Applied Computer Science, vol. 18, no. 3, pp. 5–18
doi: 10.35784/acs-2022-17

Submitted: 2022-07-28 | Revised: 2022-09-13 | Accepted: 2022-09-20

Keywords: FMS, Petri Nets, Performance evaluation

Grzegorz BOCEWICZ [0000-0002-5181-2872]*, *Robert WÓJCIK* [0000-0003-3860-7011]**, *Paweł SITEK* [0000-0001-6108-0241]***, *Zbigniew BANASZAK* [0000-0001-7219-3903]*

TOWARDS DIGITAL TWIN-DRIVEN PERFORMANCE EVALUATION METHODOLOGY OF FMS

Abstract

The paper presents a method of automated modelling and performance evaluation of concurrent production flows carried out in Flexible Manufacturing Systems. The method allows for quick assessment of various variants of such systems, considering their structure and the organization of production flow of possible ways of their implementation. Its essence is the conditions imposed on the designed model, limiting the space of possible variants of the production flow only to deadlock-free variants. The practical usefulness of the model implemented in the proposed method illustrates the example, which describes the simultaneous assessment of alternative variants of the flexible machining module's structure and the planned multi-assortment production. The ability of the method to focus on feasible solutions offers attractive perspectives for guiding the Digital Twin-like scenario in situations caused by the need to change the production flow.

1. INTRODUCTION

Designing Flexible Manufacturing Systems (FMSs) and planning technological processes generate complex multi-criteria optimization problems. These problems are related to decision-making in terms of resource allocation, process scheduling, and resolving resource conflicts of processes competing for access to shared resources.

Decision Support Systems (DSSs), based on online modelling and simulation, play a crucial role in solving the problems under consideration (Vaisi, 2022; Bujari et al., 2021; Makris, Michalos & Chryssolouris, 2012; Sliwa & Patalas-Maliszewska, 2016). The methods used in DSSs design are usually problem-oriented, which limits the range of their possible applications (Bakar, Henry & Ali, 1991; Banaszak, 1992; Banaszak, Skolud & Zaremba, 2003; Viswandham & Narahari, 1992). In particular, this implies the need for modelling and evaluating FMSs functioning based on the digital twin (DT) approach employed in interactive FMSs prototyping (Vaisi, 2022; Makris, Michalos & Chryssolouris, 2012).

* Faculty of Electronics and Computer Science, Koszalin University of Technology, Poland, grzegorz.bocewicz@tu.koszalin.pl, zbigniew.banaszak@tu.koszalin.pl
** Faculty of Information and Communication Technology, Wrocław University of Science and Technology, Poland, robert.wojcik@pwr.edu.pl
*** Department of Information Systems, Kielce University of Technology, Kielce, Poland, sitek@tu.kielce.pl

Since the dominant role in FMSs is played by the interaction of various processes taking place in them, i.e., the flow of workpieces, jigs, and fixtures, tool exchange, and chip removal, data and energy flows, and others, the natural choice of representation to model their behavior is using the formalism of the Petri nets (Reisig, 1982; Reutenauer, 1988).

Models of this type allow both to assess alternative variants of the FMS structure (i.e., the configuration of autonomous production modules that create it) and to choose how to organize the flow of production carried out in it.

The rest of the article is organized as follows: Section 2 elaborates on related works. Section 3 presents the example of a Flexible Machining Module (FMM) and introduces the issues of planning multi-assortment production carried out in it. Section 4 describes the terminology and essential modelling principles of the Petri net representation used in the FMM reference model. Section 5 includes a diagram of the methodology for the construction of methods implementing the digital twin-driven decision support concept (DTDSC), in particular, the method of determining the configuration of the FMS structure and assessing alternative ways of organizing the flow of production carried out in it. Section 6 presents the concluding remarks.

2. CURRENT-STATE

In recent years, there has been a rapid increase in the use of FMS in the automotive, electrical, and electronic industries (Janardhanan et al., 2019). The increase in the variety of FMSs applications is accompanied by new solutions to their structures and ways of functioning.

The concept of FMS that merges the ideology of flow shop and batch shop manufacturing system is based on three major components, i.e., workstations, automated material handling with a storage system, and a central computer. The interaction of these components determines the scope of the flexibility of the FMS, which can be assessed by different tests such as part variety, schedule change, error recovery, and new part test (Jonsson, 2000; Manu et al., 2018; Rachamadugu & Stecke, 1994). To effectively exploit this potential for flexibility, different methods are used. Among the available methods such as analytical methods (using mathematical programming), heuristic methods (employing production sequencing rules), artificial intelligence (using evolutionary and (or) population algorithms), and computer simulation (especially discrete event simulation). The latter is most often used in practice, which is due, among other things, to the fact that mathematical computations are cumbersome (as well as very time-consuming) and evolutionary algorithms are not very precise. It is also worth noting that computer simulation methods make it possible to analyze the transient states associated with the start-up and termination of production processes.

In this context, a digital twin concept referring to a digital replica of physical processes and systems seems to be well suited to model various FMSs solutions regarding their configuration and the organization of production carried out in them. Notice that the digital twin combines a physical object (e.g., FMS) and its digital representation in virtual space (e.g., discrete event system model). Therefore, the FMS simulation model implemented with the aid of it can be used both to plan the processes implemented virtually in it and to correct previously planned and then physically implemented in FMS. Indeed, this approach to FMSs design and control is increasingly common and finds its applications in systems supporting preventive maintenance scheduling and proactive job-shop as well as dynamic scheduling in manufacturing (Coito et al., 2022; David, Lobov & Lanz, 2018; Hatono et al., 1989; Neto et al., 2021; Nielsen, Michna & Do, 2014; Nielsen, Sung & Nielsen, 2019; Patalas-Maliszewska & Kłos, 2019; Zhang, Bai & Yang, 2022; Stączek et al., 2021; Świć & Gola, 2013).

