PROPERTIES OF THROUGHPUT
IN KANBAN LINES

by

Sridhar R. Tayur

This work was supported in part by National Science Foundation Grant DDM-8819542.
Properties of throughput in Kanban Lines

Sridhar R. Tayur
Cornell University

Abstract

We show concavity property of throughput as a function of kanbans, which coupled with the reversibility property implies schur concavity under certain conditions. This helps reduce the computational effort required in designing systems, and explains the unusually high accuracy of a previous heuristic. We provide a proof of a version of the bowl phenomenon, by a functional characterization of the departure times out of the system. This characterization helps in determining whether or not different sequences of machines yield different throughputs. Next, we bound throughput (above and below) by throughputs of closed queuing networks that are asymptotically convergent. Finally, we use a detailed proof of reversibility that has implications for bottleneck management.

1 Introduction

To aid in optimal design of stochastic serial lines, it is helpful to identify first and second order properties of performance measures with respect to the design parameters. The design parameters include the sequencing of the machines, the partitioning of the line into cells, and the allocation of kanbans to cells. The result of these properties is a reduction in the number of possibilities to be considered, and to theoretically justify optimization by gradient search procedures. This has become an established approach (for example, Shanthikumar and Yao(1990), Glasserman and Yao(1990)). In an earlier work (Tayur(1991a)), we had provided some properties of kanban lines in the same spirit.

For a network of queues in series (not limited to kanban control), we characterize (functionally) the conditions under which the sequence of machines does not affect the throughput. With markovian assumptions, it appears that this implies that the balance equations have a product form solution. This explains many of the cases (most notably the closed queuing network) when the product form solution exists although the flows in the network are non-poisson. It will be shown that the departure epoch of the \( n \)th customer (for all \( n \)) out of the network has a max-min representation using the service times of the customers (1 \( \leq k \leq n \)) at each machine. This max-min representation, considered as a function of random variables (service times),
determines the structure of the network. This condition is called the \textit{exchangeability} condition. Informally this condition means that, given a set of machines \( \{1 \ldots M\} \), \( g_n() \), the departure epoch out of the \( n \)th customer out of the system does not depend on the particular permutation of the machine sequence.

Further, our goals in this paper are the following:

1. In an earlier work (Tayur(1990b)), a heuristic (\textit{state-space}) was developed that was a surrogate for throughput. The heuristic was shown to be very accurate in identifying good kanban allocations, and possessed the combinatorially pleasing property of schur-concavity. What we show here is that the throughput itself possesses such a property. This will justify the solution indicated by the heuristic.

2. We give a proof of the following version of the bowl phenomenon: the \textit{bottleneck} should be placed away from the center of the sequence to maximize the throughput. This has been observed by simulation by many researchers (Hillier and Boling(1979), Rao(1976)) but has not been proved in a general setting.

3. Alternate designs are known to provide the same mean throughput, but at different \textit{inventory} costs. This become important in the presence of \textit{bottlenecks}. We provide a partial characterization to help compare competing designs by providing rules to sequence the machines. Informally, the idea is to place the bottleneck as much upstream as possible.

4. We bound the throughput (above and below) of a kanban line by closed queuing networks, which possess a product form structure under markovian assumptions. The bounds converge asymptotically, and so for larger problems (where exact solution for non-product form networks becomes computationally infeasible) we can provide a more accurate estimate.

This work is part of the stream of work (Tayur(1990a)-(1990d), Tayur(1991a)-Tayur(1991b)) whose main objective is to understand kanban control mechanism, and provide heuristics for the effective design of serial lines.

\section{Notation, Definitions, Background}

The model we analyze is described in Mitra and Mitrani(1990) and Tayur(1991a). We will use \( N/(M_1 \ldots M_N)/(C_1 \ldots C_N) \) to denote a kanban line with \( N \) cells, \( M_i \) machines in cell \( i \), and \( C_i \) kanbans in cell \( i, i = 1 \ldots N \). By \textit{allocation} we mean the vector \( (C_1 \ldots C_N) \), and by \textit{partition} we mean \( (M_1, \ldots, M_N) \). The set \{ \( N/(M_1, \ldots, M_N)/(C_1, \ldots, C_N) \):
\[ \sum_{i=1}^{N} M_i = M_i, M_i \geq 1, \sum_{i=1}^{N} C_i = C, C_i \geq 1, N \leq M \] contains all the configurations for a line with \( M \) machines and \( C \) kanbans. Buffering, in our context, means the sequencing of machines, the partitioning of the line, and the allocation of cards to the resulting cells.

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.

A bottleneck is a machine that has a higher mean processing time than the others with the same coefficient of variation, or a higher variance than the others with the same mean, or both.

A machine that is closer to the end of the processing sequence than another is said to be downstream of the other. We define upstream analogously.

A line is said to be \( D \)-reversible if the departure epoch of the \( n \)th job out of the line has the same distribution if the line was reversed.

