Analysis of the Lead Time Distribution in multi-product systems with dedicated buffers

Article · January 2018

DOI: 10.1016/j.ifacol.2018.08.224

2 authors:

Alessio Angius
Politecnico di Milano

Marcello Colledani
Politecnico di Milano

Analysis of the Lead Time Distribution in multi-product systems with dedicated buffers

Alessio Angius [*] **Marcello Colledani** [*]

[*] *(alessio.angius, marcello.colledani)@polimi.it , Mechanical Engineering Department, Politecnico di Milano, Milan, Italy*

Abstract: Perishable products that deteriorate before leaving the production system are common in industry. The most classic example is the food industry but several cases can be found in semiconductor manufacturing, and in polymers forming processes. These systems often require the scrapping of parts whose lead-time has exceeded a certain threshold. Previous works have considered single-product systems and have shown that the size of the buffers and actions dedicated to improve the machine availability may strongly affect the percentage of scrapped parts. In this paper, we model the dynamics of this phenomenon in a multi-product system composed of two machines that are connected through dedicated buffers. Furthermore, the model allows the use of three different policies for the mixing of products. The main contribution is a method for the calculation of the lead time distribution of each product that can be used to determine the effective throughput of the system. The relevance of the method is shown by means of numerical results that provide important insights on the problem and show counterintuitive behaviors.

Keywords: lead-time, multi-product, production line

1. INTRODUCTION

Perishable, or obsolete, or deteriorating products are allowed to spend a limited amount of time inside manufacturing systems. If the system flow time, or lead time, exceeds certain fixed thresholds, the product has to be considered as a defect and has to be scrapped by the system. Food production is a typical example. For instance, the production of yogurt is pervaded by strict requirements on delivery precision along the entire production sequence (mixing/standardizing of milk, pasteurization, fermentation, cooling, addition of fruit additives and packaging) due to the maximum allowed storage time before packaging. If the the product exceeds this limit, it has to be scrapped. Another example comes from the production of multi-layer, multi-lumen polymeric microtubes 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.

One of the first work in the context of deteriorating components and products has been addressed in Liberopoulos and Tsarouhas (2002) where cost-efficient ways of speeding up the croissant processing lines of Chipita International Inc. are 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. Liberopoulos et al. (2007) focused on the production rate of asynchronous production lines in which machines are subject to failures. The work pointed out that if the failure of a machine is long enough, the material under processing in the upstream machines must be scrapped by the system. Wang et al. (2010) proposed a transient analysis to design the size of the buffers needed in dairy filling and packaging lines. In Subbaiah et al. (2011) an inventory model for perishable products with random perishability and alternating production rate is proposed. As shown in these works, buffers should be designed by using an integrated approach.

At the best of our knowledge, the only noticeable contributions that address explicitly the lead time in manufacturing systems are: Shi and Gershwin (2012) that considered the time spent a system composed of two machines having geometrically distributed failures and repairs; Angius et al. (2014) that extends the work by considering general Markovian machines and providing a closed-form solution for the geometric case; iii) Angius et al. (2015c) that analyzes a real manufacturing system producing microcatheters for medical applications; iv) Angius et al. (2016) and Angius et al. (2015b) that consider systems having N general machines and different system layouts (Lines, Closed Loop, Assembly); v) Angius et al. (2015a) and Angius et al. (2017) that suggests alternative production policies to reduce the scrapping of parts.

All the works mentioned above take in consideration single-product systems where machines have a single upstream buffer and a single downstream buffer. This setting is becoming rarer and rarer because flexible machines are becoming increasingly common to augment the productivity. Flexible machines are usually equipped with general purpose tooling allowing different products to be worked on without having to perform a retooling operation. This type of system is frequently used in the food industry, for instance in the production of ice cream or juices, in bottling lines in which several types of packs are produced, and, in general, in packaging lines. In this case part types can be differentiated in terms of their size or other properties (e.g., color, weight).

Despite the importance of the problem, literature does not contain works where the problem of the lead time is solved in the context of multi-product machines.