It should be emphasized that methods implementing simulation models following ***IF ... THEN*** rule paradigm do not guarantee acceptable solutions, e.g., scenarios of production processes execution that do not lead to starvation and (or) deadlocks. Therefore the simulation models implemented in digital twin solutions are burdened with a similar deficiency, implying the need to eliminate unnecessary analysis of unacceptable solutions, e.g., leading to deadlocks of modelled processes. This observation is the main inspiration for the research undertaken in this work showing the possibilities of building simulation models limiting the implemented scenarios of system behavior only to acceptable ones.

The multi-criteria nature of production-planning problems, the complexity of these problems, and the need to make decisions online spur the development of techniques and methods for building dedicated DSSs (Banaszak, 1992; Bujari et al., 2012; Jensen, 1987; Laemmle & Gust, 2019). Hence, the relevant computer-aided tools should be designed to allow an integrated online analysis of alternative scenarios for completing production orders and early detection of errors in the order execution method used (Alexopoulos et al., 2022). Similar expectations apply to solutions focused on computer-aided modelling, exploring feasible alternatives, and evaluating the functioning of operational control algorithms implemented in real-time industrial controllers (Heiner et al., 1992).

Due to the variety of production decision-making problems, the large number of decision variables characterizing them, and the multi-criteria nature of the problems solved, computer simulation methods are most often used to solve them. Many papers in the scope of modelling and simulation of production flow especially scheduling and routing, prefer the Petri nets framework usage (Recalde et al., 2022; Zhou & Zain, 2016). Some already proposed approaches, such as colored, inhibitor, fuzzy, timed, predicate-transition, and hierarchical Petri nets representation, have been used to model complex systems with outstanding results (Laemmle & Gust, 2019; Van der Aalst, 1992). Their main advantage is the easy implementation of procedures for flow control of the processes performed in the modeled systems. In other words, the essential advantage of Petri network representation comes down to the possibility of prototyping alternative material flow scenarios modeled as a procedure for the relevant control flows (Bocewicz et al., 2022; He et al., 2022).Control flow procedures usually boil down to the implementation of different priority dispatching rules such as Longest Processing Time, Shortest Processing Time, and First in, First Out, and Last in, First Out, and many others, including Earliest Due Date, Critical Ratio, Dynamic Least Slack Rules, and other (Silva et al., 2012; Zanchettin, 2021).

The main shortcoming of the simulation approach is the considerable amount of time needed to build an appropriate model. In this respect, an approach aimed at an automated generation of the simulation model using domain-oriented data of production systems seems to be the most promising. An example of such a type of approach is presented in this work.

3. EXAMPLE OF HYPOTHETICAL FMM SPECIFICATION

The FMS class under consideration includes systems in which:

– a set of pipeline processes $PP = \{PR_i \mid i = 1 \dots v\}$, such that: $PR_i = ((d_{i_k}, O_{i_k}) \mid k = 1 \dots \mathcal{K})$, where: d_{i_k} – the machine used to process the i_k-th product, O_{i_k}– the th operation of the i-th process carried out on the machine $d_{i_k} \in D$, where D – is the set of FMS resources (machines), $\mathcal{K}$ – means number of operations of the PR_i process,

– an inter-operational storage buffer B_i with a capacity of BP_i is assigned to each i-th machine ($d_i \in D$),

– to each operation O_i of the i-th process carried out on the d_i machine, the time of its execution t_i is assigned.

Moreover, it is assumed that each odd operation of the i-process is carried out by an appropriate device(s) moving the workpieces between the buffers of successive machines; each operation of the i-th process is carried out on a machine from the set D.

To illustrate the introduced specification of the processes carried out in the example FMS, let's consider FMM with the configuration as in Fig. 1.

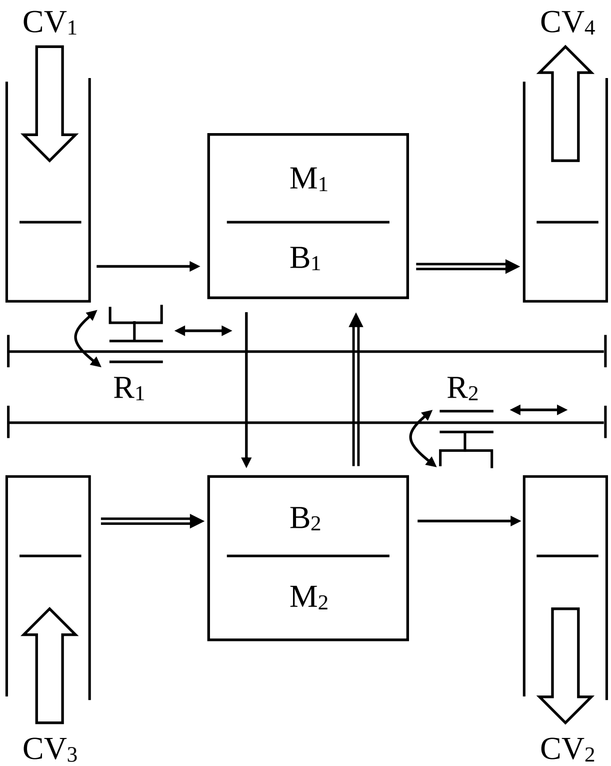

Fig. 1. Diagram of an example FMM

Process sequences specifying processes in the considered FMM are given in the following forms:

$$PR_1 = ((R_1, O_1), (M_1, O_2), (R_1, O_3), (M_2, O_4), (R_2, O_5)), \quad (1)$$
$$PR_2 = ((R_1, O_6), (M_2, O_7), (R_2, O_8), (M_1, O_9), (R_1, O_{10})),$$

(R_i, O_j) denotes the j-th operation of the robot R_i and (M_k, O_j) the j-th operation of the M_k machine.