A line \( N/(M_1 \ldots M_N)/(C_1 \ldots C_N) \) with a total of \( M \) machines is symmetric if \( C_k = C_{N-k+1} \) and \( M_k = M_{N-k+1} \) \( \forall k = 1 \ldots N \), and machines in positions \( j \) and \( M-j+1 \) have the same processing time distribution for \( j = 1 \ldots M \).

We now describe a generalized semi-markov process, which is central to the development of our results in this paper. We borrow heavily from Yao(1989). A generalized semi-markov scheme is a mathematical description of a system that evolves due to discrete events at random points in time. We describe a scheme with deterministic routing. The scheme is described by \((S, A, \epsilon, p, P, r)\), where \( S \) is the state-space, \( A \) is the set of possible events, \( \epsilon : S \to 2^A \) 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 \( \alpha \). The input to the scheme is \( \{\omega_\alpha(n) : \alpha \in A, n = 1, 2, \ldots\} \); \( \omega_\alpha(n) \) is the \( n \)th clock sample for \( \alpha \). 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 \( n \)th occurrence of \( \alpha \) and \( D_\alpha(t) \) the number of occurrences of \( \alpha \) 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 occurred.
Example 1 In a $2/(1,1)/(C_1,C_2)$ system, we have $A = \{\alpha_1, \alpha_2\}$, where $\alpha_i$ 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 $\alpha$ 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 $s_1$ (= $\phi(s_0, \beta^0)$) only for events $\epsilon(s_1) - \{\epsilon(s_0) - \beta^0\}$. This procedure is continued until some predetermined epoch, such as the departure of a fixed number of customers out of the system or clock time.

We finish this section with two definitions regarding ordering of random variables.

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

Non-negative random variables $X, Y$ are ordered by likelihood ratio, represented by $X \leq_{lr} Y$, if $\frac{P(X=a)}{P(Y=a)}$ is decreasing as $a$ increases.

3 The Main Results

The following result is known (Tayur(1991a)).

Theorem 1 The $N/(1 \ldots 1)/(\cdot)$ line is stochastically concave with respect to kanbans in any cell.

We will show the following.

Theorem 2 The $N/( M_1 \ldots M_N )/(\cdot)$ line is stochastically jointly concave with respect to kanbans.

We have shown the following (Tayur(1990a)).

Theorem 3 The $N/( M_1 \ldots M_N )/(\cdot)$ line is $D$-reversible.
We will show a more detailed version of the above theorem (theorem 5) that
connects the start times of every job on every machine in the forward sequence to the
finish times in the reverse sequence. This will help in comparing lines that have the
same throughput, but have different average inventory (or equivalently flow times).
Theorems 2 and 3 imply the following for a line with identical exponential machines,
an important result. In fact, this result holds whenever the line is symmetric, namely
machine \( m \) and \( M - m + 1 \) are identical for all \( m = 1 \ldots M \).

**Theorem 4** The throughput in a \( N/(M_1 \ldots M_N)\) line with \( M_j = M_{N-j+1} \ \forall j \) and \( C_j = C_{N-j+1} \ \forall j \neq k \) is schur-concave in \( (C_k, C_{N-k+1}) \).

The state-space heuristic (Tayur(1990b)) has the schur concavity (see Marshall
and Olkin(1979)) property too. This implies that the search is being done on a set
of allocations that contain the optimal, and that for balanced lines with 4 machines
or less the heuristic provided the optimal solution. The next result is the detailed
version of theorem 3.

**Theorem 5** Let \( T(n) = (T_{[1]}(n) \ldots T_{[M]}(n)) \) be the finish times of the \( n^{th} \) job on
the various machines in the forward sequence ([m] stands for the \( m^{th} \) machine in
the sequence). Let \( S'(n) = (S'_{[1]}(n) \ldots S'_{[M]}(n)) \) be the start times on machines in the
reversed sequence. Let \( n^* \) be the number of jobs that have left both systems. Then,
for \( 1 \leq n \leq n^* \), we have \( T_{[m]}(n) =_{st} (T_{[M]}(n^*) - S'_{[M-m+1]}(n) - n + 1) \).

Although a line and its reverse provide the same throughput, the average flow times
may be different. Thus, we are interested in ordering between the start-times of jobs
on the first machine in the sequence. We have the following.

**Theorem 6** Consider a \( N/(M_1 \ldots M_N)/(C_1 \ldots C_N) \) line with \( M_k = M_{N-k+1} \) and \( C_k = C_{N-k+1} \ \forall k = 1 \ldots N \). Let \( F_{[m]}(.) \), \( m = 1 \ldots M - 1 \) be i.i.d processing time distributions in the forward sequence, and \( F_{[M]}(.) \) be the distribution of the last machine. Let \( 1 - F_{[M]}(a) \geq 1 - F_{[m]}(a) \ \forall a \geq 0 \), \( m = 1 \ldots M - 1 \). Then \( S'(n) \geq_{st} S(n) \ \forall n \),
where \( S'(n) \) and \( S(n) \) are the start times of the \( n^{th} \) job in the reverse and the forward
sequences respectively.

