

PRODUCTION SYSTEMS FLOW MODELLING USING DECOMPOSITION METHOD AND REQUIRED BUFFERS

Gingu (Boteanu), E. I.^{*}; Zapciu, M.^{**} & Cavalieri, S.^{***}

^{*} University POLITEHNICA of Bucharest, Faculty of Engineering and Management of Technological Systems, Machines and Production Systems Department, Splaiul Independentei nr. 313, Bucharest, Romania

^{**} Academy of Romanian Scientists, Splaiul Independentei nr. 54, Bucharest, Romania

^{***} CELS Research Group Department of Management, Information and Production Engineering, University of Bergamo, Viale Marconi 5, 24044 Dalmine, Italy

E-Mail: iuliana_boteanu@yahoo.com, miron.zapciu@upb.ro, sergio.cavalieri@unibg.it

Abstract

This paper aims to provide an introduction to the analytical modelling and the discrete-events simulation for a manufacturing system. A general method of constructing a Markov chain and the simulation with discrete events using Delmia Quest software, are presented. In the present paper, those interested can find a real case study: modelling and simulation of the headrest support manufacturing line. The scientific contribution of this paper consists in the development of a theoretical model implemented in a real case study. The theoretical model estimates the production rate of the line and then, using this model we can establish how many parts in the buffers are required in order to achieve a production rate of X parts per minute. In the first phase, the manufacturing line is analysed, then the Markov chains for subsystems with two machines and a buffer are constructed. As the Markov chains need more time to provide solutions in queues theory, they can be used only for small production lines. To find the production rate for all the system, it is necessary to apply the Markov chains and the decomposition method. Those methods are implemented in the C/C++ programming. To validate the analytical model, the discrete-event simulation with Delmia Quest software is used. The article presents some comparisons between modelling with Markov chains and simulation with Delmia Quest software and provides a method of optimization of the buffers for the case study.

(Received in April 2016, accepted in January 2017. This paper was with the authors 4 months for 2 revisions.)

Key Words: Modelling, Markov Chain, Decomposition Method, Simulation, Buffers Optimization

1. INTRODUCTION AND RELATED WORK

Being given the developing product variety, the intensifying division of labour, and the growing size of manufacturing companies, a clear representation of the logistic behaviour of manufacturing process becomes increasingly important. Modelling and simulation can give detailed insights into the behaviour and performance of planned production systems.

Many companies want to know the performance of linear flow lines with stochastic processing times and the determination of the optimum buffer configuration. The focus of this work is on the practical implementation of an analytical approach to a real case study. The analytical modelling method with Markov chains and the Delmia Quest software represent tools for this practical implementation. Practically, the mathematical theory of production systems analysis represents the queuing theory; production lines can be modelled as queuing networks and solved by means of Markov chains analysis. This analysis works only for solving small production lines, because, as long as the number of machines and the capacities of the buffers increase, the results of states (idle, busy and blocked) explode and it is very difficult to handle it analytically. For larger production lines, there is another method – “decomposition method” – which can be used.

Many researchers have been studying this subject. The exact solution of small systems was initiated, among others, by Hunt [1], followed by Buzacott [2], Gershwin and Berman [3]

and Gershwin and Schick [4]. Approximation methods have been developed to overcome this major limitation (small systems) of the Markovian analysis technique. Such a method is *the decomposition method*, originally introduced by Gershwin [5] and further developed by Dallery et al. [6]. Gershwin developed an approximation method that decomposes a K -stage line into $K-1$ two-stage line. Dallery improved the equations of Gershwin in order to make the algorithm faster and to avoid possible convergence problems. The development of approaches for the design problem of allocating buffer in lines and the optimization of stochastic manufacturing systems is presented in the articles of Dallery and Gershwin [7], Papadopoulos [8], Tempelmeier [9], etc. The authors offered solutions for performance analysis of flow production systems, first of all for two-three machines with finite buffers and then, they used recursive matrix and geometric matrix for larger lines.

In this research work, we analyse a real production line consisting of a number of 8 machines, displayed in series, with a finite buffer between each pair of machines. The main objectives of this research are:

- Modelling of a manufacturing line using Markov chains and decomposition method,
- Simulation of the manufacturing line using Delmia Quest software,
- Results (analytical modelling versus discrete event simulation),
- Buffer optimization: How many stocks (parts) in the buffers are required in order to achieve a production rate of X parts per minute? (In accordance with market demand).

The contribution consists in achieving an analytical model using probabilistic states of the system (busy, idle and blocked) and then this model is implemented in a real case study. The method based on Markov chains analysis with probabilistic states and decomposition method is developed to estimate the production rate of a flow line with finite buffers. The position and size of buffers – which are the storage areas between machines in a production system – can have a significant influence on the performance of a production system.

2. THE MODELLING METHODOLOGY FOR A FLOW LINE

Models in a manufacturing line consist of machines and the buffers between them. The Processing Time (TCY), Mean Time Between Failures ($MTBF$) and Mean Time To Repair ($MTTR$) define the Machines. The Buffers are defined by their size. If there is one upstream product, the layout of the line is always the same: Machine; Buffer; Machine; Buffer; Machine ... It is assumed, that the first machine is never starved and the last machine is never blocked and machine failures are exponentially distributed [10, 11].

