DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

An Exact Model to Determine the Lead-Time Distribution of Perishable Goods in a Kanban-Controlled Production System

Ein exaktes Modell zur Bestimmung der Durchlaufzeitverteilung von umgebungsempfindlichen Gütern in einem Kanban-gesteuerten Produktionssystem

Paolo Pagani
Martin Epp
Kai Furmans

Karlsruhe Institute of Technology
Institute for Material Handling and Logistics

In many production and logistics systems the lead-time distribution of the transported products and goods plays a major role for what concerns the system performances. In particular, when the sojourn time in a part of the supply chain exceeds the target range, some quality and rework costs may arise, for example in case of perishable goods. This paper presents an analytical model to exactly determine the lead-time distribution in closed loop systems and to compute the correspondent costs by means of discrete-time Markov chains.

[Keywords: Discrete-Time, Markov Chain, Kanban, Lean Production, Perishable Goods, Lead-Time, Closed-Loop]

In vielen Produktions- und Logistiksystemen spielt die Durchlaufzeitverteilung der transportierten Waren und Produkte eine wichtige Rolle hinsichtlich der Systemleistung. Insbesondere, wenn die Verweilzeit in einem Teil der Supply Chain eine bestimmte Grenze überschreitet, können Qualitäts- bzw. Nacharbeitskosten entstehen, z.B. im Falle von umgebungsempfindlichen Waren. Dieses Paper stellt ein analytisches Modell vor, das mittels zeitdiskreter Markov-Ketten die Durchlaufzeitverteilung in geschlossenen Netzwerken exakt bestimmt und die dazugehörigen Kosten berechnet.

[Schlüsselwörter: Zeitdiskrete Markov Kette, Kanban, Lean Produktion, Umgebungsempfindliche Güter, Durchlaufzeit, Geschlossenes Netzwerk]

1 Introduction

In many different logistics and production contexts, the lead-time, i.e. the period of time that elapses between two arbitrary checkpoints in the material flow, plays a very important role. That happens not only due to inventory costs, which increase as the mean material lead-time increases, but also due to quality and rework costs, which may arise when perishable goods stays in a certain environment for a too short or too long period of time. In real logistics and production environments, there are plenty of examples. In some cases, the elapsed time between the production and sale of the products is relevant, for instance, when the products have a short date of expiry (e.g., food industry, chemistry, …). In other cases, the lead-time between two intermediate production stages of the production systems may be relevant, if particular prescribed environmental conditions (e.g., temperature, contamination, …) must be ensured for a certain time interval. Examples are thermal treatments for metal and food or drug and food conservation in refrigerated containers. Lastly, in the field of transport logistics, it is important to deliver the sold products on the agreed time interval. For each day of delay, the company incurs increasing quality costs (e.g., online shops …).

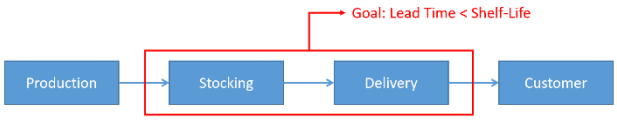

Figure 1. Example of lead-time problematic

In order to keep the quality and rework costs caused by the lead-time as low as possible without strongly affecting the other system performance measures, especially the throughput, lean production strategies such as Kanban systems are used. One of their major goals is to avoid over-production, which could overfill the warehouses and, as a result, increase the mean lead-time [AlA10]. Moreover, lean production techniques make the demand pull the production, i.e. to produce a certain product only when a customer takes it from the supermarket.

2 Motivation

In all those examples, the goal is to reduce the percentage of material that expires the target lead-time range between two checkpoints in material flow, either within one company or in the supply chain. For that reason, the

DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

need for tools that are able to estimate the lead-time distribution arises. By means of them, the system performance can be computed and the correspondent costs estimated. Finally, different possible lean strategies or even different system configurations for the same strategy can be compared and improvement measures can be quantitatively identified and quantitatively supported.

