Lead-time-oriented production control policies in two-machine production lines

Published version:

DOI:10.1080/24725854.2017.1417654

(Article begins on next page)

To appear in the *International Journal of Production Research*
Vol. 00, No. 00, 00 Month 20XX, 1–18

Lead-Time Oriented Production Control Policies in Two-Machine Production Lines

Alessio Angius[a*] and Marcello Colledani[a] and András Horváth[b]

[a]*Politecnico di Milano,Department of Mechanics, Via la Masa 1, Milan, Italy;*
[b]*Universitá degli studi di Torino,Department of Computer Science,Turin,Italy*

(*v5.0 released June 2015*)

The ability of meeting the target production lead times is of fundamental importance in modern manufacturing systems producing perishable products, where the product quality or value deteriorates with the time parts spend in the system, and in manufacturing contexts where strict lead time constraints are imposed due to tight shipping schedules. In these settings, traditional manufacturing system engineering methods and token-based production control policies loose effectiveness as they aim at achieving target production rates while minimizing the inventory, without directly taking into account the effect on the lead time distribution. In this paper a production control policy for unreliable manufacturing systems that aims at maximizing the throughput of parts that respect a given lead time constraint is proposed for the first time. The proposed policy jointly considers the actual level of the buffer and the state of the second machine in the system and stops the part loading at the first machine if there is unacceptable risk of exceeding the lead-time constraint. The effectiveness of this new policy against the traditional kanban policy is quantified by numerical analysis. The results show that this new policy outperforms the kanban policy by providing a tighter control on the production lead time. This approach paves the way to the introduction of new lead-time oriented production control policies to maximize the effective throughput in real manufacturing systems.

Keywords: manufacturing systems, lead time, perishable products, production control policy, multiple failure mode

1. Introduction and Objectives

Product quality and delivery reliability are key factors for success in modern manufacturing industries. To achieve desired target performance in both indicators, advanced production planning models have been widely analyzed in the past. Manufacturing System Engineering methods have been developed in the last decades for investigating the dynamic behaviour of manufacturing systems, for estimating their performance and for supporting their efficient design, operation and improvement. Traditional system design approaches aim at maximizing the production rate of the system, i.e. the average number of parts produced in a given time. However, by only looking at the production rate maximization, undesired side effects can be generated on the system lead time, also referred as system flow time, waiting time or residence time. In particular, since increasing the buffers increases the average production rate of the system but also increases the average lead time, a serious risk of undermining delivery reliability by excessive lead times is encountered.

Limiting the lead time is of particular interest in production systems where strict lead time constraints are imposed due to tight finished product distribution and shipping schedules (Biller, Meerkov, and Yan 2013) imposed by the customer and in systems producing perishable products. Perishable, or obsolete, or deteriorating products are allowed to spend a limited amount of time

*Corresponding author. Email: alessio.angius@polimi.it

inside the manufacturing system. If the system lead time exceeds certain fixed thresholds, the product has to be considered as a defect and has to be scrapped by the system. For example, the food production, and specifically the production of yogurt, is pervaded by strict requirements on hygiene and delivery precision. The production plants have to quickly respond to the market demands and current order situation. A typical production sequence for yoghurt includes mixing/standardizing of milk, pasteurization, fermentation, cooling, addition of fruit additives and packaging. The production planning involves very complex problems due to the maximum allowed storage time before packaging. If the time the product flows in the system exceeds this limit, it has to be scrapped. Another example comes from the production of multi-layer, multi-lumen polymeric micro-tubes for medical vascular catheters. During the phase of preparation of the raw material, in forms of granules, the moisture level is measured and if the value exceeds a fixed limit, the granules are dried to reduce the moisture level before the downstream micro extrusion process. The lead time between the drying and extrusion processes should not exceed a certain limit to avoid increase of the moisture level by exposure to the air. Excessive moisture level affects the material viscosity. This results in fluctuations of the material shear rate and instability in the extrusion process, thus affecting the quality of the tube section. Other examples of systems with strict lead time requirements due to tight shipping schedules involve, for example, water bottling systems or manufacturing systems integrated into highly responsive supply chains. In these situations, the production system should be operated at the maximal possible production rate under a fixed constraint on the lead time value.

In the literature, production control policies have been proposed to perform an indirect control on the production lead time, by regulating inventory levels. Problems that have been deeply investigated are the optimal selection of parameters in token-based policies (Liberopoulos 2013) including basestock, Kanban, Hedging Point Policy (HPP) and hybrid solutions, and the performance comparison of these policies under different scenarios. Moreover, policies for the real-time control of machine parameters, i.e. the processing rates, to regulate the work in process have been proposed (Ma and Koren 2004), (Toshniwal et al. 2004). Although these policies allow to reduce the production lead time, no direct control of this performance measure is performed. Recently, raw material release policies have been proposed to maximize the throughput under an average lead time constraint. For example, in (Biller, Meerkov, and Yan 2013) the average production rate in isolation of the raw material release stage is controlled in Bernoulli serial lines. In these contributions, the analysis is focused only on the average lead time and the distribution of lead time is not taken into consideration.

In this paper we focus for the first time on production control policies for unreliable manufacturing systems that aim at maximizing the throughput of parts that respect a given lead time constraint, namely *effective throughput*. The polices proposed in this paper are of threshold type and regulate the part loading in order to find the best possible trade-off between the goal of maximizing the system's production rate and the need of respecting the imposed lead time constraint. Two policies are investigated in the paper for maximizing the throughput of parts that respect desired lead time constraints in a two-machine one buffer system. The first is the traditional single stage Kanban policy. In this policy, the part release at the upstream stage is stopped if the downstream buffer reaches a certain threshold (buffer capacity). The second is a completely new state-dependent policy. This policy jointly considers the actual level of the buffer and the state of the second machine in the system and decides wether to load the part at the first machine based on this observable states. The effectiveness of these policies is compared by numerical analysis. The results show that this new policy outperforms the Kanban policy by providing a tighter control on the production lead time. In our knowledge, this is the first time a similar approach is investigated and validated in the literature.

In practice, if these policies are applied to systems producing perishable products, where excessive lead times cause the production of defective parts, the proposed approach aims at finding the right trade-off between product quality and production logistics performance of the system. Recently, the importance of considering the bilateral relations between production system design and product quality has been highlighted. For example, in (Inman et al. 2013) and (Colledani et al. 2014) the most recent approaches integrating the two areas are framed and the main results in this area are highlighted. As a matter of fact, only few contributions addressed this problem in a context where perishable products are produced. The design of in-process buffers in industries producing perishable or deteriorating components and products has been addressed in (Liberopoulos and Tsarouhas 2002). A project to determine cost-efficient ways of speeding up the croissant processing lines of Chipita International Inc. is reported. The installation of a properly sized in-process buffer at a specific point of the line led to a reduction in failure impact on product quality and an increase of the system efficiency. In (Liberopoulos, Kozanidis, and Tsarouhas 2007) the author focused on the production rate of asynchronous production lines in which machines are subject to failures. If the failure of a machine is long enough, the material under processing in the upstream machines must be scrapped by the system. In (Wang, Hu, and Li 2010) a transient analysis is proposed to design the size of the buffers needed in dairy filling and packaging lines. In (Subbaiah, Rao, and Rao 2011) an inventory model for perishable products with random perishability and alternating production rate is proposed. As shown in these works, inventory capacity should be designed by using an integrated quality-logistics approach.