This paper tries to fill this gap. On the one hand, we introduce a model of a system composed of two machines connected with N dedicated buffers. Each buffer contains a different type of product and products are processed by each machine according to a policy that can follow one of three different criteria: fair-sharing, priority and probability. On the other hand, we provide a method that allows the the computation of the lead time of any type of product. The method allows us to derive predictions of the effective throughput and scrap rate.

Numerical results show meaningful insights and counter-intuitive behaviors. For example, a multi-buffer system might outperform the corresponding single-buffer system in terms of production rate of conforming products when products deteriorate fast within the system.

The paper is organized as follows. In Section 2 the problem is formulated and the specific modeling assumptions and performance measures are presented. In Section 3 the mathematical derivation of the lead time distribution calculation is provided. In Section 4 relevant results are shown and discussed. Finally, in Section 5 the conclusions are drawn and future areas of research are highlighted.

2. MODELING ASSUMPTIONS

We consider discrete time manufacturing systems, i.e., assume that time is divided into slots. The system is composed of two machines, namely M_1 and M_2 and N buffer of finite capacity $B_1, B_2, \ldots, B_N$. Figure 1 provides a graphical representation of the system. Machine M_i,

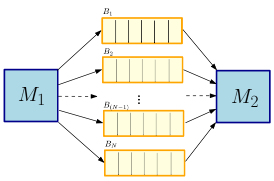

Fig. 1. System composed of two machines connected by N dedicated buffers.

$i = 1, 2$, is characterized by a set of states S_i with dimensionality L_i. The dynamics of each machine in these states are captured in a L_ixL_i probability matrix denoted with T_i. Moreover, a quantity reward vector μ_i is considered, with dimensionality L_i and binary entries: $\mu_i^j = 1$ if the machine is operational and it processes 1 part per time unit while in state j; $\mu_i^j = 0$ if the machine is down and it does not process parts in state j. Therefore, if operational, both machines have the same processing time scaled to the time unit. The states of machine M_i are partitioned according to μ_i into up states (the machine is operational), denoted as U, and down states (the machine is not operational), denoted as D. Without loosing generality, we assume that the states are ordered in such a way that T_i can be decomposed in the following four blocks:

$$T_i = \begin{vmatrix} \bar{P}_i & P_i \\ R_i & \bar{R}_i \end{vmatrix} \tag{1}$$

where, considering machine i, the block $\bar{P}_i$ contains the transition probabilities among the up states, $\bar{R}_i$ among the down states, P_i from up states to the down states (leading to a break down) and R_i from down states to up states (leading to repair).

The generic state indicator for this system assumes the form $s = (\mathbf{b}, \alpha_1, \alpha_2) \in \mathcal{S}$, where α_j assumes values in the set S_j and $\mathbf{b}$ is a vector of the form $|b_1, b_2, \ldots, b_N|$ where b_j describes the number of parts on buffer j.

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. Machines produce N types of items and each type has a dedicated buffer, i.e., each buffer contains only one type of items. In sake of simplicity, all along the paper we will identify the type of items with the number of the buffer in which it is placed; thus, products of type j will be placed on buffer j.

The upstream machine is never starved and is blocked when all the buffers are full. The downstream machine is never blocked and is starved when all the buffers are empty. Operational Dependent Transitions are assumed, i.e. a machine cannot make transitions to other states if it is starved or blocked. Machine blocking is determined before the service (BBS mechanism). Machine M_1 (M_2) is blocked (starved) for products j if buffer j is full (empty). Full blocking (starving) occurs to machine M_1 (M_2) only when all the buffers are full (empty).

Every time a machine is operative, it takes a decision about the type of product to process. Such decision is taken before the service on the basis of a policy and the blocking conditions on the buffers. Machines spend no time to decide. Policies will be described in Section 2.1.

As last, we assume that processed parts of type j undergo deterioration if they spend more than h_j^* time units inside the system. Therefore, products of type j having a lead time greater than h_j^* time units need to be scrapped. Scrapping takes place at the end of the line and does not affect the system dynamics.