For those reasons, the scientific community has tackled the problem of computing the lead-time distribution in lean systems more and more. In particularly, a large number of contributions analyzed the problem by means of simulation tools ([Gur11], [Abd07]). Nevertheless, simulation models return an inexact solution and that implies that extra statistical tools are required to deal with confidence intervals. Moreover, they perform badly when the optimization problem has a continuous solution space, since most of the algorithms are based on the computation of objective function gradients to determine the improvement direction until the optimum for the objective function has been found. The second major problem of simulation models is that, if the user is interested in a steady-state solution for the problem, long simulation times are required to get a pseudo-steady-state solution that still contains variability.

As a result, many scientific contributions have focused on analytical models, which can return an exact and quick solution. Most of the effort in that direction has been dedicated to estimate the first moment of performance measures (e.g., average throughput, average lead-time, average inventory level …). However, models that are able to compute the second moment of performance measures, such as performance variability and distribution, have been only coarsely investigated. For example, [Ass14] presents a general methodology to analyze the variability of the output of unreliable single machines and small-scale multi-stage production systems modelled as General Markovian structure. Moreover, in [Dol15] 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.

Few papers have applied those concepts to the production of perishable goods. In particular, [Col14] introduces an analytical method to model the dynamics of lead-time dependent quality deterioration of goods in a buffered two-machine line with general Markovian machines. In [Col15] a similar concept is extended for the calculation of lead-time distribution under a large set of different system architectures, including serial lines, closed-loop systems and assembly lines. Although both contribution presents method which can return a quick and exact solution, they model a Markovian behavior of the machines which implies strong model assumption for what concerns the production rate distribution.

Some other research works focused on modelling arbitrary service distributions for the servers with exact analytical models. For instance, [Epp16] introduced a methodology to compute the lead-time distribution in closed queueing networks with an arbitrary topology and arbitrary distribution of the server service times. However, the possibility to model different topologies and service distribution makes the computation effort strongly increase and, for that reason, some decomposition methodologies have been introduced, like for instance in [Epp15], in order to reduce it. Although this approach does not require dealing with confidence intervals or other statistical features and, as a result, it is still suitable for optimization problems, an approximation in the computation is introduced.

This paper presents an exact analytical model to compute the lead-time distribution in a closed-loop network composed by an arbitrary number of servers with arbitrary service time distributions. The algorithm is based on discrete-time Markov Chains as in [Epp16] but, unlike it, no branches and merges are considered in the topology. As a result, the algorithms is simplified and the computation times are drastically lowered. Moreover, it is possible to compute the lead-time not only through the entire system but also through a smaller part of the system.

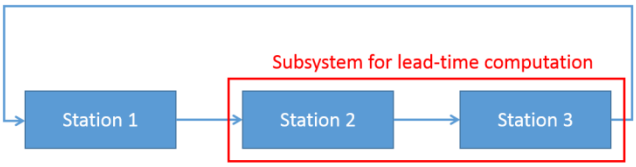

Figure 2. Example of problem solvable with the model presented in this paper

3 MODEL

3.1 CHARACTERISTICS

The system under investigation is closed-loop queueing system in discrete-time domain. It consists of V stations with one server and one waiting room each, as well as $\widehat{K}$ customers that circulate in the system. The routing of the customers to the subsequent station is defined by the station numbering, i.e. that the customers that have just been processed in Station *i* go to station *i+1* and from the last station V they return back to station 1. The system is observed at equally spaced time periods with a length of t_{inc}. It is assumed that the beginning and the end of service as well as the routing to the subsequent station take place immediately prior to the periods. The customers that cannot be processed immediately stay in the waiting room, which has infinite queueing capacities, and are served based on a first come first serve discipline. The service time at station *i* is assumed to be independent from the system state and defined by the random variable B_i,

DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

where $b_{i,j}$ denotes the probability that B_i assumes value j times t_{inc} (see Figure 3) with $j \in \{1,\ldots,J_i\}$.

