

Jarosław STAŃCZYK¹

MAX-PLUS ALGEBRA AS A TOOL FOR THE MODELLING AND PERFORMANCE ANALYSIS OF MANUFACTURING SYSTEMS

This contribution discusses the usefulness of $(\max, +)$ algebra as a mathematical framework for a class of manufacturing systems. This class can be described as dynamic and asynchronous, where the state transitions are initiated by events that occur at discrete instants of time. An event corresponds to the start or the end of an activity. Such systems are known as discrete event systems (DES). An overview of the concepts of modelling and analysis using the $(\max, +)$ algebra approach to DES has been given. Also, examples of manufacturing systems have been provided to illustrate the potential of this approach. The type of production process used, such as serial line, assembly line, etc., influences the modelling of different basic manufacturing systems. We have also presented the impact of the capacity of interoperable buffers. Based on an analytical model, effectiveness and performance indexes have been evaluated.

Keywords: *discrete event system, max-plus algebra, performance evaluation, manufacturing systems*

1. Introduction

Many phenomena from manufacturing systems, telecommunication networks or transportation systems can be described as so-called discrete event systems (DES), or discrete event dynamic systems (to emphasize their dynamic character). These are systems made by humans, which consist of a finite amount of resources (e.g., processing units, buffers, communications channels, etc.), which are shared between a certain number of processes (e.g., tasks, product types, information packets, etc.). Processes coop-

¹Wroclaw University of Environmental and Life Science, Department of Genetics, ul. Kozuchowska 7, 51-631 Wroclaw, Poland, e-mail address: jaroslaw.stanczyk@upwr.edu.pl

erate or compete to reach a goal or an objective (e.g., final product, transmission, calculation in a distributed computing system, etc.). A DES is a dynamic asynchronous system where the state transitions are initiated by events that occur at discrete instants of time. An event corresponds to the start or the end of an activity. A common feature of such processes is that the start of an activity may require the termination of several other activities. Such systems cannot conveniently be described by differential or difference equations, and naturally exhibit periodic behaviour.

DES dynamics are characterized by two phenomena:

- synchronization – consists of bringing two or more processes into conformity over time,
- concurrency – simultaneous use of the system by several processes, leading to the phenomenon of competition for access to resources.

An introduction to DES has been given, e.g., in [2]. Many frameworks exist to study DES. DES theory can be presently divided into three main approaches:

- logical which considers the occurrence of events or the impossibility of their occurrence and sequences of such events, but does not consider the precise time of these occurrences, i.e., does not consider performance, e.g., automata theory [14] or Petri nets [13],
- deterministic which addresses the issue of performance evaluation (evaluated by the number of events occurring in a given period of time), and that of performance optimization, e.g., timed Petri nets or $(\max, +)$ algebra [1, 6],
- stochastic – used when certain statistical characteristics of the system are known, e.g., Markov chain [9], queueing theory [5] or computer simulation.

However, the most widely used technique to analyse DES is computer simulation. An important drawback of this approach is that it often does not give a real understanding of how parameter changes affect important properties of a system, such as stability, robustness and optimality of system performance. Analytical techniques can provide a much better insight in this respect. Therefore, formal methods are preferred as tools for modelling, analysis and control of DES.

The aim of this paper is to discuss problems of modelling and analysing a certain class of DES using $(\max, +)$ algebra. The considered class of DES is limited to such systems whose models are $(\max, +)$ linear and stationary, i.e., deterministic systems in which there is no possibility of choosing a resource while it is involved in executing a process. Examples of such systems include, but are not limited to, production systems [14], transport [18] and batch chemical processes [11].

The article is organized as follows: Section 2 presents an example of a production system, a valve assembly line, which can be configured in several ways at the planning stage. At the end of this section the research problem is formulated, which is how to choose the best solution from those proposed, in terms of some quantitative criterion. Section 3 contains the basics of $(\max, +)$ algebra, where we will consider only a few aspects of the theory, required for the systems modelling considered. Section 4 gives

some exposition of the modelling of selected production systems using $(\max, +)$ algebra. The analysis of the case study introduced in Section 2 is presented in Section 5. Finally, Section 6 summarizes this paper, presents the results, and outlines the direction of further research.

2. An example manufacturing system as a case study

We will consider an assembly line for producing back flushing control valves [4]. Each valve consists of parts shown in Fig. 1.

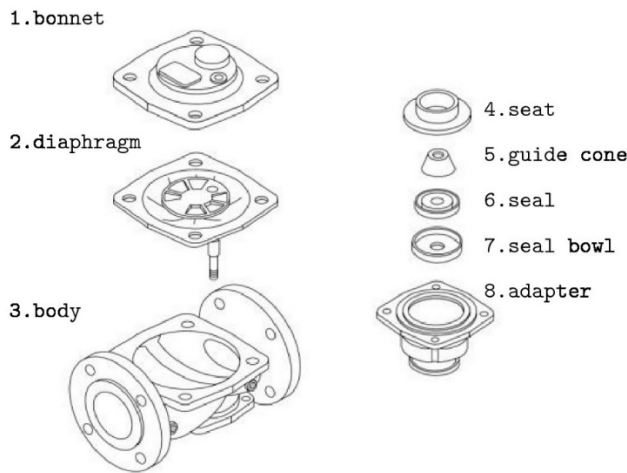


Fig. 1. Valve components (from [4])

The sequence tree is shown in Fig. 2. In this tree, each node represents an independent assembly operation, e.g., in the highlighted part: assembly of part number 3 with part number 4, and next with 5.

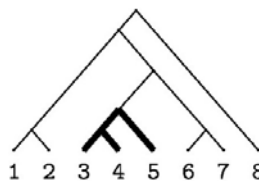


Fig. 2. Assembly sequence tree (after [7])

Translation of the sequence defined by the sequence tree into possible configurations of the actual production line depends on many factors, such as available space,

available staffing, the tools required to perform a given operation, etc. It is accomplished through appropriate production planning techniques. Assume that, at the production planning stage, 3 potential configurations of the assembly line have been obtained. All of them consist of a different number of production units. These configurations are shown in Fig. 3.