Machines M_1, M_2 are associated with buffers B_1, B_2 assigned to them with appropriate capacities: $BP_1 = 1$, $BP_2 = 2$. Operation times are given in Table 1.

Tab. 1. Delivery times t_i for transport operations and processing of process specifications (1) calculated in contractual units of time (t.u.)

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
t_i [t.u.]	5	20	5	10	5	10	25	5	10	5

According to the PR_1 specification, the items delivered by the CV_1 feeder are first processed on the M_1 machine, then on the M_2. Next, they are stored on the CV_2 feeder. According to the PR_2 specification, the items supplied by the CV_3 feeder are processed first on the M_2 machine, then on the M_1. Finally, the items are transported to the CV_4 feeder.

In other words, the example PR_1 specification refers to a process sequence in which the R_1 robot picks up an object from feeder CV_1 and transfers it to buffer B_1 (from where it is automatically transferred to the M_1 machine). After the workpiece is handed over, it is processed from the B_1 buffer to the M_1 machine. When the operation on machine M_1 is completed, the workpiece is deposited to B_1. The workpiece deposited in B_1 (after processing on M_1) is then picked up by the robot R_1 and deposited into buffer B_2 of M_2. After the workpiece is handed over, it is processed from the B_2 to the M_2 machine. When the operation is completed, the workpiece is deposited in B_2. The workpiece deposited in B_2 (after processing on M_2) is then picked up by R_2 and deposited on the CV_2 receiver.

It is worth noting that the presented method of specification of production processes can be treated as a task-oriented language of communication between the operator (planner, dispatcher) and the DSS he used. In particular, the given specification of the processes can lead to unacceptable variants of production flows. In the case under consideration, this situation occurs when the first process occupying buffer B_1 is waiting for buffer B_2 to be released while the second process occupying buffer B_2 is waiting for buffer B_1 to be released.

4. PETRI NETS MODELING FRAMEWORK

To make the paper self-contained, let's enter a set of basic concepts constituting the Petri nets framework (Reisig, 1982; Reutenauer, 1988) used in the following two sections. A Petri net is formally defined as a six-tuple $PN = (P, T, E, W, K, M_0)$, where:

- $P = \{p_1, \dots, p_n\}$ and $T = \{t_1, \dots, t_m\}$ are the finite non-empty sets of places and transitions, such that $P \cap T = \emptyset$;
- $E \subset (P \times T) \cup (T \times P)$ is a flow relation, such that the following condition holds $\mathrm{dom}(E) \cup \mathrm{cod}(E) = P \cup T$;
- $W: E \to N$ is a weight function; the weight of one is assigned to an arc as a default;
- $K: P \to N$ is a place capacity function;
- $M_0: P \to N_0$ is the initial marking, $\forall p \in P, M_0(p) \le K(p)$.

The Petri net structure is a bipartite graph that comprises a set of places drowned as boxes, a set of transitions drowned as bars, and a set of arcs E. Places usually represent some conditions or resources. When the place represents a resource, it is assumed that a token in it means the machine's readiness to operate. Places may contain tokens that are drowned as black dots. Transitions represent events. Transitions transfer tokens from one place to another. During this process, called firing *t* transition, the tokens removed from their input places are stored in their output places.

An example of a reference Petri net model determined by the specifications of the production routes of the form (1) is shown in Fig. 2. Places $p_1 – p_8$ model the current locations of the moving elements – for example, $p_1(p_2)$ corresponds to the location of the element waiting in buffer B_1 for machining in the M_1 machine. Places p_9 and p_{10} map buffer states B_1 and B_2, respectively. The $p_{11} – p_{14}$ sites correspond to the FMM resource states (robots and machines). In particular, the p_{13} and p_{14} sites model the machine standby, M_1 and M_2, respectively, and the p_{11} and p_{12} robot standby, R_1 and R_2, respectively.

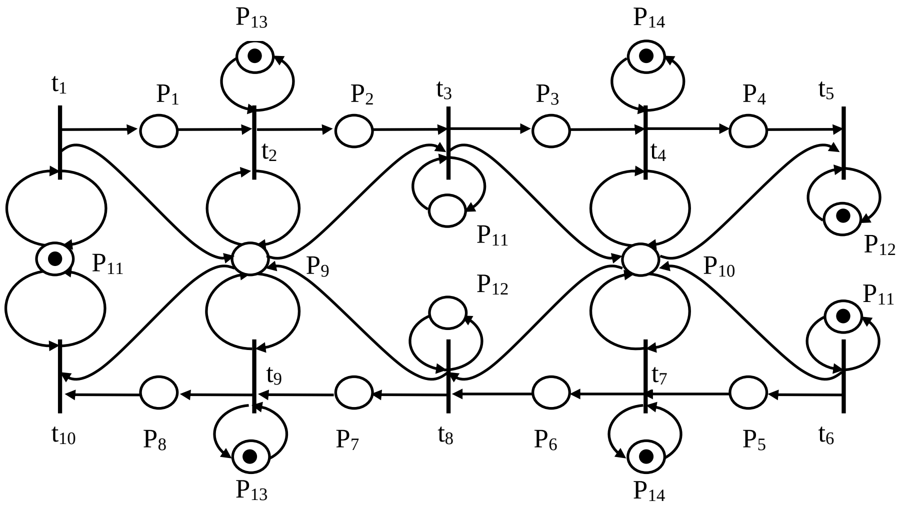

Fig. 2. Petri net-based reference model of FMM from Fig.1

Transitions t_1, t_3, t_6, and t_{10} correspond to the element movement operations carried out by the R_1 robot, while the t_5 and t_8 transitions are associated with the R_2 robot operations. The t_2 and t_9 transitions model the machining operations performed on the M_1 machine, and the t_4 and t_5 transitions correspond to the machining operations carried out on the M_2 machine.