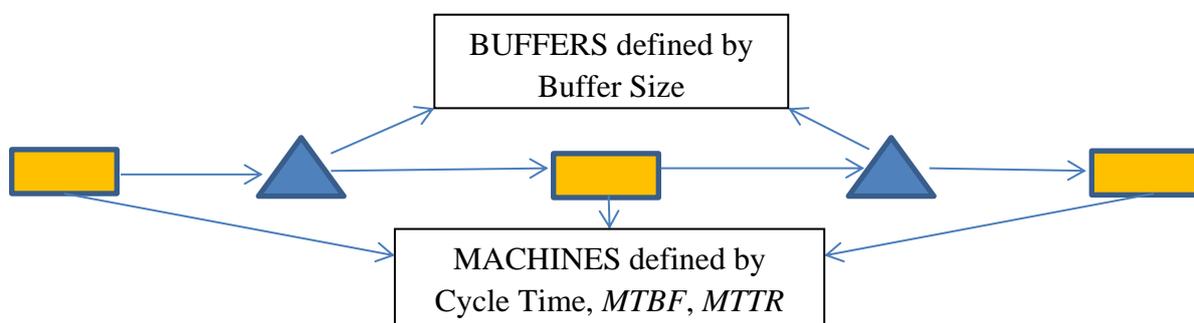


Figure 1: Parameters of machines and buffers.

The analytical modelling methods are based on the stochastic analysis and analytical results of queuing theory of a manufacturing system.

In this paper, the following notations are used:

- $P[. | .]$ – conditional probability,

- $X(t)$ – random variable indexed by the parameter $t \in (-\infty, +\infty)$,
- $X = \{x_1, \dots, x_n\}$ – domain of the variable; x_i, x_j – states,
- dt – an “infinitely small” amount of time,
- τ_{ij} – transition rate from state x_i to x_j ,
- n – vertices in the graph,
- λ – failure rate; μ – repair rate,
- w – working; b – busy; i – idle – the states of a machine,
- $\pi(t)$ – probability distribution.

A *stochastic process* is a random variable indexed by a parameter; we denote it by $X(t)$. A *Markov chain* is a stochastic process where the domain of the variable is a countable set and the following relation is satisfied:

$$P[X(t_k) = x_k | X(t_{k-1}) = x_{k-1}, \dots, X(t_1) = x_1] = P[X(t_k) = x_k | X(t_{k-1}) = x_{k-1}] \quad (1)$$

$\forall k, \forall t_1 \leq \dots \leq t_k$ and $\forall x_1, \dots, x_k$ in the domain of the variable [12].

The property (1) simply means that the knowledge of the values taken by the variable at some times in the past gives us the same information on the future behaviour. In other words, the last measure contains all the information on the past. This property is called the *no memory* property of Markov chains. We will assume that the domain of the variable is finite, that is the variable takes values in the set:

$$X = \{x_1, \dots, x_n\}. \quad (2)$$

The “*no memory property*” says that when we are in some state x_i at time t , the probability to travel to another state x_j in the very near future can be written as:

$$P[X(t + dt) = x_j | X(t) = x_i] = \tau_{ij}(t)dt, \quad (3)$$

where,

$$\tau_{ij}(t) = \tau_{ij}, \quad (4)$$

It is important to notice that transition rates have the dimension of the inverse of time. The value taken by the variable at time t gets the name of *state* since what is to happen depend only on this state. Usually, we will apply the following states: idle, busy, blocked. For the state x_i , we define the *outgoing rate* as the sum of all transition rates from x_i to all x_j 's where $i \neq j$ [12, 13].

To a finite Markov chain, one can associate a graph as follows. The graph has n vertices, each one corresponding to a state, and there is an edge (x_i, x_j) , if $\tau_{ij} \neq 0$. For example, we consider a machine with two states: *up* and *down*. When the machine is up, it works and may have failures; the breakdowns appear with rate λ , called the failure rate. When the machine is down, we repair it with a repair rate μ . [13].

The graph of the Markov chain is as follows:

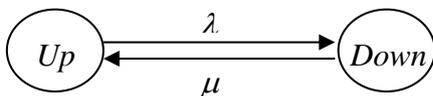


Figure 2: Graph of the Markov chain of the two-state machines.

Flow lines comprising only two machines (Fig. 3) or three machines (Fig. 4) can be evaluated with exact methods under certain conditions.

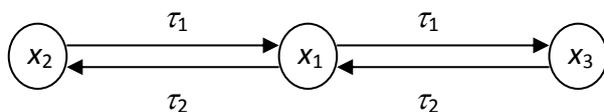


Figure 3: Markov model with stock 0 (2 machines).

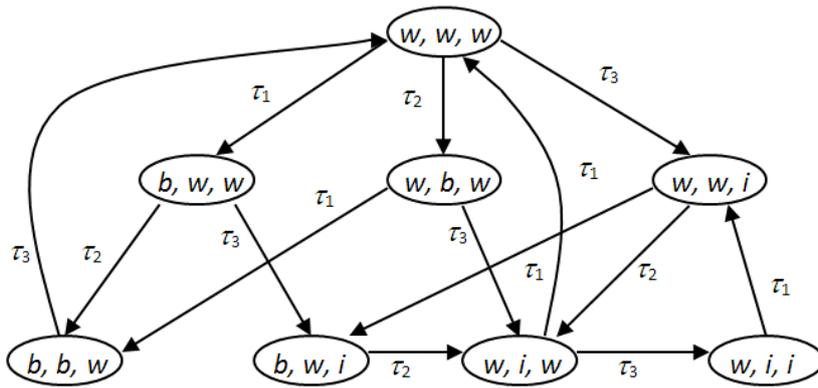


Figure 4: Markov model with stock 0 (3 machines).

Nevertheless, for larger flow lines this method is not available. For these systems the decomposition method is applied.