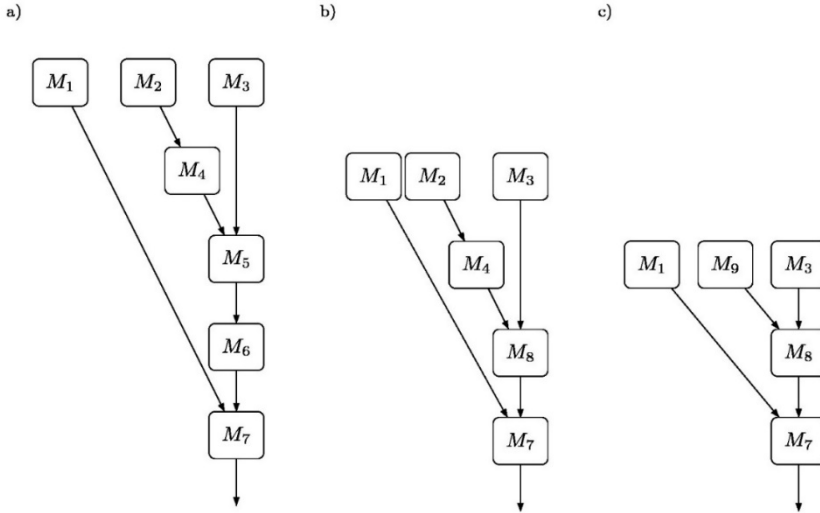


Fig. 3. Three possible assembly line configurations for the sequence tree of Fig. 2

Table 1. List of valve assembly operations

Production unit	Assembly process	Parts	Time units
M_1	assemble bonnet and diaphragm	(1, 2)	$d_1 = 43$
M_2	assemble body and seat	(3, 4)	$d_2 = 20$
M_3	assemble seal and seal bowl	(6, 7)	$d_3 = 15$
M_4	add guide cone to body and seat	((3, 4), 5)	$d_4 = 6$
M_5	assemble body subassembly with seal	(3, 4, 5), (6, 7))	$d_5 = 18$
M_6	add adapter to body and seal	((3, 4, 5, 6, 7), 8)	$d_6 = 5$
M_7	add bonnet and diaphragm to the assembly	((3, 4, 5, 6, 7, 8), (1, 2))	$d_7 = 21$
M_8	assemble body subassembly with seal and then adapter (in place of M_5 and M_6)	((3, 4, 5), (6, 7), 8)	$d_8 = 25$
M_9	assembly body and seat then add guide cone (in place of M_2 and M_4)	(3, 4, 5)	$d_9 = 28$

A description of each operation, together with their execution times, is shown in Table 1. For any configuration, a production unit with the same index performs the same operation with the same operation time. The differences between the configurations are limited to the way in which selected operations are combined. Because the individual

configurations are similar, the problem is reduced to choosing the optimal configuration in the sense of the accepted criterion. Considering the set of given configurations, as shown in Fig. 3, the problem is formulated as follows: from the possible configurations of the assembly line choose the optimal configuration in the sense of the given criterion. For the analysis of the problem, the fastest execution time for an order consisting of N pieces was chosen as an example criterion. The solution based on the shortest down-time is also presented. Of course, you can employ other criteria and many restrictions may be defined, such as, e.g., the capacity of interoperable buffers – as will be shown later in this paper.

3. Max-plus algebra

The $(\max, +)$ algebra was first introduced in [3]. A standard reference is [1], a brief survey of the methods and applications of this algebra is given in [6] and [8]. In certain aspects, the $(\max, +)$ algebra is comparable to conventional algebra. In the $(\max, +)$ algebra the addition $(+)$ and multiplication (\times) operators from conventional algebra are replaced by the maximization (\max) and addition $(+)$ operators, respectively. Using these operators, a linear description (in the $(\max, +)$ algebra sense) of certain non-linear systems (in conventional algebra) is achieved. In recent years, the theory of $(\max, +)$ algebra and its applications has been constantly developing and it suffices to mention a few directions: optimal control [10] predictive control [16], production modelling and planning [15, 12].

3.1. The basics

The $(\max, +)$ algebra is defined as follows:

- $R_\varepsilon = R \cup \{\varepsilon\}$, where R is the field of real numbers and $\varepsilon \stackrel{\text{def}}{=} -\infty$,
- $\forall a, b \in R_\varepsilon : a \oplus b \equiv \max(a, b)$,
- $\forall a, b \in R_\varepsilon : a \otimes b \equiv a + b$.

The algebraic structure $R_{\max} = (R_\varepsilon, \oplus, \otimes, \varepsilon, e)$, $e \stackrel{\text{def}}{=} 0$, is called the max-plus algebra. In this paper, the notation presented in [1] is used. This means ε and e are used instead of $-\infty$ and 0 , respectively (to emphasize their special meaning and to avoid confusion with their roles in conventional algebra). Additionally, the notation ab is used instead of $a \otimes b$ everywhere where it does not cause ambiguity.

Now, we extend the $(\max, +)$ algebra operations to matrices in the following way. The sum \oplus of matrices $\mathbf{A}, \mathbf{B} \in R_\varepsilon^{m \times n}$ is defined to be the matrix $(\mathbf{A} \oplus \mathbf{B}) \in R_\varepsilon^{m \times n}$ obtained by adding corresponding entries. That is,

$$(\mathbf{A} \oplus \mathbf{B})_{ij} = (\mathbf{A})_{ij} \oplus (\mathbf{B})_{ij}, \quad i = 1, \dots, m, j = 1, \dots, n \quad (1)$$

The product \otimes of matrices $\mathbf{A} \in R_{\varepsilon}^{m \times p}$ and $\mathbf{B} \in R_{\varepsilon}^{p \times n}$ is defined to be the $m \times n$ matrix whose (i, j) -entry is the inner product of the i th row of \mathbf{A} with the j th column in \mathbf{B} . That is,

$$(\mathbf{A} \otimes \mathbf{B})_{ij} = \bigoplus_{k=1}^p ((\mathbf{A})_{ik} \oplus (\mathbf{B})_{ik}) \equiv \max_k ((\mathbf{A})_{ik} \otimes (\mathbf{B})_{ik}), \quad i = 1, \dots, m, j = 1, \dots, n \quad (2)$$

where: $\bigoplus_{j+1}^m a_j$ is short-hand for $a_1 \oplus \dots \oplus a_m$.

