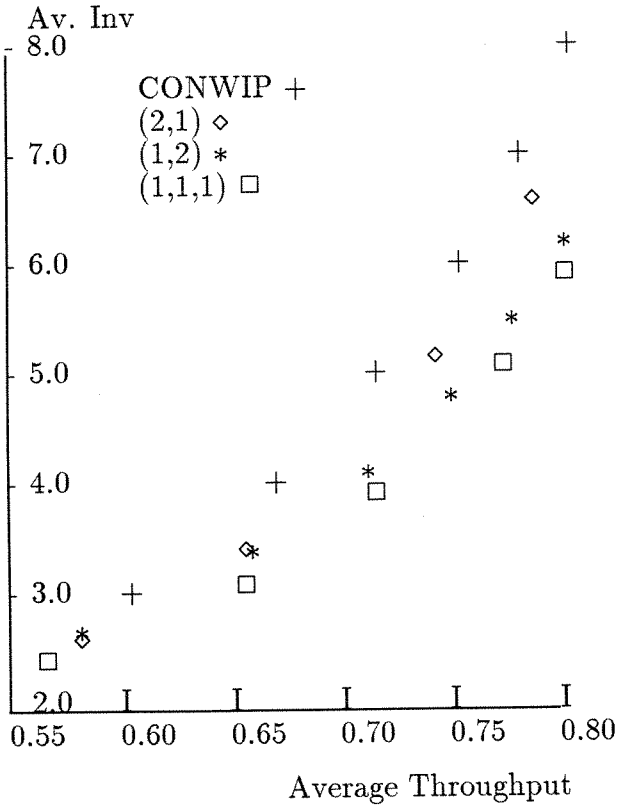

Figure 2: Average inventory vs. mean throughput for a 3 machine line with exponential machines

Proof It suffices to show that $\mathcal{L}$ satisfies (C) in case there are no yield losses. To do this, consider a $N/(1,\ldots 1)/(C_1,\ldots C_N)$ line, currently in state $s = (a_1,\ldots a_{N-1})$, where a_i is the difference between the contents of the *output hopper* of cell i and the *bulletin board* of cell $i+1$. We show that $\phi(s,\alpha_k\alpha_l) = \phi(s,\alpha_l\alpha_k)$ where α_j for $j = 1\ldots N$ is the end-of-service event at machine j.

Without loss of generality, assume $l > k$. Then, $\phi(s,\alpha_k\alpha_l) = \phi(\phi(s,\alpha_k),\alpha_l) = \phi(s',\alpha_l) = s''$, where $s' = (b_1,\ldots b_{N-1})$, and $s'' = (c_1,\ldots c_{N-1})$. Note that $b_j = a_j \forall j \geq k+1$, $b_k = a_k + 1$ and $b_i = a_i + \mathcal{I}\ (a_{i+1} < 0, k-i \text{ odd}) + \mathcal{I}\ (a_{i+1} > 0, k-i \text{ even})$ $\forall i \leq k-1$. ($\mathcal{I}$ (x) = -1 if x is true, and 0 otherwise). Similarly, $c_j = b_j \forall j \geq l+1$, $c_l = b_l + 1$, and $c_i = b_i + \mathcal{I}\ (b_{i+1} < 0, k-i \text{ odd}) + \mathcal{I}\ (b_{i+1} > 0, k-i \text{ even})\ \forall i \leq l-1$.

We can write the state $\phi(s,\alpha_l\alpha_k)$ in the same manner. A straight forward comparison of terms will show that (C) is satisfied. In a similar manner, (PC) can be verified too (to account for yield losses). The same procedure can be used for the general $\mathrm{N}/(M_1,\ldots M_N)/(C_1,\ldots C_N)$ lines.

Q.E.D.

Theorem 5 *The departure epochs out of the line are* stochastically *convex and increasing with respect to the processing times on the machines.*

Proof It suffices to show that (CX) is satisfied by $\mathcal{L}$. Since (PM) is satisfied (because (PC) $\Rightarrow$ (PM)), it is sufficient to show that the score function is min-closed. Again, consider a $\mathrm{N}/(1,\ldots,1)/(C_1,\ldots C_N)$ line. We have, for a feasible score $\mathbf{x}$ (this is a

vector in Z_+^N, with subscripts α_i for $i = 1 \ldots N$ corresponding to end of service event on machine i),

$$x_{\alpha_i} \leq x_{\alpha_{i-1}} \bigwedge (x_{\alpha_{i+1}} + C_i + C_{i+1}) \bigwedge (x_{\alpha_{i+2}} + C_i + C_{i+1} + C_{i+2}) \quad (1)$$

$$x_{\alpha_i} \leq x_{\alpha_{i-1}}. \quad (2)$$

Now if $\mathbf{x}$ and $\mathbf{y}$ satisfied (1) and (2), so does $\mathbf{x} \bigwedge \mathbf{y}$. This completes the proof.

