Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Structural Decomposition Approach to Design of No-Wait Cyclic Schedules for Repeatedly Operating Transport System Dedicated to Supply Loops

Robert Wójcik*, Arkadiusz Gola, Jarosław Pempera***, Justyna Patalas-Maliszewska******

**Department of Computer Engineering, Faculty of Electronics, Wrocław University of Science and Technology, Wrocław, Poland (e-mail: robert.wojcik@pwr.edu.pl).*
***Institute of Technological Systems of Information, Lublin University of Technology, Lublin, Poland (e-mail:a.gola@pollub.pl).*
****Department of Control Systems and Mechatronics, Faculty of Electronics, Wrocław University of Science and Technology, Wrocław, Poland (e-mail: jaroslaw.pempera@pwr.edu.pl).*
*****Institute of Mechanical Engineering, University of Zielona Góra, Zielona Góra, Poland (e-mail: j.patalas@iizp.uz.zgora.pl).*

Abstract: The paper presents a method allowing to construct no-wait cyclical schedules for repetitive transport systems (e.g. the milk-run) servicing cyclic material supply loops in the production system using selected means of transport (e.g. AGVs). The transport means are following established routes and given arrival times. The routes are composed of sectors linking workstations. Transport trolleys may share specific sectors of the route in mutual exclusion mode and must wait in a given sector to enter the next sector of the route when another trolley occupies it. The job-shop repetitive transportation system is a system of cyclic processes with a fixed structure that are executing sequences of operations (routes) using shared resources (sectors). The work aims to find a no-wait cyclic schedule that guarantees the required delivery dates or establish that such a schedule does not exist. It considers cyclic process systems for which each resource can be used by at most two operations, and the deadlock state cannot occur as a result of waiting processes on shared resources. For specified initial operations of cyclic processes and their start times (the initial system state), the problem of determining no-wait cyclical schedules decomposes into subproblems. Each subproblem consists of the verification of necessary and sufficient conditions for the existence of solutions for each of 2-process subsystems composed of one shared resource and two processes using this resource. The method aims of prototyping various variants of process starting times for which the conditions guaranteeing no-wait property of the system hold simultaneously for each of the 2-process subsystems. It allows designing cyclic schedules for complex systems composed of 2-process subsystems that are structurally deadlock-free. The class of cyclical processes considered in this article is broader than the class of cascade-like (chain-like, sequential) process systems analysed so far in the literature. In this context, the results obtained are an extension of the existing ones.

Keywords: repetitive delivery systems, milk-run routing and scheduling, vehicle routing problem, job-shop type transportation systems, resource blocking and deadlock problem, no-wait cyclic schedule, two cyclic processes with a shared resource, decomposition of the system structure.

1. INTRODUCTION

Continuous development and the efficiency increase of production systems that perform cyclical tasks related to the concurrent implementation of a series of products it requires solving some problems. These problems are related to the design of technological and transport routes (Pinedo, 2005), supply chain and transport fleet size planning (Patel et al., 2014; Bocewicz et al., 2019) and capacity planning of storage buffers (Bocewicz et al., 2014; Sitek et al., 2019), as well as various problems of the production jobs optimal scheduling (Brucker et al., 2008; Levner et al., 2010).

The most frequently solved optimization problems in the scope of jobs scheduling in repetitive production systems are formulated as minimizing the system cycle time (Abadi et al., 2000; Smutnicki, 2009). Also, decision-making problems arise in searching for answers to the question about the existence of cyclical schedules with cycle times not exceeding the required value for a given system (Pinedo, 2005; Bocewicz et al., 2014; Zhang et al., 2019), or schedules for systems with fuzzy processing times constraints (Bocewicz, 2014; Bocewicz et al., 2016).

In many cases, additional problems constraints are taken into account to increase the efficiency of the production system. The constraints take into account the number and capacity of resources used, e.g. storage buffers (Abadi et al., 2000; Smutnicki, 2009). Also transport vehicles (Bocewicz et al., 2014; Patel et al., 2014), and the selection of storage capacity for materials, and synchronization of delivery dates in a way that eliminates machine downtime (Sitek et al., 2019; Hall et al., 1996; Mascis et al., 2002; Wójcik, 2018). Solving the appropriate problems of constraints satisfaction leads to the optimization of production systems' operation taking into account selected evaluation criteria (Abadi et al., 2000; Pinedo, 2005; Kampmeyer, 2006; Smutnicki, 2009; Zhang et al., 2019).