The state of the Petri net usually called its marking, is defined by the number of tokens in each place and is denoted by vector $M = (M(p_1), \dots, M(p_i), \dots, M(p_n))$, where $n = ||P||$, is the cardinality of the set P. The number and position of tokens may change during the execution of a Petri net by firing transitions according to the following rules:

1. **Enabling Rule**: A transition t is said to be enabled when the following conditions hold:

– $M(p_n) \geq W(p, t)$, $\forall p \in \cdot\, t$,

– $M(p) \leq K(p) - W(p, t) + W(t, p)$, $\forall p \in \cdot\, t \cup (\cdot\, t \cap t\, \cdot)$, where $\cdot\, t = \{p | (p, t) \in E\}$,

– $t \cdot = \{p | (t, p) \in E\}$.

2. **Firing Rule**: An enabled transition t can fire, thus removing $W(p,t)$ tokens from each input place $p \in\cdot\, t$ and placing $W(t,p) - W(p,t)$ tokens in each output place $p \in t\,\cdot$.

The so-called reachability graph is used for the analysis of Petri net models. The graph's vertices (modelling states) represent a set $R(M_0)$ of states reachable in this network. For example, in Petri net from Fig. 3, the initial state is:

$$\begin{array}{cccccccccccccccc} & & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & p_{10} & p_{11} & p_{12} & p_{13} & p_{14} \\ M_0 & =(& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \quad) \end{array}$$

because $(\forall i = 1 \ldots 10)$ $(M_0(R_i) = 0)$ and $((\forall i = 11 \ldots 14)$ $(M_0(R_i) = 1)$. In this state, two transitions t_1 and t_6 are simultaneously enabled. The firing of transitions is random, and transitions that do not have common places may fire but not simultaneously. This limitation results from the principle of non-simultaneity of events adopted in physics.

The nodes (or states) of a Petri net's reachability graph represent the net's reachable markings. The set of states reachable in the Petri net from its initial marking M_0 is denoted as $R(M_0)$. Figure 3 shows the reachability graph of Petri net from Fig. 2.

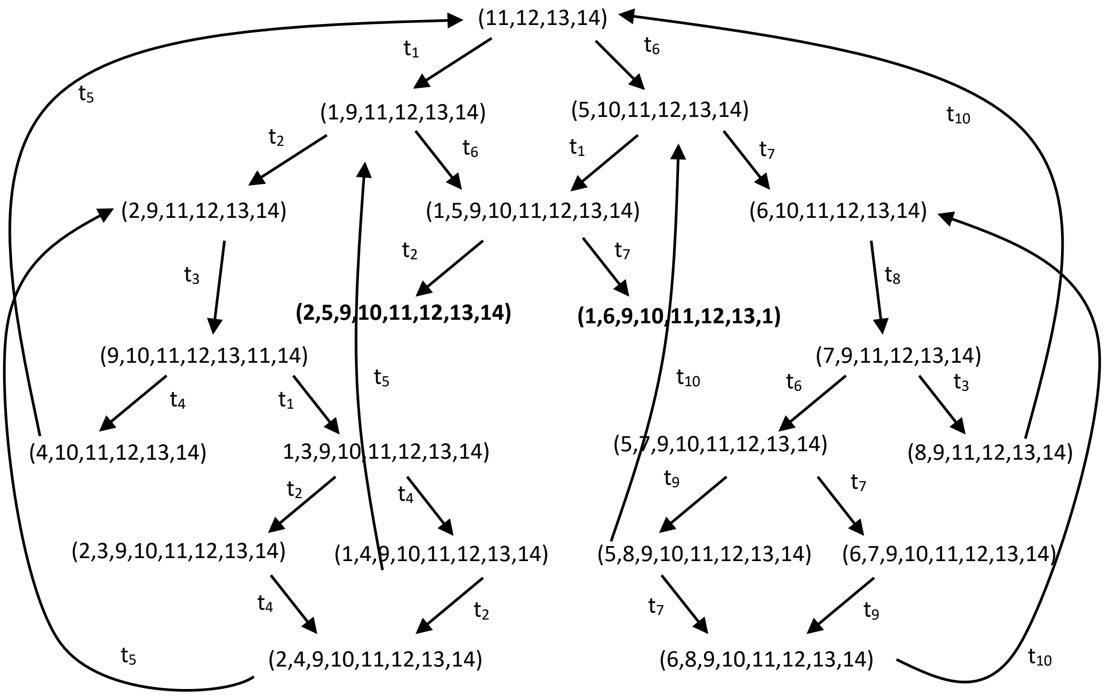

Fig. 3. The reachability graph of Petri net from Fig. 2. States in bold, i.e. (2,5,9,10,11,12,13,14) and (1,6,9,10,11,12,13,14) are states that illustrate deadlocks.

A simplified notation of states can increase the readability of the graph. In the adopted notation, we omit the positions taking the value "0" in the sequences specifying the states, and the corresponding coordinates are entered in place of the positions taking the value "1". In the adopted notation, the state $M_0 = (0,0,0,0,0,0,0,0,0,0,1,1,1,1)$ takes the form: $\boldsymbol{M_0} = (11,12,13,14)$.