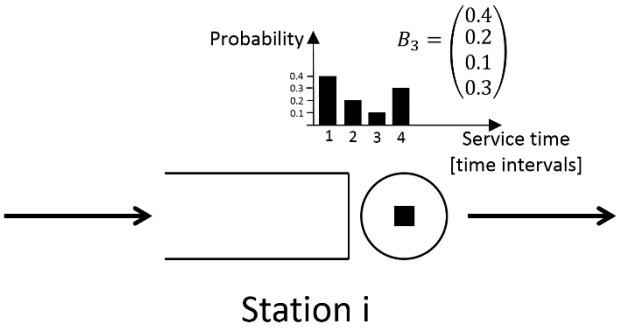

Figure 3. Single station with discrete service time distribution

The model is able to exactly compute all the main first-degree performance measures at each station (i.e. average throughput, queue length and so on) along with the lead-time distribution between two checkpoints in the network under steady-state conditions. The entry checkpoint lays just before the entry in the waiting room of one station, while the exit checkpoint lays just after the server of another station.

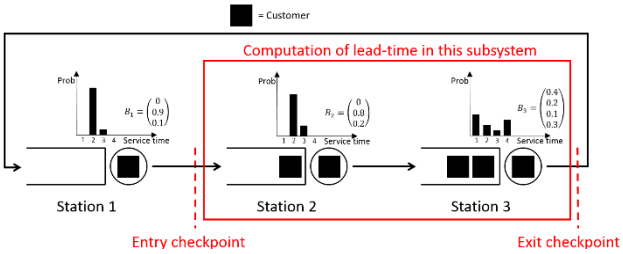

Figure 4. System example with entry and exit checkpoints

3.2 COMPUTATION

The system performance measures are computed by means of a discrete-time Markov Chain that models all the system states and the possible state transitions that can occur at each time step. In order to define a system state, a vector denoted as z with $2 \cdot V$ elements is required.

$$z = (r_1 , r_2 , \ldots , r_V , k_1 , k_2 , \ldots , k_V)$$
$$\text{with } r_i \in \{0,1,\ldots, J_i\},\ k_i \in \{0,1,\ldots,\widehat{K}\},\ i \in \{1,\ldots,V\}$$

For instance, let us consider the system in figure 5. That system will be considered as a reference system for the algorithm explanation. That system is in a state that is defined by $z = (2\ 1\ 1\ 1)$. The black boxes represent the customers (last two numbers of the state vector) while the white numbers on the boxes represent the remaining service times (first two numbers of the state vector).

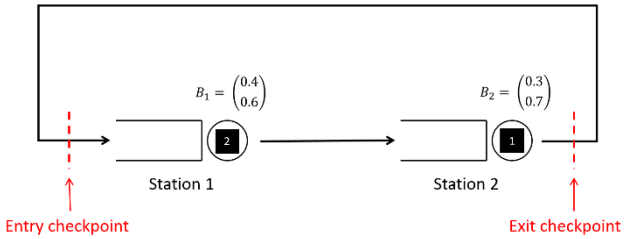

Figure 5. Reference system configuration for the algorithm explanation

3.2.1 STEP 1: STEADY STATE PROBABILITIES

The first step is to compute the steady-state probabilities of all system states. The computation is performed with the traditional discrete-time Markov Chain approaches as soon as the state transition matrix is known.

In order to compute the transition matrix, a simplified methodology in comparison to the one presented in [Epp16] is used. The simplification is because no branches and merges are considered in the model.

The beginning of each new time period reduces the residual time of each customer by 1 and the end of a service (r_i from 1 to 0) triggers the transfer of the customer to the next station. Since the routing of the customers is deterministic, there is only one possible state transition at each time step, as long as no new service start. If a new service starts at station i, the residual time of the new customer in service is given by the random variable B_i. Since B_i can take different values, more than one state transition will be possible. The combination of all possible new service times determines the transition probabilities.

By applying this methodology, the following probabilities are obtained for the system in figure 5:

Table 1. State set and steady-state probabilities for system in figure 5

System state	Probability
(2 1 1 1)	9.6 %
(1 1 1 1)	28.7 %
(1 2 1 1)	14.9 %
(2 2 1 1)	22.3 %
(1 0 2 0)	9.6 %
(0 1 0 2)	14.9 %

In general, N states are found, denoted as z_n and included in the set Z.

DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

3.2.2 STEP 2: COMPUTATION OF LEAD-TIME CONTRIBUTION OF THE SINGLE STATES

Similarly to [Epp16], the total lead-time distribution, denoted as S, is computed as the weighted summation of the contributions S_n of each system state. Even in this case, the methodology has been modified and simplified to be suitable for closed-loop networks without branches and merges. The computation starts when a customer passes through the entry checkpoint and stops when it passes through the exit checkpoint. Since no branches and merges are present in the system, the customers cannot overtake each other and this system property is used to simplify and speed up the computation.

The computation steps are as follows:

1. For each state z_n where a customer goes through the entry checkpoint in the next time interval, the set of the following states Z_n^0 (referring to the period t = 0 of the cycle time computation) is computed along with their probability $p_{n,m}^0$ (with m ∈ {1, … , M_n^0}) and M_n^0 the state number contained in Z_n^0).

2. From the set Z_n^0, the generic subsequent set Z_n^t is generated by taking all the states of the set Z_n^{t-1} and by including the correspondent subsequent states that are generated by it. Each time one or more states are generated from one state of the previous set, the probability must be also split correspondently. Since the customers cannot overtake each other, it does not matter which customers will enter the entry checkpoint after the lead-time computation has already began. As a result, at the beginning of the lead-time computation, it is possible to dispose and not to consider the customers that are outside the limits of the considered subsystem and the ones that exit it in the following time intervals. With this computation algorithm, no extra indexes to track the just entered customer are required as in [Epp16], and the computation will stop when the number of customers in the system becomes zero similarly to [Col14].

3. Whenever the customer number of the subsequent state $z_{n,m}^t$ of the set Z_n^t becomes zero, the correspondent probability $p_{n,m}^t$ is assigned to the t^{th} position of the lead-time contribution S_n.

Figure 6 provides an example for the computation of the lead-time contribution of the customer which is arriving at the first station out of the first state for the system depicted in figure 5.

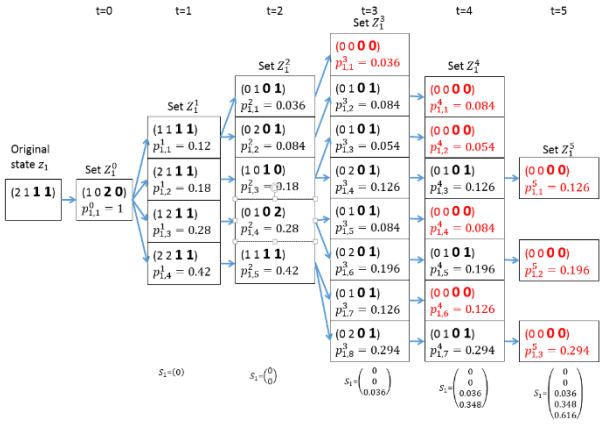

Figure 6. Example of computation for the lead-time contribution

3.2.3 STEP 3: COMPUTATION OF THE TOTAL LEAD-TIME DISTRIBUTION

Once all the lead-time distribution contributions S_n have been computed, they must be summed and weighted by the correspondent steady-state probability with the following formula:

$$S = \frac{\sum_{n=1}^{N} p_n \cdot S_n}{\sum_{n=1}^{N} p_n}$$

In this case, the following lead-time distribution is obtained:

$$S = \begin{pmatrix} 0 \\ 0.02952 \\ 0.29184 \\ 0.56776 \\ 0.11088 \end{pmatrix}$$

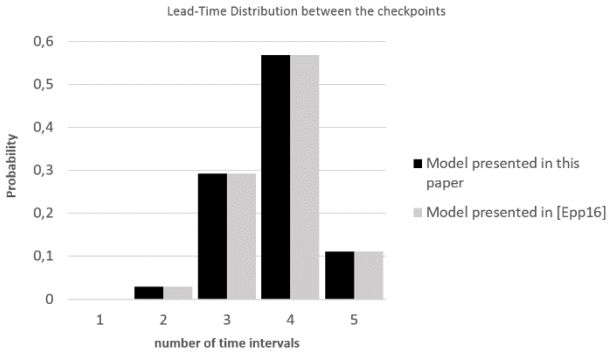

Figure 7. Lead-time distribution for the system depicted in figure 5

Figure 7 presents graphically the results of the lead-time computation. Those results have been also verified by comparing them with the ones obtained with the method presented in [Epp16].

DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

4 USE CASE

4.1 PROBLEM DESCRIPTION

In this chapter a typical industrial problem from the automotive industry, where the model can be effectively used, is presented.

Particularly, the investigated system consists of a production plant where several products made of cast iron are produced with a 3-shift daily schedule. The raw parts are received from the foundry and are worked by means of lathes (station 1: turning). After the chip removal, they are partially painted and completely immersed in an oil bath (station 2: painting) to provide extra rusting protection until they are sold to the final customer. Finally, they are stocked in the final supermarket (station 3: supermarket). In all stations, a FIFO policy is applied and every station has some intermediate buffer areas.

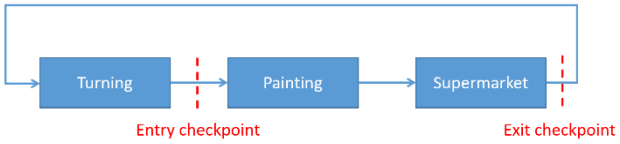

Figure 8. Representation of the Kanban cycle in the use case

The disks circulate in the system in batches which are physically represented by standardized containers and it is assumed that each of them contains 50 parts on average. The WIP is controlled by means of virtual Kanbans that are assigned to each container. As soon as a customer withdraws a container from the supermarket, the Kanban is virtually detached from it and a new batch of products is allowed to enter the system. Figure 8 represents a schematic representation of the Kanban loop.

The relevant system performance measures in this case are:

- the stock-out costs, which the company incurs when a final product is required but not available in the final supermarket
- the WIP costs, which are proportional to the average amount of material circulating in the system
- the quality costs, which occur when the target lead-time is exceeded.

In the following paragraphs, the total relevant costs, i.e. the summation of all three costs, will be evaluated by means of an objective function and the optimal number of Kanbans will be analytically computed.

4.2 MODELLING

Since the intermediate buffers and the supermarket are big enough to assume an infinite capacity, the three above-mentioned stations can be modelled like the station depicted in figure 3, and connected to each other as shown in figure 8. Since the rusting process of the turned surfaces starts after the turning station, the entry checkpoint will be placed immediately after that station. Due to the fact that the supermarket is the last station in the company that the products visit, the exit checkpoint is located after it.

Moreover, since the model just considers the Kanban flow, it is assumed that the first station is never in starvation for raw parts.

Since the turning and painting machines are dedicated for each single product, it is possible to consider the Kanban loop of each product separately.

The Kanban number for the considered product is 15 in the default configuration and the service time distributions are given as in figure 9.

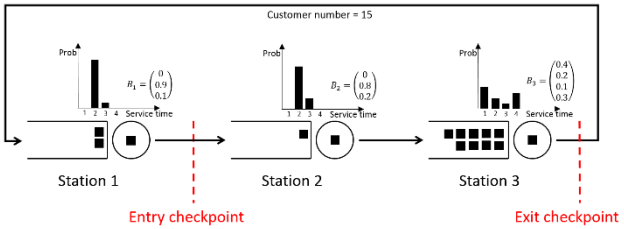

Figure 9. Modelling of the uses case

In particular, the servers of station 1 and 2 model the lathe and the painting plan which can be considered as two clocked stations that process one container every 2 time intervals (corresponding to 20 minutes), if no problem occurs and 3 time intervals otherwise. The server of station 3 represents a turbulent demand with a generic distribution. Finally, each time the demand is not satisfied (WIP at station 3 equal to 0), the demand is considered as lost. The company sets internally a target lead-time of 1 day (corresponding to 72 time units), which assures that the products arrive without rust at the customer company.