2.1 Considered Policies

In order to complete the description of the system dynamics, we need to introduce the policies that define how machines select the type of product that will be processed in the next time unit. Policies affect the mixing of the products and, as a consequence, they determine how the overall throughput of the system is split between the type of products.

We present three different policies. All of them are static over time i.e. neither the policy nor its parameters change during production. Furthermore, the policies depend only on the buffer level and they can be applied at machine level. Therefore each machine is allowed to have its own policy and can apply it without having knowledge of the state of the other machine.

We defined the policies in such way that the upstream machine can process every time there is at least one empty slot in one of the buffers and the downstream machine can produce if there is at least one not empty buffer. This has two main consequences. Firstly, the overall throughput is guaranteed to be the maximum possible. Secondly, several performance measure obtained from the multi-product system are comparable with those provided by a system having machines with the same efficiency connected through a single buffer with capacity equal to the sum of all the buffer capacities of the multi-product system.

Fair-sharing The first policy considered in this paper is called *Fair-sharing* (*FS* from now-on). It considers the machine as coupled with a circular scheduler that choose the next buffer by following the indexing of the buffers. In practice, this means that the state of M_i becomes a couple (α_i, x) where $x \in \{1, \ldots, N\}$ and α_i is the state of the machine as described at the beginning of Section 2. Because the component x represents the type of the last processed item, it is updated only when the machine arrives in an operative state and is not blocked. The scheduler decides the next buffer at the beginning of the time unit by considering only the buffers that do not lead to blocking (or starvation). Formally, we have a function:

$$f_m(\mathbf{b}, x) = \begin{cases} x & \sum_{j=1}^{N}(B_j - b_j) = 0 \land m = 1 \\ next(x) & B_{next(x)} \neq b_{next(x)} \land m = 1 \\ f_m(\mathbf{b}, x+1) & B_{next(x)} = b_{next(x)} \land m = 1 \\ x & \sum_{j=1}^{N} b_j = 0 \land m = 2 \\ next(x) & b_{next(x)} \neq 0 \land m = 2 \\ f_m(\mathbf{b}, x+1) & b_{next(x)} = 0 \land m = 2 \end{cases} \tag{2}$$

where $next(x) = (x + 1 \ mod \ N) + 1$ is a function that determines the next buffer on a circular queue. The first term of equation (2) corresponds to the case in which the machine is the upstream (m=1) and all the buffers are full. Therefore, the upstream machine is blocked and cannot produce. The second term corresponds to the case in which the upstream machine can process parts of the type that is sequentially after x because the corresponding buffer is not full. For example, if $x = 3$ and $N = 3$ then $next(3) = 1$. The third term describes the inductive step that is needed to find the first free buffer if $next(x)$ is full. The last three terms follow the same path for the downstream machine.

Prioritized The second policy, referred as *Prioritized* (*PR*), considers the buffers as associated with priorities. Priorities are collected by means of a vector $\mathbf{y}_i$ whose entries are $y_{i,k}$, $1 \leq k \leq K$. We assume that $\mathbf{y}_i$ always coincides with a permutation of $\{1, \ldots, N\}$. This guarantees that every buffer is associated with one and only one priority.

The production is modulated in such a way that the upstream machine processes parts of type $y_{1,k}$ only if buffers $B_{y_{1,1}}$, $B_{y_{1,2}}, \ldots, B_{y_{1,k-1}}$ are full. For example, if $\mathbf{y}_1 = |3, 1, 2|$ then M_1 can process parts of type 1 every time that buffer 3 is full and parts of type 2 every time buffers 3 and 1 are full. Vice versa for the downstream machine.

Probabilistic The third and last policy, called *Probabilistic* (*PB*), assumes that the production of a machine is governed by a random number generator that extracts the type of product randomly according to a multinomial distribution. The distribution that governs the production of M_i in a given state is fully described by: i) a vector of weights $\mathbf{w}_i$ having entries $w_{i,k} \in \mathbb{R}^+$ that determine the weight of the product k for the machine i, $1 \leq k \leq K$; ii) the buffer level in the state $\mathbf{b}$.

