Les Cahiers du GERAD ISSN: 0711–2440

Performance Analysis and Optimization of Kanban Based Production Policies in Multi-Part Unreliable Transfer Lines

S. Youssef
R.P. Malhamé

G–2010–58

October 2010

Performance Analysis and Optimization of Kanban Based Production Policies in Multi-Part Unreliable Transfer Lines

Stéphane Youssef
Roland P. Malhamé

GERAD and Département de génie électrique
École Polytechnique de Montréal
C.P. 6079, succ. Centre-ville
Montréal (Québec) Canada, H3C 3A7
{stephane.youssef;roland.malhame}@polymtl.ca

October 2010

Les Cahiers du GERAD
G–2010–58

Abstract

Optimization of single machine, single part-type dedicated Kanban policies in multi-part transfer lines with unreliable machines is considered. At each production stage, available machine production time is shared according to either one of two modes: (i) a synchronized mode in which distinct part-type work-in-process (wip) levels at any given stage can be shown to remain proportional to each other at all times, thus in effect reducing the analysis to that of a single part-type transfer line problem (ii) a more general prioritized mode such that at any stage the wip levels instead provably reach their maxima and minima in a fixed priority dependent sequence. A standard cost function which is a combined measure of long term storage and backlog costs under a constant vector of demand rates for different part types is employed to measure performance for any given choice of Kanban parameters. For the synchronized mode, approximate performance computation is achieved via a modification of an existing transfer line decomposition /aggregation technique, while the prioritized mode requires the development of a new approximation technique. Both approximate performance evaluation algorithms are validated against Monte Carlo simulations and are subsequently incorporated within dynamic programming schemes which compute best choices of Kanban parameters.

Résumé

Nous considérons le problème d'optimisation des seuils de Kanbans dédiés à des machines individuelles et pour chaque type de pièces particulier, dans des lignes de transfert multi-pièces en présence de machines non fiables. À chaque étape de la production, le temps utile d'une machine peut être exploité selon deux modes distincts : (i) un mode synchrone selon lequel les encours des pièces de différent type à chaque étape de production demeurent proportionnels et solidaires, réduisant l'analyse à celle d'une ligne de transfert mono-pièce, (ii) un mode prioritaire plus général tel que, à chaque étape de production, il est théoriquement établi que les encours de chaque type de pièce atteindront leurs maxima et minima dans un ordre fixe correspondant à la priorité. Une fonction coût comportant à la fois des coûts de stockage et des pénalités de retard de livraison en présence d'un vecteur fixe de taux de demande de pièces est adoptée en vue d'évaluer la performance de la ligne pour un choix donné de seuils de Kanbans. Dans le cas du mode de production synchrone, une approximation de la performance est obtenue à partir d'une modification de méthode de décomposition-agrégation existante, alors qu'une technique d'approximation nouvelle est développée pour analyser la performance du mode prioritaire. Les deux techniques d'évaluation sont validées par comparaison avec des résultats de simulation de Monte-Carlo, et sont ensuite incorporées à l'intérieur d'un schéma de programmation dynamique visant à calculer les meilleurs choix de seuils de Kanbans.

1 Introduction

The flow control problem has drawn considerable attention for unreliable transfer lines composed of machines in tandem separated by part type dedicated buffers. Such systems are subject to numerous random phenomena such as machine failures and repairs and variability in the level and nature of the demand. In this context, positive inventories, or work-in-process (wip) within intermediate machine buffers can act as an insurance policy against sources of uncertainty. Intermediate storage does indeed play a key role in the decoupling of machines within the line by allowing partial continuation of production when isolated machine failures occur in the line. Nevertheless, the stored wip or inventory can be associated with high storage costs, immobilized capital, and in general high parts transit times within the transfer line. On the other hand, overly reducing wip and inventory levels can significantly affect the productivity of the transfer line.

For a given production mode, i.e. for the purposes of this paper a rule specifying how productive machines time is shared between the manufacturing of different part types, our ultimate objective is to determine optimum kanban sizes (virtual or real buffer sizes) for each part type and at each stage in the production process, so that given a constant vector of demand rates for different part types, a long term average measure of combined storage and backlog costs is minimized. This is achieved in two steps: First a computationally effective approximation method for transfer line performance evaluation for an arbitrary fixed choice of kanban parameters is developed; secondly, this approximation tool is used within a dynamic programming scheme aimed at discovering a choice of kanban parameters which effectively optimizes performance. Two production modes will be considered in this paper. The first one, called *synchronized production*, is one such that all wip levels at any given stage within the transfer line, or inventories/backlogs of different part types move in unison in that they maintain constant relative sizes at all time. This mode of production is particularly attractive because in effect, it corresponds to a periodic production pattern between different part types at each machine, and is amenable to analysis via single part transfer line decomposition/aggregation techniques. The second production mode analyzed here is *prioritized production*; it is more general in that it allows for synchronized production as a special case. It requires the devising of a more complex new decomposition technique for approximation purposes, but holds the potential of better performance than synchronized production schemes.

The Kanban production policy, besides its simplicity, turns the flow control problem into a parameters optimization problem where parameters correspond to different part type Kanban levels (buffer sizes). For a single unreliable machine system, it is equivalent to a so-called hedging policy early on identified as a candidate for optimality in [13], and later rigorously proved optimal and precisely characterized for various instances of single machine single part systems (see [1], [4], [12], [21], [15], [14] and [8]).

For single part unreliable transfer lines under Kanban production policies, it is a fact that mean values of inventories for multi-machine lines can be obtained in closed form only for the cases where buffers are inexistent (machines could be aggregated) or infinite (machines completely decoupled). Furthermore, Monte Carlo simulations can be computationally very expensive and difficult to implement for the performance evaluation of long transfer lines. This is why a large number of decomposition/aggregation techniques have been developed in the literature of multi-machine single part lines. The main goal of these techniques is to make the computation of performance evaluation tractable in the approximated line model. Computational algorithms for different decomposition techniques have undergone vast improvements. [11] proposed an important algorithm for buffer optimization in decomposed multi-machine single-part production system based on the algorithms initially proposed in [10]. Also, a competing decomposition approach was developed in [6]. An efficient and quite precise decomposition technique based on two main approximations, the machine decoupling approximation and the demand averaging principle, was proposed in [18] and [19] for the case of transfer lines with respectively so called *partially-homogeneous* and *nonhomogeneous* machines.
In this paper, we follow this technique which leads to a simpler causality structure in the sense that influence propagates from upstream towards downstream, thus defining decision stages *best suited for a dynamic programming optimization formulation*. Furthermore, some structural properties of optimal buffering profiles for the special case of transfer lines with identical machines can be rigorously established (see [18]). While the analysis of this decomposition technique is limited to single-part systems, we develop here its extension to multi-part systems.

The special case of a single unreliable machine producing multiple parts has been also studied in the literature ([5], [9], [17], [20], [22] and [23]), while the body of work on unreliable transfer lines producing multiple part types remains very limited. Notable exceptions are [13] where a general modeling framework was developed for the optimal control of multi-part multiple machine manufacturing systems and qualitative insights and heuristic algorithms were derived.
More recently, Colledani et al [7] have considered performance analysis of multi-part unreliable manufacturing systems with transfer line architectures as well as more general architectures, starting from approximate aggregation tools developed for the queuing theory context ([2]). The current work shares with [13] the modeling set up while it differs from [7] in that in the latter, the models are defined in discrete time, and for a given part type, the processing times are identical on all machines; more importantly however, it appears that the resulting approximate analysis cannot be easily incorporated within an optimization scheme, while in our view, the main virtue of the framework proposed in this paper resides in the fact that dynamic programming emerges as a natural optimization tool.

The rest of this paper is organized as follows. In Section 2, the mathematical model is presented and the Kanban optimization problem is formulated. In Section 3, we recall two transfer lines related approximations which are key to the rest of the analysis. In Section 4, the synchronized production mode is introduced and some of its mathematical properties are established. In Section 5, an approximate model of the synchronized production mode is presented and validated via Monte Carlo simulation. In Section 6, the corresponding dynamic programming problem is formulated and solved, with numerical results reported in Section 7. Sections 8, 9, 10, 11 closely parallel Sections 4, 5, 6, 7 respectively, for the alternate prioritized production mode. Section 12 is our Conclusion.

2 Problem formulation

We consider in this paper a production line consisting of n machines in tandem separated by buffers and producing m part types. Every machine can be in one of two modes: an operational mode or a failure mode. For $i = 1, \cdots, n$, α_i is a binary variable indicating the mode of machine M_i. Thus for $\alpha_i = 1$, M_i is operational, while for $\alpha_i = 0$, M_i is in a failure state. Each α_i is assumed to evolve according to a two-state continuous time Markov chain with failure rate p_i and repair rate r_i. The production line is considered to be nonhomogeneous in the sense that machines have different failure and repair rates. When $\alpha_i = 1$, machine M_i can produce part j (for $j = 1, \cdots, m$) with a production rate lying between 0 and a maximum rate of k_{ji}; and its production rate is 0 when $\alpha_i = 0$. Note that the maximum production rate of k_{ji} could be attained only in the case of single part production, that of part j. The instantaneous rate of demand for finished parts j is assumed to be a constant d_j ($j = 1, \cdots, m$).

2.1 Assumptions

We make the following assumptions:

- Raw material is always available for the first machine. This means that the first machine is never starved.
- For all $j = 1, \cdots, m$, maximum production rates for part j respect the following inequality: $k_{j1} \geq k_{j2} \geq \cdots \geq k_{ji} \geq \cdots \geq k_{jn}$. This condition guarantees that the level of inventory for part j in buffer i (x_{ji}) will increase as long as machine M_i produces part j with the maximal rate k_{ji}, no matter the production rate of part j by machine M_{i+1}.
- Backlog is permitted only for the last machine (M_n). x_{ji} is the inventory level of part j in the storage bin downstream of machine M_i. Let c_{ji} be the instantaneous storage cost per unit x_{ji} for $j = 1, \cdots, m$ and $i = 1, \cdots, n$ - 1. For the last machine, its instantaneous storage (respectively backlog) cost per unit inventory x_{jn} is c_{jn}^{+} (respectively c_{jn}^{-}).
- We assume an *isolated machine demand feasibility condition* (see [19]): For $j = 1, \cdots, m$ and $i = 1, \cdots, n$:

$$\frac{r_i}{r_i + p_i} k_{ji} > d_j \tag{1}$$

This condition is easily interpreted as follows. If machine M_i has to satisfy a demand d_j for part j, it is necessary that the production of part j by machine M_i at full capacity (k_{ji}) during the mean time of operational mode of M_i be higher than the demand d_j.

2.2 System dynamics

The work in process $x_{ji}(t)$, for $j = 1, \cdots, m$ and $i = 1, \cdots, n$, evolves according to:

$$\frac{dx_{ji}(t)}{dt} = u_{ji}(t) - u_{j,i+1}(t) \tag{2}$$

where $u_{ji}(t)$ denotes the instantaneous production rate of machine M_i and $u_{j,n+1}(t) = d_j$.