Basically, these methods decompose a line consisting of L machines into $(L-1)$ subsystems consisting of only two machines, as shown in Fig. 5. These two-machine subsystems are analysed in isolation.

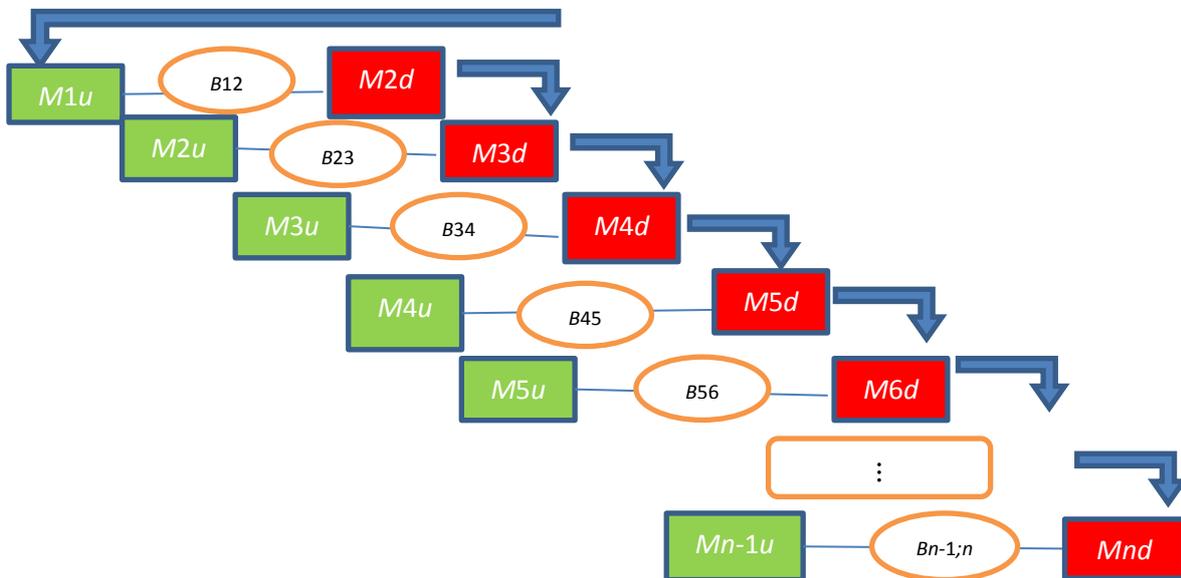


Figure 5: Machine level decomposition.

Before analysing the two-machine subsystems, it is required to set the following assumptions:

- The first machine is never starved and the last machine is never blocked,
- The cycle times are exponentially-distributed,
- The buffer sizes are finite,
- No breakdowns [14, 15].

Generally, the most difficult part is in the modelling process itself. First, the Markov states must be identified. To do this, the suggestion is to list the physical stable situations of the system and for each of them ask the questions:

- Can this situation be represented by a state?
- In other words, does the future only depend on the fact we are in this situation?

If the answer is clearly negative, the situation must be decomposed in several states, like communicating states, two states x_i and x_j , which are such that

$$P[X(t') = x_j | X(t) = x_i] > 0, \text{ for } t' > t \text{ and} \tag{5}$$

$$P[X(t') = x_i | X(t) = x_j] > 0, \text{ for } t' > t \tag{6}$$

This means that starting from one of them we have a non-zero probability to be in the other one in the future.

In applications, it is important to know the Markov chain and the stationary probabilities, by using the *balance equations*:

$$\sum_{\substack{i \in I \\ j \in J}} \pi_i \tau_{ij} = \sum_{\substack{k \in I \\ l \in J}} \pi_l \tau_{lk} \tag{7}$$

where, $\pi_i(t) = P[X(t) = x_i]$ and the *normalising equation*:

$$\sum_{i=1}^n \pi_i = 1 \tag{8}$$

The boundary conditions of the decomposition method are:

$$\begin{aligned} \tau_1^u &= \tau_1 \\ \tau_{k-1}^d &= \tau_k \end{aligned} \tag{9}$$

The general steps of the decomposition method are the following:

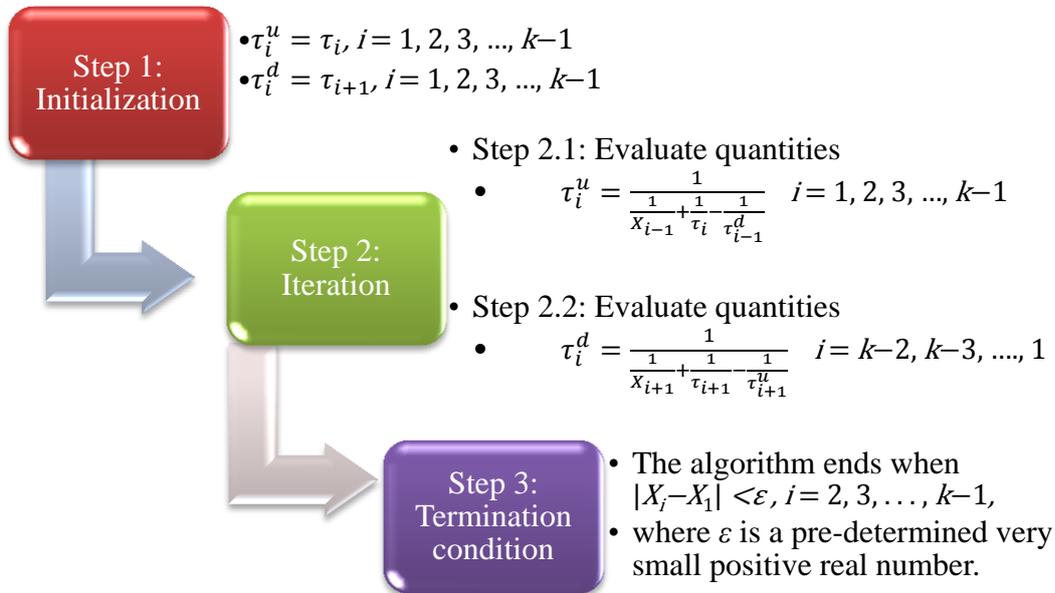


Figure 6: Algorithm to calculate the production rate.