Besides a reachability graph, the Petri net reachability analysis can be conducted through the state equation:

$$M' = M + e(i)C,$$

where:

- $e(i)$ – is a unit row-vector of size $1 \times m$, which is zero everywhere, except the i-th component corresponding to the transition t_i enabled at the marking M,
- $C = C^+ - C^-$ – is the Petri net incidence matrix defined as $n \times m$ matrix of c_{ij}'s, where $m = ||T||$ and $n = ||P||$, and

$$C^+ = (c_{ij})_{n\times m}, c_{ij} = \begin{cases} W(t_i, p_j) \text{ for } (t_i, p_j) \in E \\ 0 \text{ otherwise} \end{cases}$$

$$C^- = (c_{ij})_{n\times m}, c_{ij} = \begin{cases} W(p_j, t_i) \text{ for } (p_j, t_i) \in E \\ 0 \text{ otherwise} \end{cases}$$

The incidence matrix (i.e., an algebraic representation) provides a well-suited helpful form for the simulation of a net execution. In the case of the C matrix, it is easy to determine the same space of reachable states, as shown in Fig. 3.

C^+

	R_1	R_2	R_3	P_1	P_2
$O1_1$	1			1	
$O1_2$		1		1	
$O2_3$			1		1
$O3_1$	1				
$O3_3$			1		

$C = C^+ - C^-$

C^-

	R_1	R_2	R_3	P_1	P_2
$O1_1$	1				
$O1_2$		1			
$O2_3$			1	1	
$O3_1$	1				1
$O3_3$			1		1

The subject of the analysis of the created Petri net models is decision properties commonly addressed in the modelling of manufacturing systems, such as liveness, reachability, conservativeness, persistency, and boundedness. Their detailed description can be found in (Reisig, 1982; Reutenauer, 1988). The evaluation of prototype model variants carried out in this context aims to search for answers, among others, to questions such as: Whether the modelled system can reach a specific state as a result of required functional behavior? And: Whether a given transition t is live? To answer the first question formulated, we must find a transition firing sequence that would transform a marking M_0 to M_i, where M_i represents the specific state, and the firing sequence represents the required functional behavior. In this context, the paper is devoted to the problem of liveness-enforcing supervision in manufacturing systems where deadlocks arising from poor settlements of resource conflicts may arise.

Due to the problem NP-completeness of avoiding deadlocks in systems of concurrently executed processes competing in access to shared resources, the existing effective deadlock avoidance algorithms are based on sufficient conditions. These algorithms may lead to control strategies such that some of the allowable allocations of resources (which do not lead to a blockage of the processes using them) are omitted. In other words, using sufficient conditions to prevent the formation of blockages does not guarantee the maximum permissiveness of the considered class of processes. Therefore, in addition to differing in the number of accepted states of resource distribution, algorithms that implement different prevention conditions also differ in computational complexity. To sum up, taking into account additional information about the structure of concurrently executed processes allows one to use alternative methods of resolving resource conflicts dedicated to selected criteria for assessing the performance evaluation of the currently considered system. Among the most commonly relevant evaluation criteria are the following: degree of resource utilization, waiting time of processes for access to resources, flow time, inventory and manufacturing costs, makespan, unbalance, tardiness, and others.

The problem of deadlocks occurrence means that it becomes necessary adopting appropriate dispatching priority rules to prevent the deadlocks. In turn, the algorithms implementing such an approach are not maximally permissive, however, at the cost of low computational complexity (Yang & Hu, 2021). One such "deadlocks prevention" rule follows from the observation that the production flows follow the ordered lists of resources, indicating the sequences in which resources must be allocated to complete order execution. This assumption enables decomposing each production route into zones, where each zone is a sequence of sections consisting of shared resources followed by those consisting of unshared resources. Shared resources are used by more than one process, while other (unshared) resources are used by only one process.

The adoption of a rule prohibiting the simultaneous use of resources in individual sections of shared resources and a rule prohibiting the complete occupation of a zone of resources not shared by processes occurring in the subsequent busy section of repeating resources prevents the fulfillment of one of the four conditions necessary for the occurrence of a deadlock, i.e., the condition of a closed loop of resource requests. These rules guarantee that only resources in non-shared resource zones are used, i.e., states allowing all tasks to be completed. It is easy to see that leaving resources waiting for processes in such zones avoids deadlocks associated with situations in which all resources of all zones are used simultaneously (Banaszak, 1992; Claes & Tuyls, 2018; Reutenauer, 1988).

5. THE METHODOLOGY IMPLEMENTING THE DTDSC CONCEPT

Assuming that the execution times of individual machining operations and inter-station transport operations are known (as in Table 1) and remain unchanged for subsequent variants, it is possible to choose the solution that best meets the accepted expectations. In the experiments carried out, repeated for different priority selection rules, the same size of production batches was adopted. The experiments included planning serial (pipeline) production of two products carried out following two priority selection rules: Longest Processing Time (LPT) and Shortest Processing Time (SPT). The LPT (SPT) rule organizes tasks in the order of reducing (increasing) processing time. This approach means that each time a machine is released, the next task started on it is the longest (shortest) of the others ready. The LPT rule is most commonly used to determine the minimum cycle of a production process, and the SPT rule is used to determine the weighted time of production completion.

An illustration of the variants of the production flow corresponding to the above rules is presented in Fig. 4. The implementation of the LPT rule, compared to the SPT rule, allowed to shorten the production tact by 5 t.u. and consequently shorten the production cycle time by 20 t.u. It is also easy to see that using the LPT rule reduces the buffer capacity B_2 to $PB_2 = 1$.

Examples of computer-aided systems solutions enabling a comprehensive approach to modeling and assessing the effectiveness of FMS functioning, including flexible assembly systems, are presented in the papers (Viswandham & Narahari, 1992). The systems discussed there can support the user in the following tasks:

– design of FMS configuration (layout arrangement, selection, and placement of workstations, conveyors, buffers, and others),
– planning of production carried out in FMS (in terms of both tact and production cycle, as well as efficiency, effectiveness, and productivity).