2.3 Objectives

In this paper, we have two main goals. On the one hand, to construct a simple and effective mathematical model able to represent the system behavior and thus help in its understanding and its analysis. On the other hand, to develop a feedback control law, so as to minimize an adequate measure of the expected long term average storage and backlog costs. We look for optimality within a restricted parameterized class of feedback control policies: that of Kanban policies. Such policies are characterized by a set of critical inventory or work in process levels to be maintained whenever possible and have been proven optimal in single machine two-state manufacturing systems amongst a specific class of admissible control policies (see [12]). The critical inventory level is extended here as an insurance policy against potential machine failures within the line, and the ensuing costs associated with backlogged demand. Each work-in-process i for every part j, x_{ji}, is associated with its own Kanban level z_{ji}. If x_{ji} is less than z_{ji}, machine M_i should produce part j at the maximum rate k_{ji}; and if x_{ji} exactly equals z_{ji}, machine M_i should produce part j at the same rate as M_{i+1} ($u_{j,i+1}(t)$) so as to maintain x_{ji} at the level z_{ji}. Consequently, x_{ji} will never exceed the critical level z_{ji}. The optimization problem is thus reduced to the search for hedging levels (Kanban levels) minimizing an appropriately defined combined measure of storage and backlog costs. This measure is generally defined as follows.

$$J^{\{z\}}(x_0, \alpha_0^T) = \lim_{t\to\infty} \frac{1}{t} \Bigg\{ \int_0^t E\Big[\sum_{i=1}^{n-1} \sum_{j=1}^{m} c_{ji} x_{ji}(\tau) + \sum_{j=1}^{m} (c_{jn}^{+} x_{jn}^{+}(\tau) + c_{jn}^{-} x_{jn}^{-}(\tau)) | x_0, \alpha_0^T \Big] d\tau \Bigg\} \tag{3}$$

where z is the matrix of hedging levels, $\alpha^T(t) = [\alpha_1(t), \alpha_2(t), \cdots, \alpha_n(t)]^T$ the vector of machine states at instant t and the index 0 correspond to the initial time where the line is considered. Note that under an ergodicity assumption on the controlled processes, the above limit will be independent of the initial system state.

3 Two key approximations in the decomposition/aggregation technique

For multi-machine manufacturing systems, average levels of the work-in-process and inventories can be calculated neither analytically nor numerically. It is thus essential to resort to an approximate decomposition method. In addition to making numerical performance evaluation possible, line decomposition can play a key role in simplifying buffer sizes optimization. It allows decomposing an n-machine line into n approximately decoupled machines. In this paper, we use the decomposition method initially proposed in [16] and enhanced in [18] and [19]. This decomposition method is based on two key approximations: the machine decoupling approximation (*MDA*) and the demand averaging principle (*DAP*).

3.1 The Machine decoupling approximation (MDA)

This approximation helps in decoupling a given machine (except the first machine in the line) from its upstream counterparts. It aims at efficiently summarizing the impact of the universe upstream of this machine on its operation. In the context of transfer lines, the universe upstream of a given machine acts like an unreliable supply of parts. Thus from the point of view of the ability of machine M_{i+1} to produce, what matters is the value of the binary part type j supply state $I_{ji}(t)$, $j = 1, \cdots, m$.

$$\begin{aligned} I_{ji}(t) = I\Big[&\big\{x_{ji}(t) > 0\big\} \cup \\ &\big\{x_{ji}(t) = 0, \alpha_i(t) = 1, I_{j,i-1}(t) = 1\big\}\Big] \end{aligned} \tag{4}$$

where $I(\bullet)$ is the indicator function and $I_{j0}(t) \equiv 1$. The machine decoupling approximation is the statement that $I_{ji}(t)$ is a random process which is independent of machine M_{i+1} operating state $\alpha_{i+1}(t)$, for $j = 1, \cdots, m$ and $i = 1, \cdots, n$ - 1.

3.2 Demand averaging principle (DAP)

Let the process x_{ji} be qualified as ergodic if the unreliable machine M_i, fed by the unreliable work-in-processes of part j at stage i - 1 ($x_{j,i-1}$), can satisfy on average the demand from its downstream work-in-processes of part j, x_{ji}. Note that for a constant rate d_j of demand for finished part type j and under ergodic controls, every machine in the transfer line will be responding to the same average rate of demand for part type j; otherwise inventory would build up to infinity or deplete to zero in some parts of the transfer line. The aim of *DAP* is to use this observation to achieve an approximate but compact representation of the effect of machines *downstream* of a given buffer i, on the dynamics of the associated storage level x_{ji} for part j and supply state $I_{ji}(t)$. Indeed, it allows one to develop a simple model of the evolution of x_{ji} and $I_{ji}(t)$ as a semi-Markov or Markov chain. *DAP* states the following: "*Under ergodic Kanban production controls in a transfer line subjected to a constant rate of demand d_j of part j (j = 1, $\cdots$, m), the steady-state mean value of stock level $x_{ji}(t)$ over the active portions of the supply cycle (periods where $I_{ji}(t)$ = 1) depends only on the steady-state mean value of the stochastic instantaneous rate at which parts j are drawn from buffer i by machine M_{i+1} when $I_{ji}(t)$ = 1 and is independent of its higher moments*". This means that $u_{j,i+1}(t)$ could be replaced during the active portions of the x_{ji} supply cycle by an appropriate constant level *without* affecting the mean of $x_{ji}(t)$. Let $\tilde{x}_{ji}$ be the fictitious process approximating x_{ji}, i.e. the work-in-process of the fictitious markovian machine $\tilde{M}_i$ subjected by virtue of *DAP* to a constant demand when the supply x_{ji} is active. Let a_{ji} be defined as the expected steady-state value of the instantaneous availability coefficient of supply x_{ji}. Given that the steady-state mean of the $u_{j,i+1}(t)$ process must be d_j, one can write

$$\begin{aligned} d_j &= \lim_{t\to\infty} \left\{ \begin{matrix} E[u_{j,i+1}(t)|I_{ji}(t) = 1]Pr[I_{ji}(t) = 1] \\ +E[u_{j,i+1}(t)|I_{ji}(t) = 0]Pr[I_{ji}(t) = 0] \end{matrix} \right\} \\ &= u^+_{j,i+1} a_{ji} \end{aligned} \tag{5}$$

where $u^+_{j,i+1}$ is the steady-state mean of $u_{j,i+1}(t)$ during the active portions of x_{ji}. From (5), the constant appropriate level of $u^+_{j,i+1}$ is d_j/a_{ji}. This means that machine $\tilde{M}_i$ is subjected to a constant demand of d_j/a_{ji} for part j (d_j/a_{ji} is higher than d_j since a_{ji} lies between 0 and 1). Thus, the division by factor a_{ji} serves to compensate for the loss of demand from machine M_{i+1} whenever x_{ji} is not available.

4 The synchronized production mode

The synchronized production mode assumes an idealized simultaneous production of the m parts at every stage in the sense that, at any given time, every machine if operational not starved and not blocked, produces all the parts (see [24]). In practise, this is equivalent to assuming that every machine dedicates a certain percentage of its production rate, when it is operational, to each part, and it does so in a cyclic way, with negligible setup times. Thus, the synchronized production mode is a class of periodic production rules, in

which we consider only the special case where, for any given machine, the maximal simultaneous production rates and the Kanban levels of all of the parts are *consistent* with the demand rates ratio. More specifically, this could be mathematically expressed by the following equalities. For $i = 1, \cdots, n$:

$$\frac{z_{ji}}{z_{li}} = \frac{\bar{k}_{ji}}{\bar{k}_{li}} = \frac{d_j}{d_l} \quad \text{for } j,\ l\ (\neq j) = 1, \cdots, m \tag{6}$$

where $\bar{k}_{ji}$ is the new maximum production rate of part j by M_i for the parallel synchronized production scheme. Note that $\bar{k}_{ji}$ must obviously be less than k_{ji}. The maximum production rates of the synchronized production ($\bar{k}_{ji}$'s) represent the intersection of the m-dimensional demand vector direction with the upper boundary of the feasible production rates space (see Figure 7 below). Indeed, this upper boundary is the hyper surface formed by the feasible sets of simultaneous maximum production rates (k_{ji}'s). The $\bar{k}_{ji}$'s are thus given, for $i = 1, \cdots, n$, by:

$$\begin{aligned} \bar{k}_{1i} &= \frac{\prod_{l=1}^{m} k_{li}}{\sum_{j=1}^{m} \frac{d_j}{d_1}\Big(\prod_{l=1, l\neq j}^{m} k_{li}\Big)} \\ \bar{k}_{ji} &= \frac{d_j}{d_1}\bar{k}_{1i} \quad \text{for } j = 2, \cdots, m \end{aligned} \tag{7}$$

The following proposition establishes the solidarity property of all part type paths under the synchronized production mode.

Proposition 1 *Under assumption (6) of the synchronized production mode, wips associated with different part types at every stage evolve in unison in the sense that their instantaneous levels remain consistent with the ratio of the demand rates for each part type. More specifically, for $i = 1, \cdots, n$ and $j, l = 1, \cdots, m$ $(l \neq j)$: $x_{ji}(t)/x_{li}(t) = d_j/d_l$ for all $t > 0$ whenever $x_{ji}(t) \neq 0$ and $x_{li}(t) \neq 0$.*

Proof. See Appendix A □

This behavior has the immediate consequence that the availability coefficients for all part types at any given stock level are identical, i.e. for $i = 1, \cdots, n$, $I_{1i}(t) = I_{2i}(t) = \cdots = I_{mi}(t) = I_i(t)$ and hence $a_{1i} = a_{2i} = \cdots = a_{mi} = a_i$.

5 Approximate performance analysis of the synchronized production mode

In this section we use the previously described approximations to produce a simplified model of wip evolution under the Kanban based synchronized production mode.The simplified model is subsequently validated via monte Carlo simulations.