The matrix $\mathbf{I}_n \in R_{\varepsilon}^{n \times n}$ with e 's on the main diagonal and ε 's else-where is called the identity matrix of order n . The matrix $\varepsilon \in R_{\varepsilon}^{n \times n}$ with $\varepsilon_{ij} = \varepsilon$ for all i, j is the zero matrix. The operator $*$ is defined for square matrix $\mathbf{A} \in R_{\varepsilon}^{n \times n}$ by:

$$\mathbf{A}^* \oplus_{k \in N_0} \mathbf{A}^k \quad (3)$$

where: $\mathbf{A}^k = \mathbf{A} \otimes \mathbf{A}^{k-1}$, $\mathbf{A}^0 = \mathbf{I}$, N_0 is the set of nonnegative integers. (3) is only meaningful if the right-hand side converges [3]. The operator $*$ will be needed later, because it can be used to solve the following implicit equation for \mathbf{x} :

$$\mathbf{x} = \mathbf{A}\mathbf{x} \oplus \mathbf{b} \quad (4)$$

so

$$\mathbf{x} = \mathbf{A}^* \mathbf{b} \quad (5)$$

The proof can be found, e.g., in [1].

3.2. Description of the state space

One of the best-known equations defining a dynamic system is

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t-1), \quad t = 1, 2, \dots \quad (6)$$

where the vector $\mathbf{x} \in R^n$ is the state of the system considered, and the matrix $\mathbf{A} \in R^{n \times n}$ is the state (or system) matrix. If the starting conditions are known, i.e., $\mathbf{x}(0) = \mathbf{x}_0$, then

the behaviour of the system is determined. When $\mathbf{x} \in R_\varepsilon^n$, $\mathbf{A} \in R_\varepsilon^{n \times n}$, Eq. (6) written in $(\max, +)$ algebra is as follows:

$$\forall k \in N : \mathbf{x}(k) = \mathbf{A} \otimes \mathbf{x}(k-1) \quad (7)$$

In Equation (7) k is used instead of t , since k is not the time of an event, but the index of the cycle in which an event takes place. The most general state-space representation of a $(\max, +)$ -linear system is given by:

$$\forall k \in N : \mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) \oplus \mathbf{B}\mathbf{u}(k) \quad (8)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \oplus \mathbf{D}\mathbf{u}(k) \quad (9)$$

where $\mathbf{u} \in R_\varepsilon^r$, $\mathbf{B} \in R_\varepsilon^{n \times r}$, $\mathbf{y} \in R_\varepsilon^m$, $\mathbf{C} \in R_\varepsilon^{m \times n}$, and $\mathbf{D} \in R_\varepsilon^{m \times r}$.

In a general case, for an N th order system, i.e., where the N previous iterations affect the current behaviour of the system given by an implicit formula for $\mathbf{x}(k)$, the model is represented by:

$$\forall k \in N : \mathbf{x}(k) = \bigoplus_{i=0}^N \mathbf{A}_i \mathbf{x}(k-i) \oplus \bigoplus_{i=0}^{N-1} \mathbf{B}_i \mathbf{u}(k-i) \quad (10)$$

$$\mathbf{y}(k) = \bigoplus_{i=0}^{N-1} (\mathbf{C}_i \mathbf{x}(k-i) \oplus \mathbf{D}_i \mathbf{u}(k-i)) \quad (11)$$

After removing $\mathbf{x}(k)$ from the right hand side of Eq. (10) (assuming that \mathbf{A}_0^* is convergent) and after introducing the new vectors

$$\tilde{\mathbf{x}} = [\mathbf{x}(k) \quad \mathbf{x}(k-1) \quad \dots \quad \mathbf{x}(k-N+1)]^T, \quad \tilde{\mathbf{u}} = [\mathbf{u}(k) \quad \mathbf{u}(k-1) \quad \dots \quad \mathbf{u}(k-N+1)]^T$$

and matrices:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0^* \mathbf{A}_1 & \mathbf{A}_0^* \mathbf{A}_2 & \dots & \mathbf{A}_0^* \mathbf{A}_N \\ \mathbf{I} & \varepsilon & \dots & \varepsilon \\ \varepsilon & \mathbf{I} & \dots & \varepsilon \\ \vdots & \vdots & \vdots & \vdots \\ \varepsilon & \varepsilon & \dots & \varepsilon \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{A}_0^* \mathbf{B}_0 & \dots & \mathbf{A}_0^* \mathbf{B}_{N-1} \\ \varepsilon & \dots & \varepsilon \\ \vdots & \vdots & \vdots \\ \varepsilon & \dots & \varepsilon \end{bmatrix}$$

$$\mathbf{C} = [\mathbf{C}_0 \quad \dots \quad \mathbf{C}_{N-1}], \quad \mathbf{D} = [\mathbf{D}_0 \quad \dots \quad \mathbf{D}_{N-1}]$$

where \mathbf{I} and ε are the (max, +)-algebraic identity and zero matrix of the appropriate sizes, a 1st order model, described by (8) and (9), is obtained.

In the rest of the paper, the part $\mathbf{D}u(k)$ in (9) is omitted and the equation describing takes the form:

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (12)$$

4. Modelling manufacturing systems

Before we go back to the production system described in Section 2, let us look at modelling the individual fragments of such a system. These fragments will be then combined into an entire production system. Modelling simple fragments is needed to gain intuition and experience for modelling more comprehensive systems.

We start by modelling a sequential production line, i.e., a line in which individual production sites are connected in series. Then we move on to modelling an assembly line in which flow can come from several lines. Finally, we will look at the modelling of interoperability buffers.

4.1. Modelling of serial flow

Let us consider a production line with n production units in series, as shown in Fig. 4. The input buffer (warehouse), u , from which materials are taken, is associated with this line. This type of line has one input station, i.e., one that gets materials from the input warehouse, and one output station, i.e., the production unit from which the finished (or intermediate) product leaves the line. The products leaving the line go to the output warehouse, y .

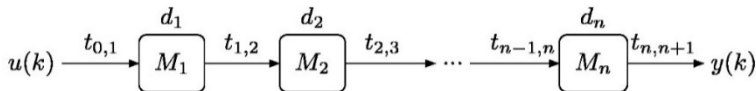


Fig. 4. Production line with n production units in series

Let $u(k)$, $y(k)$ and $x_i(k)$ be the time at which the incoming parts are made available to the line, the time at which the finished product leaves the line and the starting time of processing by the production unit M_i , respectively, for the k th job. If the material enters the system in the k th iteration (i.e., at time $u(k)$), then it is available at M_1 at time $t = u(k) + t_{0,1}$, where $t_{0,1}$ is the time at which the item is transported from the input warehouse to the production unit M_1 . But, M_1 can start work on this item only if it has already finished work on the previous one, i.e., on the item from the $(k - 1)$ th iteration. Assuming that the processing

time at M_1 is d_1 , then the item from the $(k-1)$ th iteration will leave the production unit at time $t = x_1(k-1) + d_1$. We assume that this item will leave the production unit immediately after processing is finished and processing of the next item will start as soon as the production unit is available.