3. CASE STUDY – THE MODELLING METHODOLOGY FOR HEADREST SUPPORT LINE USING MARKOV CHAINS AND SIMULATION METHOD

The case study is related to a manufacturing line of a car headrest support work piece (Fig. 7).



Figure 7: Headrest support work piece.

The operations of the line manufacturing (headrest support) and the cycle time for each machine are shown in Table I.

Table I: Sequence of operations.

| Machine | Operation name | Cycle Time (min) |
|---------|----------------|------------------|
| M_1 | Cutting | 0.077 |
| M_2 | Edge Milling | 0.565 |
| M_3 | Bending I | 0.200 |
| M_4 | Milling B | 0.343 |
| M_5 | Milling A | 0.343 |
| M_6 | Deburring | 0.600 |
| M_7 | Bending II | 0.220 |
| M_8 | Retouch | 0.190 |

It is considered the following line (Fig. 8).

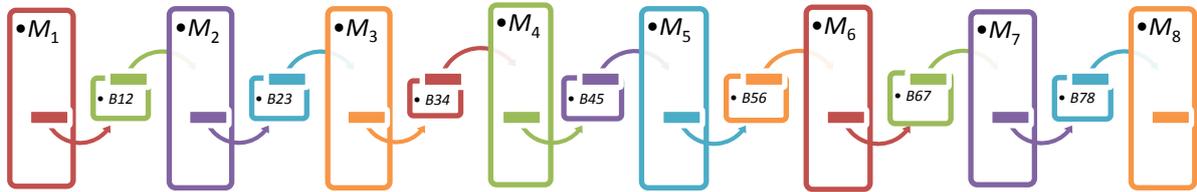


Figure 8: Flow line with buffers.

3.1 Production rate of 2 machines using buffers (stock 1)

By using the decomposition method with two machines and one buffer, several possible issues were identified in order to analyse and optimize flow lines with limited buffers displayed among the machines [14, 15].

Assumptions:

➤ There are always raw parts available before M_1 and when the machining on M_2 is finished the part leaves the system. The machining times on M_1 (resp. M_2) are exponentially distributed with rate τ_1 (resp. τ_2). With a stock place we have one more state corresponding to one part present in stock, while the two machines are working. We denote this state x'_1 .

➤ The states are:

- x_1 : M_1 and M_2 are working;
- x'_1 : M_1 and M_2 are working and work piece is in buffer;
- x_2 : M_1 is working and M_2 is waiting (idle);
- x_3 : M_1 is blocked and M_2 is working.

These situations are indeed states because of the assumption of exponential distribution for the machining times.

It will get the graph of the Markov chain:

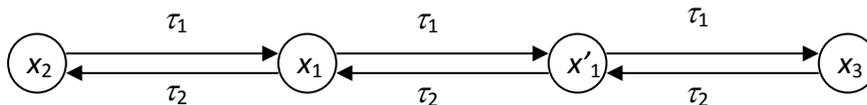


Figure 9: Markov model with stock 1.

The stationary probabilities $\pi_1, \pi_2, \pi_3, \pi_4$ are obtained by writing the balance equations relative to the cuts between x_1 and x_2 , x_1 and x'_1 , x'_1 and x_3 :

$$\sum_{j \in J} \pi_i \tau_{ij} = \sum_{l \in J} \pi_l \tau_{lk} \Rightarrow \pi_1 \tau_2 = \pi_2 \tau_1, \quad \pi_1 \tau_1 = \pi_3 \tau_2, \quad \pi_3 \tau_1 = \pi_4 \tau_2. \quad (10)$$

Then, the normalising equation is used:

$$\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1. \tag{11}$$

For M_1 with the operation Cutting and M_2 with the operation Edge Milling:

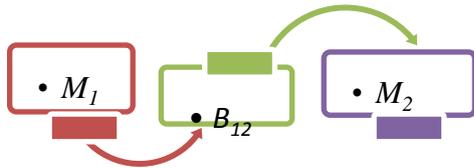


Figure 10: Flow line with 2 machines and 1 buffer.

Production rates:

$$\tau_1 = \frac{6000 \text{ units}}{480 \text{ min}} = 12.5 ; \tau_2 = \frac{3368 \text{ units}}{480 \text{ min}} = 7.01 ; B = 1 \tag{12}$$

The utilisation rate of M_1 is:

$$\pi_1 + \pi_2 + \pi_3 = 0.512591, \tag{13}$$

as the sum of the probabilities of the states where M_1 is working.

The production rate of the system is the production rate of one of the machines, in our case, M_2 , which delivers the finished parts. The production rate of M_2 is the product of its production rate when it works, by the proportion of time M_1 is working, which is: 6.407391 products/minute.

To validate the analytical model, we apply a discrete-event simulation by using Delmia Quest software. Simulation of production lines is a powerful tool in obtaining the performance measures.