One of the methods of optimizing the operation of inter-workstation transport in repeatable production systems is the milk-run method (Patel et al., 2014; Bocewicz et al., 2019). The assumption of the method is the cyclical delivery of appropriate amounts of materials to workstations by specific means of transport (e.g. AGVs, tugger trains, logistic trains) to guarantee the flow of production tasks with a fixed production cycle. In this case, each mean of transport is loaded and unloaded multiple times during one trip along its fixed route that connects several workstations into a supply loop. During supply tours, logistic trains deliver a variable quantity of materials and products to the particular workstations at regular time intervals. The milk-run set-up allows reducing variability by running tours that are cyclically repeated according to a fixed sequence of operations and fixed cyclic schedule, and simultaneously allow to reduce inventory and storage buffer capacity within a supply network.

In milk-run systems, wheelchair routes consist of sectors that are successively visited by wheelchairs at fixed arrival times. Some sectors can be shared by multiple cars, which can lead to a wheelchair in its sector waiting to enter the next sector occupied by another wheelchair. Due to the lack of additional buffers between sectors within the transportation routes, sectors can become blocked, and the vehicles waiting in sectors can occur, which in turn leads to delays in arrival times in sectors and disruption of the agreed delivery schedule (AitZai et al., 2012; Allahverdi et al., 2016; Aschauer et al., 2017; Louaqad et al., 2018). The lack of resources requires the inclusion of additional constraints in the classic cyclic scheduling problems of milk-run systems and other repetitive transport systems. The restrictions to be taken into account relate to blocking resources by transport jobs (no-store, no-buffer constraint) and no-wait constraint. The last restriction means that each job does not wait for any transport or production operation (Hall et al., 1996; Schuster et al., 2003; Mascis et al., 2002; Brucker et al., 2008; Wójcik, 2018; Wójcik et al., 2019).

In a cyclic job-shop transportation system with resource blocking jobs that perform operations may wait for the resources and some of them may be deadlocked (Banaszak et al., 1990). For systems with deadlock possibility (Bocewicz et al., 2014, 2019), even the problem of any cyclic schedule design is a difficult one since the solution may not exist.

The objective of this paper is to develop a method that allows fast prototyping of no-wait cyclic schedules for repetitive transport system with fixed transportation routes and fixed operations times. Our goal is to calculate the start times of the transport jobs that belong to the no-wait cyclic schedule of the system? The presented results extend the method used to determine no-wait cyclic schedules for systems consisting of n cyclic processes sharing one resource (Wójcik, 2018), and cascade-like (chain-like) systems (Fig. 2) (Wójcik et al., 2019) to structurally deadlock-free systems (definition - see next section) of any configuration, being a composition of the extended 2-process subsystems (Fig. 3).

1.1 Properties of transport systems considered

The cyclical transport tasks are using trolleys (e.g. AGVs) enabling the handling of supply loops. The loops consist in the delivery of specific materials and products to workstations following rigidly determined transport routes and at fixed moments resulting from the times of operations carried out in sectors belonging to particular routes (Fig.1).

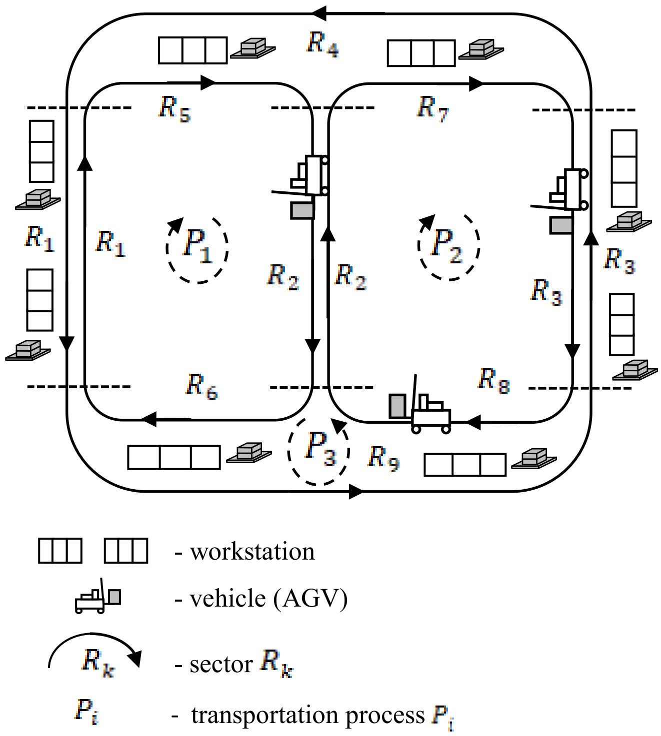
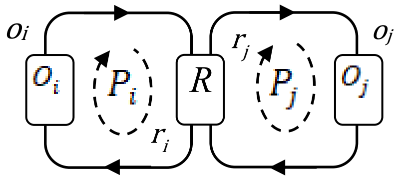


Fig. 1. A transportation system with repetitive tasks.