The above conditions can be translated into the following equation describing the time at which the k th item will begin to be processed on M_1 :

$$\forall k \in N : x_1(k) = \max \left(d_1 + x_1(k-1), t_{0,1} + u(k) \right) \quad (13)$$

which in $(\max, +)$ algebra is as follows:

$$\forall k \in N : x_1(k) = d_1 x_1(k-1) \oplus t_{0,1} u(k) \quad (14)$$

Similarly, for production unit M_i , the k th item will start being processed by this unit when:

- processing on M_{i-1} has finished and it is delivered to M_i (the transportation time is $t_{i-1,i}$),
- processing of the $(k-1)$ th item on M_i has been completed.

In the $(\max, +)$ algebra this is expressed as follows:

$$\forall k \in N : x_i(k) = t_{i-1,i} d_{i-1} x_{i-1}(k) \oplus d_i x_i(k-1) \quad (15)$$

By combining the above equations into matrix notation:

$$\forall k \in N : \mathbf{x}(k) = \mathbf{A}_0 \mathbf{x}(k) \oplus \mathbf{A}_1 \mathbf{x}(k-1) \oplus \mathbf{B}_0 \mathbf{u}(k) \quad (16)$$

where:

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \dots & \varepsilon \\ t_{1,2} d_1 & \varepsilon & \dots & \varepsilon \\ \vdots & \vdots & \vdots & \vdots \\ \varepsilon & \varepsilon & t_{n-1,n} d_{n-1} & \varepsilon \end{bmatrix}$$

$$\mathbf{A}_1 = \begin{bmatrix} d_1 & \varepsilon & \dots & \varepsilon \\ \varepsilon & d_n & & \varepsilon \\ \vdots & \vdots & \vdots & \vdots \\ \varepsilon & \varepsilon & \dots & d_n \end{bmatrix}, \quad \mathbf{B}_0 = \begin{bmatrix} t_{0,1} \\ \varepsilon \\ \vdots \\ \varepsilon \end{bmatrix}, \quad \mathbf{u}(k) = [u(k)]$$

By applying the $*$ operator, Eq. (16) can be described as Eq. (8), where: $\mathbf{A} = \mathbf{A}_0^* \mathbf{A}_1$ and $\mathbf{B} = \mathbf{A}_0^* \mathbf{B}_0$. The output of the production line is described by Eq. (12), in which:

$$\mathbf{y}(k) = [y(k)] \text{ and } \mathbf{C} = [\varepsilon \quad \dots \quad \varepsilon \quad t_{n,n+1}d_n]$$

Example 4.1. Let us consider a production line with of 3 stations M_1 , M_2 and M_3 in series, as shown in Fig. 5.

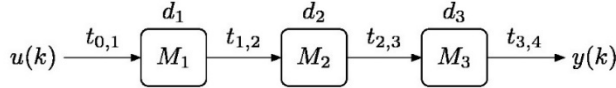


Fig. 5. An exemplary production line with 3 production units in series

Raw materials are fed into production unit M_1 from input buffer u , where they are processed and sent to M_2 , then to M_3 and finally leave the system. The processing times for the individual production units are respectively: $d_1 = 3$, $d_2 = 2$ and $d_3 = 6$ time units. The transportation times are respectively: $t_{0,1} = 1$, $t_{1,2} = 2$, $t_{2,3} = 0$, $t_{3,4} = 0$ time units. At the input of the system and between the production units there are buffers with a capacity that is large enough to ensure that no over-flow occurs. Initially, all the buffers are empty and none of the production units contain any raw materials or intermediate products. A production unit can start working on a new product only after it has finished processing the previous one. We assume that each production unit starts working as soon as all of the parts are available. This means:

$$\begin{aligned} \forall k \in N : x_1(k) &= \max(d_1 + x_1(k-1), t_{0,1} + u(k)) \\ x_2(k) &= \max(d_2 + x_2(k-1), t_{1,2} + d_1 + x_1(k)) \\ x_3(k) &= \max(d_3 + x_3(k-1), t_{2,3} + d_2 + x_2(k)) \\ y(k) &= \max(t_{3,2} + d_3 + x_3(k)) \end{aligned} \tag{17}$$

The system of Eqs. (17) is given in (max, +) algebra by:

$$\begin{aligned} \forall k \in N : x_1(k) &= d_1 x_1(k-1) \oplus t_{0,1} u(k) \\ x_2(k) &= d_2 x_2(k-1) \oplus t_{1,2} d_1 x_1(k) \\ x_3(k) &= d_3 x_3(k-1) \oplus t_{2,3} d_2 x_2(k) \\ y(k) &= t_{3,4} d_3 x_3(k) \end{aligned} \tag{18}$$

Using matrix notation, this system can be written in the form of (16), then of (8), and the output equation is in the form of (12), where:

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ t_{1,2}d_1 & \varepsilon & \varepsilon \\ \varepsilon & t_{2,3}d_2 & \varepsilon \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon \\ \varepsilon & \varepsilon & d_3 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ t_{1,2}d_1^2 & d_2 & \varepsilon \\ t_{1,2}d_1^2t_{2,3}d_2 & t_{2,3}d_2^2 & d_3 \end{bmatrix}, \quad \mathbf{B}_0 = \begin{bmatrix} t_{0,1} \\ \varepsilon \\ \varepsilon \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} t_{0,1} \\ t_{0,1}t_{1,2}d_1 \\ t_{0,1}t_{1,2}d_1t_{2,3}d_2 \end{bmatrix}$$

$$\mathbf{u}(k) = [u(k)], \quad \mathbf{y}(k) = [y(k)], \quad \mathbf{C} = [\varepsilon \quad \varepsilon \quad t_{3,4}d_3]$$

After substituting in the appropriate numerical values, we obtain:

$$\mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 3 & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon \\ \varepsilon & \varepsilon & 6 \end{bmatrix}, \quad \mathbf{A}_0^* = \begin{bmatrix} 0 & \varepsilon & \varepsilon \\ 5 & 0 & \varepsilon \\ 7 & 2 & 0 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 3 & \varepsilon & \varepsilon \\ 8 & 2 & \varepsilon \\ 10 & 4 & 6 \end{bmatrix}, \quad \mathbf{B}_0 = \begin{bmatrix} 1 \\ \varepsilon \\ \varepsilon \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 6 \\ 8 \end{bmatrix}, \quad \mathbf{C} = [\varepsilon \quad \varepsilon \quad 6]$$

with the initial conditions: $\forall k \in N : \mathbf{u}(k) = [0]$, and $\mathbf{x}(0) = [\varepsilon \quad \varepsilon \quad \varepsilon]^T$. The behaviour of the modelled production line for the first 12 iterations is presented in Table 2, although it is better illustrated by the Gantt chart shown in Fig. 6.