Quest is a discrete-event simulation system from DELMIA (Digital Enterprise Lean Manufacturing Interactive Applications) with integrated 3D capabilities. Quest combines a 3D CAD geometry with material flow logics grouped in modules for: sources, buffers, machines, controllers, labours, conveyors, automated guided vehicles (AGVs), kinematics, power and free path systems automated storage and retrieval systems (AS/RS) and sinks. Delmia Quest includes real production variables within a plant layout, such as the machine cycle time, labourers' movement and/or operating speeds. Using these variables the users can analyse the effects on material handling equipment and labour [16-19].

So, after a simulation of 480 minutes using Delmia Quest software, the production rate is: 6.40 products/minute (see Fig. 11).

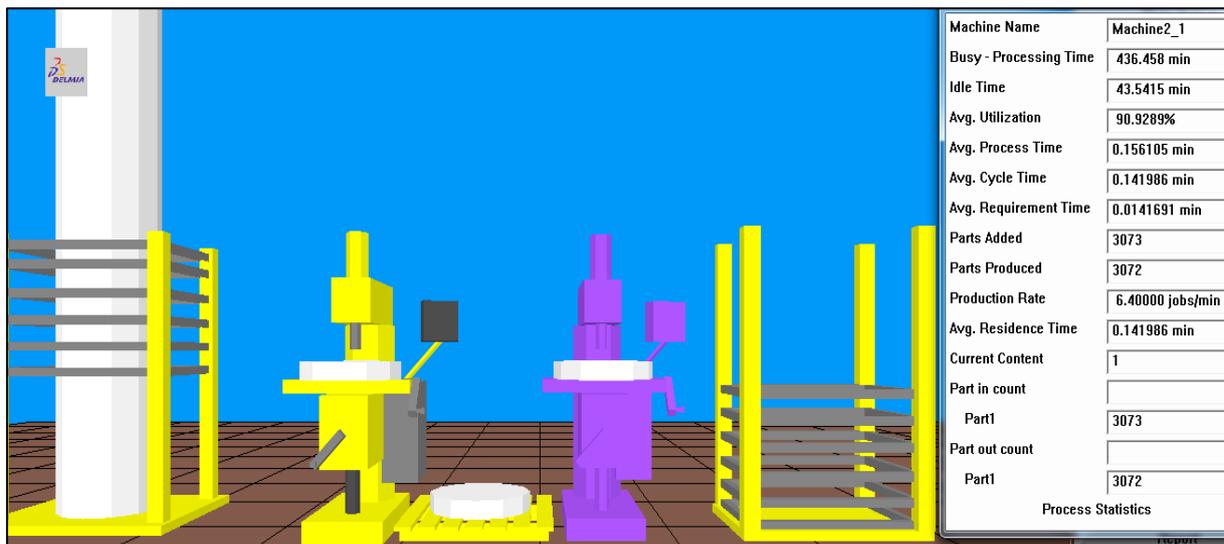


Figure 11: Simulation of a flow line with 2 machines and 1 buffer (stock 1).

3.2 Production rate for headrest support line with 8 machines using buffers (stock 1)

The flow line Headrest support is consisting of $M = 8$ machines and 7 buffers. The system is decomposed into $(M-1) = 7$ subsystems consisting of two machines each and a buffer with 1 piece.

The production rates are:

$$\begin{aligned} \tau_1 &= \frac{6000 \text{ units}}{480 \text{ min}} = 12.5; \quad \tau_2 = \frac{3368 \text{ units}}{480 \text{ min}} = 7.01; \quad \tau_3 = \frac{2400 \text{ units}}{480 \text{ min}} = 5; \\ \tau_4 &= \frac{2798 \text{ units}}{480 \text{ min}} = 5.83; \quad \tau_5 = \frac{2798 \text{ units}}{480 \text{ min}} = 5.83; \quad \tau_6 = \frac{1600 \text{ units}}{480 \text{ min}} = 3.33; \\ \tau_7 &= \frac{2182 \text{ units}}{480 \text{ min}} = 4.55; \quad \tau_8 = \frac{2525 \text{ units}}{480 \text{ min}} = 5.26; \quad B_1=B_2=\dots=B_7= 1 \end{aligned} \tag{14}$$

Given the solution of the subsystem with two machines and a buffer, approximate methods for the large lines were developed. The case study analysed in this paper is a system decomposed into $(M-1) = 7$ subsystems. Each subsystem is analysed with the help of Markov chain method and then, in an iterative procedure, the decomposition method is applied, in order to find the production rate for whole the system.

Implementing the decomposition approach and Markov chains in C++ programming, the following results (Table II, Fig. 12) are obtained.

Table II: Results of production rates.

| Machine | τ^u | τ^d | Work in progress (WIP) |
|---------|----------|----------|------------------------|
| M_1 | 12.5 | - | 1.7584 |
| M_2 | 6.8819 | 2.5407 | 1.5072 |
| M_3 | 4.6782 | 2.6145 | 1.1968 |
| M_4 | 4.7469 | 2.8003 | 1.2118 |
| M_5 | 4.7814 | 2.7884 | 1.2192 |
| M_6 | 2.9658 | 2.7827 | 0.5006 |
| M_7 | 2.7191 | 4.0281 | 0.2827 |
| M_8 | - | 5.26 | - |

The production rate of the flow line headrest support with 8 machines and 7 buffers (stock 1) is 2.5236 products/minute and $WIP = 7.6767$ products/minute.

```

\Program Files (x86)\CodeBlocks\MinGW\Juliana\
k =8
n=1
tau[1]=12.5
tau[2]=7.01
tau[3]=5
tau[4]=5.83
tau[5]=5.83
tau[6]=3.33
tau[7]=4.55
tau[8]=5.26
x=2.5236
    
```

Figure 12: Result of the model implemented in C++ by CodeBlocks.