Of course, the abovementioned issues do not exhaust all possible applications of similar tools based on Petri net models.

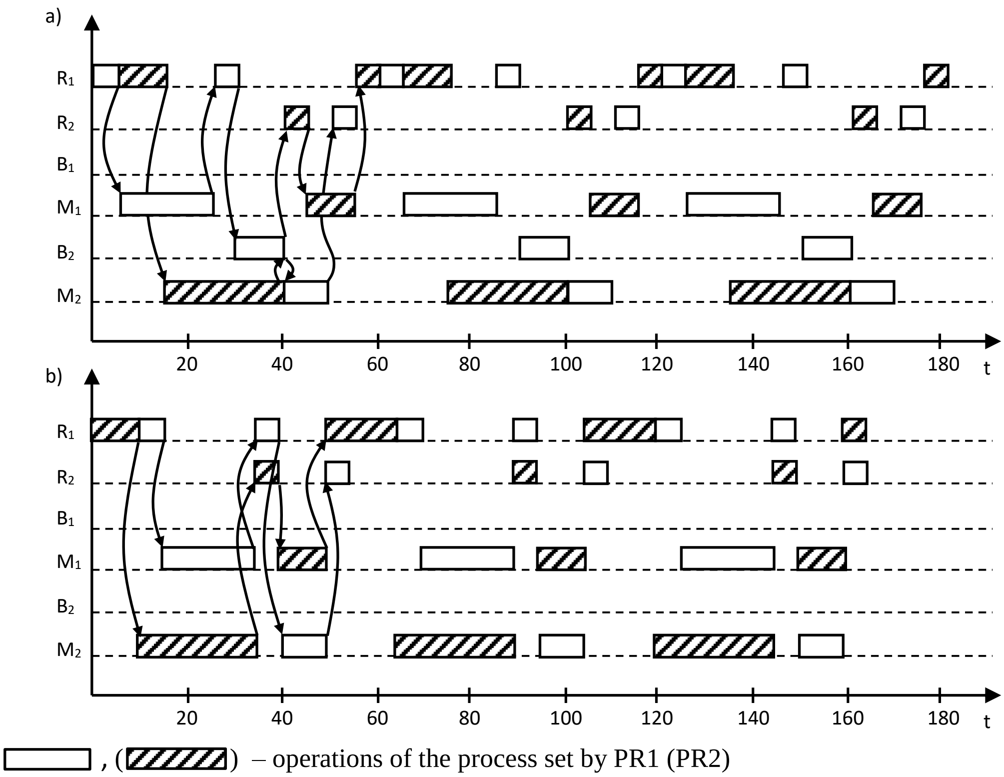

Fig. 4. Gantt diagrams of the given production flow in the accepted production route specification (1) carried out according to the rule of the shortest processing time (a) and according to the rule of the longest processing time b).

In the presented context, we notice that any attempt at a more comprehensive approach to FMS modeling, allowing for simultaneous analysis of different variants of the arrangement of its elements and acceptable scenarios of the flow of production processes implemented in them, leads to complex, non-linear problems of multi-criteria optimization. The proposed approach overcomes this limitation. Focusing on the resources of the system and the processes implemented with their help makes it possible to consider their various options but is limited only to feasible solutions. The guarantee that only admissible variants of the production flow can be evaluated shortens searching for a solution that meets the set expectations. A diagram of the iterative process of alternating data specification and evaluation of their results-oriented toward the search for a variant that meets the required evaluation criteria is presented in Fig. 5.

The idea of the methodology presented in Fig.5, implementing the DTDSC concept, boils down to iteratively run stages: planning the FMS configuration and planning the production carried out in it, as well as the correction of the production plan implemented. In short, it can be reduced to stages: automatic determination of the Petri net model of the considered class of processes set in the adopted notation, modification of the model, which boils down to the implementation of the mechanism of synchronization of modeled processes (guaranteeing their deadlock-free execution), implementation of an arbitrarily chosen method of resolving resource conflicts and assessment of the quality indicators implied by it. Consequently, the presented methodology provides a robust framework for simultaneous optimization of the production facility layout and the production flow scenarios.

At the first of these stages (elements highlighted in Fig. 5 with a solid line), the layout design of the system (in particular, the material handling system) and the order execution plan (in particular: the input sequencing the order in which parts of various types are released into the system, tact and cycle time, makespan, and production flow schedule) are determined.