Table 2. The evolution of the state vector and output for Example 4.1

k	1	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{x}(k)$	1	4	7	10	13	16	19	22	25	28	31	34
	6	9	12	15	18	21	24	27	30	33	36	39
	6	14	20	26	32	38	44	50	56	62	68	74
$\mathbf{y}(k)$	14	20	26	32	38	44	50	56	62	68	74	80

According to this model, an item will leave a production unit immediately after processing. This means that there must be space for this part in the interoperability

buffer. As shown in Fig. 6, the number of items stored in the buffer between M_2 and M_3 increases with each iteration. The interoperability of buffer capacity is the subject of Section 4.3.

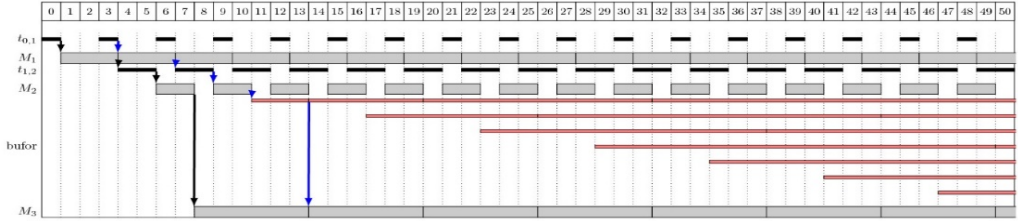


Fig. 6. The Gantt chart for Example 4.1

4.2. Modelling the merging of lines

Let us consider the production system presented in Fig. 7. Items from more than one production unit go to the merging station M_{n+1} where processing (e.g., assembly) takes place. Then one finished product leaves the system from the merging station.

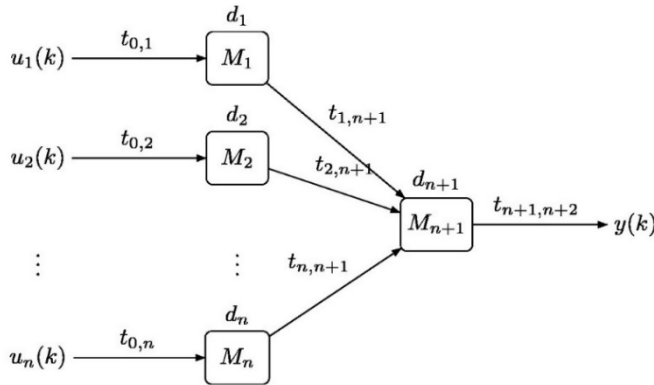


Fig. 7. Production system with n merging lines

Let d_i denote the operation time on production unit M_i . Let $u_i(k)$ for $i \leq n$ denote the time at which the materials required by the i th unit appear at the input in the k th cycle. Additionally, let $t_{0,i}$ denote the transportation time of these materials from u_i to M_i , and $t_{i,n+1}$ denote the transportation time from M_i to M_{n+1} . Hence, $x_i(k)$ (i.e., the start time of processing on production unit M_i in the k th iteration) $\forall k \in \mathbb{N}$: and $\forall i \leq n$, is the maximum value of the following:

- the time at which the materials from u_i are shipped to M_i ,
- the time at which the $(k - 1)$ th operation at production unit M_i is finished.

Analogously, $x_{n+1}(k)$ is the maximum of the following:

- the time at which the k th operation at M_i is both finished and the item transported to M_{n+1} , ($i = 1, 2, \dots, n$),
- the $(k - 1)$ th operation at M_{n+1} is finished.

Thus, an equation of the form (16) has been obtained, where:

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_{n+1}(k) \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} \varepsilon & \dots & \varepsilon & \varepsilon \\ \vdots & & \vdots & \vdots \\ \varepsilon & \dots & \varepsilon & \varepsilon \\ t_{1,n+1}d_1 & \dots & t_{n,n+1}d_n & \varepsilon \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} d_1 & \varepsilon & \dots & \varepsilon \\ \varepsilon & d_2 & & \varepsilon \\ \vdots & & \cdot & \vdots \\ \varepsilon & \varepsilon & \dots & d_{n+1} \end{bmatrix}$$

$$\mathbf{B}_0 = \begin{bmatrix} t_{0,1} & \varepsilon & \dots & \varepsilon \\ \varepsilon & t_{0,2} & & \varepsilon \\ \vdots & & & \vdots \\ \varepsilon & \varepsilon & & t_{0,n} \\ \varepsilon & \varepsilon & \dots & \varepsilon \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_n(k) \end{bmatrix}$$

The finished product reaches the output warehouse $y(k)$ when the k th operation at production unit M_{n+1} is finished and this item is transported to the warehouse, so the equation of output is expressed as in (12), in which:

$$\mathbf{y}(k) = [y(k)]$$

and

$$\mathbf{C} = [\varepsilon \quad \dots \quad \varepsilon \quad t_{n+1,n+2}d_{n+1}]$$

Example 4.2. Let us look at the specific example of the assembly line shown in Fig. 8.