5.1 A lower order Markovian model

The binary state of the first machine M_1, $\alpha_1(t)$, is represented by a two-state markovian chain with repair rate r_1 and failure rate p_1 which according to *DAP* could be considered subjected to a constant demand of d_j/a_1 of parts type j. Also, the binary stock part type availability processes associated with intermediate buffer part type j in buffer i ($i = 1, \cdots, n$ - 1, $j = 1, \cdots, m$), $I_{ji}(t)$, could be represented by a two-state Markov chain evolving according to a certain failure and a certain repair rate. For index i, the repair rate, denoted $\tilde{r}_i$, has to take into consideration both the real repair rate of machine M_i (r_i) and all repair rates of machines upstream of stage i. Let p_{s_i} be the failure rate of any of the (synchronized) part type availability processes in buffer i and $\tilde{r}_i$ its repair rate. Let $t_{rz_{ji}}$ and $\bar{t}_{rz_{ji}}$ be the first return time to zero of the x_{ji} process and its mean value respectively. Proposition 1 implies that $\bar{t}_{rz_{1i}} = \bar{t}_{rz_{2i}} = \cdots = \bar{t}_{rz_{mi}} = \bar{t}_{rz_i}$. p_{s_i} is considered to be equal to $1/\bar{t}_{rz_i}$. Considering the probabilistic characteristics of the process $I_i(t)$, the proportion of time

where the buffer $x_{ji}(t)$ (for all $j = 1, \cdots, m$) is positive is given by $(1/p_{s_i})/((1/p_{s_i})+(1/\tilde{r}_i))$. This proportion is also equal to a_i, thus yielding the following value of p_{s_i}.

$$p_{s_i} = \frac{\tilde{r}_i(1 - a_i)}{a_i} \tag{8}$$

Figure 1 corresponds to a low order Markov chain representation of the wip availability process associated

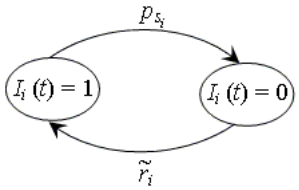

Figure 1: Markovian model of wip availability process associated with buffer $x_{ji}(t)$, $j = 1, \cdots, m$ and $i = 1, \cdots, n$ - 1.

with buffer $x_{ji}(t)$. By virtue of MDA, the Markov chain state-space of the decomposed equivalent machine $\tilde{M}_i$ is the cartesian product of the $I_{i-1}(t)$ Markov chain and that of the mode $\alpha_i(t)$ of machine M_i. This results in a four-state Markov chain with a single operational mode and three failure modes (Figure 2). The failure mode due to the simultaneous failures of I_{i-1} and α_i is neglected because it occurs very rarely in general. Thus, only two failures remain. The first one $Fail1$ represents machine M_i operational ($\alpha_i = 1$) and process $x_{j,i-1}$ unavailable ($I_{i-1} = 0$); while $Fail2$ represents machine M_i in failure ($\alpha_i = 0$) and process $x_{j,i-1}$ available ($I_{i-1} = 1$). In view of the assumed independence of processes α_i and I_{i-1} (MDA), probabilities of $Fail1$ and $Fail2$ are given as:

$$P[Fail1] = \frac{r_i}{r_i + p_i}(1 - a_{i-1}) \tag{9}$$

$$P[Fail2] = \frac{p_i}{r_i + p_i}a_{i-1} \tag{10}$$

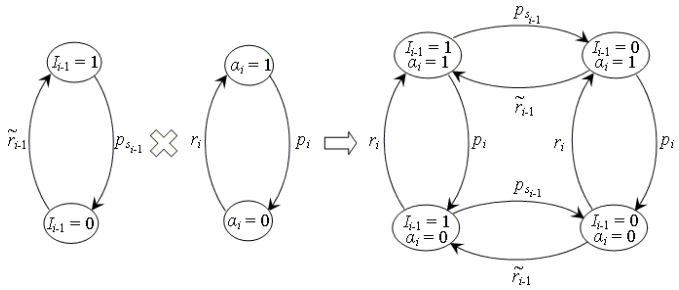

Figure 2: Cartesian product of the two models of I_{i-1} and M_i.

It follows that the probability that machine $\tilde{M}_i$ is in failure due to $Fail1$ (respectively $Fail2$) is given by (11) (respectively (12)).

$$\begin{aligned} P[Fail1|Failure] &= \frac{\frac{r_i}{r_i+p_i}(1 - a_{i-1})}{\frac{r_i}{r_i+p_i}(1 - a_{i-1}) + \frac{p_i}{r_i+p_i}a_{i-1}} \\ &= \frac{r_i(1 - a_{i-1})}{r_i(1 - a_{i-1}) + p_i a_{i-1}} \end{aligned} \tag{11}$$

$$\begin{aligned} P[Fail2|Failure] &= \frac{\frac{p_i}{r_i+p_i}a_{i-1}}{\frac{r_i}{r_i+p_i}(1 - a_{i-1}) + \frac{p_i}{r_i+p_i}a_{i-1}} \\ &= \frac{p_i a_{i-1}}{r_i(1 - a_{i-1}) + p_i a_{i-1}} \end{aligned} \tag{12}$$

If the equivalent machine $\tilde{M}_i$ is in failure because of $Fail1$ (respectively $Fail2$), it returns to the operational mode with a repair rate $\tilde{r}_{i-1}$ (respectively r_i). The two remaining failure modes could then be aggregated together yielding an approximate single failure mode machine (Figure 3). In order to minimize the impact of the loss of information resulting from the aggregation of the two failure modes, the repair rate of the resultant machine $\tilde{M}_i$, $\tilde{r}_i$, is computed as an expectation conditional on being in a failure mode. This yields the expression in (13):

$$\tilde{r}_i = \frac{r_i(1-a_{i-1})}{r_i(1-a_{i-1})+p_i a_{i-1}}\tilde{r}_{i-1} + \frac{p_i a_{i-1}}{r_i(1-a_{i-1})+p_i a_{i-1}} r_i \tag{13}$$

for $i = 2,\cdots, n$. In addition, we impose that the steady-state probability of the state "$\tilde{\alpha}_i = 1$" be $(\tilde{r}_{i-1}/(p_{s_{i-1}}+\tilde{r}_{i-1}))(r_i/(r_i+p_i))$, the latter being an exact probability under MDA. This yields:

$$\tilde{p}_i = \left[\frac{r_i+p_i}{r_i a_{i-1}} - 1\right]\tilde{r}_i \tag{14}$$

Relying on DAP, a_i is computed from the single unreliable machine with no backlog expressions derived in [12], considering that isolated equivalent machine $\tilde{M}_i$ is subjected to demand d_j/a_i of part type j.

$$a_i = 1 - \frac{\tilde{p}_i}{\tilde{r}_i+\tilde{p}_i}\left[\frac{1-\rho_i}{1-\rho_i e^{-\mu_{ji}(1-\rho_i)z_{ji}}}\right] \tag{15}$$

with:

$$\mu_{ji} = \frac{\tilde{p}_i}{\bar{k}_{ji} - \frac{d_j}{a_i}} \tag{16}$$

$$\rho_i = \frac{\tilde{r}_i(\bar{k}_{ji} - \frac{d_j}{a_i})}{\tilde{p}_i \frac{d_j}{a_i}} \tag{17}$$

Due to (6), $\mu_{ji} z_{ji} = \mu_{li} z_{li}$ for any j, $l = 1, \cdots, m$ and ρ_i does not depend on the part type (the index j in (17) could be replaced by 1, 2, ... or m). Solving (15) for z_{ji}, using (16) and (17), yields the following expression of z_{ji}.

$$z_{ji} = \frac{1}{\lambda_{ji}} \ln \gamma_i \tag{18}$$

with:

$$\lambda_{ji} = -\mu_{ji}(1-\rho_i) \tag{19}$$

$$\gamma_i = \frac{\tilde{p}_i\left[\tilde{r}_i \frac{\bar{k}_{ji}}{d_j} - (\tilde{r}_i+\tilde{p}_i)\right]}{\tilde{r}_i(\tilde{r}_i+\tilde{p}_i)(1-a_i)(\frac{\bar{k}_{ji}}{d_j} - \frac{1}{a_i})} \tag{20}$$

The storage cost for machine $\tilde{M}_i$, for $i = 1, \cdots, n$ - 1, can also be obtained from [12]. This cost is a function of three variables $\tilde{r}_i, \tilde{p}_i$ and a_i (via (16) and (17)).

$$T_{ji}(\tilde{r}_i, \tilde{p}_i, a_i) = \frac{c_{ji}\rho_i}{(\tilde{r}_i+\tilde{p}_i)(1-\rho_i\gamma_i)}\left[\bar{k}_{ji}\frac{1-\gamma_i}{1-\rho_i} - \frac{\gamma_i}{\lambda_{ji}}(\tilde{r}_i+\tilde{p}_i)\ln\gamma_i\right] \tag{21}$$

The last equivalent machine $\tilde{M}_n$ allows backlog and is also considered as approximately isolated and evolving according to a two-state Markov chain with rates $\tilde{p}_n$ and $\tilde{r}_n$. This machine has been studied and optimized in [4]. Let us define $\bar{z}_n = z_{jn}/d_j \;\forall\; j = 1, \cdots, m$. According to [4], the total storage and backlog costs of this machine for part j are given below.

$$T_{jn}(\tilde{r}_n, \tilde{p}_n, \bar{z}_n) = c^+_{jn} d_j \bar{z}_n + \frac{\tilde{p}_n \bar{k}_{jn}}{\sigma_{jn}(\tilde{r}_n+\tilde{p}_n)(\bar{k}_{jn}-d_j)}\left[(c^+_{jn}+c^-_{jn})e^{-\sigma_{jn} d_j \bar{z}_n} - c^+_{jn}\right] \tag{22}$$

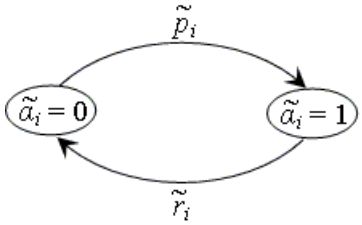

Figure 3: Simplified equivalent two-state model of $\tilde{M}_i$.

with:

$$\sigma_{jn} = \frac{\tilde{r}_n(\bar{k}_{jn} - d_j) - \tilde{p}_n d_j}{d_j(\bar{k}_{jn} - d_j)} \tag{23}$$

Note that the last buffer does not have an availability coefficient as it can go to -∞, so the variable to be optimized is $\bar{z}_n$. The ergodicity condition requires σ_{jn} to be positive for all $j = 1, \cdots, m$; otherwise, the demand d_j will not be satisfiable by machine $\tilde{M}_n$.

5.2 Model validation

In order to evaluate the accuracy of the approximate mathematical model presented in Section 5.1, we will compare its performance to that of Monte-Carlo simulations for the same transfer lines. Over fifty different transfer lines were tested. We report the results for only three lines considered as representative samples. For these three lines, we fixed $n = m = 3$; the demand vector for the three part types $d = [1\ 1.2\ 0.9]$, the unit storage cost $c_{ji} = 2$ for $j = 1, 2, 3$ and $i = 1, 2$, the unit storage cost of the last machine $c^+_{j3} = 2$ for $j = 1, 2, 3$. Finally, the backlog cost for the three part types at the last buffer $c^-_{j3} = 10$ for $j = 1, 2, 3$. Remaining transfer line data is specified in Tables 1, 2 and 3.

Table 1: Data for the first line.

p_i			r_i		
0.1	0.08	0.15	0.5	0.45	0.6

k_{ji}			z_{ji}		
9	8.7	8.5	5	3	7
11	10.5	10	6	3.6	8.4
7.955	7.786	7.532	4.5	2.7	6.3

Table 2: Data for the second line.

p_i			r_i		
0.05	0.07	0.075	0.45	0.55	0.5

k_{ji}			z_{ji}		
7.5	8	8.5	2	2.72727	6
8.7	7.5	6.5	2.4	3.27273	7.2
7.5	6.5	6	1.8	2.45455	5.4

Table 3: Data for the third line.

p_i			r_i		
0.2	0.15	0.1	0.6	0.45	0.4

k_{ji}			z_{ji}		
9	8.7	8.5	3	5	7
11	10.5	10	3.6	6	8.4
7.95536	7.78551	7.53165	2.5	4.5	6.3

Let T_{tot} be the mean total storage and backlog cost of the line, i.e. $T_{tot} = \sum_{j=1}^{3}\sum_{i=1}^{2} T_{ji}(\tilde{r}_i, \tilde{p}_i, a_i) + \sum_{j=1}^{3} T_{j3}(\tilde{r}_3, \tilde{p}_3, \bar{z}_3)$. Tables 4, 5 and 6 show the results of the comparison between approximate analytic evaluation of the system performance using our model and that of Monte-Carlo simulations for the three lines.

Table 4: Comparison of theory based and simulation based results, Line 1.