4.3 PERFORMANCE EVALUATION

Since the first goal is to evaluate the overall costs, an objective function is defined as follows:

$$OF = C_{lostSales} + C_{WIP} + C_{quality} =$$

$$= P_{stock-out} \cdot D \cdot NP + WIP \cdot C_u \cdot C_c + P_{(LT>\widehat{LT})} \cdot TH \cdot p_u \cdot C_f$$

$P_{stock-out}$ = probability that no customers are in station 3. It corresponds to the percentage of the lost demand.

$\overline{B_3}$ = average service time of station 3 $[2.3 \frac{time\ intervals}{container}]$

$D = \frac{1}{\overline{B_3}}$ = average demand rate $[\frac{1}{2.3} \frac{containers}{time\ interval}]$

NP = net profit $[500 \frac{€}{container}]$

DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

WIP = work in progress in the system, which also corresponds to the total number of Kanbans [containers]

C_u = unitary cost [2000 $\frac{€}{container}$]

C_c = cost of capital [0.0000044 $\frac{€/_{time\ interval}}{€}$]. It is computed by considering an annual cost of capital of 8% and time intervals of a length equal to 20 minutes, as given in the previous paragraph.

$P_{(LT>\widehat{LT})}$ = probability that the lead-time exceeds the target lead-time

$\widehat{LT}$ = target lead time [*72 time intervals (1 day)]*

TH = system throughput [$\frac{container}{time\ interval}$]

p_u = unitary price [2500$\frac{€}{container}$]

C_f = complain factor [200%]. Percentage of the unitary price that the company incurs in case of quality problems.

With the given parameters, following results are obtained:

Table 2. Performance measures of the use case with the default number of Kanbans

Stock-out probability (Lost sales)	0.2%
WIP	15 containers
Probability of exceeding the target LT	0%
TH	0.434 $\frac{containers}{time\ interval}$
Lost sales costs	7770 $\frac{€}{year}$
WIP costs	2376 $\frac{€}{year}$
Quality costs	0 $\frac{€}{year}$
Total costs	10146 $\frac{€}{year}$

The results suggest that the company incurs much more costs due to the lost sales than for WIP and quality costs. As a result, a slightly higher number of Kanban could probably help the company to lower the total costs but for a rigorous and quantitative estimation, further computations are required.

4.4 Optimization

If the number of Kanban in the system is increased, it influences the parameter of the objective function as follows:

- $P_{stock-out}$ decreases because more finished products will be preventively stocked on the shelves of the supermarket (see figure 10).
- WIP will proportionally increase (see figure 11).
- $P_{(LT>\widehat{LT})}$ increases because the average queue length at each station increases (see figure 12).
- TH increases because the utilization of the servers increases (see figure 13).

The following graphs can be plotted for a number of Kanban between 6 and 30:

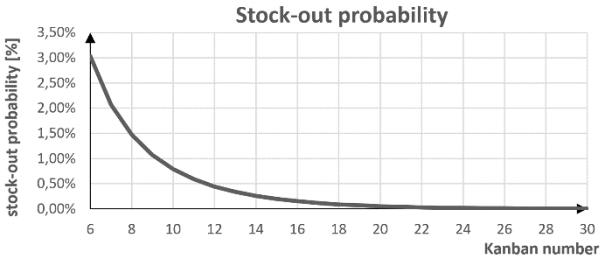

Figure 10. Stock-out probability as a function of the Kanban number

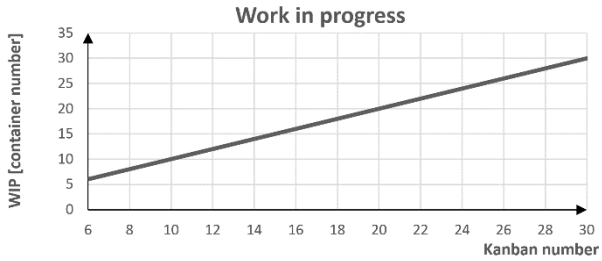

Figure 11. WIP as a function of the Kanban number

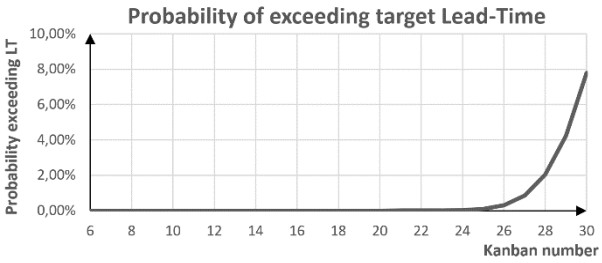

Figure 12. Probability of exceeding the target LT as function of the Kanban number

DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

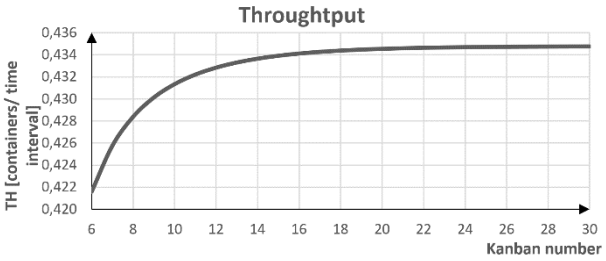

Figure 13. Throughput as a function of the Kanban number

If the performance parameters are converted in costs, the trends plotted in figure 14 can be observed. Particularly, the WIP costs increase slowly and proportionally the number of Kanban, since they are directly linked. On the other hand, the lost sales costs decrease. Lastly, the quality costs are equal to zero, if the Kanban number is smaller than 19, since it is impossible to exceed the target lead-time. On the contrary, if it is greater or equal to 19, quality costs arise and increase quickly.

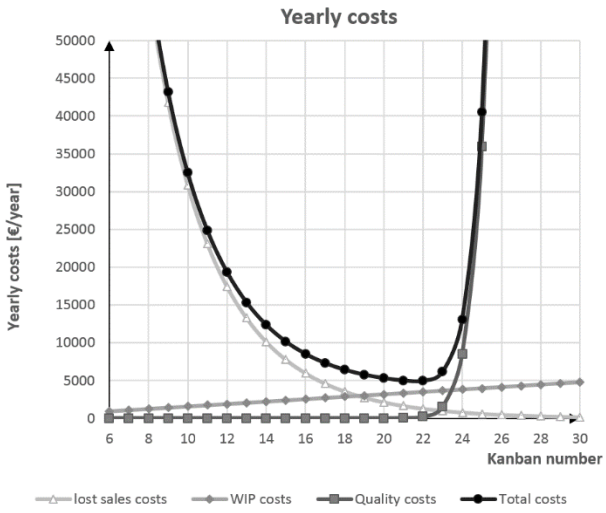

Figure 14. Yearly costs as function of the Kanban number

The optimum is found for a Kanban number equal to 22, where the best compromise between quality, lost sales and WIP costs is achieved. The yearly cost saving per product that the company would achieve by just changing the Kanban number for the considered product family is 5200 $\frac{€}{year}$ that, extended to all other products could bring a significant overall cost saving.

As shown in figure 15, the computation time (processor Intel Core i7-5600U CPU 2.60GHz) also rapidly increases by increasing the Kanban number. The reason is that the number of possible system state increases exponentially with the number of customers in the system and that, since the queue lengths increase, the mean number of time intervals to empty the system during the computation of the lead-time distribution contribution increases.