Defined is a model, in which each transport task being a cyclical process, which involves a sequence of operations, where each operation requires access to a specific resource (sector) of the production system. In particular, the process represents a means of transport (e.g. AGV, tugger train, logistic train) that uses the resources defined in its route in a cyclical way. Each resource is used by a process on an exclusive basis (unit capacity of the resource) for a specific period resulting from the time of the trolley's journey and the loading/unloading time. A process (vehicle) can wait for a resource (sector) if it is used (occupied) by another one (Bocewicz et al., 2014; Wójcik et al., 2019).

The paper considers a problem of determining no-wait cyclic schedules for cyclical process systems (CPSs) being free from deadlock, concerning waiting on resources. The other assumption is using each resource by no more than two processes. The two processes (2-P), consisting of two cyclic processes sharing a resource (Fig.2) (Alpan et al., 1997; Wójcik, 2001), are examples of process systems that meet the presented limitations. Other ones are systems of cyclic processes with cascade-like (C-L) (chain-like, sequential) structure (Alpan et al., 1998; Zaremba et al., 1998; Wójcik, 2001, 2018) that consist of several 2-P subsystems creating a chain structure such that neighbouring subsystems have only one shared resource. The class of cyclic processes considered in this work is a broader class than C-L systems (Wójcik et al., 2019). It includes systems of processes with any configurations that are structurally free of blockades, and in which each shared resource may be used by only two processes (two operations). Thanks to the last property, it is possible to analyse process systems of this type using 2-P systems properties.



P_i - cyclic process

R - shared resource

O_i - unshared resource

r_i - time of using the shared resource

o_i - time of using the unshared resource

Fig. 2. A basic system of two cyclic processes (2-P).

The objective of this work is to present a fast method which allows determining all no-wait cyclic schedules for a given system of the cyclic process with fixed routes and fixed operation times, or stating, that it is not possible to construct such schedules (there is no initial state of the processes that belong to a no-wait cyclic schedule).

1.2 Review of the literature

Results on cyclic scheduling with blocking constraint (no-store, no-buffer) and (or) no-wait constraint concern examples of different variants of job-shop systems (AitZai et al., 2012; Aschauer et al., 2017; Louaqad et al., 2018; Mascis et al., 2002) and flow-shop (Abadi et al., 2000; Allahverdi et al., 2016; Hall et al., 1996) and using different numbers of resources (machines) (Kumar et al., 2000; Levner et al., 2010). Both, cyclic scheduling problems (Kamoun et al., 1993; Brucker et al., 2008; Kampmeyer, 2006; Smutnicki, 2009; Levner et al., 2010; Bocewicz et al., 2014) and non-cyclic problems (e.g. Hall et al., 1996; Kumar et al., 2000; Schuster et al., 2003; Schuster, 2006; Allahverdi et al., 2016) are analysed.

The cyclic scheduling problems with no-buffer and (or) no-wait constraints are NP-hard ones (Levner et al., 2010; Kamoun et al., 1993), so that determination of a deadlock-free schedule is, in general, a computationally difficult problem, and for many cases with no solution (Banaszak et al., 1990; Kampmeyer, 2006; Bocewicz et al., 2014). Methods and algorithms used for solving cyclic and non-cyclic job-shop scheduling problems take into account conditions sufficient for the existence of solutions as well as different types of heuristics (Abadi et al., 2000; Aschauer et al., 2017) and metaheuristics (Smutnicki, 2009; Schuster et al., 2003; Schuster, 2006). They allow us to find suboptimal solutions, which satisfy given constraints. Some approaches are graph-based methods (Mascis et al., 2002; Brucker et al., 2008; Louaqad, 2018), extensions of the critical path approach (Smutnicki, 2009), and block analysis (Kampmeyer, 2006), variants of Tabu Search (Schuster, 2006; Smutnicki, 2009), a branch and bound method (AitZai et al., 2012), genetic algorithms (Kumar et al., 2000), and linear programming (Smutnicki, 2009; Louaqad et al., 2018). In our previous works, linear modulus equations are used to design cyclic schedules for systems of n cyclic processes sharing one resource (Wójcik, 2018), and for C-L (Fig.2) systems (Wójcik et al. 2019). In this work, these results are extended for more complex systems of cyclic processes.