	Model based estimate	M.-C. simulation	% relative error
a_1	0.97746	0.98334	0.60156
a_2	0.94462	0.95198	0.77915
T_{tot}	83.62864	84.09018	0.55189

Table 5: Comparison of theory based and simulation based results, Line 2.

	Model based estimate	M.-C. simulation	% relative error
a_1	0.95757	0.95681	0.07937
a_2	0.95099	0.95809	0.74659
T_{tot}	59.54076	61.0687	2.56621

Table 6: Comparison of theory based and simulation based results, Line 3.

	Model based estimate	M.-C. simulation	% relative error
a_1	0.93627	0.94692	1.13749
a_2	0.92741	0.95931	3.43969
T_{tot}	81.24034	81.24801	0.00944

These results suggest that the theoretical estimates of the availability coefficients and total storage and backlog costs can be quite accurate. The maximal error of the evaluation of the mean total cost is around 2.5%. Furthermore, numerous other Monte Carlo based validations were made. They do tend to confirm the accuracy of the theoretical estimates based on the approximate model. The only exceptions are cases where the transfer line is strongly stressed i.e., when it has to respond to part type demand rates at the limits of its capacity set. In such cases, random sample paths of the backlog process display significant variance and even the Monte Carlo simulations display extremely slow convergence.

6 Kanbans optimization via dynamic programming

6.1 Optimization problem

In Section 5.1, we developed a simplified approximate two-state model for the equivalent machine $\tilde{M}_i$, for $i = 1, \cdots, n$ (with $\tilde{M}_1 = M_1$, $\tilde{p}_1 = p_1$ and $\tilde{r}_1 = r_1$). States ($\tilde{r}_i$,$\tilde{p}_i$) at every stage i are a subset of $\mathbb{R}^2$. The decision variable for $i = 1, \cdots, n$ - 1 is the availability coefficient a_i, which is bounded with a lower and an upper bound; while for the last stage the decision variable is $\bar{z}_n$ which has only an upper bound. The global optimization problem separates naturally into two sub-problems, the first one from 1 to n - 1 and the second one for stage n. We will solve the two sub-problems separately.

We start with the last equivalent machine and assume that its parameters which are a consequence of sizing decisions taken at previous production stages, are given. Recall that the last equivalent machine $\tilde{M}_n$ evolves according to a two-state Markov chain with rates $\tilde{p}_n$ and $\tilde{r}_n$ and is subject to a demand d_j of part type j ($j = 1, \cdots, m$). Bielecki and Kumar [4] developed an analytical expression for the optimal hedging level minimizing the total storage and backlog costs for this machine in the case of single part production. Because of the solidarity of evolution amongst the inventories/backlogs of distinct part types under the synchronized

production mode, the Bielecki-Kumar result can be extended in a straightforward way to this case. Indeed, the total cost related to the last stage of production is as follows:

$$T_F(\tilde{r}_n, \tilde{p}_n, \bar{z}_n) = \sum_{j=1}^{m} T_{jn}(\tilde{r}_n, \tilde{p}_n, \bar{z}_n) \tag{24}$$

It can be aggregated into the cost associated with a single "macro" part with adequately augmented running costs. Following [4], if $\partial T_F(\tilde{r}_n, \tilde{p}_n, \bar{z}_n)/\partial \bar{z}_n > 0$ at $\bar{z}_n = 0$, the optimal value of $\bar{z}_n$ will be zero; otherwise, a positive optimal value of $\bar{z}_n$ is to be found for which $\partial T_F(\tilde{r}_n, \tilde{p}_n, \bar{z}_n)/\partial \bar{z}_n = 0$ and $\partial^2 T_F(\tilde{r}_n, \tilde{p}_n, \bar{z}_n)/\partial \bar{z}_n^2 > 0$. Note that a negative value of $\bar{z}_n$ can never be optimal (see [4]). Finally, let β be defined as follows.

$$\beta = \frac{(\tilde{r}_n + \tilde{p}_n)(\bar{k}_{1n} - d_1) \sum_{j=1}^{m} c_{jn}^{+} d_j}{\tilde{p}_n \bar{k}_{1n} \sum_{j=1}^{m} (c_{jn}^{+} + c_{jn}^{-}) d_j} \tag{25}$$

From the above discussion, the optimal value of $\bar{z}_n$ minimizing (24) is given by the following equation.

$$\bar{z}_n^* = \begin{cases} 0 & \text{if } \beta \geq 1 \\ \frac{1}{\sigma_{1n} d_1} ln(\frac{1}{\beta}) & \text{otherwise} \end{cases} \tag{26}$$

Consequently, the optimal storage and backlog costs of the last machine $\tilde{M}_n$ ($T_F^*(\tilde{r}_n, \tilde{p}_n, \bar{z}_n^*)$) is obtained by substituting the value of $\bar{z}_n^*$ of (26) in (24) according to the value of β (25). This cost is a function of $\tilde{r}_n$ and $\tilde{p}_n$ which depend on a_{n-1} (see (13) and (14)) and thus has to be taken into consideration when solving the global dynamic programming problem.

With the last stage optimal cost expression thus obtained as a function of the last machine parameters $(\tilde{r}_n, \tilde{p}_n)$, it becomes possible to address the second subproblem of optimizing the rest of the sizing decisions upstream. The solution of this second sub-problem ultimately leads to the solution of the first one given that $\bar{z}_n^*$ depends indirectly on a_{n-1}. Again, one uses dynamic programming to determine the sequence a_i^* (for $i = 1, \cdots, n$ - 1), if it exists, that allows one to attain the lower bound of the following total storage and backlog costs.

$$J^* = \inf_{a_i \in A_i(\tilde{r}_{i+1}, \tilde{p}_{i+1})_{i=1,..,n-1}} \left\{ \sum_{j=1}^{m} \sum_{l=1}^{n-1} T_{jl}(\tilde{r}_l, \tilde{p}_l, a_l) + T_F^*(\tilde{r}_n, \tilde{p}_n) \right\} \tag{27}$$

with $T_{jl}(\tilde{r}_l, \tilde{p}_l, a_l)$ given by (22) and $T_F^*(\tilde{r}_n, \tilde{p}_n)$ given by (24) but substituting $\bar{z}_n$ with $\bar{z}_n^*$ of (26) according to the value of β.
$A_i(\tilde{r}_{i+1}, \tilde{p}_{i+1})$, for $i = 1, \cdots, n$ - 1, are the sets of admissible availability coefficients (a_i). Two main constraints determine an upper and lower bound of $A_i(\tilde{r}_{i+1}, \tilde{p}_{i+1})$. As a_i is an availability coefficient, so 0 ¡ a_i ¡ 1. In addition, for the demand d_j to be satisfiable by machine $\tilde{M}_i$, the following inequality must hold :

$$\frac{\tilde{r}_{i+1}}{(\tilde{r}_{i+1} + \tilde{p}_{i+1})} \bar{k}_{j,i+1} > d_j \tag{28}$$

with $\tilde{r}_{i+1}$ and $\tilde{p}_{i+1}$ are given respectively by (13) and (14) with index $i + 1$ replacing i.

6.2 Solution of the dynamic programming problem

In the following, we propose an algorithm for the solution of the dynamic programming problem (27). A discretization of the decision variable a_1 over its complete admissible range leads to the generation of the state space at stage 2 $[\tilde{r}_2\ ,\ \tilde{p}_2]^T$ based on (13) and (14) and using the fact that $\tilde{r}_1 = r_1$ and $\tilde{p}_1 = p_1$. In turn, the generated state space $[\tilde{r}_2\ ,\ \tilde{p}_2]^T$ with a discretization of the decision variable a_2 over its admissible range leads to the generation of the state space at stage 3 $[\tilde{r}_3\ ,\ \tilde{p}_3]^T$; and so on until the state discretized space at the last stage n is obtained. Once the state space has been generated, dynamic programming backwards costs could be calculated from $i = n$ to 1 to identify an optimal trajectory (see [3] for further insights on dynamic programming). The optimal hedging levels z_{ji}'s could then be calculated by substituting, in (18), a_i's with the optimal a_i's found by the algorithm. This method is actually very expensive in terms of operating time

and memory as it generates an exact discretized state space. Indeed, the state space grows exponentially as i increases from 1 to n.

In order to avoid the exponential growth of the state space, we define an *Abscissa reduced state space generation method* specifically tailored for the current problem. This method is based on the observation, from (13) and (14), that the two-dimensional state at stage $i+1$ depends only on $\tilde{r}_i$ and a_i, but *not directly* on $\tilde{p}_i$. A way of generating an approximate state space of *fixed* size say N^2 at every stage (from $i=3$ to n) is defined as follows; a_1 is discretized into N values over its admissible range. The N values of a_1 generate, given r_1, N two-dimensional states at stage 2 using (13) and (14). These N states at stage 2 with N discretized a_2 generate N^2 two-dimensional states at stage 3. The range of abscissas ($\tilde{r}_3$) is then divided into N equidistant intervals and only N states representing these N intervals are retained from the N^2 states generated at stage 3 (one state representing each interval). The N *representative* states are chosen according to abscissas closest possible to the centers of the N intervals. The same procedure is repeated until one has generated an N^2 size state space at every stage (stages 3 to n). Figures 4 and 5 illustrate the procedure of the *Abscissa reduced state space generation method* for $N=3$ where a_i^k represents the k^{th} discretized value of a_i. The

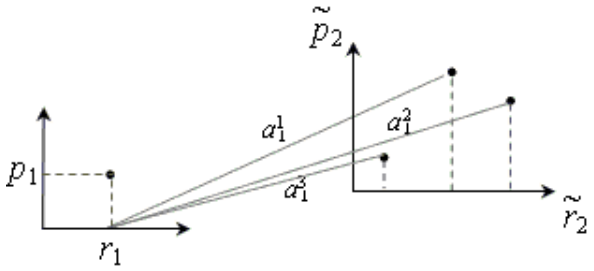

Figure 4: Generation of state space at stage 2 from r_1 and the discretized values of a_1, for $N=3$.

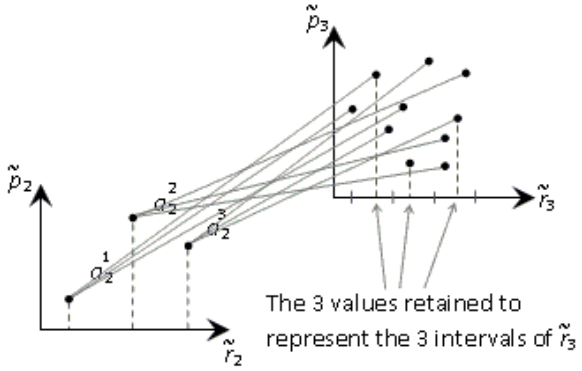

Figure 5: Generation of state space at stage 3 from $\tilde{r}_2$ and the discretized values of a_2, for $N=3$.

proposed computational algorithm relies on *Abscissa reduced state space generation method*. In phase 1 of the algorithm, one uses the *Abscissa reduced state space generation method* with a certain N^2 space size in order to quickly generate an initial rough discrete state space discretization and identify a corresponding initial suboptimal dynamic programming trajectory. In a second phase, one generates N^2 size discrete state space at each stage, but this time concentrated in the neighborhood of the initial suboptimal trajectory of phase 1 by selecting upper and lower bounds of a_i (for $i=1,\cdots,n$ - 1) around the optimal a_i of phase 1. Then, an enhanced dynamic programming solution is obtained according to the state space generated in phase 2. Note that in this phase a different value of N could be used. Phase 2 of the algorithm is repeated until one reaches the desired accuracy. This algorithm appears to be quite fast while maintaining very good accuracy, when compared to the exponentially growing state search. The latter suffers from both an explosion in computing requirements and a non uniform state space accuracy of state space discretization from one stage to another.