The paper is structured as follows. In the next section the basic modeling assumptions are introduced and the main performance measures of interest in this context are outlined. In Section 3 the new production control policy is formulated. It can be easily seen that the single stage Kanban policy is a sub-case of the proposed policy. In Section 4 the effect of the proposed policy on the system state-space structure is investigated. In Section 5 the system lead time distribution is calculated by an exact analytical approach. In Section 6 the derivation of the optimal policy is discussed, while in Section **??** the proposed policies are compared with respect to several performance measures. The conclusions and future research directions are highlighted in Section 8.

2. System Modeling

2.1 *Modeling Assumption*

The system is composed of two machines, namely M_1 and M_2, and a finite capacity buffer. Finite capacity buffers are used for modeling the single stage Kanban control policy. Thus, to modify the capacity of the buffer, B, is equivalent to change the number of Kanbans at the production stage. A discrete time system is considered, i.e., time is divided into slots. Both machines have equal and constant processing times. Time is scaled so that the processing cycle of each machine takes one time unit. If operational, machines start their operations at the same time instant. Machine M_i, with $i = 1, 2$, is unreliable and characterized by F_i failure modes. In particular, machine M_i may fail with probability $p_{i,j}$ whenever it is operational and begins to process a part, where $j \in \{1, .., F_i\}$. Consequently, for each failure mode, times to failure (TTFs) follow a geometric distribution with mean $1/p_{i,j}$. If the machine is down in mode j it gets repaired in a slot with probability $r_{i,j}$. Thus, for each failure mode, times to repair (TTRs) follow a geometric distribution with mean $1/r_{i,j}$. For each machine, failure modes are mutually exclusive, in the sense that a machine cannot go down in a certain mode without being repaired from a different failure mode. Therefore, each machine is characterized by a set of states S_i with dimensionality $L_i = F_i + 1$. We will denote by U the up state and by D_i the ith failure mode. When the machine is operational it processes 1 part per time unit while it does not process parts if the machine is down. The dynamics of each machine can be captured in the transition probability matrix T_i, which is a square matrix of size L_i. An example of transition probability matrix for a machine with 3 failure modes is provided below.

$$\mathbf{T_1} = \begin{vmatrix} 1-p_{1,1}-p_{1,2}-p_{1,3} & p_{1,1} & p_{1,2} & p_{1,3} \\ r_{1,1} & 1-r_{1,1} & 0 & 0 \\ r_{1,2} & 0 & 1-r_{1,2} & 0 \\ r_{1,3} & 0 & 0 & 1-r_{1,3} \end{vmatrix}$$

The generic state indicator for this system assumes the form $s = (b, \alpha_1, \alpha_2)$, where b is the number of parts in the buffer and α_i assumes values in the set S_i. Under these assumptions, in there are $(B+1)L_1L_2$ states in total. We assume that, in each time slot, the state of the machine is determined at the beginning of the time unit and the buffer content is changed accordingly at the end of the time unit. The upstream machine is never starved and the downstream machine is never blocked. Machine M_1 is blocked if the downstream buffer is full. Similarly, machine M_2 is starved if there are no parts in the upstream buffer. Operational Dependent Failures are assumed, i.e., a machine cannot make transitions to other states if it is starved (M_2) or blocked (M_1). The Blocking Before Service (BBS) mechanism is assumed, i.e., machine M_1 can start processing a part only if the buffer is not full. With these assumptions some states, for example state (B, U, U), are transient and thus do not have to be considered during the analysis.

We assume that the lead time is constrained and the processed parts are allowed to spend a maximum of N time units inside the system. Therefore, if the lead time is larger than N, the lead time constraint is violated and the part cannot be delivered in output from the system, thus it is disposed, when it leaves M_2.

2.2 *Performance Measures*

The performance measures of interest for this kind of systems are summarized in the following:

- The total production rate of the system, E, calculated neglecting the lead time constraint.
- The probability that the lead time LT exceeds the threshold N: $D = P(LT > N)$.
- The effective production rate, E^{Eff}, only considering parts that respect the lead time constraint. It is given by $E^{Eff} = E \times (1 - D)$.
- The defective production rate, E^{Def}, i.e. the throughput of parts that do not respect the lead time constraint. It is given by $E^{Def} = E \times D$.
- The yield, Y, i.e. the fraction of parts respecting the lead time limit. It is given by $Y = E^{Eff}/E$.
- The average inventory in the system, or work in progress, WIP, i.e. the average amount of parts in the system.
- The average lead time in the system.

3. Structure of the Production Control Policies

If the system behaves as described in Section 2.1, a traditional single stage Kanban production policy is implemented. In other words, machine M_1 is allowed to load a raw part, to process it and to release it in the downstream buffer, whenever there is at least one slot available in the downstream buffer and independently of the state of the second machine. However, in case machine M_2 is characterized by multiple failure modes with significantly different repair times, the Kanban level shall be set as a compromise between the effect of long and short failures, thus resulting in a non-negligible fraction of parts exceeding the lead-time constraint.

In order to reduce the amount of parts that exceed the target lead time, in this paper a new state-dependent production control, or part release, policy is proposed for machine M_1. The rationale of the policy is explained in the following. Whenever the machine M_1 is operational and machine M_2 is down in failure mode $j = 1, .., F_2$ if there are more than b_j parts in the buffer the first machine is prevented from loading, processing and releasing the part into the buffer. Therefore, for each failure mode of the second machine M_2, a specific threshold b_j is defined that limits the number of parts that should be deposited in the buffer, under that specific condition.

More formally, the proposed state-dependent production control policy is characterized by two actions:

- Action 0: load the part. If the machine M_1 is operational, the part is loaded, processed and released in the buffer by M_1.
- Action 1: stop part loading. If the machine M_1 is operational, machine M_1 does not load, process, and release the part in the buffer.

The policy u acts on the loading of parts at the first machine M_1 only if the second machine M_2 is failed. It is defined by the vector $\mathbf{b_u}$ of critical buffer levels, or thresholds. In detail, for each $j = 1, .., F_2$:

- If the buffer level $b \leq b_j$ and M_2 is down in mode j: implement action 0.
- If the buffer level $b > b_j$ and M_2 is down in mode j: implement action 1.

It is worth to notice that the implementation of the proposed policy does not entail additional complexity with respect to the single stage Kanban policy. Indeed, in addition to the level of the downstream buffer, also the state of the second machine is taken into consideration to implement the selected action. Both state variables are observable in the system. It is also worth to highlight that the single stage Kanban policy can be seen as a sub-case of the new state-dependent policy. Indeed, if $b_j = B - 1$ for $j = 1, .., F_2$ then the state-dependent policy reduces to the Kanban policy. However, the application of the state-dependent policy affects the behavior of the system. This phenomenon will be investigated in the next section.

4. Effect of the Policy on the System

A system applying the policy introduced in the previous section can be modeled by a Discrete Time Markov Chain (DTMC). Let us first consider the state space of the DTMC. As long as M_2 is up, the buffer level cannot increase. If M_2 is in failure mode j, with $j \in \{1, 2, ..., F_2\}$, and we have $b > b_j$, then the buffer level cannot increase because of the control policy. Accordingly, the maximal buffer level that can be reached in the system is $\max_{j=1,2,...,F_2}(b_j + 1)$. Let us assume hence $B = \max_{j=1,2,...,F_2}(b_j + 1)$. As said before, the number of states to consider is approximately $(B+1)L_1L_2$ because at the boundaries (where the buffer is empty or full) some states are transient.