Using simulation model, a production rate of 2.6625 products/minute is obtained.

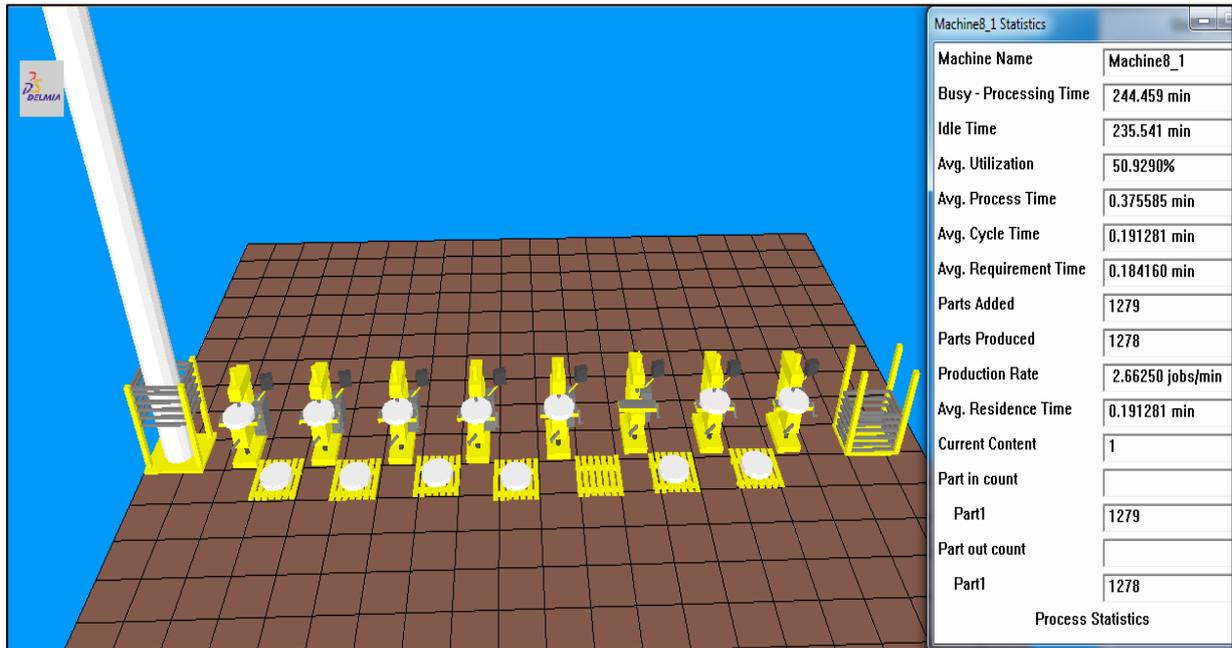


Figure 13: Simulation of headrest support line with 8 machines and 7 buffers (1 stock).

4. RESULTS OF THE CASE STUDY

Applying the algorithm (Fig. 6) to calculate the production rate using Markov chains and having the tool for simulation, we can make some comparisons, as shown in Fig. 14.

This algorithm was coded in C++ and it is used to run the probabilistic states of the system (busy, idle and blocked) and to find a way for solving the linear equations system in order to calculate the production rate. The results have been calculated for the Delmia Quest models using a simulation time of 480 minutes, as a shift.

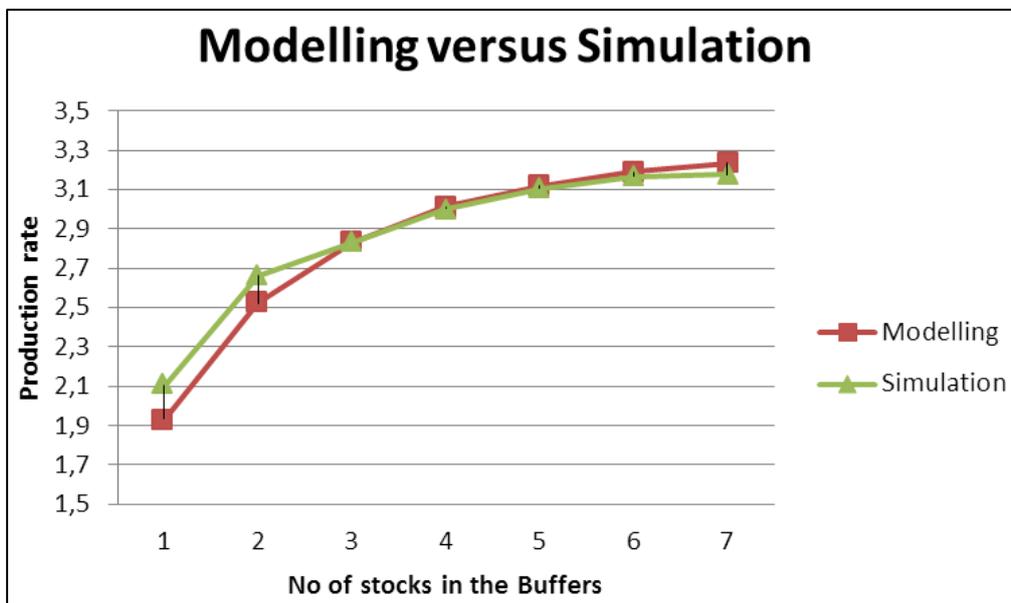


Figure 14: Comparison between modelling using Markov chains and simulation using Delmia Quest.