If we had a choice to place the machines in any sequence, then it has been found (by
extensive simulation studies by various researchers) that the bottleneck must be away
from the middle of the sequence. This is a manifestation of the bowl phenomenon.
To clarify, suppose we had 3 machines (A,B,C), all exponential, 2 of them (B, C)
identical in distribution and the third (A) larger in the mean. Then, we do not want the sequence BAC. In general, we have the following (a similar result holds for longer lines too).

**Theorem 7** Let \( D_{ABC}(t) \) be the number of departures out of the line with sequence ABC up to time \( t \), and \( D_{BAC}(t) \) the same for a line with sequence ACB. Then, if \( \mu_A \leq \mu_B \), the rates on machines A and B, we have \( D_{ABC}(t) \geq_{st} D_{BAC}(t), \forall t \geq 0 \).

As the lines become larger, it becomes computationally infeasible to compute the throughput as the kanban system in general does not possess a product form structure. However, it is possible to bound (both above and below) any kanban line by closed queuing networks, which are relatively easier to solve, and have been studied extensively. We have the following.

**Theorem 8** (a) Let \( N/(1 \ldots 1)/(C_1 \ldots C_N) \) be the line in question, say line A with departure epochs \( \lambda T(n) \). Let \( C = \sum_{i=1}^{N} C_i \), and let \( C^* = \min\{C_i + C_{i+1} : i = 1 \ldots N - 1, M_i = M_{i+1} = 1\}, \{C_j : j = 1 \ldots N, M_j > 1\} \). Then, let line B be \( 1/(M)/(C) \) with departure epochs \( \beta T(n) \) and line C be \( 1/(M)/(C^*) \) with departure epochs \( \gamma T(n) \). We have \( \beta T(n) \leq_{st} \lambda T(n) \leq_{st} \gamma T(n) \) \( \forall n \geq 1 \).

The above result is a consequence of the fact that the departure epochs out of each machine has a min-max representation using the processing time variates generated up to that departure. To be precise, given \( \{X_i^{m_1}, i \geq 1\} \) and \( \{X_i^{m_2}, i \geq 1\} \) the processing times for machines \( m_1 \) and \( m_2 \), the finish time of job \( n \) can be written as \( g_n(\{X_i^{m_1}, i \geq 1\}, \{X_i^{m_2}, i \geq 1\}) \). We have the following.

**Theorem 9** (a) If \( g_n(\{X_i^{m_1}, i \geq 1\}, \{X_i^{m_2}, i \geq 1\}) = g_n(\{X_i^{m_2}, i \geq 1\}, \{X_i^{m_1}, i \geq 1\}) \), then there is no gain in throughput by interchanging \( m_1 \) and \( m_2 \).
(b) (Conjecture) If \( g_n(\{X_i^{m_1}, i \geq 1\}, \{X_i^{m_2}, i \geq 1\}) = g_n(\{X_i^{m_2}, i \geq 1\}, \{X_i^{m_1}, i \geq 1\}) \) for all \( m_1, m_2 \), then under the markovian assumption we have product form of steady state probabilities.

In case all the machines had exponentially distributed processing times, then the line B and C correspond to product form networks. We have the following, our last result in this section.

**Theorem 10** \( \lim_{n \to \infty} \left( \frac{n}{\beta T(n)} - \frac{n}{\gamma T(n)} \right) \to 0 \). If the machines were identical and exponential, and \( C \) the total number of cards:

(a) \( \lim_{n \to \infty} \left( \frac{n}{\beta T(n)} - \frac{n}{\gamma T(n)} \right) \to O(C^{-2}) \) if \( M = 3 \) as \( C \to \infty \).
(b) \( \lim_{n \to \infty} \left( \frac{n}{\beta T(n)} - \frac{n}{\gamma T(n)} \right) \to O(C^{-1}) \) if \( M \geq 4 \) as \( C \to \infty \).

We now proceed to prove all our claims.
4 Proofs

The fundamental concepts that will be used in the proofs below are sample-path comparisons and coupling. All queues, in particular queues in series, can be described by a GSMP. It is beneficial, however, to have a visual aid in the proofs. This is provided by transition graphs, the nodes of which correspond to states of the GSMP and the arcs show the event list at a state and point to the result of the event completion. The properties of throughput of a queuing system are determined by the structure of the transition graph that describes it. It is, therefore, only natural that we study kanban systems by studying the underlying transition graphs. In Tayur (1991a), for example, many of the structural properties in allocation and partitioning issues were derived in a very general setting by exploiting the structure if the transition graphs.

A technical point. The departure epochs in our networks admit a min-max representation (see Glasserman and Yao (1989)) because the language generated by our GSMP satisfies property (M) (see Tayur (1991a), for details).

4.1 The Bowl Phenomenon

We are interested in determining whether the sequence of machine affects the throughput, and if so which are the preferred sequences. We begin with a simple example to motivate the main result of this section and its proof approach.