In order to have a more compact description of the transition probabilities of the resulting Markov chain we introduce some further notation:

$$\overline{p_i} = 1 - \sum_{j=1}^{F_i} p_{i,j} \qquad i = 1, 2$$

is the probability that machine M_i does not break down if it is up, and

$$\overline{r_{i,j}} = 1 - r_{i,j} \qquad i = 1, 2, \quad j = 1, ..., F_i$$

is the probability that machine M_i does not get repaired if it is in state D_j.

A generic portion of the state space away from the borders $(1 < b < B-1)$ is depicted in Figure 1 assuming a single failure mode for machine M_1 and two failure modes for M_2 (not all transitions are drawn). Let us consider first state (b, U, U) in which there are b parts in the buffer and both machines are operating. Since both machines are up, the proposed policy does not intervene and the following transitions are possible:

- with probability $\overline{p_1}\ \overline{p_2}$ the machines remain operating and thus both machines process a part and the system remains in state (b, U, U);
- with probability $p_{1,1}\ \overline{p_2}$ machine M_1 breaks down while M_2 remains operative processing a part and thus the buffer level is decreased to $b - 1$;
- if machine M_2 enters failure mode 1 (or 2) and M_1 remains operative then the next state is $(b + 1, U, D_1)$ with probability $\overline{p_1}\ p_{2,1}$ (or $(b + 1, U, D_2)$ with probability $\overline{p_1}\ p_{2,2}$)
- if both machines fail then the buffer level remains the same and the next state is either (b, D_1, D_1) or (b, D_1, D_2) with probability $p_{1,1}p_{2,1}$ and $p_{1,1}p_{2,2}$, respectively.

We turn our attention now to a state in which M_1 is up and M_2 is down, namely state (b, U, D_1). In this state M_1 is either blocked by the policy, if $b > b_1$, or can process a part, if $b \leq b_1$. If M_1 is blocked by the policy, then only two transitions are possible (drawn by dashed line):

- with probability $r_{2,1}$ machine M_2 gets repaired, processes a part and the next state is $(b - 1, U, U)$;
- with probability $\overline{r_{2,1}}$ machine M_2 does not get repaired and the system remains in state (b, U, D_1).

If M_1 is not blocked by the policy then four cases must be distinguished:

- M_1 remains operative and M_2 gets repaired and thus both machines process a part and the system goes to state (b, U, U) (with probability $\overline{p_1}r_{2,1}$);
- M_1 remains operative and M_2 does not get repaired and thus only M_1 processes a part and the system goes to state $(b + 1, U, D_1)$ (with probability $\overline{p_1}\ \overline{r_{2,1}}$);
- M_1 breaks down and M_2 gets repaired and thus only M_2 processes a part and the system goes to state $(b - 1, D_1, U)$ (with probability $p_{1,1}r_{2,1}$);
- M_1 breaks down and M_2 does not get repaired and thus no parts are processed and the system goes to state (b, D_1, D_1) (with probability $p_{1,1}\overline{r_{2,1}}$).

Outgoing transitions of state (b, U, D_2) can be determined in a similar manner. We consider now state (b, D_1, U) in which, since machine M_2 is operative, the policy does have an impact. The cases to consider are symmetric to those above (associated transitions are not drawn in Figure 1):

- M_2 remains operative and M_1 gets repaired and thus both machines process a part and the system goes to state (b, U, U) (with probability $\overline{p_2}r_{1,1}$);
- M_2 remains operative and M_1 does not get repaired and thus only M_2 processes a part and the system goes to state $(b - 1, D_1, U)$ (with probability $\overline{p_2}\ \overline{r_{1,1}}$);
- M_2 breaks down and M_1 gets repaired and thus only M_1 processes a part and the system goes to state $(b + 1, U, D_1)$ or $(b + 1, U, D_2)$ (with probability $p_{2,1}r_{1,1}$ and $p_{2,2}r_{1,1}$, respectively);
- M_2 breaks down and M_1 does not get repaired and thus no parts are processed and the system goes to state (b, D_1, D_1) or (b, D_1, D_2) (with probability $p_{2,1}\overline{r_{1,1}}$ and $p_{2,2}\overline{r_{1,1}}$, respectively).

We consider now a state in which both machines are down, namely (b, D_1, D_2). Also in this case the behavior of the system depends on the threshold associated with the failure mode of machine M_2, namely b_2. If $b > b_2$ then M_1 can get repaired but cannot put parts into the buffer until M_2 does not get repaired. The possible transitions in case of $b > b_2$ are the following (drawn by dashed lines in Figure 1):

- with probability $\overline{r_{1,1}}\ \overline{r_{2,2}}$ both machines remain down and the system remains in state (b, D_1, D_2);
- with probability $r_{1,1}\overline{r_{2,2}}$ machine M_1 gets repaired (but cannot process a part) and M_2 remains down which leads to state (b, U, D_2);
- with probability $\overline{r_{1,1}}\ r_{2,2}$ machine M_1 does not get repaired, M_2 turns operative and processes