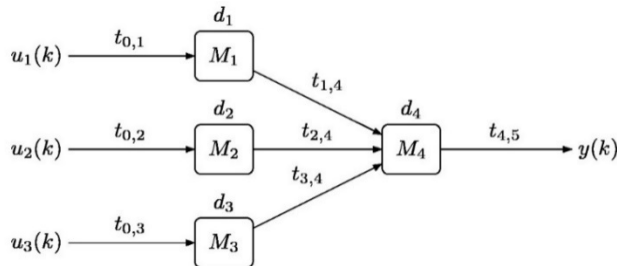


Fig. 8. Assembly line from Example 4.2

Assume that the processing times are: $d_1 = 3, d_2 = 2, d_3 = 6$ and $d_4 = 2$. The transportation time $t_{2,4} = 1$, the other transportation times are negligible. Hence, the matrices from (16) and (12) are respectively:

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}, \mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 & 3 & 6 & \varepsilon \end{bmatrix}, \mathbf{A}_0 = \begin{bmatrix} 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 6 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 2 \end{bmatrix}, \mathbf{B}_0 = \begin{bmatrix} 0 & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$\mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}, \mathbf{y}(k) = [y(k)], \mathbf{C} = [\varepsilon \quad \varepsilon \quad \varepsilon \quad 2]$$

The state and output vectors for the first 12 iterations are shown in Table 3, assuming that the initial conditions are given by $\mathbf{x}(0) = [\varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon]^T$. The input vector for each iteration does not change and is $\mathbf{u}(k) = [0 \quad 0 \quad 0]^T$. The production schedule is shown in Fig. 9.

Table 3. The evolution of the state and output vector for Example 4.2

k	1	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{x}(k)$	0	3	6	9	12	15	18	21	24	27	30	33
	0	2	4	6	8	10	12	14	16	18	20	22
	0	6	12	18	24	30	36	42	48	54	60	66
	6	12	18	24	30	36	42	48	54	60	66	72
$\mathbf{y}(k)$	8	14	20	26	32	38	44	50	56	62	68	74

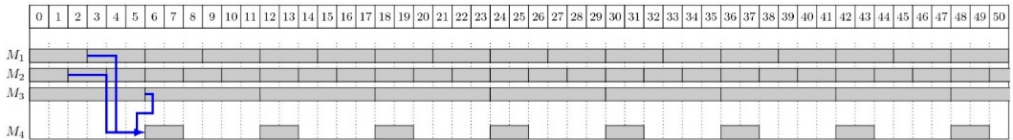


Fig. 9. The Gantt chart for Example 4.2

4.3. Modelling the capacity of interoperability buffers

In the previous discussions, we assumed that the capacity of the interoperability buffers was sufficient. Let us now take a closer look at the modelling of systems for which this assumption does not hold.

4.3.1. No buffers at all

Assume that there are no buffers between any production units. This means that an item from M_i goes to the next production unit, M_{i+1} , only if it is free, or the item must wait for M_i until such a time that M_{i+1} would be ready to operate. To illustrate this situation, consider again the system illustrated in Fig. 4. The raw materials entering the system in the k th iteration are available at M_1 at time $t = u(k) + t_{0,1}$. M_1 can start processing the materials when the previous operation is finished, i.e., at time $t = x_1(k-1) + d_1$. In addition, production unit M_1 must be ready to load the materials for the k th iteration. This means that the $(k-1)$ th item has already left the production unit and processing on M_2 has started. Hence:

$$\forall k \in N : x_1(k) = d_1 x_1(k-1) \oplus 0 x_2(k-1) \oplus t_{0,1} u(k) \quad (19)$$

The remaining equations describing the system are formulated in the same way. If the transport time of an item from M_i to M_{i+1} is significant and denoted $t_{i,i+1}$, then processing at M_i should be completed before processing at M_{i+1} is started, more specifically at time $= -t_{i,i+1} x_{i+1}$ instead of $0 x_{i+1}$ for the case where the transportation time is insignificant. So, $x_1(k)$ is given by:

$$\forall k \in N : x_1(k) = d_1 x_1(k-1) \oplus -t_{1,2} x_2(k-1) \oplus t_{0,1} u(k) \quad (20)$$

In the matrix description given by (16), the only difference occurs in the matrix \mathbf{A}_1 . When there are no buffers

$$\mathbf{A}_1 = \begin{bmatrix} d_1 & -t_{1,2} & \varepsilon & \dots & \varepsilon \\ \varepsilon & d_2 & -t_{2,3} & \dots & \varepsilon \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \varepsilon & \varepsilon & \varepsilon & \dots & d_n \\ \varepsilon & \varepsilon & \varepsilon & \dots & \dots \end{bmatrix}.$$

Example 4.3. Let us consider the production system from Example 4.1 with no interoperation buffers. The evolution of the output vector is shown in Table 4. The schedule is presented in Fig. 10.

Table 4. Evolution of the state and output vector for Example 4.3

k	1	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{x}(k)$	1	4	7	12	18	24	30	36	42	48	54	60
	6	9	14	20	26	32	38	44	50	56	62	68
	8	14	20	26	32	38	44	50	56	62	68	74
$\mathbf{y}(k)$	14	20	26	32	38	44	50	56	62	68	74	80

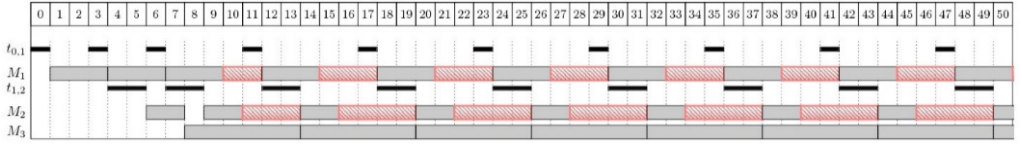


Fig. 10. The Gantt chart for Example 4.3

4.3.2. Buffers with a finite capacity

The modelling of finite buffers is an extension of the no buffers case. This means that if there is a buffer of capacity n between M_i and M_{i+1} , then $(\mathbf{A}_{n+1})_{i,i+1} = 0$ (if the transportation time $t_{i,i+1}$ is significant, then $(\mathbf{A}_{n+1})_{i,i+1} = -t_{i,i+1}$). The modelled system is described by the equation:

$$\forall k \in N : \mathbf{x}(k) = \mathbf{A}_0 \mathbf{x}(k) \oplus \mathbf{A}_1 \mathbf{x}(k-1) \oplus \dots \oplus \mathbf{B}_0 \mathbf{u}(k) \quad (21)$$

This approach, as can be seen, increases the number of matrices in the model, and transformation into a model described by (16) significantly increases the size of the matrices.

Example 4.4. Let us consider the behaviour of the production system from Example 4.1 with finite buffers:

- between M_1 and M_2 there is a buffer with capacity 1,
- between M_2 and M_3 there is a buffer with capacity 2.