By contrast, the proposed algorithm combined with abscissa reduction at every stage, maintains the same accuracy of discretization at all stages, and computes a steadily improving sub-optimum.

7 Numerical results for the synchronized production mode

In the following, we apply the algorithm proposed in Section 6.2 in order to optimize kanban sizing decisions in a multi-part transfer line. The algorithm was applied to optimize a large collection of transfer lines; as an example we consider the following line: a 6-machine 4-part type nonhomogeneous transfer line with repair rates of [0.5 0.8 0.7 0.85 0.7 0.6]; failure rates of [0.4 0.4 0.45 0.285 0.28 0.5]; demand rates for the 4 parts of [1.5 , 1.25 , 1.75 , 1]; storage costs of $c_{ji} = 2$ (for $i = 1, \cdots, 5$ and $j = 1, \cdots, 4$) and $c^+_{j6} = 2$ (for $j = 1, \cdots, 4$); backlog costs of $c^-_{j6} = 10$ (for $j = 1, \cdots, 4$) and maximal production rates for $i = 1, \cdots, 6$ of $k_{1i} = 15.5$ - (0.01 * (i - 1)), $k_{2i} = 16$ - (0.01 * (i - 1)), $k_{3i} = 16.5$ - (0.01 * (i - 1)) and $k_{4i} = 15$ - (0.01 * (i - 1)). The resulting optimal availability coefficients are shown in Figure 6 with their lower and upper bounds. The optimal solution was obtained after 58 cycles and achieves an accuracy (size of the discretization step) of 10^{-8} with an initial state-space size (N^2) of 75^2. The overall running time is 5.7841 minutes using a *Pentium 4 CPU - 3.00GHz - 512 MB of RAM* computer. With the classic (exponentially growing state space) discretization algorithm, it takes around 6 days of calculation on the same computer to obtain an optimal solution with a precision of only 10^{-4}.

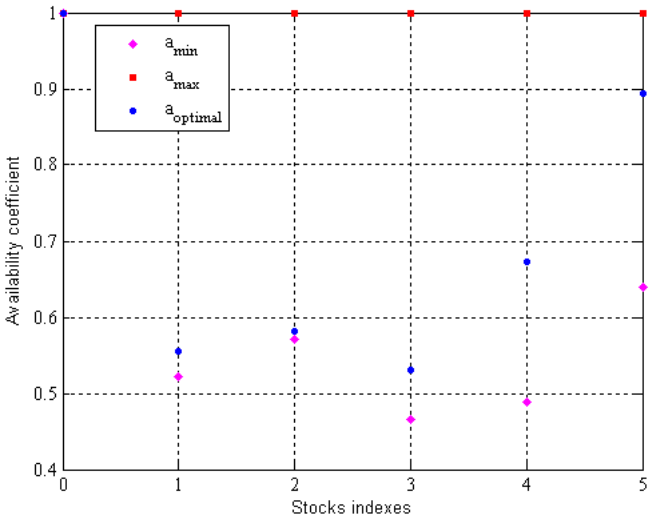

Figure 6: Optimal availability coefficients of buffer i for $i = 0, \cdots, 5$.

8 Prioritized production mode

The class of synchronized production strategies (Sections 4 to 7), although interesting both from a practical standpoint (cyclic character of the production) and because of its analytical and computational tractability, represents however only a very particular case of the more complex so-called *Prioritized production* mode, which for ease of exposition we shall develop for a 2-machine 2-part type transfer line. The two machines are M_1 and M_2, each followed by a buffer sharing two part types. x_{ji} represents level of part j in buffer i for $j = 1, 2$ and $i = 1, 2$. The prioritized production mode is a class of production strategies aiming at enforcing a certain priority ordering for each part type. This order imposes that the highest priority part always attains its maximum kanban level first, and then the next highest priority part type and so on. In addition, the same priority structure must hold for different part type wip levels in the sense that at any given production stage, the wip associated with any particular part type can never be zero unless wips of lower priority part types are all zero. In other words, the availability of the wip of any given part type is always greater than or equal

to that of a lower priority part type. The assumptions of Section 2.1 continue to hold here. Furthermore, let us assume that part type 1 has higher priority than part 2. The prioritized production mode with priority given to part type 1 is characterized by the following (design) parameters at machine M_i ($i = 1, 2$): (i) Two vectors of maximum Kanban levels $[z_{1i}\ ,\ z_{2i}]^T$. (ii) Two vectors of parts production rates, $[k'_{1i},\ k'_{2i}]^T$, which also satisfy the rates monotonicity assumption (second assumption in Section 2.1). Note that it is assumed that the chosen production vectors are maximal in the sense that the following holds: $(k'_{1i}/k_{1i}) + (k'_{2i}/k_{1i}) = 1$, $i = 1, 2$. Let $u_{ji}(t)$ be the instantaneous production of part j by machine M_i for $j, i = 1, 2$. When machine M_i can produce, it produces part 1 with a rate k'_{1i} and part 2 with a rate k'_{2i} until either x_{1i} attains its maximum level z_{1i} or machine M_i fails. Note that, if $(k'_{1i}/k'_{2i}) > (d_1/d_2)$, x_{1i} attains z_{1i} before x_{2i} attains z_{2i} (see further proof of Proposition 2). At that point, part 1 is produced with a rate $u_{1,i+1}(t)$ (with $u_{13}(t) = d_1$ and $u_{23}(t) = d_2$ for all $t > 0$ where d_1 and d_2 are the demand rates for the two part types) required to maintain x_{1i} at its maximal level z_{1i}. By the monotonicity assumption, this rate must be less than k'_{1i} and thus some production capacity is freed. It is then entirely dedicated to an increase in the production rate of part type 2 which is given by:

$$k^a_{2i} = (1 - \frac{u_{1,i+1}}{k_{1i}})k_{2i} \tag{29}$$

Sufficient conditions on the design parameters of prioritized production policies so as to achieve the required pathwise prioritized behavior are given in Proposition 2 for a general m-part n-machine transfer line.

Proposition 2 *The following set of inequalities will insure a part-type priority ordering with the highest priority associated with part-type index 1, and lowest priority associated with part-type index m:*

$$\sum_{j=1}^{m} \frac{k'_{ji}}{k_{ji}} = 1 \quad (for\ i = 1, \cdots, n)$$

$$\frac{k'_{j1}}{k'_{l1}} = \frac{z_{j1}}{z_{l1}} > \frac{k'_{j2}}{k'_{l2}} = \frac{z_{j2}}{z_{l2}} > \cdots > \frac{k'_{j,n-1}}{k'_{l,n-1}} = \frac{z_{j,n-1}}{z_{l,n-1}} > \frac{k'_{jn}}{k'_{ln}} > \frac{d_j}{d_l}$$

$$(for\ j,\ l\ (\neq j) = 1, \cdots, m) \tag{30}$$

Proof. See Appendix B □

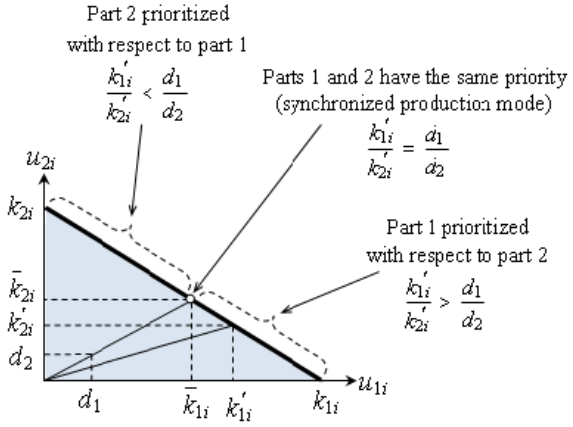

Figure 7: Space of maximum production rates for machine M_i in the case of two parts.

Let a_{11} (respectively a_{21}) be the availability coefficient of part 1 wip (respectively part 2 wip) at the end of the first production stage, Proposition 2 guarantees that $a_{11} > a_{21}$, provided that the design parameters of the prioritized production policy satisfy the inequalities stated in the Proposition.

9 Approximate performance analysis in prioritized mode

In this section, we shall approximate the transfer line in the part type 1 prioritized mode as a collection of isolated equivalent Markovian machines, each associated with work on a given part type at a given production stage. Estimates of various quantities based on the approximate model are subsequently validated against the results of Monte Carlo simulations.

9.1 Approximate mathematical Model

9.1.1 Machine M_1 producing part type 1

The first machine is preceded by an infinite stock. It evolves according to a Markov chain with failure rate p_1 and repair rate r_1. Following the decomposition technique proposed for the synchronized production mode, this machine is considered as subjected to a constant rate of demand for part type 1, d_1/a_{11} with a_{11} the unknown availability coefficient of wip x_{11}. It depends on the choice of Kanban parameter z_{11}. It is calculated through the implicit equation [12]:

$$a_{11} = 1 - \frac{p_1}{r_1 + p_1}\left[\frac{1-\rho_{11}}{1-\rho_{11}e^{-\mu_{11}(1-\rho_{11})z_{11}}}\right] \tag{31}$$

with :

$$\mu_{11} = \frac{p_1}{k'_{11} - \frac{d_1}{a_{11}}} \tag{32}$$

$$\rho_{11} = \frac{r_1}{\mu_{11}\frac{d_1}{a_{11}}} \tag{33}$$

Let c_{11} be part type 1 unit storage cost per unit time within the first production stage. This storage cost for part type 1 by the isolated machine with backlog not allowed is obtained via the following equation ([12]).

$$T_{11}(p_1, r_1, a_{11}) = \frac{c_{11}\rho_{11}}{(r_1+p_1)(1-\rho_{11}\gamma_{11})} \left[k'_{11}\frac{1-\gamma_{11}}{1-\rho_{11}} - \frac{\gamma_{11}}{\lambda_{11}}(r_1+p_1)\ln\gamma_{11}\right] \tag{34}$$

where

$$\lambda_{11} = -\mu_{11}(1-\rho_{11}) \tag{35}$$

$$\gamma_{11} = \frac{p_1\left[r_1 k'_{11} - d_1(r_1+p_1)\right]}{r_1(r_1+p_1)(1-a_{11})(k'_{11} - \frac{d_1}{a_{11}})} \tag{36}$$

9.1.2 Machine M_1 producing part type 2

The production rate of part type 2 by the first machine depends on the machine M_1 operating state α_1 and wip x_{11}. The proposed approximate Markovian model of machine M_1 producing part type 2 designated by $\tilde{M}_{21}$ is shown in Figure 8. The model has three states 0, 1 and 2 where the only failure state is 0. Let Π_0, Π_1 and Π_2 be the probabilities at steady state to be respectively in state 0, 1 and 2. Π_0 is the steady-state failure probability of machine M_1 ($\Pi_0 = p_1/(p_1+r_1)$) while $\Pi_1 + \Pi_2$ is the steady-state operational probability of machine M_1 ($\Pi_1 + \Pi_2 = r_1/(p_1+r_1)$). Π_2 is the probability that $x_{11} = z_{11}$ and α_1 =1. It is given as follows [12].

$$\Pi_2 = P_{z_{11}} = \frac{r_1}{r_1+p_1}\frac{(1-\rho_{11})e^{\lambda_{11}z_{11}}}{1-\rho_{11}e^{\lambda_{11}z_{11}}} \tag{37}$$

From the flow balance equations at steady state, one obtains:

$$r^a_{21} = \frac{p_1 P_{z_{11}}}{\frac{r_1}{r_1+p_1} - P_{z_{11}}} \tag{38}$$

The stationary probabilities of this isolated machine as well as its storage cost for part 2 are calculated via the solution of the corresponding Kolmogorov equations ([15] and [8]). This cost depends on p_1, r_1, a_{11} and a_{21}.