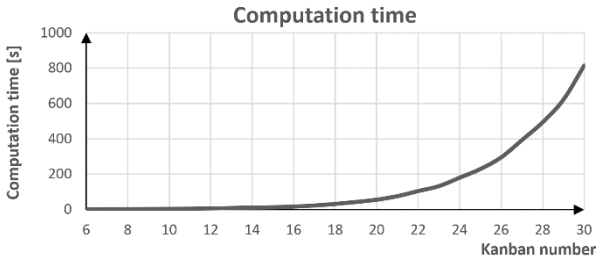

Figure 15. Computation time as a function of the Kanban number

5 CONCLUSIONS

The work has presented an exact method to determine the lead-time distribution in any closed-loop queueing system with general discrete service times. In comparison to the already existing works in the literature, the model is able to handle any discrete service time distribution at each station. Furthermore, we are able to choose a smaller subset of stations, where the lead-time will be computed. The algorithm returns an exact solution, which is particularly helpful in optimization problems and in problems, where the distribution must be determined accurately, e.g., percentage of parts exceeding the target lead-time.

Moreover, the application to the real case has shown how, in a real industrial context, the model can be used to quantitatively support decisions, e.g., the choice of the proper Kanban number for each product. Particularly, it can be used to show how the trade-off between lost sales, work in progress and quality costs influence the optimal Kanban number.

LITERATURE

[Abd07] Abdulmalek, Fawaz A.; Rajgopal, Jayant: *Analyzing the benefits of lean manufacturing and value stream mapping via simulation: A process sector case study*. International Journal of Production Economics, Special Section on Building Core-Competence through Operational Excellence, 2007.

[AlA10] Al-Araidah, Omar; Momani, Amer; Khasawneh, Mohammad; Momani, Mohammed: *Lead-Time Reduction Utilizing Lean Tools Applied to Healthcare: The Inpatient Pharmacy at a Local Hospital*. Journal for Healthcare Quality, Volume 32, Issue 1, 2010.

DOI: 10.2195/lj_Proc_pagani_en_201610_01
URN: urn:nbn:de:0009-14-44594

[Ass14] Assaf, Ramiz; Colledani, Marcello; Matta, Andrea: *Analytical evaluation of the output variability in production systems with general Markovian structure.* OR Spectrum, Volume 36, Issue 3, 2014.

[Col14] M. Colledani; A. Angius; A. Horvath: *Lead time distribution in unreliable production lines processing perishable products.* Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, 2014.

[Dol15] Dolgui, Alexandre; Sasiadek, Jurek; Zaremba, Marek; Angius, Alessio; Colledani, Marcello; Horváth, András: *Lead-time oriented production control policies in two-machine production lines*. 15th IFAC Symposium oninformation Control Problems in Manufacturing, Volume 48, issue 3, 2015.

[Epp16] Epp, Martin; Pagani, Paolo; Stoll, Judith; Scherer, Sebastian; Rohlehr, Carsten; Furmans, Kai: *Performance evaluation of closed-loop logistics systems with generally distributed service times*. Proceedings of the Karlsruhe Service Summit Research Workshop, Karlsruhe, Germany, 2016.

[Epp15] Epp, Martin; Stoll, Judith; Scherer, Sebastian; Pagani, Paolo; Furmans, Kai: *A decomposition approach for the calculation of the cycle time distribution of closed queueing systems*. 10th conference on Stochastic Models for Manufacturing and Service Operations SMMSO, Volos, Greece, 2015.

[Gur11] Gurumurthy, Anand; Kodali, Rambabu.: *Design of lean manufacturing systems using value stream mapping with simulation.* Journal of Manufacturing Technology Management, Volume 22, Issue 4, pages 444 – 473, 2011.

M. Sc. Paolo Pagani, Research Associate at IFL Institute for Material Handling and Logistics, Mechanical Engineering Faculty, KIT Karlsruhe Institute of Technology.

Dipl. –Ing. Martin Epp, Research Associate at IFL Institute for Material Handling and Logistics, Mechanical Engineering Faculty, KIT Karlsruhe Institute of Technology.

Prof. Dr. –Ing. Kai Furmans, Head of IFL Institute for Material Handling and Logistics, Mechanical Engineering Faculty, KIT Karlsruhe Institute of Technology.

Address: IFL Institute for Material Handling and Logistics, KIT Karlsruhe Institute of Technology, Gotthard-Franz-Straße 8, 76131 Karlsruhe, Germany, Phone: +49 721 608-48640, E-Mail: paolo.pagani@kit.de