The model of such a system is described by the equations:

$$\begin{aligned} \forall k \in N : \mathbf{x}(k) &= \mathbf{A}_0 \mathbf{x}(k) \oplus \mathbf{A}_1 \mathbf{x}(k-1) \oplus \mathbf{A}_2 \mathbf{x}(k-2) \oplus \mathbf{A}_3 \mathbf{x}(k-3) \oplus \mathbf{B}_0 \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_0 \mathbf{x}(k) \end{aligned} \quad (22)$$

where

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ t_{1,2}d_1 & \varepsilon & \varepsilon \\ \varepsilon & t_{2,3}d_2 & \varepsilon \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon \\ \varepsilon & \varepsilon & d_3 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \varepsilon & -t_{1,2} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{B}_0 = \begin{bmatrix} t_{0,1} \\ \varepsilon \\ \varepsilon \end{bmatrix}, \quad \mathbf{u}(k) = [u(k)], \quad \mathbf{y}(k) = [y(k)], \quad \mathbf{C}_0 = [\varepsilon \quad \varepsilon \quad t_{3,4}d_3]$$

After introducing a new state vector $\tilde{\mathbf{x}}$, Eqs. (22) can be reduced to (16), and the output Eq. (12), where

$$\tilde{\mathbf{x}}(k) = [x_1(k) \quad x_2(k) \quad x_3(k) \quad x_4(k) \quad x_5(k) \quad x_6(k) \quad x_7(k) \quad x_8(k) \quad x_9(k)]^T$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0^* \mathbf{A}_1 & \mathbf{A}_0^* \mathbf{A}_2 & \mathbf{A}_0^* \mathbf{A}_3 \\ \mathbf{I} & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{I} & \varepsilon \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{A}_0^* \mathbf{B}_0 \\ \varepsilon \\ \varepsilon \end{bmatrix}, \quad \mathbf{C} = [\mathbf{C}_0 \quad \varepsilon \quad \varepsilon]$$

The behaviour of the modelled system, i.e., the behaviour of the first three state variables $\mathbf{x}_i(k)$ and output $\mathbf{y}(k)$ is shown in Table 5 and Fig. 11.

Table 5. Evolution of the state and output vector for Example 4.4

k	1	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{x}(k)$	1	4	7	10	13	16	19	22	25	28	31	34
	6	9	12	15	18	21	26	32	38	44	50	56
	8	14	20	26	32	38	44	50	56	62	68	74
$\mathbf{y}(k)$	14	20	26	32	38	44	50	56	62	68	74	80

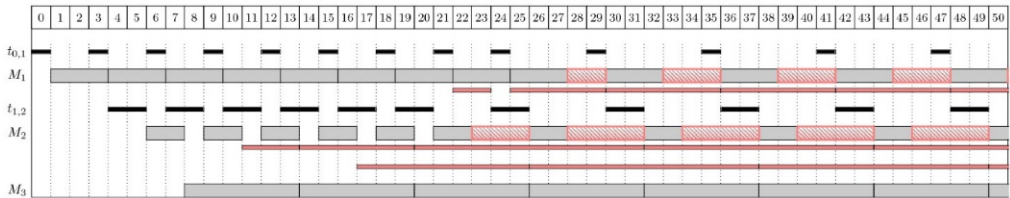


Fig. 11. The Gantt chart for Example 4.4

The capacity of the interoperability buffers can also be modelled differently. A buffer of capacity n can be treated as n machines with 0 operation time and no buffers in between them. The matrices describing models generated in this way are usually smaller than those derived using the previous approach. This case is illustrated by Example 4.5.

Example 4.5. Let us consider the behaviour of the production system from Example 4.1 with interoperability buffers as in Example 4.4. To model the interoperable buffer between M_1 and M_2 , an artificial production unit $M_4(x_4)$ is added. Similarly, to model the buffer with capacity 2 between M_2 and M_3 , two additional production units are added: M_5 and M_6 . The system obtained after such a modification is shown in Fig. 12.

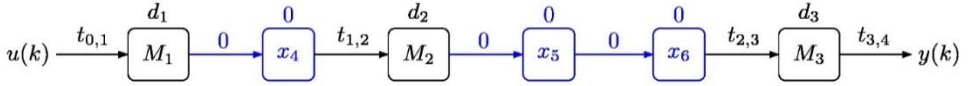


Fig. 12. Production line with 3 production units in series and interoperability buffers modelled as zero-time operations

The model of this system is described by Eqs. (16) and (12), where:

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & t_{1/2} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_{2/3} \\ d_1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} d_1 & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & d_3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & -t_{1,2} & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & 0 \\ \varepsilon & \varepsilon & -t_{2,3} & \varepsilon & \varepsilon & 0 \end{bmatrix}$$

$$\mathbf{B}_0 = \begin{bmatrix} t_{0,1} \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix}, \quad \mathbf{u}(k) = [u(k)], \quad \mathbf{y}(k) = [y(k)], \quad \mathbf{C} = [\varepsilon \quad \varepsilon \quad t_{3,4}d_3 \quad \varepsilon \quad \varepsilon \quad \varepsilon]$$

The behaviour of the modelled system, more specifically, the behaviour of the first three state-space variables of $\mathbf{x}(k)$ and output $\mathbf{y}(k)$ are identical to the previous example. Being able to model selected classes of production systems, we can return to the case study considered in Section 2.

5. Case study. Part 2

Before proceeding to analyse the configuration of the production line described in Section 2 (Fig. 3), the following variants were modelled for each configuration:

- execution time for orders of 10, 100 and 1000 items,
- buffer capacity 0, 1, 10 and sufficient (i.e., infinite).

After modelling, it turns out that almost all of the configurations give the same result – column $\mathbf{y}(k)$ in Table 6. Almost, because, as can be seen, cases with no interoperability buffers give a longer lead time for the configurations (b) and (c). Hence, you can opt for

any configuration and any variant (except for the two variants mentioned above) or define an additional criterion for selection. To choose one of the configurations, let us adopt minimisation of the total downtime of production units as an additional criterion. Let the downtime p_i of production unit M_i in the course of m cycles be expressed by:

$$p_i = \sum_{k=1}^m (x_i(k) - (x_i(k-1) + d_i)) \quad (23)$$

where $x_i(k)$ – start time of the k th operation on production unit M_i , d_i – processing time on M_i . It is assumed that $(x_i(k-1) + d_i) = 0$ for $k = 1$.