The buffer level is required because blocking conditions might not allow the production of some types of products; therefore, the distribution must be scaled to sum to one. Formally:

$$\nu_m(k, \mathbf{b}) = \begin{cases} \dfrac{w_{1,k}}{\sum_{k:b_k<B_t} w_{1,k}} & b_k < B_k \land m = 1 \\ \dfrac{w_{2,k}}{\sum_{k:b_k \neq 0} w_{2,k}} & b_k \neq 0 \land m = 2 \\ 0 & otherwise \end{cases} \tag{3}$$

The first term of equation (3) states that the probability with which the upstream machine process parts of type k when the buffers are in state $\mathbf{b}$ is equal to $w_{1,k}$ divided by the sum of all the weights corresponding to buffers that are not full. The second term follows the same path but takes in consideration the downstream machine and possible starvations.

2.2 Performance Measures

The problem tackled in this paper can be formulated as follows. Given the system described by the assumptions detailed in the previous section, calculate the following performance measures:

- E_j : the system production rate of parts of type j, considering both good and deteriorated parts.
- $P(LT_j = h)$: the probability that the lead time of the type j is equal to the number of time units h.
- E_j^{Eff} : the effective production rate of type j, only considering good parts. It is given by $E_j^{Eff} = E \times P(LT_j \leq h_j^*)$
- E : the total production rate of the system, it is equivalent to $\sum_{j=1}^{N} E_j$.
- E^{Eff} : the overall effective production rate of the system. It is given by $E^{eff} = \sum_{j=1}^{N} E_j^{Eff}$.

3. NUMERICAL METHOD

In this section we describe the procedure to calculate the distribution of the lead time. The procedure is composed of three steps. In the first step, we build the DTMC that describes the system and computes its steady-state. The second step consists in computing the initial condition for the computation of the lead time. Finally, in the third step, we build another DTMC representing the absorbing chain whose time to absorption corresponds to the lead time of a part of type j.

Construction DTMC and computation steady-state The construction of the DTMC describing the system can be done by applying the assumptions described in Section 2 and the rules described in Buchholz and Telek (2010) that provides a detailed description of concurrent systems enhanced with Phase-type distributions. The resulting DTMC is a block-matrix similar to the one described in Angius et al. (2014) where the single-product case is introduced. The multi-product case requires only small adaptations but a proper description requires the introduction of a notation that is too heavy for being contained in this paper. Thus, we describe only the most indicative rules.

First of all, the DTMC cannot move between two states $(\mathbf{b}, \alpha_1, \alpha_2)$ and $(\mathbf{b}', \alpha_1', \alpha_2')$ if $\mathbf{b}'$ cannot be expressed as

$$\mathbf{b}' = \mathbf{b} + \bar{\mathbf{b}}^+ - \bar{\mathbf{b}}^- \tag{4}$$

where $\bar{\mathbf{b}}^+$ and $\bar{\mathbf{b}}^-$ are vectors having at the most one entry equal to one and the remaining entries equal to zero. The not zero entry points to the buffer that has been accessed by the upstream ($\bar{\mathbf{b}}^+$) and downstream machine ($\bar{\mathbf{b}}^-$). By definition $|\bar{\mathbf{b}}_i| = 0$ whenever: i) $\alpha_i' = D$ because the machine is not operative; ii) if $i = 1$ and $b_j = B_j$ then the jth entry of $\bar{\mathbf{b}}^+$ must be zero because the corresponding buffer is full; iii) if $i = 2$ and $b_j = 0$ then the jth entry of $\bar{\mathbf{b}}^-$because the corresponding buffer is empty.

All the transitions that satisfy equation (4) can be expressed in the form $\beta_{1,j}A_1 \otimes \beta_{2,j}A_2$ where $\otimes$ is the Kronecker product, matrices A_i describes the change of state of machine i and the $\beta_{i,j}$ are scalar factors that indicate if machine i was processing a part of type j in the source state, $i = 1, 2$. Matrix A_i can assume five different values; the first four correspond to the internal transitions of the machine i as described in equation (1) whereas the fifth is an identity matrix (I) having dimension equal to $\bar{P}_i$. The matrix I is used in place of matrix $\bar{P}_i$ whenever machine i is operative but blocked. This is necessary to avoid failures when a machine is not processing items.