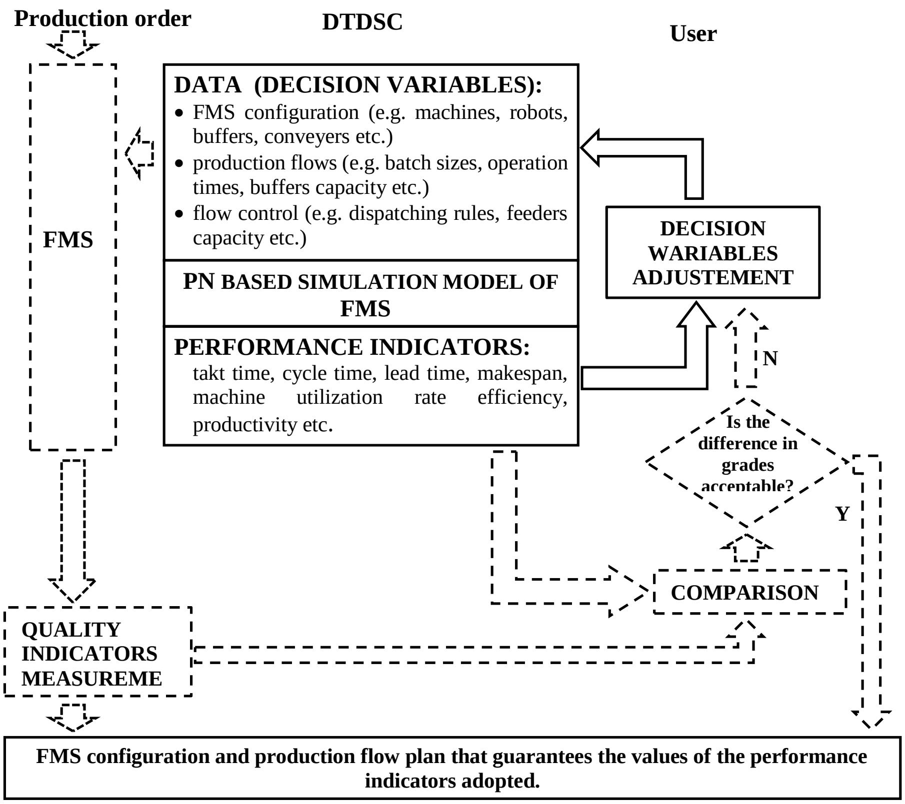

Fig. 5. Conceptual framework of the iterative searching process aimed at solution following the assumed criteria

At the next stage (including elements highlighted by a dashed line), based on the assessment of the adopted quality indicators of the implemented production course, or not the correction of the previously adopted plan is made.

It is worth noting that in the proposed search process, besides modelling control procedures coordinating cooperation of workstations, transport and storage equipment as well as robots and whole production processes, can also be determined schedules of workspaces, tools, waste, and auxiliary fastening devices flows as well as the functioning of modelled robots and auxiliary devices used in these processes.

6. CONCLUSIONS

Most of the methods used to model systems and the course of production processes carried out in them are based on techniques that implement Petri net formalisms. The advantage of this model type is the possibility of using computer simulation techniques to assess alternative scenarios of the modeled processes. Unfortunately, a significant shortage of such solutions, occurring in most studies of interactions of asynchronously occurring events, is associated with the suspension of simulation programs. The approach proposed in our paper fills this gap by offering the method of automatic synthesis of network models implementing mechanisms that prevent the deadlocks occurrence and consequently prevent the suspension of simulation programs. To sum up, our proposal significantly increases the effectiveness of appropriate decision support systems used in designing FMS class systems.

The presented methodology makes it possible to comprehensively cover various tasks of design and operation (control and management) of FMS. The proposed approach provides a robust framework for simultaneous optimization of the layout of machining centers, conveyors, robots, buffers, and the production flow scenario.

In the general case, it can be implemented in DSSs supporting the operational planning of production orders. It can also be used in online batching and routing production orders, resource allocation, and task scheduling, among many other applications.

However, among its more essential shortcomings, one should mention the lack of possibility to analyze manufacturing processes, in which flows of workpieces form the structure of a partially ordered graph (occurring, for example, in car assembly processes) and the lack of possibility to analyze the influence of stochastic disturbances. These issues can be addressed in future studies.

REFERENCES

Alexopoulos, K., Anagiannis, I., Nikolakis, N., & Chryssolouris, G. (2022). A quantitative approach to resilience in manufacturing systems. *International Journal of Production Research*, *60*(13), 4342–4360.

Bakar, B. A., Henry, R. M., & Ali, M. (1991). An alternative approach in batch process control implementation using hierarchical Petri nets, World Scientific. *Proc. of the International Conference on Computer Integrated Manufacturing* (pp. 171–174).

Banaszak, Z. (1992). Synchronisation of robots in flexible assembly systems. *Archiwum Budowy Maszyn*, *39*(1–2), 117–133.

Banaszak, Z., Skolud, B., & Zaremba, M. B. (2003). Computer-aided prototyping of production flows for virtual enterprise. *Journal of Intelligent Manufacturing*, *14*, 83–106.

Bocewicz, G., Wójcik, R., Witczak, M., & Banaszak, Z. (2022). Petri Net Approach to Automated Modelling and Performance Evaluation for Robotic Assembly Systems. *2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR)* (pp. 306–311). IEEE. https://doi.org/10.1109/MMAR55195.2022.9874291

Bujari, A., Calvio, A., Foschini, L., Sabbioni, A., & Corradi, A. (2021). A Digital Twin Decision Support System for the Urban Facility Management Process. *Sensors*, *21*(24), 8460. https://doi.org/10.3390/s21248460

Claes, D., & Tuyls, K. (2018). Multi robot collision avoidance in a shared workspace. *Autonomous Robots*, *42*, 1749–1770. https://doi.org/10.1007/s10514-018-9726-5

Coito, T., Faria, P., Martins, M. S. E., Firme, B., Vieira, S. M., Figueiredo, J., & Sousa, J. M. C. (2022). Digital Twin of a Flexible Manufacturing System for Solutions Preparation. *Automation*, *3*(1), 153–175. https://doi.org/10.3390/automation3010008

David, J., Lobov, A., & Lanz, M. (2018). Leveraging Digital Twins for Assisted Learning of Flexible Manufacturing Systems. *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)* (pp. 529-535). IEEE. https://doi.org/10.1109/INDIN.2018.8472083

Hatono, I., Katoh, N., Yamagata, K., & Tamura, H. (1989). Modelling of FMS under uncertainty using stochastic Petri Nets. *Proc. of the 3rd International Workshop on Petri nets and performance models* (pp. 122–130).