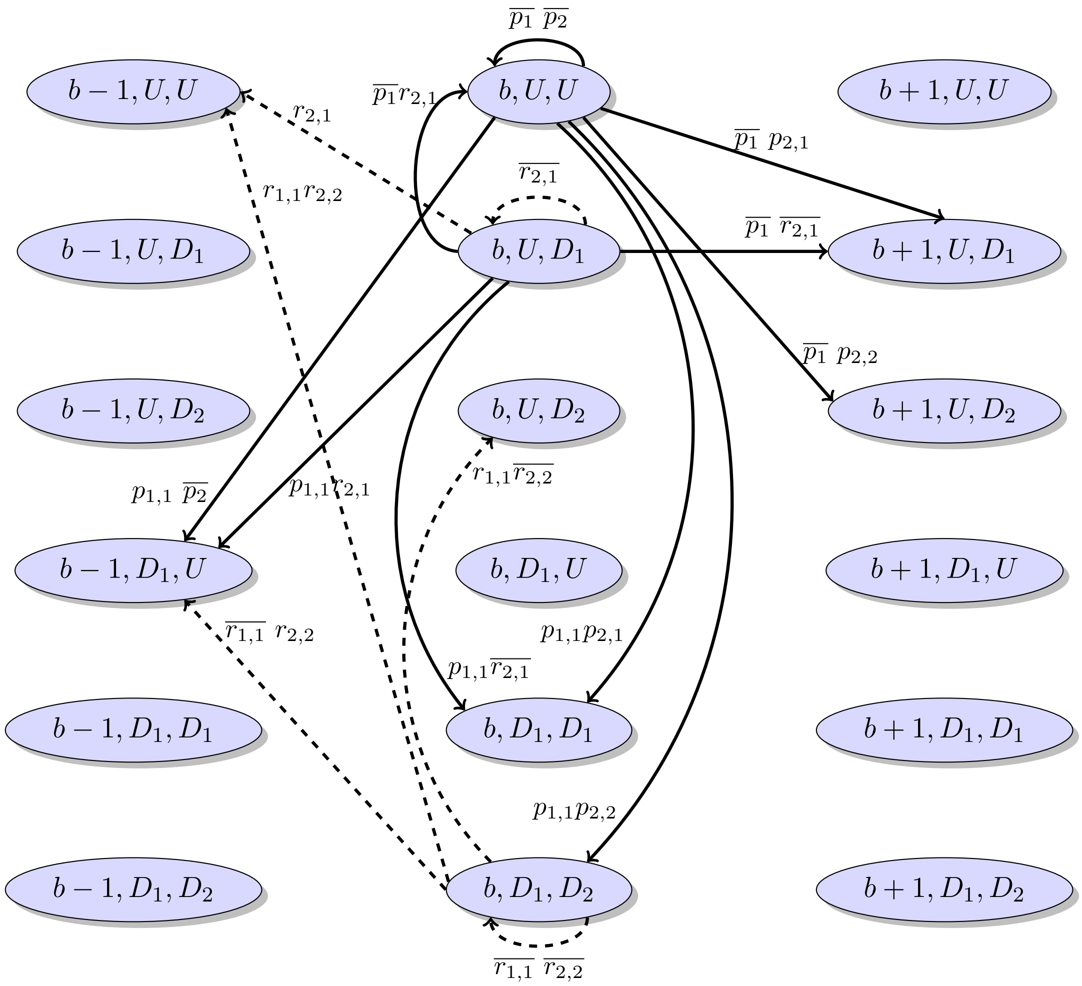

Figure 1. A portion of the state space away from the borders with $F_1 = 1$ and $F_2 = 2$. Note that not all transitions are drawn in the figure. Dashed lines indicate transitions that are present when machine M_1 is blocked by the proposed threshold based policy.

a part and thus the next state is $(b-1, D_1, U)$;
- with probability $r_{1,1}r_{2,2}$ both machines get repaired but only M_2 can process a part and the system goes to state $(b-1, U, U)$.

On the other hand, if $b \leq b_2$ then machine M_1 can process a part if it gets repaired. We have the same probabilities associated with the transitions as in the previous case but the next state of the system is different if M_1 gets repaired (these transitions are not depicted in Figure 1):

- with probability $\overline{r_{1,1}}\ \overline{r_{2,2}}$ both machines remain down and the system remains in state (b, D_1, D_2);
- with probability $r_{1,1}\overline{r_{2,2}}$ machine M_1 gets repaired, processes a part and M_2 remains down which leads to state $(b+1, U, D_2)$;
- with probability $\overline{r_{1,1}}\ r_{2,2}$ machine M_1 does not get repaired, M_2 turns operative and processes a part and thus the next state is $(b-1, D_1, U)$;
- with probability $r_{1,1}r_{2,2}$ both machines get repaired, process a part and the system goes to state (b, U, U).

Outgoing transitions of state (b, D_1, D_1) can be determined in a similar manner.

The above described procedure can be adopted easily to the general case in which both machines have multiple failure modes and the policy is characterized by the vector of thresholds $\mathbf{b_u}$. The resulting transition probability matrix of the DTMC will be denoted by P. Moreover, we will denote by P' the matrix which contains only those transition probabilities of the DTMC that correspond to placing a part in the buffer. I.e., $P'_{i,j} = P_{i,j}$ if in state j M_1 is up and in state i M_1 is not blocked (due to the control policy or because the buffer is full) and otherwise $P_{i,j} = 0$.

5. Performance Analysis

The lead time of a part can be computed following the approach described in Colledani, Horvath, and Angius (2015) for lines with machines of general complexity and in Shi and Gershwin (2012) for lines with up-down machines. In this paper we give a brief overview of this approach. The first step is to compute the steady state probabilities of the DTMC characterized by P. Denoting by π the vector of steady state probabilities, this corresponds to solve the following set of linear equations

$$\pi P = \pi, \quad \pi\vec{1} = 1 \tag{1}$$

where $\vec{1}$ is a column vector of ones. Efficient techniques to find the solution of (1) can be found, for example, in Stewart (2009). Having determined π, the production rate of the system can be obtained by

$$E = \pi P'\vec{1}$$

The second step is to characterize the system when a part is put in the buffer. This can be done by calculating

$$\pi' = \frac{\pi P'}{E}$$

which is a stochastic vector and its entry π'_i gives the probability that the system is in state i when a randomly chosen part enters the buffer.

The third step is to build a DTMC that models the way a part, which is already put in the buffer, leaves the system. A state in this DTMC is a couple (b, α_2) where b is the number of parts in the system and α_2 is the state of machine M_2 (M_1 has not to be considered in this DTMC because it has no impact on a part that is already in the system). This DTMC, whose transition probabilities can be derived from the parameters of M_2, is depicted in Figure 2. The transition probabilities are as follow. When the system is in state (b, U), i.e., M_2 is operational, we have the following possibilities: M_2 remains operational with probability $\overline{p_2}$ and the next state is $(b-1, U)$ or M_2 goes into failure mode $i, i \in \{1, ..., F_2\}$, with probability $p_{2,i}$ and the next state is (b, D_i). On the other hand, if M_2 is not operational, i.e., the state is (b, D_i) with $i \in \{1, ..., F_2\}$, then there are two possibilities: M_2 gets repaired with probability $r_{2,i}$ and the next state is $(b-1, U)$ or M_2 remains down with probability $\overline{r_{2,i}}$ and the system remains in the same state (b, D_i). Let us denote the resulting probability transition matrix by P''. Note that the DTMC characterized by P'' has an absorbing state, $(0, U)$, which signals the moment when the last part is removed from the system.

The initial probability vector for this DTMC, denoted by π'', will be chosen in such a way that it reflects the situation when a random part is put in the buffer. Accordingly, π'' must be derived from π' (they are not the same because π' and π'' do not refer to the same state space). Given a state i, we denote by $\mathcal{B}(i)$, $\mathcal{M}_1(i)$ and $\mathcal{M}_2(i)$ the number of parts in the buffer, the state of M_1 and the state of M_2 in i, respectively. With this notation the relation between π' and π'' can be

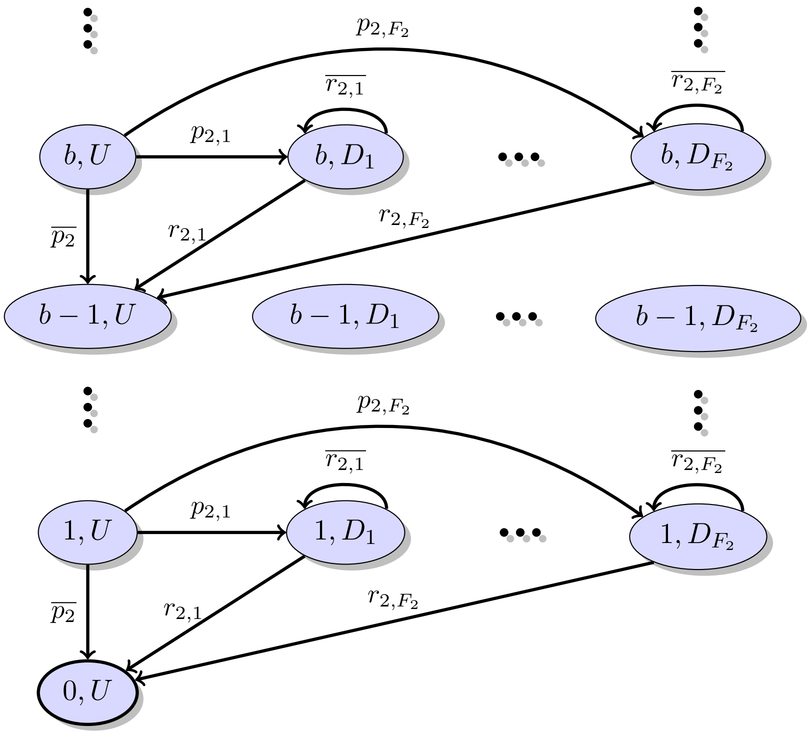

Figure 2. Absorbing Markov chain used to characterize the lead time distribution.

formalized as

$$\pi''_i = \sum_{\forall j: \mathcal{B}(j)=\mathcal{B}(i) \wedge \mathcal{M}_2(j)=\mathcal{M}_2(i)} \pi'_j$$

i.e., an entry in π'' covers all those entries of π' that correspond to states which the number of parts in the buffer and the state of machine M_2 are identical.