Example 2 Consider a three machine line with machines $M_a$, $M_b$, and $M_c$ with exponential processing time distributions with rates $\mu_a, \mu_b, \mu_c$ respectively. Let $\Delta_{ab}$ and $\Delta_{bc}$ be the number of customers between machines $M_a$ and $M_b$, and between $M_b$ and $M_c$ respectively. Suppose the transitions allowed were restricted to be those in the figure below.

The question is whether there is a benefit in changing the sequence of the machines? Specifically, is there a reason to prefer $(M_b, M_a, M_c)$ over $(M_a, M_b, M_c)$. The claim is that the former is preferred if and only if $\mu_a \geq \mu_b$. To explain, we want the faster machine in the middle.

The explanation is by considering the evolution of the GSMP corresponding to this network with the sequence $(M_a, M_b, M_c)$. Let $\{\alpha_a, \alpha_b, \alpha_c\}$ be the events that correspond to end of service on $M_a, M_b, M_c$ respectively. Starting at $(0,0)$, we place $\alpha_a$ on the event list. After $\omega_{\alpha_a}(1)$ time units, we schedule two events: $\alpha_a$ (for the second time), and $\alpha_b$. After another $\omega_{\alpha_b}(1)$ time units, we are ready to place (for the first time) $\alpha_c$ on the event list. The first output out of the system occurs at $\omega_{\alpha_a}(1) + \omega_{\alpha_b}(1) + \omega_{\alpha_c}(1)$.
time units, denoted by \( g_1(a, b, c) \). Similarly, \( g_n(a, b, c) = T_{\alpha_c}(n) \) under the sequence \((M_a, M_b, M_c)\) can be obtained.

The key point to notice is that the difference in times between consecutive scheduling of event \( \alpha_b \) is \( \max(\omega_{a_b}(), \omega_{c_b}()) + \omega_{a_b}() \) under the sequence \( M_a, M_b, M_c \), which determines the throughput (inverse of consecutive times). Note that this time is smaller than \( \max(\omega_{a_b}(), \omega_{c_b}()) + \omega_{a_b}() \) (the consecutive times between scheduling of event \( \alpha_a \) in the sequence \( M_b, M_a, M_c \)) if \( \mu_a \leq \mu_b \). To be precise, \( g_n(a, b, c) \leq g_n(b, a, c) \forall n \geq 1 \), if \( \mu_b \geq \mu_a \).

The point of the above example is that we are interested in comparing \( g_n(\sigma) \) with \( g_n(\pi) \) for different permutations \( \sigma, \pi \) of \((1 \ldots M)\). The reason this is sufficient is because it is possible to write any departure time out of each machine (equivalently, the times \( T(n) \)) as min-max functions of the processing time variates until that departure epoch (Glasserman and Yao(1990), Tayur(1990a)). If it so happens that \( g_n(\sigma) \geq g_n(\pi) \) \( \forall n \geq 1 \), then \( \pi \) is preferred over \( \sigma \). In general, it is worthwhile to exchange positions \( i \) and \( j \) if \( a_i \geq a_j \Rightarrow g_n(\ldots a_i \ldots a_j \ldots) \geq g_n(\ldots a_j \ldots a_i \ldots) \). We need to identify this property, given any network of queues. That for a \( 3/(1,1,1)/() \) we have \( g_n(a, b, c) \leq g_n(b, a, c) \forall n \geq 1 \), if \( \mu_b \geq \mu_a \) is shown next, which proves theorem 7. The following result from Shanthikumar and Yao(1990) is central to our proof.
**Lemma 1** Let $\mathcal{G}_{xy} = \{g(x, y) : g(x, y) \geq g(y, x) \text{ if } x \geq y\}$. If $X \geq_{tr} Y$, then $g(X, Y) \geq_{st} g(Y, X) \forall g(\cdot) \in \mathcal{G}_{xy}$.

**Lemma 2** For a 3/(1,1)/(1,C,1) line, we have $g_n(a, b, c) \leq_{st} g_n(b, a, c) \forall n \geq 1$, if $\mu_b \geq \mu_a$.

**Proof** The transition graph for the line for the sequence $M_a, M_b, M_c$ is given in figure 2; as before $\Delta_{ab}$ and $\Delta_{bc}$ be the number of customers between machines $M_a$ and $M_b$, and between $M_b$ and $M_c$ respectively. Let $\hat{\omega}_a(\cdot), \hat{\omega}_b(\cdot), \hat{\omega}_c(\cdot)$ be the random variables corresponding to the remaining time of the events $\{\alpha_a, \alpha_b, \alpha_c\}$ if they are on the event list when an event occurs. To be precise, at the $n$th event occurrence, they are the remaining clock times of $\epsilon(s_n) - \{\epsilon(s_{n-1}) - \beta^{n-1}\}$. We begin at $(0,0)$ as in example 2. It is easily verified that $g_1(a, b, c) = g_1(b, a, c)$. The key observation to make is that it is sufficient to look at the time between the consecutive schedulings of event $\alpha_b$. This time difference can take four types of forms depending on the state of the system: $\hat{\omega}_a(\cdot), \hat{\omega}_b(\cdot), \max(\hat{\omega}_a(\cdot), \hat{\omega}_b(\cdot))$ and $\hat{\omega}_b(\cdot) + \max(\hat{\omega}_a(\cdot), \hat{\omega}_c(\cdot))$. The only form that is asymmetric (compared to a line with sequence BAC) is the term $\hat{\omega}_b(\cdot) + \max(\hat{\omega}_a(\cdot), \hat{\omega}_c(\cdot))$, which happens at node $(C+1,0)$. 