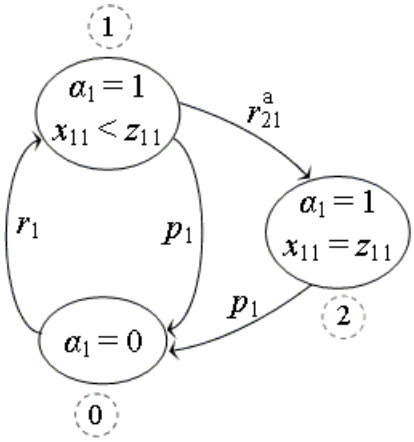

Figure 8: Equivalent machine $\tilde{M}_{21}$. It approximates machine M_1 producing part type 2.

9.1.3 Machine M_2 producing part type 1

This machine designated as $\tilde{M}_{12}$ is modeled using exactly the same approximations as for the last machine downstream in the synchronized mode decomposition method (Equations (13), (14), with $i = n$). Note that state $\tilde{\alpha}_{12} = 1$ corresponds to machine M_2 effectively able to produce part type 1 (i.e. it is operational and wip x_{11} is available), while $\tilde{\alpha}_{12} = 0$ corresponds to either x_{11} unavailable or machine M_2 not operational. Furthermore, The Bielecki-Kumar optimization result [4] still applies, and thus, the optimal Kanban parameter z_{12} is given by:

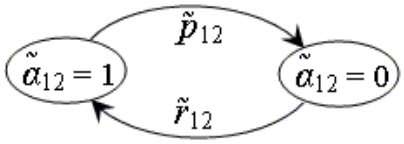

Figure 9: Equivalent machine $\tilde{M}_{12}$ which is an isolated Markovian machine version of M_2 producing part type 1.

$$\bar{z}_{12}^* = \begin{cases} 0 & \text{if } \beta_{12} \geq 1 \\ \frac{1}{\sigma_{12}} ln(\frac{1}{\beta_{12}}) & \text{otherwise} \end{cases} \tag{39}$$

with

$$\sigma_{12} = \frac{\tilde{r}_{12}(k_{12}^{'} - d_1) - \tilde{p}_{12} d_1}{d_1(k_{12}^{'} - d_1)} \tag{40}$$

$$\beta_{12} = \frac{c_{12}^{+}(\tilde{r}_{12} + \tilde{p}_{12})(k_{12}^{'} - d_1)}{(c_{12}^{+} + c_{12}^{-})\tilde{p}_{12} k_{12}^{'}} \tag{41}$$

where c_{12}^{+} (respectively c_{12}^{-}) is the finished parts storage (respectively backlog) cost per unit of time and product for part type 1 within the second stock. The optimal cost of storage and backlog of part type 1 for this machine is given by:

$$\bar{T}_{12}^*(\tilde{p}_{12}, \tilde{r}_{12}, a_{11}) = \begin{cases} \frac{c_{12}^{-} \tilde{p}_{12} k_{12}^{'}}{\sigma_{12}(\tilde{r}_{12} + \tilde{p}_{12})(k_{12}^{'} - d_1)} & \text{if } \beta_{12} \geq 1 \\ \frac{c_{12}^{+} d_1}{\tilde{r}_{12} + \tilde{p}_{12}} + \frac{c_{12}^{+}}{\sigma_{12}} ln \frac{1}{\beta_{12}} & \text{otherwise} \end{cases} \tag{42}$$

9.1.4 Machine M_2 producing part type 2

Production rate of part type 2 by the second machine depends on the state of M_2 and the levels of buffers x_{21} and x_{12}. We propose a two step procedure to approximately model the dynamic behavior of machine

M_2 producing part type 2. Figure 10 is a first approximate Markovian representation of machine M_2 feeding from wip x_{21}. It is denoted $\bar{M}_{22}$ using equations similar to those of $\tilde{M}_{21}$. More specifically:

$$\bar{r}_{22} = \frac{r_2(1-a_{21})}{r_2(1-a_{21})+p_2a_{21}}r_1 + \frac{p_2a_{21}}{r_2(1-a_{21})+p_2a_{21}}r_2 \tag{43}$$

$$\bar{p}_{22} = \Big(\frac{r_2+p_2}{a_{21}r_2} - 1\Big)\bar{r}_{22} \tag{44}$$

Its states are $\bar{\alpha}_{22} = 1$ which corresponds to "$\alpha_2 = 1$ and $x_{21} > 0$" while $\bar{\alpha}_{22} = 0$ corresponds to either

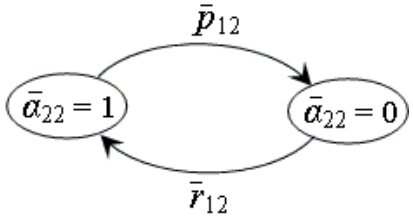

Figure 10: The equivalent machine $\bar{M}_{22}$.

"$\alpha_2 = 1$ and $x_{21} = 0$" or "$\alpha_2 = 0$ and $x_{21} > 0$". Machine $\bar{M}_{22}$ captures the effective operational state of machine M_2 when producing part type 2. Indeed, M_2 can produce type 2 parts only if both M_2 itself is operational and wip M_{21} is available. However, $\bar{M}_{22}$ as such does not capture the influence of stock x_{12} on the production rate of M_2, this is why we require an additional modeling step. In Figure 11, state $\bar{\alpha}_{22}$ of $\bar{M}_{22}$ is disaggregated into two states: in state 1, M_2 will produce at maximum rate k'_{22}, while in state 2 machine M_2 can redirect the extra capacity freed whenever x_{12} reaches its maximum level z_{12}, towards production of part type 2. This results in the augmented production rate k^a_{22}, and the resulting Markovian machine is designated as $\tilde{M}_{22}$. In Figure 11, we set $\tilde{r}_{22} = \bar{r}_{22}$, and $\tilde{p}_{22} = \bar{p}_{22}$.

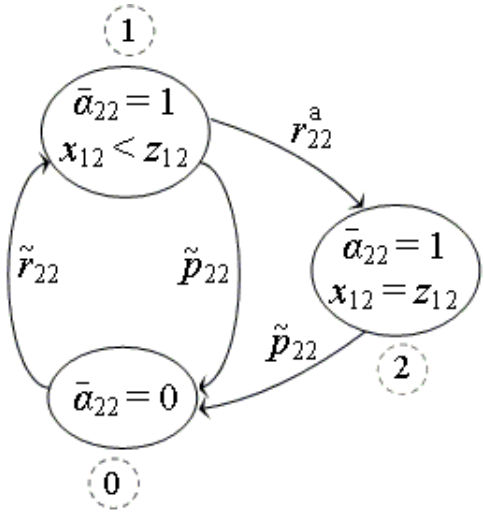

Figure 11: Equivalent machine $\tilde{M}_{22}$ which is an isolated Markovian machine version of M_2 producing part type 2.

Let Π_0, Π_1 and Π_2 be the steady state probabilities to be in states 0, 1 and 2 of Figure 11. $\Pi_1 + \Pi_2$ represents the probability that $\bar{\alpha}_{22} = 1$, i.e. $\alpha_2 = 1$ and $x_{21} > 0$, and thus $\Pi_1 + \Pi_2 = a_{21}r_2/(r_2 + p_2) = \tilde{r}_{22}/(\tilde{r}_{22}+\tilde{p}_{22})$. Let $P_{z_{12}}$ be the probability that $\tilde{\alpha}_{12} = 1$ ($\alpha_2 = 1$ and $x_{11} > 0$) and $x_{12} = z_{12}$. This probability can be calculated as follows [4]:

$$P_{z_{12}} = \frac{\sigma_{12}d_1}{\tilde{r}_{12}+\tilde{p}_{12}} \tag{45}$$

State 2 represents $\bar{\alpha}_{22} = 1$ ($\alpha_2 = 1$ and $x_{21} > 0$), $x_{11} > 0$ and $x_{12} = z_{12}$. Thus, the stationary probability to be in state 2 (Π_2) is given by :

$$\begin{aligned}\Pi_2 &= P[x_{21} > 0, \alpha_2 = 1, x_{11} > 0, x_{12} = z_{12}] \\ &= P[x_{21} > 0 | \alpha_2 = 1, x_{11} > 0, x_{12} = z_{12}] P_{z_{12}}\end{aligned} \tag{46}$$

According to MDA (Section 3), the binary availability process for x_{21}, and state α_2 are independent. We further assume also that events $x_{21} > 0$ and $x_{12} = z_{12}$ are independent. In addition, because part type 1 has priority over part type 2, if x_{21} is positive, x_{11} will also be positive. So, we obtain the following equation:

$$\begin{aligned}P[x_{21} > 0 | \alpha_2 = 1, x_{11} > 0, x_{12} = z_{12}] &= P[x_{21} > 0 | x_{11} > 0] \\ &= \frac{P[x_{21} > 0, x_{11} > 0]}{P[x_{11} > 0]} \\ &= \frac{a_{21}}{a_{11}}\end{aligned} \tag{47}$$

Finally, the steady state probability flow balance equations in state 2 yield $r^a_{22}\Pi_1 = \tilde{p}_{22}\Pi_2$, thus leading to the following value of r^a_{22}.

$$r^a_{22} = \frac{\tilde{p}_{22}\frac{P_{z_{12}}}{a_{11}}}{\frac{r_2}{r_2+p_2} - \frac{P_{z_{12}}}{a_{11}}} \tag{48}$$

The stationary probabilities of this isolated machine as well as the storage and backlog costs for part type 2 are calculated by the solution of the corresponding Kolmogorov equations ([21]). This cost depends on $\tilde{p}_{22}$, $\tilde{r}_{22}$, a_{11} and a_{21}.

9.2 Monte Carlo validation of the approximate performance models

In this section, we verify the performance of our approximate mathematical model by comparison with the results of Monte-Carlo simulations for given choices of prioritized production mode design parameters. Tables 7 and 8 show respectively data for two distinct transfer lines, both with $d_1 = 1.25$ and $d_2 = 1.5$. For the first line: $p_1 = 0.2$, $p_2 = 0.25$, $r_1 = 0.45$ and $r_2 = 0.75$; for the second line: $p_1 = 0.22$, $p_2 = 0.25$, $r_1 = 0.42$ and $r_2 = 0.6$.