Q.E.D.

8 Concavity of Throughput with respect to number of Kanbans

In this section we show that the throughput is stochastically concave with respect to number of kanbans in a given cell. It is possible to prove the result by *uniformization* arguments and anti-matroid properties that were developed earlier. However, for the sake of variety, we show the result for the case of exponential machines using sample-path arguements and coupling methods (see [1]). An alternative method, following [5] is shown in [12]. For ease of exposition, we will restrict to one machine per cell and assume no yield losses.

The basic idea is to construct three processes corresponding to the following three systems, where we fix m, $m \in \{1, \ldots, M\}$. System 0 has C kanbans in cell m, system 1 has C+1 kanbans in cell m, and system 2 has $C+2$ kanbans in cell m. All other cells

have one card in them. Let $D_i^M(t)$ be the number of outputs out of system i upto time t, $i = 0, 1, 2$, and $t \geq 0$. We need to show that $D_2^M(t) - D_1^M(t) \leq_{st} D_1^M(t) - D_0^M(t)$, $\forall t \geq 0$. The crux of the proof rests on the fact that it is possible to construct a *virtual* (S_j^m) service process for each server, and have them be synchronous across the three systems. Recall that this process is simply a poisson process, and a customer departs from a queue only if the queue was non-empty. Thus, it is sufficient to look at the system only at times $\{\tau_n\}$=$\{S_j^m\}$. The proof is by induction.

We need some notation. Let $N_i^j(t)$ be the number of customers in front of machine j in system i at time t, and $H_i^j(t)$ be the number in the output hopper of cell j in system i at time t.

Define for i = 0,1,2 and $t \geq 0$,

$$X_i^j(t) = D_i^M(t) + \sum_{p=j}^{M}(N_i^p(t) + H_i^{p-1}(t)) \quad j = 1 \ldots M$$

$$X_i^{M+1}(t) = D_i^M(t)$$

The inductive hypothesis is, for $j = 1 \ldots M+1$ and $k = 0 \ldots n$,

$$\Delta_2^j(\tau_n) = X_2^j(\tau_k) - X_1^j(\tau_k) \quad \leq \quad \Delta_1^j(\tau_n) = X_1^j(\tau_k) - X_0^j(\tau_k)$$

At $\tau_0 = 0$, the inductive hypothesis is trivially satisified. Let us now suppose that it is true at τ_n. If it is not true at τ_{n+1}, we have the following cases to consider (and each will yield a contradiction).

1. $\tau_{n+1} = S_b^M$. Then the contradiction can occur only at $M+1$. We have

$$D_2^M(\tau_n) - D_1^M(\tau_n) = D_1^M(\tau_n) - D_0^M(\tau_n)$$

and $N_2^M(\tau_n) > 0, \text{and} N_1^M(\tau_n) = 0 (\Rightarrow H_1^{M-1}(\tau_n) = 0)$. Then,

$$\begin{aligned} X_2^M(\tau_n) - X_1^M(\tau_n) &= \Delta_2^{M+1}(\tau_n) + N_2^M(\tau_n) - N_1^M(\tau_n) + H_2^{M-1}(\tau_n) - H_1^{M-1}(\tau_n) \\ X_1^M(\tau_n) - X_0^M(\tau_n) &= \Delta_1^{M+1}(\tau_n) + N_1^M(\tau_n) - N_0^M(\tau_n) + H_1^{M-1}(\tau_n) - H_0^{M-1}(\tau_n) \end{aligned}$$

which is a contradiction.

2. $\tau_{n+1} = S_b^j, j \in \{m+1, \ldots M-1\}$. We have

$$\Delta_2^{j+1}(\tau_n) = \Delta_1^{j+1}(\tau_n)$$

and $N_2^j(\tau_n) > 0 (\Rightarrow H_2^j(\tau_n) = 0), \text{and} N_1^j(\tau_n) = 0$. Then,

$$\begin{aligned} X_2^j(\tau_n) - X_1^j(\tau_n) &= \Delta_2^{j+1}(\tau_n) + N_2^j(\tau_n) - N_1^j(\tau_n) + H_2^{j-1}(\tau_n) - H_1^{j-1}(\tau_n) \\ X_1^j(\tau_n) - X_0^j(\tau_n) &= \Delta_1^{j+1}(\tau_n) + N_1^j(\tau_n) - N_0^j(\tau_n) + H_1^{j-1}(\tau_n) - H_0^{j-1}(\tau_n) \end{aligned}$$

If $H_1^{j-1}(\tau_n) = 0$, then we are done. Else, if $H_1^{j-1}(\tau_n) > 0 (\Rightarrow H_1^j(\tau_n) = 1)$, we have

$$\begin{aligned} X_2^{j+1}(\tau_n) - X_1^{j+1}(\tau_n) &= \Delta_2^{j+2}(\tau_n) + N_2^{j+1}(\tau_n) - N_1^{j+1}(\tau_n) + H_2^j(\tau_n) - H_1^j(\tau_n) \\ X_1^{j+1}(\tau_n) - X_0^{j+1}(\tau_n) &= \Delta_1^{j+2}(\tau_n) + N_1^{j+1}(\tau_n) - N_0^{j+1}(\tau_n) + H_1^j(\tau_n) - H_0^j(\tau_n) \end{aligned}$$

and this will take you to stage $j+1$. Continuing this way, we will reach stage M, which will yield a contradiction.