Figure 2: Transition graph for a 3/(1,1)/(1,C,1) line
Clearly $x + \max(y, z) \geq y + \max(x, z)$ whenever $y \leq x$. This implies that $g_n(a, b, c) \in \mathcal{G}_{ab}$. Note that $\omega(a)$ and $\hat{\omega}_a()$ have the same distribution because of the exponential distribution. Similarly in the cases of $\omega_b()$ and $\hat{\omega}_b()$ and $\omega_c()$ and $\hat{\omega}_c()$. Further, $\mu_b \geq \mu_a \Rightarrow \omega_b() \leq_{tr} \omega_a()$. From lemma 1 above, the desired result follows.

Q.E.D.

Remark: The exponential distribution is required to prove the stochastic order; in case the distributions were not exponential, a weaker ordering can be obtained. Also, note that the transition graph in figure 2 holds for $3/(1,1,1)/(C_1, C_2, C_1)$ lines as well ($C_1 + C_2 = C + 1$). Similarly, bowl phenomenon can be shown to exist for $N \geq 4$. Explicit expressions for $T_c(n)$ can be obtained by using the recursions in example 3 (section 4.4), and that it belongs to $\mathcal{G}_{ab}$ can be shown by induction on $n$. Theorem 9(a) follows directly from the discussions in the above section.

### 4.2 Upper and Lower Bounds

A closer look at figure 2 will show that the transition graph of a $3/(1,1,1)/(1,C,1)$ is sandwiched between transitions graphs of $1/(3)/(C+2)$ and $1/(3)/(C+1)$. This is true for any processing time distribution or sequence of the machines. In general, a $N/(1\ldots 1)/(C_1\ldots C_N)$ line has a transition graph that is contained in the transition graphs of $1/(N)/(C)$, and contains the graph of $1/(N)/(C^*)$, where $C = \sum_{i=1}^{N} C_i$ and $C^* = \min\{C_i + C_{i+1} : i = 1\ldots N-1\}$.

**Proof (Theorem 8)** Let $A$ be the line $N/(1\ldots 1)/(C_1\ldots C_N)$, $B$ the line $1/(N)/(C)$, and $C$ the line $1/(N)/(C^*)$ with transition graphs $G_A, G_B, G_C$ respectively. Let $A T(n)$, $B T(n)$, and $C T(n)$ correspond to the end epochs of GSMP's of lines $A, B$ and $C$ respectively. Similarly, define $\epsilon(s), \phi(s, \beta), and \omega()$ for each of the lines with appropriate pre-subscripts. Note that $C S = \{s : s \in G_C\} \subseteq A S = \{s : s \in G_A\} \subseteq B S = \{s : s \in G_B\}$. Further, $\epsilon(s) \subseteq A \epsilon(s) \subseteq B \epsilon(s) \forall s \in C S$ and $A \epsilon(s) \subseteq B \epsilon(s) \forall s \in B S$. Finally, $\{A \omega()\} =_{st} \{B \omega()\} =_{st} \{C \omega()\}$, which implies that $C T(n) \geq_{st} A T(n) \geq_{st} B T(n)$. A similar argument can be made with $N/(M_1\ldots M_N)/(C_1\ldots C_N)$ lines.

Q.E.D.

This proves theorem 8. In fact we can bound from above a given $N/(M_1\ldots M_N)/(C_1\ldots C_N)$ line by $N-1/(M_1\ldots M_{k-1}, M_k+M_{k+1}, M_{k+2}\ldots M_N)/(C_1\ldots C_{k-1}, C_k+C_{k+1}, C_{k+2}\ldots C_N)$, for $k = 1\ldots N-1$. A similar argument was given in Tayur(1991a) to prove the dominance result. The bounds in theorem 10 are obtained by simple algebra to prove the results of Gordon and Newell(1967).
Figure 3: Transition graph for $1/3/C+2$. 
4.3 The Exchangeability Property

It is beneficial, from a computational point of view, to identify which permutations yield the same throughput. This motivates the search for exchangeable machines—those pairs whose location in the sequence can be interchanged without affecting the throughput of the line.

An close look at the transition graph of $1/(M)/(C)$ (figure 3) will show that the time between successive scheduling of event $\alpha_b$ takes the form $\omega_a(), \omega_b(),$ or $\omega_c()$. This means that there is no benefit in changing the positions of machines $M_a$ and $M_b$. Similarly, one can argue for any pair of machines in the sequence. This implies that there is no bowl phenomenon in $1/(M)/(C)$ systems. In fact, we have:

**Theorem 11** The departure epochs out of a $1/M/C$ system have the same distribution regardless of the sequence of the machines.