Table 7: Data for the first transfer line.

c_{j1}	c^+_{j2}	c^-_{j2}
2	2	10
2	2	10

k_{ji}		k'_{ji}		z_{ji}	
6.686	9.25	3.65	2.75	2	7.5
9.36	4.625	4.25	3.25	2.35	8.9

Table 8: Data for the second transfer line.

c_{j1}	c^+_{j2}	c^-_{j2}
1	2	4
1	2	4

k_{ji}		k'_{ji}		z_{ji}	
9.45	10.417	4.8	3.5	1.9	4.77
11.34	6.25	5.58	4.15	2.25	5.7

Let T_{tot} be the total transfer line storage and backlog costs for part types 1 and 2. Tables 9 and 10 summarize the results of comparisons of theoretical estimates based on the approximate model and Monte Carlo based estimates for availability coefficients and total costs for the two transfer lines.

Table 9: Comparison of theory based and simulation based results, Line 1.

	M-C simulation	Model based estimate	% Relative error
a_{11}	0.8288	0.8221	0.812
a_{21}	0.8263	0.8209	0.656
T_{tot}	37.64	38.099	1.221

Table 10: Comparison of theory based and simulation based results, Line 2.

	M-C simulation	Model based estimate	% Relative error
a_{11}	0.80573	0.78621	2.42
a_{21}	0.80387	0.78542	2.29
T_{tot}	22.33	23.235	3.9

The worst case error is about 4%. Over 50 different transfer lines were studied with transfer line as well as policy design parameters satisfying the assumptions stated in the paper. The results are consistent with the ones reported here.

10 Dynamic programming based optimization

When carrying out the following optimization, we consider the rate design parameters for the prioritized production policy as given (of course, ultimately, they themselves would have to be optimized). Let us rewrite (30) as follows.

$$\begin{aligned} \frac{k'_{ji}}{k'_{li}} &= \frac{\gamma_j}{\gamma_l}\frac{d_j}{d_l} \quad \text{for } i = 1, \cdots, n \text{ and } j, l = 1, \cdots, m \\ \frac{z_{ji}}{z_{li}} &= \frac{\gamma_j}{\gamma_l}\frac{d_j}{d_l} \quad \text{for } i = 1, \cdots, n\text{ - }1 \text{ and } j, l = 1, \cdots, m \end{aligned} \tag{49}$$

with $1 = \gamma_1 > \gamma_2 > \cdots > \gamma_m > 0$. Thus, the variables to be optimized are a_{1i} for $i = 1, \cdots, n$ - 1, γ_j for $j = 2, \cdots, m$, and z_{jn} for $j = 1, \cdots, m$. Note that $\gamma_1 = 1$, and the availability coefficients $a_{2i}, \cdots, a_{mi}$ (for $i = 1, \cdots, n$ - 1) do not have any degree of freedom, they are related to a_{1i} by virtue of (49). The optimization problem is divided into two sub-problems of different natures; the first one for $i = 1, \cdots, n$ - 1, and the second one for the last stage ($i = n$). For the second sub-problem, that of the last stage, the optimal value of z_{1n} is obtained directly from the result of Bielecki-Kumar [4] as a function of $a_{1,n-1}$; while no analytical expression is available for the optimal value of z_{jn} ($j = 2, \cdots, m$) for the approximate Markovian machine $\tilde{M}_{jn}$ has more than two states. Nevertheless, the optimal values of z_{jn}, $j = 2, \cdots, m$, could be calculated numerically from the optimization of the sum of the individual costs for the equivalent machine $\tilde{M}_{jn}$ ($j = 2, \cdots, m$), each obtained from the corresponding Kolmogorov equations ([15] and [8]). In other words, the optimal values of z_{jn}, $j = 2, \cdots, m$, are obtained from the minimization of $\sum_{j=2}^{m} T_{jn}(\tilde{r}_{jn}, \tilde{p}_{jn}, z_{jn})$, where $T_{jn}(\tilde{r}_{jn}, \tilde{p}_{jn}, z_{jn})$ is the equivalent machine $\tilde{M}_{jn}$ total storage and backlog cost calculated numerically from the corresponding Kolmogorov equations, and $\tilde{r}_{jn}$ (respectively $\tilde{p}_{jn}$) is calculated in a similar way to (43) (respectively (44)). This leads to an optimal cost depending on $a_{1,n-1}$ (which by its turn depends on the previous availability coefficients). Consequently, the optimal cost of the second sub-problem has to be taken into consideration into the global cost dynamic programming based optimization problem, which is thus formulated as follows: To determine the values of the availability coefficients a_{1i} ($i = 1, \cdots, n$ - 1), γ_j ($j = 2, \cdots, m$), and z_{jn} ($j = 1, \cdots, m$), if they exist, so as to minimize the total storage and backlog costs in the transfer line, J, defined as:

$$J = \sum_{j=1}^{m}\sum_{i=1}^{n-1} T_{ji}(\tilde{r}_{ji}, \tilde{p}_{ji}, a_{1i}, \gamma_j) + T_F^*(a_{1,n-1}) \tag{50}$$

where $T_{ji}(\tilde{r}_{ji}, \tilde{p}_{ji}, a_{1i}, \gamma_j)$ is calculated as presented in Section 9 with $\tilde{r}_{11} = r_1$ and $\tilde{p}_{11} = p_1$, $T^*_F(a_{1,n-1})$ is the total optimal storage and backlog cost of the last equivalent machine defined as $T^*_{1n}(a_{1,n-1})$ obtained from Bielecki-Kumar [4] plus $\sum_{j=2}^{m} T_{jn}(\tilde{r}_{jn}, \tilde{p}_{jn}, z^*_{jn}(a_{1,n-1}))$ calculated numerically. The admissible set of the coefficient a_{1i} has a lower bound of $(d_1/\ k'_{1,i+1})((r_{i+1}+p_{i+1})/r_{i+1})$ given from the ergodicity condition (the demand feasibility condition) and a higher bound of 1; both bounds are not admissible (open admissibility set). γ_j satisfies $1 = \gamma_1 > \gamma_2 > \cdots > \gamma_m > 0$. For stages 2 to n - 1 of the dynamic programming, the availability coefficient a_{ji} corresponding to z_{ji} is calculated from the iterative algorithm of Appendix C, and z_{ji} is related to z_{1i} (which is a function of a_{1i}, see (31)) by (49). Also, we impose $k'_{ji} > k^a_{j,i+1}$, for $j = 2, \cdots, m$ and $i = 1, \cdots, n$ - 1, to guarantee that wip x_{ji} increases when machine $\tilde{M}_i$ produces part type j whatever the rate at which this part type is drawn by machine $\tilde{M}_{i+1}$.

11 Numerical results

We consider a 2-machine 2-part transfer line and provide the optimization results for both synchronized and prioritized modes to be able to compare between them. As a sample of our numerous experiments, we consider the following transfer line. $c_{11} = c_{21} = c^+_{12} = c^+_{22} = 2$, $c^-_{12} = c^-_{22} = 10$, $d_1 = 1.25$, $d_2 = 1.5$, $p_1 = 0.2$, $p_2 = 0.3$, $r_1 = 0.5$, $r_2 = 0.7$, $k_{11} = 8.20857$, $k_{21} = 11.492$, $k_{12} = 10.94$ and $k_{22} = 5.75789$. For the prioritized production, the solution of this dynamic programming problem is done according to Section 10 where the variables to optimize are a_{11}, γ_2, z_{12} and z_{22}; while the synchronized production is optimized according to Section 6.

Table 11 summarizes the results obtained using the dynamic programming for both cases. T^*_{tot} is the minimal storage and backlog cost for the whole line. Note that for the synchronized production mode $\gamma_j = 1$ for all $j = 1, \cdots, m$. The two optimal solutions appear to give very close results although the details of the solutions are completely different. This situation is expected to occur given that the prioritized production mode, although distinct and much harder to optimize, includes the synchronized production mode as a special case. Table 11 shows also that the total optimal cost of the prioritized production mode is more optimal (less) than that of the synchronized production mode. Numerous comparisons appear to confirm this conclusion. It is also consistent with the result of [9] obtained for single-machine multi-part systems. On the other hand, the extreme cases where the ratio k'_{1i}/k'_{2i} gets closer to one of both extremities, $(k_{1i},0)$ or $(\bar{k}_{1i},\bar{k}_{2i})$ (see Figure 7), the prioritized strategy is less advantageous than the synchronized strategy as it produces higher total optimal costs.

Table 11: Comparing optimal solutions for the synchronized and the part type 1 prioritized modes.

	synchronized production	prioritized production
γ^*_2	1	0.9309
a^*_{11}	0.85368	0.84897
a^*_{21}	0.85368	0.84201
z^*_{12}	4.40562	4.17497
z^*_{22}	5.28674	5.82972
T^*_{tot}	31.42143	30.39928

12 Conclusion

We have proposed two consistent Kanban based production modes for multi-part unreliable transfer lines, and their corresponding approximate performance evaluation schemes. The production modes are consistent in that under an idealized transfer line model with negligible set up times and fluid part production, they provably do exactly what they were set up to do. More specifically:
(i) For the synchronized production mode: it is shown that a solidarity property holds for different part type wips at every production stage in that they reach their maxima or zero simultaneously; this permits a representation of the multi part transfer line as a single part type transfer line with modified costs.

(ii) For the prioritized production mode: sufficient conditions on production parameters are given guaranteeing that an order relation is preserved at all times and at every production stage among different part type wip sizes, consistent with the announced priority assignment.
Computationally efficient approximations schemes are proposed for both classes of production modes. These approximation schemes lead in a natural manner to dynamic programming based optimization schemes for Kanban levels; this is because of their underlying causality structure with unidirectional propagation of the effects of sizing decisions (upstream to downstream in this case), thus permitting the definition of sequential decision stages. While prioritized production strategies are more general than synchronized ones, their implementation and optimization remain harder. In this respect, it is our feeling that synchronized production strategies can still provide a viable, easily implementable (periodic sharing of machines productive time between distinct part types at every stage) and computationally tractable class of production strategies, in so far as optimization is concerned.

13 Appendices

A Synchronized production mode (Proposition 1)

Consider synchronized production policies. They are governed by the following equations, for $i = 1, \cdots, n$:

$$\begin{aligned} \frac{\bar{k}_{1i}(\vec{d})}{d_1} = \frac{\bar{k}_{2i}(\vec{d})}{d_2} = \cdots = \frac{\bar{k}_{mi}(\vec{d})}{d_m} \\ \sum_{j=1}^{m} \frac{\bar{k}_{ji}(\vec{d})}{k_{ji}} = 1 \\ \frac{z_{1i}}{d_1} = \frac{z_{2i}}{d_2} = \cdots = \frac{z_{mi}}{d_m} \end{aligned} \tag{51}$$

with $\vec{d} \equiv [d_1, d_2, \cdots, d_m]^T$. Introducing a part-type dependent normalization of wip/ inventory/ backlog variables, for $j = 1, \cdots, m$ and for $i = 1, \cdots, n$:

$$\begin{aligned} \tilde{x}_{ji} &= \frac{x_{ji}}{d_j} \qquad \tilde{k}_{ji} = \frac{\bar{k}_{ji}}{d_j} \\ \tilde{z}_{ji} &= \frac{z_{ji}}{d_j} \qquad \tilde{u}_{ji} = \frac{u_{ji}}{d_j} \end{aligned} \tag{52}$$

We notice, in view of (52) that for $i = 1, \cdots, n$:

$$\begin{aligned} \tilde{k}_{ji} &= \tilde{k}_i \\ \tilde{z}_{ji} &= \tilde{z}_i \quad \forall j = 1, \cdots, m \end{aligned} \tag{53}$$

With this normalized system of variables, the dynamics of the parts of type j evolve according to:

$$\begin{aligned} \dot{\tilde{x}}_{ji} &= \tilde{u}_{ji} - \tilde{u}_{j,i+1} \\ \dot{\tilde{x}}_{jn} &= \tilde{u}_{jn} - 1 \end{aligned} \tag{54}$$

In (54), machine operating states, machine maximum production rates and maximum Kanban levels are i dependent but independent of j. (54) clearly indicates that for identical initial conditions, the normalized sample paths will be the same for any j part-type trajectories, thus establishing the pathwise synchrony of all part-type trajectories. Furthermore, it is not difficult to see that if the constraints on z_{jn} ($j = 1, \cdots, m$) are removed in (54), one could still establish synchrony of the normalized trajectories for all part types provided one imposes that they all start for example with Kanban maximal levels initially attained.