A comparison between modelling and simulation of the state times – Busy is shown in the following chart (Fig. 15).

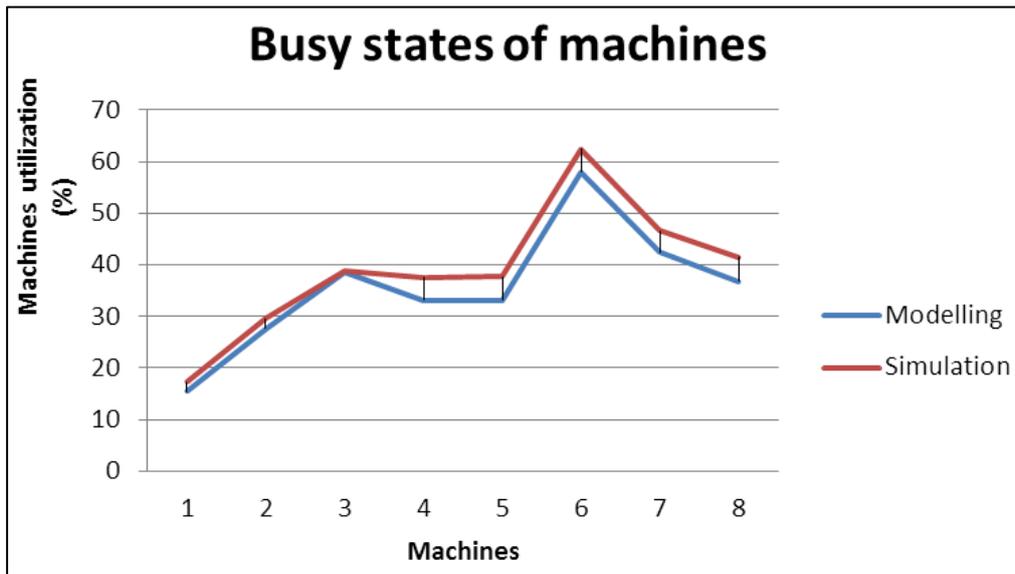


Figure 15: Comparison between modelling and simulation of the state times – Busy.

By comparison, it can be seen in the figures above, that the results for the approximations used in the mentioned models (decomposition approach, Markov chains and Delmia Quest) are much closer than expected.

As random phenomena, like the variability of the cycle time result in a loss of production rate, the machines in an asynchronous flow production system are usually decoupled with the help of buffers [20].

Experience shows that the production rate of many real-life production systems can be increased by buffers relocation.

The main goal of this research is to answer to the following question:

- How many stocks (parts) in the buffers are required in order to achieve a production rate of X parts per minute? (In accordance with market demand.)

The goal of the optimisation is to reach the 1400 products/shift plus a security reserve of 5 %. This sum is up to 1470 pieces/shift, translating into 3.0625 products/minute.

To demonstrate this effect, the model of headrest support line without any buffers is used. Gradually, one buffer after another is added to the system. The position is determined by a manual function of finding the next buffer, which will yield the greatest increase in throughput.

Table III: Production rate.

| Buffer | | Modelling | Simulation |
|--------|---|-----------|------------|
| 0 | Production rate, products/minute | 1.9334 | 2.10417 |
| 1 | | 2.5236 | 2.6625 |
| 2 | | 2.8341 | 2.83125 |
| 3 | | 3.0123 | 3.00208 |
| 4 | | 3.1207 | 3.10417 |
| 5 | | 3.1893 | 3.16667 |

So, it is predicted that the line exceeds the desired production rate of 3.0625 products/minute the first time with a total of 4 stocks in the buffers.

5. CONCLUSION

It is necessary to help the decision-maker get a quick estimated production rate, without recurring to simulation. For that, an approximation method was taken into account and applied to a real production line. It is known that a simulation model takes a lot of time, not to mention the design stage, because many alternative configurations should be considered.

We divided the research work into three parts: simulation, exact analytical methods (Markov chains), and approximate analytical methods (decomposition method). This paper provides a detailed approach of the mathematical modelling in order to get a performance analysis for a real case study – *Headrest support*.

The main contributions of this study results from the answers to the following questions:

- How can an approximate analytical method be used to evaluate the performances of the *Headrest support* flow line?
- How can an approximate analytical method be used to maximize the performances of this flow line by buffer optimisation?

The results of this case study show that the production rate can be increased by buffers relocation, i.e. the buffer optimization requires the evaluation of the system configuration, including the buffer sizes. When we added parts in the buffers, the production rate has increased; this increase must be in accordance with the market demand. For the case study of this paper it is necessary to introduce in each buffer 4 parts in order to predict a production rate of 3.0625 products/ minute.

The analytical model proposed in this article and the algorithm coded in C++ help any industry company to estimate the production rate of the line in order to *synchronize* the production with the market demand. Also the results can be integrated in the Production stage of Product Lifecycle Management (PLM). In this stage, the layout of the machines is achieved, the flows of the parts are established in the manufacturing system, and then can be made a simulation for production optimization. PLM enables companies to accelerate product innovation and to propose the exploration of a specific stabilizing production technique.

Future research effort should be oriented to include the decomposition method and Markov chains in the Maintenance stage of PLM in order to estimate the *MTBF* (Mean Time Between Failures) and *MTTR* (Mean Time To Repair), which can be used to improve the efficiency of the processes in the company.

ACKNOWLEDGEMENT

The work has been funded by the Sectorial Operational Program Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/134398.