We give another example of the benefits of looking at transition graphs. One can verify also that in a open network of queues with infinite supply in front of the first
machine in the sequence and infinite buffers between all others, the departure process has the same distribution for all permutations of the machines (figure 4). We have (see also Ananthram(1988) for a special case):

**Theorem 12** The departure epochs out of a $G/G/1$ queues in series have the same distribution regardless of the sequence of the machines, if infinite supply is available in front of the first machine.

We say a network exhibits the *exchangeability* property if for every pair of nodes $(x,y)$, the departure epochs of the jobs out of the system $g_n(x, y) = g_n(y, x)$. This approach can accommodate situations where the processing time on a machine depends on the number in the queue, or equivalently when there are parallel servers at each stage. Note that the steady state stationary distributions of (single class) markovian systems that exhibit the *exchangeability* property have product form distributions. The point is that product form is a consequence of *exchangeability* and markov properties, and not due to poisson flows or independence. This explains why closed queuing networks have product form solution in spite of the fact that flows are non-poisson and queues are not independent.

We give one last example of the power of the *exchangeability* property. In figures 5 and 6 below, transition graphs are provided for two systems. It is easily verified that for the first, exchanging machines $M_a$ and $M_b$ will not change the throughput of the line; while in the latter case it will. Note, however, that the change in position of machine $M_c$ will affect the throughput of the line in both cases. This implies that these networks do not possess the product form network, although some machine positions are interchangeable.

### 4.4 Reversibility

The bottle phenomenon simply says that the bottleneck should be placed away from the center. It does not distinguish between symmetric positions from the center, namely between location $k$ and $M - k + 1$ in the line. This is because the two locations provide the same throughput, given that all the other machines are identical. In manufacturing systems, an important criteria is to minimize flow time of a part through the system. This is because, the response to a demand will be quick, and the average inventory in the system will be lower. Hence, it is important to identify the better of the symmetric pair of permutations with regard to flow times.

Afterall, the flow time of part $n$ is the difference between $T_{[M]}(n)$ and the time it entered the system. For a $1/M/C$ system, the enter time of part $n$ is equal to the de-
Figure 5: Transition graph 1
parture time of part $n - C$, namely $T[M](n - C)$, and so there is no benefit from changing the sequence of the machines. On the other hand, for a $N/(1\ldots1)/(1, C_2 \ldots C_N)$ system, the enter time of part $n$ is the start-time of the $n$th part on machine $[1]$; similarly one can determine enter times for a general $N/(M_1 \ldots M_N)/(C_1 \ldots C_N)$ system (the notation gets annoying; see Tayur(1990a) for the appropriate epoch). We are thus interested in a relation between start times and finish times in the forward and in the reversed sequence. As the proof is similar to that in Tayur(1990a), we give an example instead of a detailed proof of the reversibility theorem, which is central to the proof of theorem 6.

**Example 3** Consider a $3/(1,1,1)/(C_1,C_2,C_3)$ line. We can represent the sample path mechanics of this system (with the sequence $M_a, M_b, M_c$) by the network in figure 7 (see Tayur(1990a) for explanation, Muth(1979)). Each node represents an AND node; all the arcs that lead into it must be satisfied. Each arc corresponds to the service time of a part on a machine, or is a dummy arc with zero time to account for precedences. All vertical and curved arcs are dummy arcs. Among the horizontal arcs, the ones that are long are service time arcs, while the small segments are dummy arcs. Each node is therefore a start time or a finish time. Consider the system until
the $n$th customer has left the system. The equations are, for $1 \leq k \leq n$:

\[
T_a(k) = \max(T_a(k-1), T_b(k - C_1 - C_2), T_c(k - C)) + \omega_a(k)
\]

\[
T_b(k) = \max(T_a(k), T_b(k-1), T_c(k - C_2 - C_3)) + \omega_b(k)
\]

\[
T_c(k) = \max(T_b(k), T_c(k-1)) + \omega_c(k)
\]

The reversed line $(3/(1,1,1)/(C_3,C_2,C_1)$ with the sequence $M_c, M_b, M_a$) is symmetric to the forward line in the sense of the equations. All one needs to do is to check what epoch each node in the network corresponds to in the forward and the reverse sequence to prove theorem 5.

It is seen at once that if the bottleneck were in the first position instead of the last position in the forward sequence in an otherwise symmetric line, then the start time of the $n$th part will be later than when the sequence is reversed. As the departure epochs out of the line are equal stochastically for every part in both sequences, we have shown theorem 6. Formally, we give an induction argument. In fact, it does not matter what positions $k$ and $M - k + 1$ are being compared, as long as the rest of the line is symmetric.