The probability that the lead time does not exceed a given threshold N is equal to the probability that the DTMC characterized by π'' and P'' is in its absorbing state after N time units. The transient probabilities of this DTMC after N time units can be obtained by calculating the vector

$$\nu^{(N)} = \pi''(P'')^N$$

Let us denote the entry of $\nu^{(N)}$ that corresponds to state $(0, U)$ by $\nu^{(N)}_{(0,U)}$. With this notation, the probability that the lead time exceeds N time units is simply

$$D = 1 - \nu^{(N)}_{(0,U)}$$

Most performance measures listed in Section 2.2 can be directly calculated based on E and D. The WIP of the system can be calculated in the standard way from π. The mean lead time and higher order moments can be calculated by considering the fact that the lead time distribution is a discrete phase type distribution (Bobbio et al. (2003)).

6. Policy Optimization

In a recent paper Colledani, Angius, and Horvath (2014), we showed how the effective throughput is affected by buffer capacity when Kanban policy is applied. In particular, we showed that often the variation of the effective throughput as function of buffer capacity describes a convex curve that admits a single maximal point that can be found easily since the optimization depends on a single paramater. In this section, we present the same optimization problem for systems that implement our failure control policy. On one hand, the search of an optimal policy is more complex because it involves as many parameters as the failures of the downstream machine i.e. the vector $\mathbf{b_u}$. On the other hand, by tuning more than one parameter, we have more freedom to search a solution that improves the effective throughput togheter with other performance measures.

In principle, the parameters of the proposed policy $\mathbf{b_u}$ should be optimized with respect to some economic performance indicators of the system, for example the expected long term profit. However, in practice it is very hard to estimate the cost coefficients to a level of confidence that is significant for the analysis. Moreover, the use of a synthetic economic objective function prevents from observing characteristics behavior of the system that can provide insights on the application of the policy. Therefore, in this paper we focus on control policy optimization problems based on production logistics related to performance indicators. Indeed, this choice penalizes our work because it significantly reduces the gap between our failure control policy and the Kanban but, at the same time, it enhances the importance of the result by showing that our method is able to provide not negligible improvements also in a "flattened" scenario.

In the following, we propose four optimization problems that will be used in Section 7 to illustrate the advantages of the failure control policy. The first three problems aim to maximize the effective throughput whereas the last one deals with WIP minimization.

Unconstrained Optimization

The first problem deals with the unconstrained optimization of the effective throughput. This means that we search for the set of parameters $\mathbf{b_u}$ that provides the highest effective throughput by disregarding any other performance measure. As a consequence, the formalization of the problem, referred as *Unconstrained* (U), coincides with its objective function:

$$E_U^{Eff} = \max_b \left[E^{Eff}(b) \right]. \tag{2}$$

In Section 7 we will show that this search for optimal solution is naive and can lead to a drastic worsening of other important performance indicators, such as the WIP, without a significant gain once compared with the results of more conservative optimization problems. For this reason, the remaining of this section is dedicated to the description of three possible alternatives.

Kanban Constrained

The second optimization problem aims to tackles the drawbacks provided by the unconstrained optimization in such a way that it maximizes the effective throughput but limits the search to systems having a buffer capacity smaller or equal than B^*, namely the optimal buffer capacity when Kanban policy is applied. We refer such optimization problem as B^* *constrained* (BC). In formula:

$$E_{BC}^{Eff} = \max_b \left[E^{Eff}(b) \right]. \tag{3}$$

subject to

$$\max_j(b_j) \leq B^*. \tag{4}$$

Since B^* constrained optimization considers a sub-set of the feasible solutions of the uncontrained optimization, by definition, we have that the relation $E_{BC}^{Eff} \leq E_U^{Eff}$ holds. However, due to the additional constraint, it is also guaranteed that WIP, $E[LT]$, $Var[LT]$ and $(1 - Y)$ cannot be greater than those obtained by using the Kanban policy. As a consequence, the BC optimization provides a policy that, in the worse case scenario, concides with the optimal Kanban but it can never happen that it provides a policy that improves the effective throughput by worsening the other performance measures in comparison with the optimal Kanban.

A possible disadvantage of BC optimization is that it is a very conservative strategy that bounds the improvements on the effective throughput by limiting the buffer size and, consequently, the decoupling between the upstream and the downstream machine. The main problem is that Kanban's optimal buffer capacity is the result of an averaging between all the MTTRs of the downstream machine. Therefore, a single, possibly rare, failure with a high MTTR can lead to a small buffer even if the other failures are characterized by small MTTRs.

WIP constrained

In order to loosen the constraint and try to further improve the effective throughput, we propose a third optimization problem that follows the same path of the second but takes in consideration the WIP obtained by using the optimal Kanban. Therefore, the problem, referred as *WIP Constrained* (WC), is defined as:

$$E_{WC}^{Eff} = \max_b \left[E^{Eff}(b)\right]. \tag{5}$$

subject to

$$WIP(b) \leq WIP(|B^*, .., B^*|). \tag{6}$$

The idea behind this strategy is to be more flexible in regard to the buffer capacity but, at the same time, do not consent an increment of the average number of parts inside the system. By acting in this way, we can set large thresholds for all the failures having small MTTRs and mantain tight thresholds for those failures that cause the scrapping with high probability.

By definition, we have that the relation $E_{BC}^{Eff} \leq E_{WC}^{Eff} \leq E_U^{Eff}$ holds whereas WIP, $E[LT]$ and $Var[LT]$ are guaranteed to be equal or smaller than the optimal Kanban but they will be greater or equal than those obtained by using BC optimization.

WIP Minimization

The last optimization problem coincides with the dual problem of the WIP constrained optimization. In fact, it aims to minimize the system WIP as long as the effective throughput is greater than the one obtained by using the optimal Kanban. This problem will be referred as *Throughput constrained* (TC) and corresponds to:

$$WIP_{TC} = \min_b \left[WIP(b)\right]. \tag{7}$$

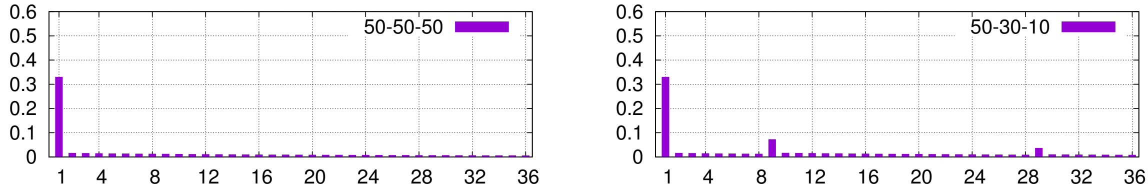

Figure 3. Lead Time distribution under the Kanban and state-dependent control policies.

subject to

$$E^{Eff}(b) \geq E^{Eff}(|B^*, .., B^*|). \tag{8}$$

As a consequence, we have that WIP, $E[LT]$, $Var[LT]$ and $(1-Y)$ obtained by using any of the optimization problem proposed above will be greater than those obtained by using the throughput constrained optimization. At the same time we have that the effective throughput cannot be smaller than the one obtained by using the optimal Kanban. Indeed, TC reflects the scenario in which the service level is not a priority. Therefore, more attention can be dedicated to the reliability of the output of the system.