He, Z., Zhang, R., Ran, N., & Gu, C. (2022). Path Planning of Multi-Type Robot Systems with Time Windows Based on Timed Colored Petri Nets. *Applied Science*, *12*(14), 6878. https://doi.org/10.3390/app12146878

Heiner, M. (1992). *Petri net based software validation (prospects and limits), Technical report No. TR-92-022.* International Computer Science Institute.

Janardhanan, M. N., Li, Z., Bocewicz, G., Banaszak, Z., & Nielsen, P. (2019). Metaheuristic Algorithms for balancing robotic assembly lines with sequence-dependent robot setup times. *Applied Mathematical Modelling*, *65*, 256–270.

Jensen, K. (1987). Computer tools for construction, modification and analysis of Petri nets. *Lecture Notes on Computer Science* (No. 255). Springer Verlag.

Jonsson, P. (2000). An empirical taxonomy of advanced manufacturing technology. *International Journal of Operations & Production Management*, *20*(12), 1446–1474.

Laemmle, A., & Gust, S. (2019). Automatic layout generation of robotic production cells in a 3D manufacturing simulation environment. *Procedia CIRP*, *84*, 316–321.

Makris, S., Michalos, G., & Chryssolouris, G. (2012). Virtual Commissioning of an Assembly Cell with Cooperating Robots. *Advances in Decision Sciences*, *2012*, 428060. https://doi.org/10.1155/2012/428060

Manu, G., Kumar, V. M., Nagesh, H., Jagadeesh, D., & Gowtham, M. B. (2018). Flexible Manufacturing Systems (FMS): A Review. *International Journal of Mechanical and Production Engineering Research and Development*, *8*(2), 323–336.

Neto, A. A., Carrijoa B. S., Brock, J. G. R, Deschamps, F., & Lima, E. P. (2021). Digital twin-driven decision support system for opportunistic preventive maintenance scheduling in manufacturing. *Procedia Manufacturing*, *55*, 439–446.

Nielsen, L. D., Sung, I., & Nielsen, P. (2019). Convex Decomposition for a Coverage Path Planning for Autonomous Vehicles: Interior Extension of Edges. *Sensors*, *19*(19), 4165. https://doi.org/10.3390/s19194165

Nielsen, P., Michna, Z., & Do, N. A. D. (2014). An Empirical Investigation of Lead Time Distributions. *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World. APMS 2014. IFIP Advances in Information and Communication Technology* (vol. 438). Springer. https://doi.org/10.1007/978-3-662-44739-0_53

Patalas-Maliszewska, J., & Kłos, S. (2019). An Approach to Supporting the Selection of Maintenance Experts in the Context of Industry 4.0. *Applied Sciences*, *9*(9), 1848. https://doi.org/10.3390/app9091848

Rachamadugu, R., & Stecke, K. E. (1994). Classification and review of FMS scheduling procedures. *Production Planning & Control*, *5*(1), 2–20. https://doi.org/10.1080/09537289408919468

Recalde, L., Silva, M., Ezpeleta, J., & Teruel, E. (2022). Petri Nets and Manufacturing Systems: An Examples-Driven Tour. *ACPN 2003. Lecture Notes in Computer Science* (vol. 3098). Springer. https://doi.org/10.1007/978-3-540-27755-2_21

Reisig, W. (1982). *Petri nets*. Springer Verlag.

Reutenauer, Ch. (1988). *The mathematics of Petri nets*. Englewood Cliffs.

Silva, E. B., Costa, M. G., Silva, M. F., & Pereira, F. H. (2012). Evaluation of Production Sequencing Rules in Job Shop and Flow Shop Environment through Computer Simulation. *ICIEOM 2012* (no. 257).

Sliwa, M., & Patalas-Maliszewska, J. (2016). A Strategic Knowledge Map for the Research and Development Department in a Manufacturing Company. *Foundations of Management*, *8*(1), 151–166.

Stączek, P., Pizoń, J., Danilczuk, W., & Gola, A. (2021). A digital twin approach for the improvement of an autonomous mobile robots (AMR's) operating environment – a case study. *Sensors*, *21*(23), 7830. https://doi.org/10.3390/s21237830

Świć, A., & Gola, A. (2013). Economic Analysis of Casing Parts Production in a Flexible Manufacturing System. *Actual Problems of Economics*, *141*(3), 526–533.

Vaisi, B. (2022). A review of optimization models and applications in robotic manufacturing systems: Industry 4.0 and beyond. *Decision Analytics Journal*, *2*, 100031. https://doi.org/10.1016/j.dajour.2022.100031

Van der Aalst, W. M. (1992). *Timed coloured Petri nets and their application to logistics*. Technische Universiteit Eindhoven.

Viswandham, N., & Narahari, Y. (1992). *Performance modelling of automated manufacturing systems*. Prentice-Hall.

Yang, B., & Hu, H. (2022). Maximally Permissive Deadlock and Livelock Avoidance for Automated Manufacturing Systems via Critical Distance. In *IEEE Transactions on Automation Science and Engineering*. IEEE. https://doi.org/10.1109/TASE.2021.3138169

Zanchettin, A. M. (2021). Robust scheduling and dispatching rules for high-mix collaborative manufacturing systems. *Flexible Services and Manufacturing Journal*, *34*, 293–316. https://doi.org/10.1007/s10696-021-09406-x

Zhang, F., Bai, J., & Yang, D. (2022). Digital twin data-driven proactive job-shop scheduling strategy towards asymmetric manufacturing execution decision. *Scientific Reports*, *12*, 1546. https://doi.org/10.1038/s41598-022-05304-w

Zhou, K. Q., & Zain, A. M. (2016). Fuzzy Petri nets, and industrial applications: a review. *Artificial Intelligence Review*, *45*(4), 405–446. https://doi.org/10.1007/s10462-015-9451-9