The total downtime, p_T , is defined as the sum of the downtimes for the individual production units (n – number of production units), i.e.,

$$p_T = \sum_{i=1}^n p_i \quad (24)$$

The average downtime is given by:

$$\bar{p}_T = \frac{p_T}{n} \quad (25)$$

A convenient indicator is the percentage downtime, defined as:

$$pp_T = \frac{\bar{p}_T}{y(m)} 100\% \quad (26)$$

where $y(m)$ is the time at which the order is completed (output time of the m th item). After adding this new selection criterion, configuration (c) is preferred. This is because the order will be executed in the same time as for the other configurations, but in this configuration, the individual production units are released the fastest. In addition, increasing the capacity of the interoperability buffers further improves this index.

6. Conclusions

We encounter many complex analytical problems when designing, analysing and planning production systems. This paper presents an application of $(\max, +)$ algebra to the modelling of manufacturing systems to obtain an analytical model. Based on this

model, various performance indicators can be easily obtained, derived from the implementation of processes over time. The (max, +) algebra is a convenient tool for modelling deterministic DES, where the operation times are constant, and the sequences of operations are known. On the other hand, this is also a fundamental drawback of this tool, since it can be applied to a narrower class of systems than other DES modelling techniques, such as automata theory or Petri nets. Nevertheless, it has the advantage of being a purely analytical model, which allows us to understand certain phenomena or optimize the parameters of the systems under investigation. The presented material is an introduction to further research, first and foremost on the optimal or suboptimal synthesis of individual single-product systems in a multi-assortment production system. In the study of such systems, there is a need to develop a tool to automate the generation of a (max, +) model to speed up simulations and calculations.

Table 6. Numerical results obtained for various configurations and variants

Configuration	k = 10			k = 100			k = 1000		
	y(k)	p _T	pp _T	y(k)	p _T	pp _T	y(k)	p _T	pp _T
(a)									
No buffers	451	1364	43.21	4321	18014	59.56	43021	184514	61.27
Buffer capacity 1	451	927	29.36	4321	17455	57.71	43021	183955	61.08
Buffer capacity 10	451	706	22.36	4321	12983	42.92	43021	179483	59.60
Infinite buffers	451	706	22.36	4321	6556	21.67	43021	65056	21.60
(b)									
No buffers	459	1071	38.89	4329	12591	48.47	43029	127791	49.50
Buffer capacity 1	451	687	25.38	4321	12199	47.05	43021	127399	49.36
Buffer capacity 10	451	413	15.26	4321	9447	36.49	43021	124647	48.29
Infinite buffers	451	413	15.26	4321	3653	14.09	43021	36053	13.97
(c)									
No buffers	461	754	32.71	4331	8224	37.98	43031	82924	38.54
Buffer capacity 1	451	489	21.68	4321	7959	36.84	43021	82659	38.43
Buffer capacity 10	451	296	13.17	4321	6239	28.88	43021	80939	37.63
Infinite buffers	451	296	13.13	4321	2546	11.78	43021	25046	11.64

All the computations for this publication were carried out using the Max-Plus Algebra Toolbox for Matlab [17].

References

[1] BACCELLI F., COHEN G., OLSDER G.J., QUADRAT J-P., *Synchronisation and Linearity, an Algebra for Discrete Event Systems*, Wiley, Chichester 1992.

[2] CASSANDRAS C.G., LAFORTUNE S., *Introduction to Discrete Event Systems*, Springer, Heidelberg 2007.

[3] CUNINGHAME-GREEN R., *Minimax Algebra. Lecture Notes in Economics and Mathematical Systems*, 1979, 166.

[4] *Dorot Control Valves*, Adopted from Dorot materials, 2001, <http://www.dorot.com/>

- [5] GROSS D., SHORTIE J.F., THOMPSON J.M., HARRIS C.M., *Fundamentals of Queueing Theory*, Wiley, Hoboken 2008.
- [6] HEIDERGOTT B., OLSDER G.J., VAN DER WOUDE J., *Max Plus at Work. Modelling and Analysis of Synchronized Systems*, Princeton University Press, Princeton 2006.
- [7] KASHKOUSH M., ELMARAGHY H., *Consensus tree method for generating master assembly sequence*, *Prod. Eng.*, 2014, 8, 233–242.
- [8] KOMEDA J., LAHAYE S., BOIMOND J.-L., VAN DEN BOOM T., *Max-plus algebra and discrete event systems*, *IFAC Papers Online*, 2017, 50 (1), 1784–1790.
- [9] LIMNIOS N., OPROŞAN G., *Semi-Markov Processes and Reliability*, Springer Science, Business Media, New York 2013.
- [10] MAIA C.A., HARDOUIN L., SANTOS-MENDES R., COTTENCEAU B., *Optimal closed-loop control of timed event graphs in dioids*, *IEEE Trans. on Automatic Control*, 2003, 12, 2284–2287.
- [11] MUTSAERS M., ÖZKAN L., BACKX T., *Scheduling of energy flows for parallel batch processing using max-plus systems*, *Proc. 8 IFAC Symposium on Advanced Control of Chemical Processes*, Singapore, 2012.
- [12] NAMBIAR A.N., IMAEV A., JUDD R.P., CARLO H.J., *Production planning models using max-plus algebra*, [In:] V. Modrak, R.S. Pandian (Eds.), *Operations management research and cellular manufacturing systems*, IGI Global, Hershey 2012, 227–257.
- [13] PETRI C.A., *Kommunikation mit Automaten*, PhD Thesis, University of Bonn, 1962.
- [14] RAMADGE P.J., WONHAM W.M., *The control of discrete event systems*, *Proc. of the IEEE*, 1989, 77, 81–98.
- [15] SELEIM A., ELMARAGHY H., *Max-plus modeling of manufacturing flow lines*, *Proc. 47 CIRP Conference Manufacturing Systems (CMS2014)*, 2014, 17, 302–307.
- [16] SEYBOLD L., WITCZAK M., MAJDIK P., STETTER R., *Towards robust predictive fault-tolerant control for a battery assembly system*, *Int. J. Appl. Math. Comp. Sci.*, 2015, 25 (4), 849–862.
- [17] STAŃCZYK J., *Max-Plus Algebra Toolbox for Matlab ver. 1.7*, 2016, <http://gen.up.wroc.pl/stanczyk/mpa/>
- [18] STAŃCZYK J., MAYER E., RAISCH J., *Modelling and performance evaluation of DES*, *Proc. International Conference Informatics in Control, Automation and Robotics*, Setubal, Portugal, 2004, 3, 270–275.

Received 9 October 2017

Accepted 14 August 2018