In order to solve these optimization problems, an exhaustive search approach has been implemented. More intelligent optimization algorithms will be tested in future works.

7. Numerical Results

7.1 *Effect of the Policy on the Lead Time Distribution*

First we illustrate the effect of the proposed policy on the lead time distribution of the system. We consider a system in which: i) the first machine has a single failure mode with parameters $p_{1,1} = 0.01$ and $r_{1,1} = 0.1$; ii) the second machine is characterized by three failure modes that occur with probability $p_{2,1} = 0.01$, $p_{2,2} = p_{2,3} = 0.005$, and are repaired with probability $r_{2,1} = 0.1$, $r_{2,2} = 0.05$ and $r_{2,3} = 0.01$ respectively; iii) the number of Kanbans is fixed at $B = 50$ items. As already shown in (Colledani, Angius, and Horvath 2014), under the single stage Kanban control policy (top box of Figure 3) the lead time distribution in a two machine line is characterized by a bi-modal distribution that has peaks at time units 1 and $B-1$. The magnitudes of these peaks depend on the difference of the efficiencies of the two machines.

With the new state-dependent policy, the lead time distribution becomes a multi-modal distribution with as many peaks as the number of failure modes to which a distinct critical threshold level is associated by the control policy. In the second box of Figure 3 we show the lead time distribution of the same system operating under state-dependent control policy with parameters $\mathbf{b_u} = (50, 30, 10)$. In this case, the lead time distribution is characterized by peaks of different magnitudes at levels $(9, 29, 49)$. This shift in the distribution can provides considerable benefits to the system behavior. In particular, we can observe that the new peaks detract probability from the tail of the distribution on the right, corresponding to the large values of lead time, thus reducing in practice the probability that the lead time exceeds the fixed constraint. It is remarkable to observe that the introduction of the new policy also reduces the first and the second moments of the distribution, thus providing benefits also to the average lead time of the system. Therefore, these graphs prove that with the introduction of the new state-dependent production control policy a more direct control on the lead time distribution can be achieved. In the next experiment, the impact of the new policy on other system performance measures is investigated.

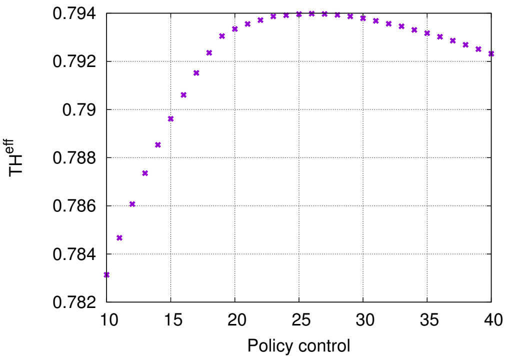

Figure 4. Effective throughput of the system as a function of the single stage Kanban level.

b	E	E^{Eff^*}	WIP	$\mathbb{E}[LT]$	$\mathbb{V}[LT]$	Y
Kanban	0.7958	0.7939	6.6339	8.3354217415	4041.59653	0.9981
U	0.7970	0.7963	6.2939	7.8963	1937.8116	0.9993
WC	0.7970	0.7963	6.2939	7.8963	1937.8116	0.9993
BC	0.7969	0.7962	5.9099	7.4159	1839.2490	0.9993
TC	0.7945	0.7939	4.9137	6.1842	1596.1407	0.9994
Δ% U	0.1499	0.3026	-5.1252	-5.2672	-52.0533	0.1212
Δ% CW	0.1499	0.3026	-5.1252	-5.2672	-52.0533	0.1212
Δ% CB	0.1301	0.2871	-10.9148	-11.0306	-54.4920	0.1246
Δ% DW	-0.1663	0.0003	-25.9306	-25.8072	-60.5071	0.1329

Table 1. Comparison of the optimal policies for the considered case

7.2 *Comparison between policies*

In the second experiment, the performance measures provided by the application of the optimal single stage Kanban and the optimal state-dependent policies are compared for a specific case. We consider a system in which: i) the first machine fails with probability $p_{1,1} = 0.2$ and is repaired with probability $r_{1,1} = 0.1$; ii) the second machine is characterized by three failure modes that occur with probability $p_{2,1} = 0.0033$, $p_{2,2} = 0.0066$ and $p_{2,3} = 0.0001$, and are repaired with probability $r_{2,1} = 0.5$, $r_{2,2} = 0.1$ and $r_{2,3} = 0.001$ respectively; iii) the lead time constraint is fixed at $N = 100$ time units.

We first solve the optimization problem formulated in Section 6 for the single stage Kanban policy. The results are reported in Figure 4. The effective production rate is maximized at $B^* = 26$.

Then, we solved the four optimization problems presented in Section 6. Since the second machine is affected by three failure modes there are three threshold levels, i.e. b_1, b_2, and b_3 to be fixed in order to perform the optimization. In reference to the Unconstrained optimization, the optimal parameters are $\mathbf{b_u}* = (22, 29, 5)$ which corresponds also to the optimal solution provided by the WIP constrained problem. Therefore, in this case the unconstrained problem does to a worsen the WIP, $\mathbb{E}[LT]$, $\mathbb{V}[LT]$ and $Yield$. For what concern the BC problem and WIP minimization we have that the optimal parameters are $\mathbf{b_u}* = (21, 26, 5)$ and $\mathbf{b_u}* = (25, 19, 5)$, respectively.

The overall set of performance measures is reported in Table 1 for all the policies, together with the percentage relative difference provided by the two policies.

We can observe that all the new state dependent policies provide an increase in the effective throughput of the system while the total throughput is decreased.

Therefore, under the new production control policy, the system produces less parts in total but more parts respect the lead time constraint. This is due to the fact that the state-dependent

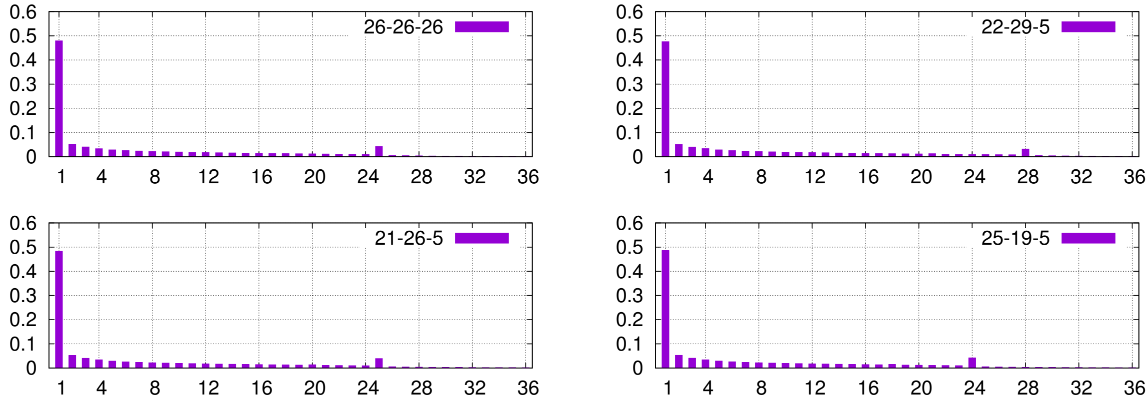

Figure 5. Lead Time distribution under the optimal Kanban and the four optimal failure control policies.