B Prioritized production mode (Proposition 2)

The above synchronized production policies can be turned into prioritized control policies by creating a "distortion" in the demand vector. To fix ideas assume, without loss of generality, that the objective is to

produce a priority ordering consistent with the part-type labeling, i.e. the highest priority part type is 1 and the lowest priority part type is m. Denote $\vec{d}^{dis} \equiv [\gamma_1 d_1, \gamma_2 d_2, \cdots, \gamma_m d_m]^T$ with:

$$1 = \gamma_1 > \gamma_2 > \cdots > \gamma_{i-1} > \gamma_i > \gamma_{i+1} > \cdots > \gamma_m > 0 \tag{55}$$

Consider the synchronized policy production rates, and the corresponding Kanban maximum levels $\bar{k}_j i(\vec{d}^{dis})$, z_{ji}, $j = 1, \cdots, m$. They satisfy the following ((51) like) constraints:

$$\begin{aligned} \frac{\bar{k}_{1i}(\vec{d}^{dis})}{\gamma_1 d_1} = \frac{\bar{k}_{2i}(\vec{d}^{dis})}{\gamma_2 d_2} = \cdots = \frac{\bar{k}_{mi}(\vec{d}^{dis})}{\gamma_m d_m} \quad \text{for } i = 1, \cdots, n \\ \sum_{j=1}^{m} \frac{\bar{k}_{ji}(\vec{d}^{dis})}{k_{ji}} = 1 \quad \text{for } i = 1, \cdots, n \\ \frac{z_{1i}}{\gamma_1 d_1} = \frac{z_{2i}}{\gamma_2 d_2} = \cdots = \frac{z_{mi}}{\gamma_m d_m} \quad \text{for } i = 1, \cdots, n\text{ - }1 \end{aligned} \tag{56}$$

While the production rates in (56) have been obtained on the basis of a "distorted" demand vector, the real demand vector remains $\vec{d}$. Under these production rates, the real demand vector, and using the entries to normalize the wip and production rate variables, the counterpart of (54) becomes in this case:

$$\dot{\tilde{x}}_{ji} = \tilde{u}_{ji} - \tilde{u}_{j,i+1} \tag{57}$$

$$\dot{\tilde{x}}_{jn} = \tilde{u}_{jn} - \frac{1}{\gamma_j} \tag{58}$$

With machine operating states, normalized maximum production rates (equal to 1), and normalized maximum Kanban levels i dependent but independent of j. These normalized sample paths have identical underlying dynamics except for the $\frac{1}{\gamma_j}$ in (58) which corresponds to a different constant demand j dependent normalized variable. However, the trajectories generated by (57), (58) decrease monotonically with $\frac{1}{\gamma_j}$. In view of inequalities (55), part-type 1 trajectories will be pathwise dominant, while in general part-type j will dominate part-type $j+1$ pathwise for $j = 1, \cdots, m$ - 1. Thus, the defining properties of prioritized policies are satisfied. As a result, part-type 1 will attain its maximum Kanban level before any other part, thus freeing up some production capacity for the next highest priority part-type. The situation will remain so as long as the wip/inventory of part-type 1 sits at its maximum level (thus securing its relatively dominant position all the time). Likewise for an arbitrary j not equal to m. It will dominate pathwise all part types of lower index, particularly with the extra help that it may get from the extra capacity freed by the part of immediately higher priority, and its dominance will remain secured even when it, itself, frees extra capacity for the next lower priority part type. At this point, if we ignore the (arbitrary) γ_j variables, (56) becomes equivalent to the following system of inequalities characterizing the parameters of potential prioritized policies:

$$\begin{aligned} \sum_{j=1}^{m} \frac{k'_{ji}}{k_{ji}} = 1 \quad (\text{for } i = 1, \cdots, n) \\ \frac{k'_{j1}}{k'_{l1}} = \frac{z_{j1}}{z_{l1}} > \frac{k'_{j2}}{k'_{l2}} = \frac{z_{j2}}{z_{l2}} > \cdots > \frac{k'_{j,n-1}}{k'_{l,n-1}} = \frac{z_{j,n-1}}{z_{l,n-1}} > \frac{k'_{jn}}{k'_{ln}} > \frac{d_j}{d_l} \\ (\text{for } j,\ l\ (\neq j) = 1, \cdots, m) \end{aligned} \tag{59}$$

C Iterative algorithm for the calculation of a_{ji} from z_{ji}

In the following, we give the main steps of the iterative algorithm for the calculation of a_{ji} from a certain value of z_{ji}.

1. Set a value for the required precision (ϵ).
2. At iteration 0, $a^0_{ji} = 1$.
3. At iteration $k > 0$, solve the system's Kolmogorov equations ([15] and [8]) as function of z_{ji} to find the stationary probability that $x_{21} = 0$ (P_0).

4. $a_{ji}^{(k)}$ = 1 - $P_0^{(k)}$ and $precision^{(k)}$ = absolute value of $a_{ji}^{(k)}$ - $a_{ji}^{(k-1)}$.
5. While $precision^{(k)} > \epsilon$:
   $i = i + 1$.
   Return to step 3.

References

[1] Akella R., and Kumar P. R., Feb. 1986, "Optimal control of production rate in failure prone manufacturing system", *IEEE Trans. Automat. Contr.*, vol. AC-31, no. 2, pp. 116–126.

[2] Baynat B., and Dallery Y., 1996, "A product-form approximation method for general closed queuing networks with several classes of customers", *Performance Evaluation*, vol. 24, pp. 165-188.

[3] Bertsekas D., *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, Massachusetts, Vol. I, 3^{rd} edition 2005.

[4] Bielecki T., and Kumar P. R., July-Aug. 1988 "Optimality of zero-inventory policies for unreliable manufacturing systems", *Operations research*, vol. 36, no. 4, pp. 532–541.

[5] Caramanis M. C., and Sharifnia A., 1991, "Near-optimal manufacturing flow controller design", *International Journal of Flexible Manufacturing Systems*, vol. 3(4), pp. 321–336.

[6] Chiang S.Y., Kuo C. T., and Meerkov S. M., 2000, "DT-Bottlenecks in Serial Production Lines: Theory and Application." *IEEE Transactions on Robotics and Automation*, 16(5), pp. 567-580.

[7] Colledani M., Gandola F., Matta A., and Tolio T., 2008, "Performance evaluation of linear and non-linear multi-product multi-stage lines with unreliable machines and finite homogeneous buffers", *IIE Transactions*, vol. 40, pp. 612–626.

[8] El-Férik S., and Malhamé R. P., 1997, "Padé approximants for transient optimization of hedging control systems in manufacturing", *IEEE Tranactions on Automatic Control*, vol. 42, no. 4, pp. 440–457.

[9] El-Férik S., and Malhamé R. P. and Boukas E.-K., 1998, "A tractable class of maximal hedging policies in multi-part manufacturing systems", *Discrete Event Dynamic Systems: Theory and applications*, vol. 8, pp. 299–331.

[10] Gershwin S. B., Mar-apr 1987, "An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking". *Operations Research Society of America*, vol. 35, no. 2, pp. 291–305.

[11] Gershwin S. B., and Schor J. E., "Efficient algorithms for buffer space allocation", *Ann. Oper. Res.*, vol. 93, 2000, pp. 117–144.

[12] Hu J. Q., Feb. 1995, "Production control for failure-prone production systems with no backlog permitted", *IEEE Trans. Automat. Contr.*, vol. 40, pp. 299–305.

[13] Kimemia J. G., and Gershwin S. B., Dec. 1983, "An algorithm for the computer control of production in flexible manufacturing systems", *IIE Transactions*, vol. AC-15, pp. 353–362.

[14] Malhamé R. P., Feb 1993, "Ergodicity of hedging control policies in single-part multiple-state manufacturing systems", *IEEE Tranactions on Automatic Control*, vol. 38, no. 2, pp. 340–343.

[15] Malhamé R. P., and Boukas E.-K., May 1991, "A renewal theoretic analysis of a class of manufacturing systems", *IEEE Tranactions on Automatic Control*, vol. 36, no. 5, pp. 580–587.

[16] Malhamé R. P., and Boukas E.-K., May 1991, "Optimization of a class of decentralized hedging production policies in an unreliable two-machine flow shop", *38th IEEE Conf. Decision Control, Tampa, FL*, pp. 2282–2287.

[17] Perkins J. R., and Srikant R., March 1997, "Scheduling multiple part-types in an unreliable single-machine manufacturing system", *IEEE transactions on Automatic Control*, vol. 42, no. 3, pp. 364–377.

[18] Sadr J., and Malhamé R. P., Jan. 2004, "Decomposition/Aggregation-based dynamic programming optimization of partially homogeneous unreliable transfer lines", *IEEE Transactions on Automatic Control*, vol. 49, pp. 68–81.

[19] Sadr J., and Malhamé R. P., 2004, "Unreliable transfer lines: Decomposition/Aggregation and optimization", *Annals of Operations Research*, vol. 125, pp. 167–190.

[20] Sethi S. P., Suo W., Taksar M. I., and Yan H., Mar 1998, "Optimal Production Planning in a Multi-Product StochasticManufacturing System with Long-Run Average Cost", *Discrete Event Dynamic Systems*, vol. 8, no. 1, pp. 37–54.

[21] Sharifnia A., 1988, "Production control of a manufacturing system with multiple machine States", *IEEE Transactions on Automatic Control*, vol. 33, pp. 620–625.

[22] Srivatsan N., and Dallery Y., Jan-Feb 1998, "Partial Characterization of Optimal Hedging Point Policies in Unreliable Two-Part-Type Manufacturing Systems", *Operations Research*, vol. 46, no. 1, pp. 36–45.

[23] Veatch M. H., and Caramanis M. C., May 1999, "Optimal Manufacturing Flow Controllers: Zero-Inventory Policies and Control Switching Sets", *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 914–921.

[24] Youssef S., and Malhamé R. P., June 2008, "An improved dynamic programming algorithm for nonhomogeneous transfer line Kanban optimization", *16th Mediterranean Conference on Control and Automation, Ajaccio, France*, pp. 204–209.