For example, if $\mathbf{b}$ is a vector such that both the machines are not blocked, the transitions that lead from the states $(\mathbf{b}, D, U)$ to the states in $(\mathbf{b} + \bar{\mathbf{b}}^+ - \bar{\mathbf{b}}^-, U, D)$ are given by $\beta_{1,j}R_1 \otimes \beta_{2,j}P_2$ because the first machine returns operative (described by R_1) and the second machine fails (described by P_2). If $\mathbf{b}$ was such that the downstream machine could not process then the only possible transition would have been towards the state $(\mathbf{b} + \bar{\mathbf{b}}^+ - \bar{\mathbf{b}}^-, U, U)$ and the corresponding block would be $\beta_{1,j}R_1 \otimes \beta_{2,j}I$. In both cases, since the downstream machine does not modify the buffers, we have that $|\bar{\mathbf{b}}^-| = 0$ whereas the jth component of $\bar{\mathbf{b}}^+$ is equal to one.

The terms $\beta_{1,j}$ and $\beta_{2,j}$ are policy-dependent and are used to enable only those transitions that are permitted by the policy of the machine. For *PB*, $\beta_{i,j} = \nu_i(j, \mathbf{b})$. For *PR*, $\beta_{i,j} = 1$ only if all the buffers with a higher priority than j are full (upstream) or empty (downstream); $\beta_{i,j} = 0$ otherwise. For *FS*, we need to consider the last item processed by the machine; assume x to be the last item that has been processed by machine i then $\beta_{i,j} = 1$ only if $j = f_i(\mathbf{b}, x)$ and $\beta_{i,j} = 0$ otherwise.

Once the DTMC has been defined, the steady state distribution of the above DTMC can be calculated by efficient numerical techniques by exploiting the fact that the process is quasi-birth-death.

Computation of the initial vector The next step consists in computing the initial condition for the computation of the lead time a.k.a. the distribution of the lead time in $h = 0$. This corresponds in catching the moment in which a new part of type j is placed on the buffer.

We force the DTMC in making an additional jump by using only those transitions that insert a new part on buffer j. Assume Q to be the matrix describing the DTMC of the system and ϕ to be the vector containing the steady steady state distribution of the DTMC. In formula this corresponds to:

$$\phi' = \phi(F^{<cond_1>} Q F^{<cond_2>}) \tag{5}$$

where $F^{<cond_1>}$ and $F^{<cond_2>}$ are filtering matrices whose entries are equal to one on the diagonal only if the logical expressions $< cond_1 >$ and $< cond_2 >$ are true in the corresponding state; otherwise, they are equal to zero. The logical expression $< cond_1 >$ determines the states where a part of type j can be processed; therefore it is policy dependent. Policy *PR* requires only that buffer j is not full a.k.a. $b_j < B_j$. The policy *PR* requires also that all the buffers with an higher priority than j are full whereas *FS* requires that buffer j is the next buffer to be processed. The logical expression $< cond_2 >$ is not policy dependent and is necessary to discriminate from those states in which the machine fails during the processing. As a consequence, the only requirement is that $M_1 = U$.

By definition, the system throughput for type j corresponds to the the sum of entries of ϕ'; thus, $E_j = |\phi'|$. The distribution obtained by applying equation (5) does not sum to one but it can be normalized by diving ϕ' by E_j. Finally, we can define the initial condition of the lead time as $\pi(0) = \phi'/E_j$.

Computation of the lead time The lead time is a duration that starts from the moment in which an item is put on the buffer by the upstream machine and ends in the moment in which the downstream machine removes it from the buffer. In practice, the computation of lead time is done by tracking the items on a buffer. In particular, we must track one item for each possible buffer level. By defining $\pi(0)$, we caught the moment in which a new part entered buffer j.