REFERENCES

- [1] Hunt, G. C. (1956). Sequential arrays of waiting lines, *Operations Research*, Vol. 4, No. 6, 674-683, doi:[10.1287/opre.4.6.674](https://doi.org/10.1287/opre.4.6.674)
- [2] Buzacott, J. A. (1972). The effect of station breakdowns and random processing times on the capacity of flow lines with in-process storage, *American Institute of Industrial Engineers Transactions*, Vol. 4, No. 4, 308-312, doi:[10.1080/05695557208974867](https://doi.org/10.1080/05695557208974867)
- [3] Gershwin, S. B.; Berman, O. (1981). Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers, *American Institute of Industrial Engineers Transactions*, Vol. 13, No. 1, 2-11, doi:[10.1080/05695558108974530](https://doi.org/10.1080/05695558108974530)
- [4] Gershwin, S. B.; Schick, I. C. (1983). Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers, *Operations Research*, Vol. 31, No. 2, 354-380, doi:[10.1287/opre.31.2.354](https://doi.org/10.1287/opre.31.2.354)

- [5] Gershwin, S. B. (1987). An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking, *Operations Research*, Vol. 35, No. 2, 291-305, doi:[10.1287/opre.35.2.291](https://doi.org/10.1287/opre.35.2.291)
- [6] Dallery, Y.; David, R.; Xie, X.-L. (1988). An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers, *Institute of Industrial Engineers Transactions*, Vol. 20, No. 3, 280-283, doi:[10.1080/07408178808966181](https://doi.org/10.1080/07408178808966181)
- [7] Dallery, Y.; Gershwin, S. B. (1992). Manufacturing flow line systems: a review of models and analytical results, *Queueing Systems*, Vol. 12, No. 1, 3-94, doi:[10.1007/BF01158636](https://doi.org/10.1007/BF01158636)
- [8] Papadopoulos, H. T. (1996). An analytic formula for the mean throughput of K -station production lines with no intermediate buffers, *European Journal of Operational Research*, Vol. 91, No. 3, 481-494, doi:[10.1016/0377-2217\(95\)00113-1](https://doi.org/10.1016/0377-2217(95)00113-1)
- [9] Tempelmeier, H. (2003). Practical considerations in the optimization of flow production systems, *International Journal of Production Research*, Vol. 41, No. 1, 149-170, doi:[10.1080/00207540210161641](https://doi.org/10.1080/00207540210161641)
- [10] Xia, B.; Xi, L.; Zhou, B.; Du, S. (2013). An efficient analytical method for performance evaluation of transfer lines with unreliable machines and finite transfer-delay buffers, *International Journal of Production Research*, Vol. 51, No. 6, 1799-1819, doi:[10.1080/00207543.2012.713137](https://doi.org/10.1080/00207543.2012.713137)
- [11] Diamantidis, A. C.; Papadopoulos, C. T.; Heavey, C. (2007). Approximate analysis of serial flow lines with multiple parallel-machine stations, *Institute of Industrial Engineers Transactions*, Vol. 39, No. 4, 361-375, doi:[10.1080/07408170600838423](https://doi.org/10.1080/07408170600838423)
- [12] Tan, B.; Gershwin, S. B. (2011). Modelling and analysis of Markovian continuous flow systems with a finite buffer, *Annals of Operations Research*, Vol. 182, No. 1, 5-30, doi:[10.1007/s10479-009-0612-6](https://doi.org/10.1007/s10479-009-0612-6)
- [13] Nof, S. Y.; Wilhelm, W. E.; Warnecke, H.-J. (2012). *Industrial Assembly*, Springer-Verlag, New York
- [14] Aziz, A.; Jarrahi, F.; Abdul-Lader, W. (2010). Modeling and performance evaluation of a series-parallel flow line system with finite buffers, *INFOR: Information Systems and Operational Research*, Vol. 48, No. 2, 103-120, doi:[10.3138/infor.48.2.103](https://doi.org/10.3138/infor.48.2.103)
- [15] Lagershausen, S.; Manitz, M.; Tempelmeier, H. (2013). Performance analysis of closed-loop assembly lines with general processing times and finite buffer spaces, *Institute of Industrial Engineers Transactions*, Vol. 45, No. 5, 502-515, doi:[10.1080/0740817X.2012.705450](https://doi.org/10.1080/0740817X.2012.705450)
- [16] Bzymek, Z. M.; Nunez, M. A.; Li, M.; Powers, S. (2008). Simulation of a machining sequence using Delmia/Quest software, *Computer-Aided Design and Applications*, Vol. 5, No. 1-4, 401-411, doi:[10.3722/cadaps.2008.401-411](https://doi.org/10.3722/cadaps.2008.401-411)
- [17] Popescu, D.; Anania, F. D.; Cotet, C. E.; Amza, C. G. (2013). Fully-automated liquid penetrant inspection line simulation model for increasing productivity, *International Journal of Simulation Modelling*, Vol. 12, No. 2, 82-93, doi:[10.2507/IJSIMM12\(2\)2.225](https://doi.org/10.2507/IJSIMM12(2)2.225)
- [18] Mohora, C.; Anania, D.; Calin, O. A., (2009). Simulations strategies using Delmia Quest, *Proceedings of the 20th International DAAAM Symposium*, Vienna, 335-336
- [19] Banks, J.; Carson, J. S.; Nelson, B. L.; Nicol, D. (2010). *Discrete-Event System Simulation*, Prentice-Hall, Upper Saddle River
- [20] Talib Bon, A.; Rahman, A. A.; Bolhassan, E. A.; Nordin, M. A. A. (2013). Optimize assembly production line using line balancing, *Proceedings of the 2nd International Conference on Global Optimization and its Applications*, 140-145