**Proof (of theorem 6)** We can write the start times of the $n$th part at various machines (for simplicity we consider a $N/(1\ldots 1)/(1,C_2,\ldots,C_{N-1},1)$ line) as a function of finish times as follows. For the forward sequence we have:

\[
S_1(n) = \max(T_1(n-1), T_2(n - 1 - C_2), \ldots, T_N(n - C_1 - \ldots C_N))
\]

\[
S_j(n) = \max(T_j(n-1), T_{j+1}(n - C_j - C_{j+1}), \ldots, T_N(n - C_j - \ldots C_N))
\]

\[
\text{for } j = 2 \ldots N, n = 1, 2 \ldots
\]

For the reverse sequence ($r$'s in the superscript),

\[
S'_1(n) = \max(T'_1(n-1), T'_2(n - 1 - C'_2), \ldots, T'_N(n - C'_1 - \ldots C'_N))
\]

\[
S'_j(n) = \max(T'_j(n-1), T'_{j+1}(n - C'_j - C'_{j+1}), \ldots, T'_N(n - C'_j - \ldots C'_N))
\]

\[
\text{for } j = 2 \ldots N, n = 1, 2 \ldots
\]

We have for all $n \geq 1$, $\omega_j(n) =_{st} \omega_{N-j+1}(n)$ $\forall j \in \{2 \ldots N - 1\}$, and $\omega'_M(n) =_{st} \omega_1(n) \leq_{st} \omega_M(n) =_{st} \omega'_1(n)$. Further, we have $C_k = C_{N-k+1} = C'_k \forall k = 1 \ldots N$.

As in the reversibility theorem, we stop at $T_M(n^*) =_{st} T'_M(n^*)$. Since $T(n) = S(n) + \omega(n)$ and $T'(n) = S'(n) + \omega'(n)$, we have desired result by straightforward comparison of expressions in forward and reverse sequences.

Q.E.D.
Figure 7: Reversibility in a 3/(1,1,1)/() system.
Intuitively, what is happening is that in the forward sequence, the cards in the later cells are more likely to be waiting on their respective bulletin boards while in the reverse sequence they are more likely to be holding semi-finished parts in their respective output hoppers. This accounts for the difference in average inventory; in fact the same proof as above will show the echelon inventory will be stochastically lower at every machine. This is because, \( E(t) = (E_1(t) \ldots E_M(t)) \), the echelon inventory is defined as \( E_j(t) = \max(n_j : S_j(n_j) \leq t) - \max(n : T_M(n) \leq t) \ \forall j = 1 \ldots M \). In English, the echelon inventory at any stage (machine) is the sum of items downstream of the stage and the item on the machine (equivalent alternate definition using only downstream items will provide the same result).

Remark: The connection between flow times and inventory has been made precise in the above discussion. The connection is the clearest in a 1/M/C system. Here the input time of the nth part into the system is \( T_M(n - C) \), resulting in a flow time of \( T_M(n) - T_M(n - C) \). Now, \( \sum_{k=1}^{n} (T_M(kC) - T_M(kC - C)) \) gives the average flow time of \( n \) tagged parts that are \( C \) apart. The total time is \( T_M(nc) \) for \( nc \) parts to leave the system. We have \( \sum_{k=1}^{n} (T_M(kC) - T_M(kC - C)) \ast \frac{nC}{T_M(nc)} = C \), which is the inventory in the system. This is simply Little’s Law.

### 4.5 Concavity

There are three (somewhat equivalent) ways of proving theorem 1. First, is to prove concavity componentwise by a sample path construction and a coupling argument as in Tayur(1991a), Ananthram and Tsochas(1989). Second, one can show that the characteristic function (see Yao(1989)) is concave using a slightly more general synchronization mechanism. The third, following Meester and Shanthikumar(1990), is to use the explicit representation of the dynamics of the system and show the (stronger) joint concavity result. We proceed by the third method.

Let \( D_j(t, C) \) be the number of departures from machine \( j \) in the interval \((0, t)\) given that the kanban allocation is \( C = (C_1 \ldots C_N) \). As the raw material is infinite in front of cell 1, it can be verified that \( \{D(t, C)\} = \{D_j(t, C), j = 1 \ldots M\} \) is a Markov process on the state space \( \{(b^0, a^1, \ldots a^1_{M_1}, b^1, a^2, \ldots a^2_{M_2}, b^2, \ldots, a^N, b^N), \sum_{j=1}^{M} a^k_j + b^k - b^{k-1} = C_k, \ k = 1 \ldots N; 0 \leq a^k_i \leq C_k, i = 1 \ldots M_k, k = 1 \ldots N; -C_{k+1} \leq b^k \leq C_k, k = 1 \ldots N - 1; b^0 = b^N = 0\} \), where \( a^k_j = D_{j-1+\sum_{i=1}^{k-1} M_i}(t) - D_{j+\sum_{i=1}^{k-1} M_i}(t), j = 2 \ldots M_k \), the number in queue in front of machine \( j \) in cell \( k \). \( b^k \), is the difference between the contents of the output hopper of cell \( k \) and the bulletin board of cell \( k + 1 \). The expressions for \( b^k \) and \( a^k \) are complex functions of \( D_j(t) \)'s (see below
for the expression, see Tayur(1990a) for a complete sample path description). The transition rates out of a state is \( \mu_j^k \cdot I(a_j^k > 0) \), resulting in one more departure out of machine \( j \) in cell \( k \).