In order to catch the removal, we need to build a modified DTMC in which the insertion of new parts of type j is inhibited. By doing this, we create an absorbing DTMC where the amount of parts on buffer j can only decrease until it reaches zero. The moment in which it reaches zero coincides with the removal of the tracked part from the system. Assume $\hat{Q}$ to be the absorbing DTMC, we carry on the computation of lead time over time by following the relation $\pi(h) = \pi(h-1)\hat{Q}$. The cumulative distribution of the lead time corresponds to the sum of all the states where buffer j is empty. In formula:

$$P(LT_j \leq h) = \sum_{\forall s \in \mathcal{S}: b_j = 0} \pi(h) \tag{6}$$

The construction of $\hat{Q}$ follows the same rules that have been used to build Q by assuming that $\bar{\mathbf{b}}^+$ is always null when the upstream machine is processing type j.

4. NUMERICAL ILLUSTRATIONS

In this section, we provide numerical illustrations to show how the lead time distribution changes by using different policies. We consider systems where machines have the same efficiency and the same type of policy. Failures and repairs are assumed geometrically distributed with parameters $p = 0.01$ and $r = 0.1$, respectively. These choices have been made to not introduce additional complexity that would have made difficult the comprehension of the results. For the same reason, we also assume that all the type of products undergo deterioration after the same time interval. However, it is important to point out that the method allows the use of more general settings.

All the tests have been computed by using a JAVA prototype on common hardware. The computation of a single experiment did not require more than one second.

The fist experiment takes in consideration machines implementing policy FS and buffers capacities: $B_1 = 5$, $B_2 = 10$ and $B_3 = 15$. Figure 2 depicts the three lead time distributions.

We observe that all the distributions are characterized by two peaks. The peak on the l.h.s. is always at 1 and the peak on r.h.s. is placed in position $3 \times B_j - 1$. Thus, the larger is the buffer the more spread is the distribution. The r.h.s. peak corresponds to the case where a part takes the last slot on the buffer. Since each buffer is served one time every three time units where the downstream machine has been operative, it will be removed after a number of time units that is three times the size of the buffer.

Figure 3 depicts the (partial and total) effective throughput as function of h^* for the multi-product system and for its corresponding single-product system ($B = 30$). We observe that the multi-product system outperforms the single-product system for any value of $h^* \leq 28$. Then the single product provides the highest E^{Eff} until both the curves start to tend asymptotically to the theoretical throughput. Furthermore, we have that $E_3^{Eff} \leq E_2^{Eff} \leq E_1^{Eff}$ for any h^*. This phenomenon can be explained by observing that, when the system is balanced, that the major part of the probability mass of the lead-time distribution is placed between the two peaks. As shown in Angius et al. (2014), the lead-time distribution of a single-product system has a peak in $B-1$. Therefore, the effective throughput of such system is penalized for $h^* < B - 1$.

The second test considers a system having three buffers with capacities $B_1 = B_2 = B_3 = 5$ and machines implementing policy PR with $\mathbf{y}_1 = \mathbf{y}_2 = |3, 2, 1|$. Figure 4 illustrates the lead time of each buffer.

We observe that B_1 is characterized by a lead time having two peaks; the first placed in 1 and the second placed in 4. This results was expected because buffer B_1 have the highest priority; thus, parts of type 1 are processed in isolation from the other types as they were passing through a single-product system having buffer equal to 5. The lead time distribution of types 2 and 3 is instead smooth and extremely heavy-tailed indicating that the totality of these parts will be scrapped with high probability. This is because parts placed on buffers 2 and 3 are processed only when M_1 is down; otherwise, M_1 would put parts of type 1 in the system by forcing M_2 to process type 1. Therefore, when M_1 returns operative, all the parts on buffer 2 and 3 must wait for another failure of M_1 to be removed.