Machine	$f_{i,1}$	$f_{i,2}$	$f_{i,3}$	$f_{i,4}$	$r_{i,1}$	$r_{i,2}$	$r_{i,3}$	$r_{i,4}$	Efficiency
1	0.003	0.003	0.003	0	0.1	0.2	0.05	0	0.9049
2	0.03	0.03	0.03	0	0.1	0.5	0.7	0	0.7128
3	0.003	0.003	0.0016	0.0016	0.1	0.5	0.07	0.01	0.8204
4	0.003	0.003	0.0016	0.0016	0.7	0.5	0.1	0.01	0.8429

Table 2. Parameters of multifailure machines

policy can prevent the first machine from producing even if it is operational, with the objective to limit the lead time of parts flowing in the system. For what concerns optimal policies that aim to maximize the effective throughput the increment is around the 0.3%. In reference to the WIP minimization such increment is instead almost null but, at the same time, we can notice a remarkable improvement on the other performance measures. In fact, work in progress and average lead time are reduced by more than 25% and the variability of the lead time is decreased by more than the 60%. In this direction, smaller but still notable results are obtained also by the other policies that show improvements of the WIP and mean lead time that oscillate around 5% and 10% and a reduction of the variability around 55%.

Figure 5 reports the lead time distribution for all the different optimal policies.

As expected, it is possible to notice that the lead time corresponding to the minimal WIP has the shortest tail.

We performed a large number of similar tests by using both real and "in silico" parameters. Similar results have been found in the major part of the cases.

We selected a set of 10 tests based on four different machines by assuming that items must be scrapped after $N = 50$ time units, Table 2 reports the parameters of each machine; we can observe that the first two machines are characterized by three failures whereas the last two by four. Machines differ also in efficiency; the first is highly efficient (around 90%), the second is poorly efficient ($\approx$ 70%), and the third and the fourth, referred as medium efficiency machines, are characterized by efficiencies around $\approx$ 82% and $\approx$ 84% respectively. By using these machines, we generated the 10 different systems that reported in Table 3. It is possible to notice that the first four systems are balanced since we use the same machine type both for the downstream and the upstream. The last six cases are instead dedicated to unbalanced in systems.

In order to provide a preliminary qualitative comparison, Table 3 provides also the optimal policy vectors obtained by applying the optimization problems presented in Section 6. We can observe that there are only three cases in which the solutions of two optimization problems coincide i.e. U and CW on system 1; CK and DW on system 2; CW and CK on system 5. Furthermore, it has

Case	M_1	M_2	B^*	U	CW	CK	DW
1	1	1	2	(23,24,19)	(23,24,19)	(22,22,19)	(21,22,16)
2	2	2	20	(19,22,22)	(19,21,21)	(19,20,20)	(19,20,20)
3	3	3	13	(20,21,18,5)	(17,18,15,5)	(13,13,13,5)	(10,7,8,5)
4	4	4	8	(20,20,20,5)	(9,10,10,5)	(8,8,8,5)	(5,6,6,5)
5	1	2	12	(11,13,13)	(11,12,12)	(11,12,12)	(10,12,12)
6	1	4	14	(20,21,19,5)	(19,19,17,5)	(14,14,14,5)	(10,9,10,5)
7	2	1	35	(38,40,34)	(38,39,34)	(35,35,34)	(35,35,33)
8	2	3	25	(45,45,42,22)	(35,35,35,22)	(25,25,25,21)	(24,21,25,16)
9	3	4	12	(26,27,25,5)	(15,15,15,5)	(12,12,12,5)	(6,8,8,5)
10	4	3	9	(15,15,13,5)	(10,11,9,5)	(9,9,9,5)	(7,6,6,5)

Table 3. Parameters of multifailure machines

never occurred that an optimization problem coincide the Kanban policy.

Since we intentionally used only basic optimization problems we argue that such variety of results points out the complexity of the problem and underlines how large could be the margin for improvements when more detailed optimization problems will be considered.

Table 4 provides the comparison between the performance measures obtained by using the optimal Kanban and the optimal policy of each optimization problem. In sake of synthesis, we reported punctual numbers only for the Kanban whereas the other policies measures are reported in terms of relative distance percentage from the Kanban.

For what concern the effective throughput of the system we can notice that:

- The gain provided by the introduction of buffer policy does not exceed 0.36% in the best case scenario i.e. test 6 by using uncontrained optimization.
- Failure control policies benefit from the number of failure of the downstream machine. The more failures characterize the downstream machine and the more large can be the difference between the optimal failure control policy and the optimal Kanban.
- The difference between the effective throughput obtained by using the optimal U and the optimal CW is most of the times negligible.

Much more significant results are obtained on WIP, average and variance of the lead-time. In fact we can observe that:

- The unconstrained optimization often provides marked worsening of these values whilst it does not provide a significant improvementof the effective throughput once compared with the optimal obtained by WC. As an example, in case 3 we have an increment of 0.3369% againsts 0.3121% but at the same time by using the optimal policy provided by unconstrained optimization the WIP increases by 13% whereas it decreases by 0.1% by using the result of WC.
- The improvements obtained on WIP (and consequently on $\mathbb{E}[LT] and \mathbb{V}[LT]$) by minimizing the WIP are always larger than those obtained on the effective throughput by applying effective throughput maximization problems.

These set of tests points out that:

- when the priority is the effective throughput, the best strategy is to apply the optimal solution provided by the WC optimization because it leads to an effective throughput that is very similar to that provided by the unonstrained optimization without having drawbacks in terms of WIP.
- when the priority is the stability of the output, the greatest gain can be achieved by using the minimization of the WIP.