We construct another process \( \{D'_j(t, C), t \geq 0\} \equiv_{st} \{D_j(t, C), t \geq 0\} \) in the following manner. Let \( \nu = \sum_{i=1}^{M} \mu_i \), and \( \{A_n, n = 1, 2, \ldots\} \) a poisson process with rate \( \nu \), and \( \{U_n, n = 1, 2, \ldots\} \) be a sequence of uniform i.i.d random variables on \((0, \nu)\) independent of the poisson process. Define

\[
Z_{j,n} = I\{\sum_{i=1}^{j-1} \mu_i \leq U_n < \sum_{i=1}^{j} \mu_i\}
\]

\[
j = 1 \ldots M, \quad n = 1 \ldots
\]

In the constructed process a service completion is allowed to take place only at time points \( \{A_n, n = 1 \ldots\} \). To be clear, a service completion takes place at machine \( j \) in cell \( k \) at time \( A_n \) only if \( Z_{j,\sum_{i=1}^{k-1} M_i:n} = 1 \) and \( a_j^k > 0 \). Then, let \( D'_{j+\sum_{i=1}^{k-1} M_i:n}(C) \)
be the number of departures out of machine \( j \) in cell \( k \) in time interval \([0, A_n]\), with \( D'_{j,0}(C) = 0 \), and

\[
D'_{j+\sum_{i=1}^{k-1} M_i:n+1}(C) = D'_{j+\sum_{i=1}^{k-1} M_i:n}(C) + Z_{j,\sum_{i=1}^{k-1} M_i:n} \cdot I(a_j^k > 0).
\]

Now, as no state change occurs in \((A_n, A_{n+1})\), we set

\[
D'_{j+\sum_{i=1}^{k-1} M_i}(t, C) = D'_{j+\sum_{i=1}^{k-1} M_i:n}(C).
\]

\( A_n \leq t < A_{n+1}, \quad j = 1 \ldots M_k, \quad k = 1 \ldots N, \quad n = 1 \ldots \)

As the initial states in of both processes \( \{D'_j(t, C), t \geq 0\} \) and \( \{D_j(t, C), t \geq 0\} \) are the same, we have \( \{D'_j(t, C), t \geq 0\} \equiv_{st} \{D_j(t, C), t \geq 0\} \) as desired.

The next lemma is the desired explicit description of the mechanics of the system (we omit the proof).

**Lemma 3**

\[
D'_{j+\sum_{i=1}^{k-1} M_i:n+1}(C) = \min(D'_{j+1+\sum_{i=1}^{k-1} M_i:n}(C), D'_{j+\sum_{i=1}^{k-1} M_i:n}(C) + Z_{j+\sum_{i=1}^{k-1} M_i:n})
\]

\( j = 2 \ldots M_k, \quad k = 1 \ldots N \)

\[
D'_{1+\sum_{i=1}^{k-1} M_i:n+1}(C) = \min(D'_{\sum_{i=1}^{k-1} M_i:n}(C), D'_{1+\sum_{i=1}^{k-1} M_i:n}(C) + Z_{1+\sum_{i=1}^{k-1} M_i:n})
\]

\[
D'_{\sum_{i=1}^{k-1} M_i:n}(C) + C_k, D'_{\sum_{i=1}^{k-1} M_i:n}(C) + C_k + C_{k+1}, \ldots
\]

\[
D'_{\sum_{i=1}^{N} M_i:n}(C) + C_k + C_{k+1} + \ldots C_N
\]

\( k = 1 \ldots N, \quad n = 1 \ldots \)
Note that $D'_{j + \sum_{i=1}^{k-1} M_i; n+1} (C)$ is increasing and concave in $C$ for every $j = 1 \ldots M_k$, $k = 1 \ldots N$ if $D'_{j + \sum_{i=1}^{k-1} M_i; n} (C)$ is increasing and concave in $C$ for every $j = 1 \ldots M_k$, $k = 1 \ldots N$. Then, by induction, the initial condition that $D'_{j + \sum_{i=1}^{k-1} M_i; 0} (C) = 0$ and equation (1), the proof of theorem 1 is complete.

5 Conclusion

We have provided properties of throughput in kanban lines in good generality. Some of the results in this paper are simply alternate proofs of known results, but provide insight into the mechanics of buffered lines. Some results are generalizations of previous results or consider an alternate blocking mechanism. The proof of the bowl phenomenon is new, as are the proofs of the bounds. The detailed statement of reversibility, and its implication for flow-time reduction are new. The methods used in this paper can be used for other types of blocking mechanisms; to prove concavity, bowl phenomenon, flow-time reduction, interchangeability of queues, upper and lower bounds, and reversibility.

References