Figure 5 depicts the effective throughput of the system as function of h^* and compares it with the effective throughput of corresponding single-product system ($B = 15$). We notice that the effective throughput of the multi-product system is higher than the one of the single-product system for $h^* \leq 25$. This is because, almost the totality of the system production is covered by the parts of type 1 whose lead time has almost all the probability mass before six time units. Furthermore, we notice that, because the wasting of parts of type 2 and 3, the multi-product system tends to the theoretical throughput very slowly. For sake of readability we truncated the plot after 220 time units where the system is still far from it.

The last experiment takes in consideration policy PB by considering a system having buffer capacities equal to $B_1 = B_2 = B_3 = 5$ and weights equal to $\mathbf{y}_1 = \mathbf{y}_2 = |1, 1, 1|$. Therefore, the overall system throughput is spit equally. Figure 6 depicts the lead time distribution. The distribution is characterized by a high peak in 1 followed by a smooth distribution that becomes almost null around 28. Figure 7 shows that the overall effective throughput of the multi-product system outtakes the one of the corresponding single-product system for $h^* < 14$.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we described a method for the computation of the lead time of multi-product systems composed of two machines and N dedicated buffers. The model allows the use of general Markovian machines and three different policies for the mixing of the products. Numerical illustrations showed that, when products deteriorate fast within the system, the effective throughput of this kind of systems is higher than the one of single-product systems having the efficiency and the same buffer capacity. This suggests that, for small thresholds, the multi-buffer layout can be used also for increasing the effective throughput of a single product by using FB, PR and FS policies instead of a common FIFO on the downstream machine. Experiments showed also that the, in presence of perishable products, the association of static priorities to the buffers lead to a waste of material. In the future, we plan to exploit the generality of the method to investigate more complex scenarios and to validate it by using a real use case. Furthermore, we aim to define new kind of priorities and to deepen the investigation about the inefficiencies of FIFO policy by using Rogiest et al. (2015) as starting point.

REFERENCES

Angius, A., Colledani, M., and Horváth, A. (2014). Lead time distribution in unreliable production lines processing perishable products. In *Proceedings of the 2014 IEEE ETFA 2014, Barcelona*, 1–8.

Angius, A., Colledani, M., and Horvath, A. (2017). Lead-time oriented production control policies in two-machine production lines. *IISE Transactions*. doi: 10.1080/24725854.2017.1417654.

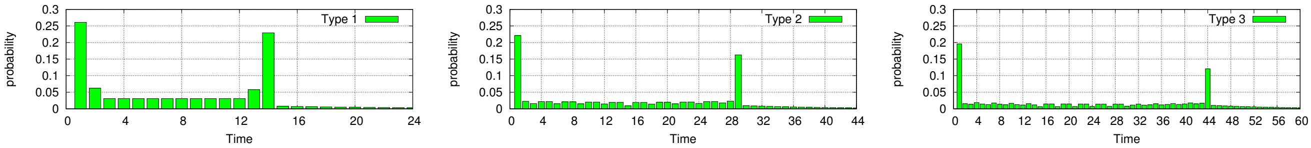

Fig. 2. Lead time distribution of a multi-product system implementing policy FS and $B_1 = 5, B_2 = 10, B_3 = 15$.

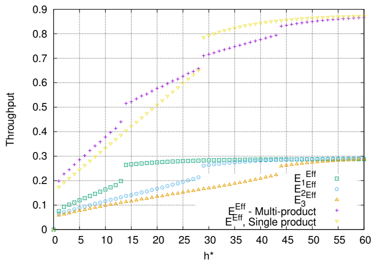

Fig. 3. Effective throughput as function of h^*: comparison between a multi-product system implementing policy FS with $B_1 = 5, B_2 = 10, B_3 = 15$ and the corresponding single-product system ($B = 30$).

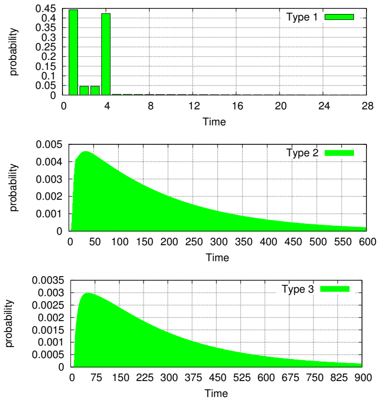

Fig. 4. Lead time distribution of a multi-product system implementing policy PR with $\mathbf{y}_1 = \mathbf{y}_2 = |3, 2, 1|$ and $B_1 = B_2 = B_3 = 5$.