3. $\tau_{n+1} = S_b^m$. This would imply, as before,

$$\Delta_2^{m+1}(\tau_n) = \Delta_1^{m+1}(\tau_n)$$

and $N_2^m(\tau_n) > 0 (\Rightarrow H_2^m(\tau_n) \leq C+1)$, and $N_1^m(\tau_n) = 0$. This means,

$$\begin{aligned} X_2^m(\tau_n) - X_1^m(\tau_n) &= \Delta_2^{m+1}(\tau_n) + N_2^m(\tau_n) - N_1^m(\tau_n) + H_2^{m-1}(\tau_n) - H_1^{m-1}(\tau_n) \\ X_1^m(\tau_n) - X_0^m(\tau_n) &= \Delta_1^{m+1}(\tau_n) + N_1^m(\tau_n) - N_0^m(\tau_n) + H_1^{m-1}(\tau_n) - H_0^{m-1}(\tau_n) \end{aligned}$$

As before, if $H_1^{m-1}(\tau_n) = 0$, we have an immediate contradiction. Else, if $H_1^{m-1}(\tau_n) > 0 (\Rightarrow H_1^m(\tau_n) = C+1)$, we have

$$\begin{aligned} X_2^{m+1}(\tau_n) - X_1^{m+1}(\tau_n) &= \Delta_2^{m+2}(\tau_n) + N_2^{m+1}(\tau_n) - N_1^{m+1}(\tau_n) + H_2^m(\tau_n) - H_1^m(\tau_n) \\ X_1^{m+1}(\tau_n) - X_0^{m+1}(\tau_n) &= \Delta_1^{m+2}(\tau_n) + N_1^{m+1}(\tau_n) - N_0^{m+1}(\tau_n) + H_1^m(\tau_n) - H_0^m(\tau_n) \end{aligned}$$

and this will yield a contradiction by appealing to a previous case.

4. $\tau_{n+1} = S_b^j, j \in \{1, \ldots m-1\}$. The same procedure as the above two cases will provide the desired result.

The case with *yield* losses can also be obtained by the same method but involves more detailed notation. We thus have,

Theorem 6 *The throughput of the line is* stochastically *concave and increasing in the number of kanbans in a given cell.*

Acknowledgements

Thanks to Prof. David Yao for early drafts of his work, Prof. Muckstadt for involving the author with real world implementation, and Levent Tuncel for correcting an earlier draft of this paper.

References

[1] V. Ananthram, and P. Tsoucas, *Concavity of throughput in tandem queues*, Working Paper, University of Maryland, 1989.

[2] D. W. Cheng, and David Yao, *Tandem Queues with general blocking: a unified model and comparison results*, Working Paper, January 1991.

[3] Paul Glasserman, and David Yao, *Monotonicity in Generalized Semi markov Processes*, Working Paper, Columbia University, October 1989.

[4] Paul Glasserman, and David Yao, *GSMP: Antimatroidal Structure and Second Order Properties*, Working Paper, Columbia University, January 1990.

[5] L. E. Meester, and J. G. Shanthikumar, *Concavity of the throughput of tandem queuing systems with finite buffer storage space*, Adv. Appl. Prob., 22, 764-767 (1990).

[6] D. Mitra and I. Mitrani, *Analysis of a Novel Discipline for Cell Coordination in Production Lines, 1.*, Mgmt. Sci., Dec. 1990.

[7] John A. Muckstadt, and Sridhar Tayur, *A Comparison of Alternative kanban control mechanisms*, Technical Report No. 962, Cornell University, April 1991. Submitted for publication.

[8] Sridhar Tayur, *Analysis of a Kanban controlled serial manufacturing system*, Ph.D Thesis, Cornell University, August 1990.

[9] Sridhar Tayur, *Heuristics for multi-stage stochastic kanban controlled system*, Technical Report No. 933, Cornell University, August 1990. Submitted for publication.

[10] Sridhar Tayur, *Controlling Serial Production Lines with yield losses using kanbans*, Technical Report No. 947, Cornell University, December 1990. Submitted for publication.

[11] Sridhar Tayur, *Analysis of a multi-product kanban controlled system*, Working Paper, Cornell University, January 1991.

[12] Sridhar Tayur, *Properties of Throughput in Kanban Lines*, Technical Report No. 963, Cornell University, April 1991.