Case	Policy	E	E^{Eff}	WIP	$E[LT]$	$[V][LT]$	Y
1	B^*	0.8603	0.8500	10.8800	12.6467	295.1206	0.9897
	Δ% U	-0.0259	0.0391	-0.9865	-0.9608	-1.8158	0.0562
	Δ% WC	-0.0259	0.0391	-0.9865	-0.96085	-1.8158	0.0562
	Δ% BC	-0.1141	0.0270	-4.9227	-4.8140	-8.7134	0.12250
	Δ% TC	-0.2881	0.0049	-11.7058	-11.4506	-19.5590	0.2547
2	B^*	0.6492	0.6354	9.9991	15.4016	409.8976	0.9862
	Δ% U	0.0123	0.0317	0.1842	0.1719	0.15656	0.0123
	Δ% WC	0.0123	0.0316	-1.1079	-1.1201	-2.0632	0.0607
	Δ% BC	-0.1632	0.0147	-2.7939	-2.6349	-4.8341	0.1169
	Δ% TC	-0.1632	0.0147	-2.7939	-2.6349	-4.8341	0.1169
3	B^*	0.7120	0.7015	6.4492	9.0568	530.6025	0.9894
	Δ% U	0.2143	0.3369	13.4975	13.2547	-4.4383	0.0846
	Δ% WC	-0.0064	0.3121	-0.1289	-0.1224	-18.5901	0.2259
	Δ% BC	-0.3300	0.2077	-18.1322	-17.8611	-35.4453	0.3848
	Δ% TC	-0.7453	0.0029	-37.9256	-37.4595	-51.2570	0.5384
4	B^*	0.7374	0.7314	3.9704	5.3840	290.5272	0.9940
	Δ% U	0.3980	0.2505	51.5942	50.9932	34.5248	-0.1108
	Δ% CW	-0.0052	0.1505	-0.5990	-0.5937	-16.3843	0.1147
	Δ% BC	-0.1269	0.09135	-13.4908	-13.3808	-26.1316	0.1614
	Δ% TC	-0.28854	0.0007	-30.3377	-30.1361	-36.9158	0.2146
5	B^*	0.6905	0.6840	9.8242	14.2275	282.2502	0.9935
	Δ% U	0.1543	0.0104	6.5715	6.4072	11.4181	-0.1000
	Δ% WC	-0.0236	0.006	-1.0167	-0.9933	-1.8772	0.0205
	Δ% BC	-0.0236	0.006	-1.0167	-0.9933	-1.8772	0.0205
	Δ% TC	-0.0564	0.0033	-2.3031	-2.2480	-4.0517	0.0415
6	B^*	0.7761	0.7636	7.40252	9.5373	577.3116	0.9875
	Δ% U	0.1037	0.36203	6.55553	6.4450	-12.9934	0.1984
	Δ% WC	-0.0068	0.3537	-0.0415	-0.0347	-19.6504	0.2789
	Δ% BC	-0.3893	0.2406	-20.1177	-19.8055	-38.2341	0.4921
	Δ% TC	-0.8419	0.0054	-39.2702	-38.7546	-53.4448	0.6659
7	B^*	0.7070	0.6988	4.3122	6.0991	168.6085	0.9918
	Δ% U	0.0013	0.0075	0.0335	0.0322	-0.0744	0.0043
	Δ% WC	-2.43E-5	0.0073	-0.0134	-0.0134	-0.1664	0.0052
	Δ% BC	-0.0321	0.0014	-1.0797	-1.0479	-2.159	0.0239
	Δ% TC	-0.0668	0.0003	-2.2147	-2.1494	-4.3773	0.0478
8	B^*	0.6562	0.6403	4.5076	6.8690	843.42003	0.98416
	Δ% U	-0.1929	0.06868	-8.0248	-7.8469	-11.2910	0.1733
	Δ% WC	-0.2062	0.0644	-8.4535	-8.2643	-11.5448	0.1794
	Δ% BC	-0.3574	0.02560	-13.1927	-12.8813	-16.2519	0.2550
	Δ% TC	-0.8578	4.3E-5	-29.5338	-28.9241	-36.2011	0.5720
9	B^*	0.7249	0.7159	5.1055	7.0430	448.9454	0.9910
	Δ% U	-0.0239	0.3072	1.0333	1.0575	-19.1250	0.2395
	Δ% WC	-0.2786	0.2298	18.9155	-18.6889	-38.6024	0.3698
	Δ% BC	-0.4045	0.1633	-27.0093	-26.7129	-45.0319	0.4138
	Δ% TC	-0.6613	0.0009	-41.5485	-41.1594	-54.8544	0.4844
10	B^*	0.7224	0.7151	5.0035	6.9261	351.8407	0.9927
	Δ% U	0.3709	0.2057	38.5345	38.0226	23.6833	-0.1217
	Δ% WC	-0.0252	0.1426	-0.9922	-0.9672	-14.9564	0.1211
	Δ% BC	-0.1309	0.1072	-10.7273	-10.6103	-23.3929	0.1725
	Δ% TC	-0.3827	0.0039	-32.5234	-32.26417	-40.0125	0.2813

Table 4. Comparison between optimal Kanban and different optimal buffer policies in terms of relative distance percentage

- The BC optimal policy represents the best trade-off between effective throughput and output stability.

8. Conclusions and Prospects

This paper proposes for the first time the analysis of production control policies for systems with strict lead time limitations. A new state-dependent policy is designed to be implemented in real manufacturing systems where the product is affected by deterioration and perishability and can turn into defective by excessive waiting time in the system. Also, it can be applied in context where the customers requirements on the accepted lead time have to be tightly met. Numerical results prove that the proposed policy can induce a controllable shift in the lead time distribution thus outperforming the application of traditional Kanban policies with respect to the effective production rate. Moreover, it also provides considerable benefits in terms of inventory levels, fraction of parts that exceed the lead time constraint and production yield.

Future research will be oriented towards the extension of this analysis for validating the policy also for production lines with more production stages. Moreover, a formal proof of optimality of the proposed policy within the described context will be investigated. Finally, more complex machine models, also involving non-geometric up and down times and non-identical processing time of machines will be investigated.

References

Biller, S., S. Meerkov, and C.-B. Yan. 2013. "Raw Material Release Rates to Ensure Desired Production Lead time in Bernoulli Serial Lines." *International Journal of Production Research* 51: 4349–4364.

Bobbio, A., A. Horváth, M. Scarpa, and M. Telek. 2003. "Acyclic Discrete Phase Type Distributions: Properties and a Parameter Estimation Algorithm." *Performance Evaluation* 54 (1): 1–32.

Colledani, Marcello, Alessio Angius, and Andras Horvath. 2014. "Lead Time Distribution in Unreliable Production Lines Processing Perishable Products." In *19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'14),* 1–8.

Colledani, Marcello, Andras Horvath, and Alessio Angius. 2015. "Production quality performance in manufacturing systems processing deteriorating products." *CIRP Annals - Manufacturing Technology* 64 (1): 431 – 434.

Colledani, M., T. Tolio, A. Fischer, B. Iung, G. Lanza, R. Schmitt, and J. Vancza. 2014. "Design and management of manufacturing systems for production quality.." *CIRP Annals-Manufacturing Technology* 63(2): 773–796.

Inman, R. R., D. E. Blumenfeld, N. Huang, and J. Li. 2013. "Survey of Recent Advances on the Interface Between Production System Design and Quality." *IIE Transactions* 45/6: 557–574.

Liberopoulos, G. 2013. "Production Release Control: Paced, WIP-Based, or Demand-Driven? Revisiting the Push/Pull and Make-to-Order/Make-to- Stock Distinctions.." *Handbook of Stochastic Models and Analysis of Manufacturing System Operations International Series in Operations Research and Management Science* 192: 211–247.

Liberopoulos, G., G. Kozanidis, and P. H. Tsarouhas. 2007. "Performance evaluation of an automatic transfer line with WIP scrapping during long failures." *Manufacturing and Service Operations* 9/1: 62–83.

Liberopoulos, G., and P. H. Tsarouhas. 2002. "Systems analysis speeds up Chipita's food processing line." *Interfaces* 32/3: 62–76.

Ma, Y., and Y. Koren. 2004. "Operation of Manufacturing Systems with Work-in- process Inventory and Production Control.." *CIRP Annals Manufacturing Technology* 53(1): 361–365.

Shi, C., and S. Gershwin. 2012. "Part Waiting Time Distribution in a Two-Machine Line." In *Proc. of the 14th IFAC Symposium on Information Control Problems in Manufacturing,* 457–462.

Stewart, William J. 2009. *Probability, Markov chains, queues and simulation: the mathematical basis of performance modeling.* 1st ed. Princeton University Press.

Subbaiah, K. V., S.V.U.M. Rao, and K.S. Rao. 2011. "An inventory model for perishable items with alternating rate of production." *International Journal of Advanced Operations Management* 3/1: 66–87.

Toshniwal, V., N. Duffie, T. Jagalski, H. Rekersbrink, and B. Scholz-Reiter. 2004. "Assessment of Fidelity of Control-Theoretic Models of WIP Regulation in Networks of Autonomous Work Systems.." *CIRP Annals Manufacturing Technology* 60(1): 485–488.

Wang, J., Y. Hu, and J. Li. 2010. "Transient analysis to design buffer capacity in dairy filling and packing production lines." *Journal of Food Engineering* 98/1: 1–12.