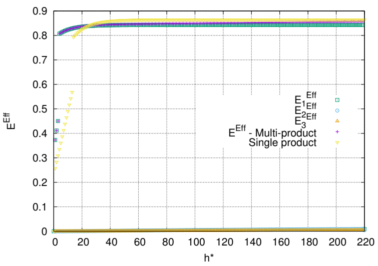

Fig. 5. Effective throughput as function of h^*: comparison between a multi-product system implementing policy PR with $\mathbf{y}_1 = \mathbf{y}_2 = |3, 2, 1|$ and $B_1 = B_2 = B_3 = 5$, and the corresponding single-product system (B=15).

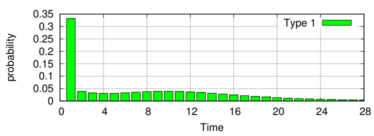

Fig. 6. Lead time distribution of a multi-product system implementing policy PB with $\mathbf{w}_1 = \mathbf{w}_2 = |1, 1, 1|$ and $B_1 = B_2 = B_3 = 5$.

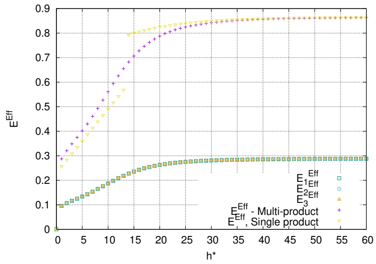

Fig. 7. Effective throughput as function of h^*: comparison between a multi-product system implementing policy PB with $\mathbf{w}_1 = \mathbf{w}_2 = |1, 1, 1|$ and $B_1 = B_2 = B_3 = 5$, and the corresponding single-product system ($B = 15$).

Angius, A., Colledani, M., Horváth, A., and Gershwin, S.B. (2016). Analysis of the Lead Time Distribution in Closed Loop Manufacturing Systems. *IFAC-PapersOnLine*. IFAC MIM'16.

Angius, A., Horváth, A., and Colledani, M. (2015a). Production Control Policies for Desired Lead-Time in Two-machine Production Lines.

Angius, A., Horváth, A., Colledani, M., and Gershwin, S. (2015b). Lead time dependent product deterioration in manufacturing systems with serial, assembly and closed-loop layout. In *SMMSO'15*.

Angius, A., Marcello, C., and Horváth, A. (2015c). Production quality performance in manufacturing systems processing deteriorating products. *CIRP Annals Manufacturing Technology*.

Buchholz, P. and Telek, M. (2010). Stochastic Petri nets with matrix exponentially distributed firing times. *Perf. Eval.*, 67, 1373–1385.

Liberopoulos, G., Kozanidis, G., and Tsarouhas, P.H. (2007). Performance evaluation of an automatic transfer line with wip scrapping during long failures. *Manufacturing and Service Operations*, 9/1, 6283.

Liberopoulos, G. and Tsarouhas, P.H. (2002). Systems analysis speeds up chipita's food processing line. *Interfaces*, 32/3, 6276.

Rogiest, W., Laevens, K., Walraevens, J., and Bruneel, H. (2015). When random-order-of-service outperforms first-come-first-served. *Oper. Res. Lett.*, 43(5), 504–506. doi:10.1016/j.orl.2015.06.010.

Shi, C. and Gershwin, S. (2012). Part waiting time distribution in a two-machine line. In *IFAC Symposium on Information Control Problems in Manufacturing*, 457–462.

Subbaiah, K.V., Rao, S., and Rao, K. (2011). An inventory model for perishable items with alternating rate of production. *International Journal of Advanced Operations Management*, 3/1, 6687.

Wang, J., Hu, Y., and Li, J. (2010). Transient analysis to design buffer capacity in dairy filling and packing production lines. *Journal of Food Engineering*.