1.3 Method of decomposition and analysis

The problem of determining cyclic schedules considered in this paper is limited to a subclass of job-shop problems, for which the graph describing the structure of resource requests of processes executing a given sequence of operations does not contain cycles composed of shared resources only. Additionally, each resource can be used by at most two processes. The method can also be applied for cyclic process systems (CPSs) with a quasi-cascading (sequential, chain-like) structure (Zaremba et al., 1998; Alpan et al., 1998; Wójcik, 2001, 2018; Wójcik et al., 2019) where such cycles occur but do not lead to deadlocks. Process systems of this type can be decomposed into extended 2-PE subsystems (Fig.3), with two cyclic processes sharing one resource and executing at least two operations, which can be independently analysed. Finding the initial state of the processes in which each of the 2-PE subsystems of the cyclic process system (CPS) satisfies the conditions guaranteeing the existence of a no-wait cyclic schedule (Wójcik, 2018; Wójcik et al., 2019) means that such a schedule also exists for the CPS system. The developed structural decomposition method consists in the analysis of process operation start times that meet the no-wait constraints defined for each 2-PE subsystem. It is worth to notice that a number of analysed subsystems is small since it is not exceeding the number of resources.

If we want to find all possible no-wait cyclic schedules, it is necessary to check the conditions guaranteeing their existence for all possible values of the starting times of the cyclic processes. The number of such variants is equal to the product of the cycle times of the component processes, which may lead to a calculation time that increases exponentially with the number of processes. In this paper, the approach developed illustrates an example.

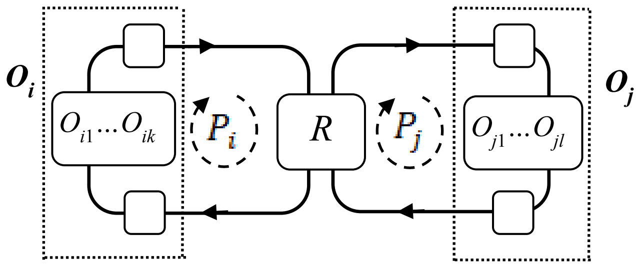


Fig. 3. The extended system of two cyclic processes (2-PE).

The novelty of the proposed method of no-wait cyclic schedules design consists in:

- decomposition of the problem of determining no-wait cyclic schedules in CPSs sharing resources in the mutual exclusion mode, in which there is no deadlock state possible, into subproblems consisting in the analysis of conditions guaranteeing the existence of no-wait cyclic schedules for extended 2-process subsystems (2-PE);

- applying to the construction of constraints guaranteeing the existence of start-up times of processes which belong to the no-wait cyclic schedules, the *necessary and sufficient* conditions for the existence of no-wait cyclic schedules for 2-PE subsystems; this is an extension to traditional methods using the sufficient conditions for the existence of cyclic schedules;

- enabling quick verification of the conditions for the existence of no-wait cyclical schedules by checking constraints whose number does not exceed the number of resources shared in the system;

- taking into account any configuration of cyclic processes, in which each shared resource is used precisely by two processes, and the graph defining the structure of resource requests of processes does not contain cycles composed of shared resources only.

In Section 2 system of cyclic processes sharing resources is defined as well as a problem of designing no-wait cyclic schedules. Section 3 presents a method of cyclic schedules design using decomposition of the system into 2-process extended (2-PE) subsystems and checking the necessary and sufficient conditions for the existence of the no-wait schedules for each component subsystem, which is the necessary condition for the existence of a no-wait cyclic schedule for the whole system. Section 4 contains an example of schedules design using the method. For a system of cyclic processes with fixed parameters (i.e. given structure and operation times), we calculate all possible starting times of the processes for which no-wait cyclic schedules exist and present one of the solutions. Finally, Section 5 presents conclusions.

2. PROBLEM FORMULATION

A system of cyclic processes CPS = $\{P_1, \ldots, P_i, \ldots, P_n\}$, ($i$=1,2,...,$n$) is considered. Each process P_i, is corresponding to one repetitive transportation task (Fig.1). The process follows a sequence of operations $ZO_i = (O_{i1}, \ldots, O_{ik(i)})$ periodically using different resources defined by $Z_i = (R_{i1}, \ldots, R_{ik(i)})$, where $k(i)$ denotes the number of P_i process operations, $O_{ij} \in OP(i)$, and $OP(i)$ is a set of operations executed by P_i (defined by ZO_i). $OP = (\cup_{i=1,\ldots,n})OP(i) = \{O_1, \ldots, O_{lo}\}$ is a set of the operations (lo – a number of the operations), $R_{ij} \in RE$, and $RE = \{R_1, \ldots, R_{lr}\}$ is a set of resources (lr – a number of the resources), each one of unit capacity (i.e. only one vehicle can be present at any sector). The set $RE = RES \cup REU$, where $RES = \{Q_1, \ldots, Q_{ls}\}$ is a set of resources shared by the processes (ls – the number of shared resources), REU is a set of unshared resources. The sequence $ZT_i = (r_{i1}, \ldots, r_{ik(i)})$ denotes operations times and a cycle time of process P_i is equal to $c_i = \Sigma_{k=1,\ldots,k(i)}(r_{ik})$, where $r_{ik} \in N$ are uniform time units given as natural numbers (N – a set of natural numbers).

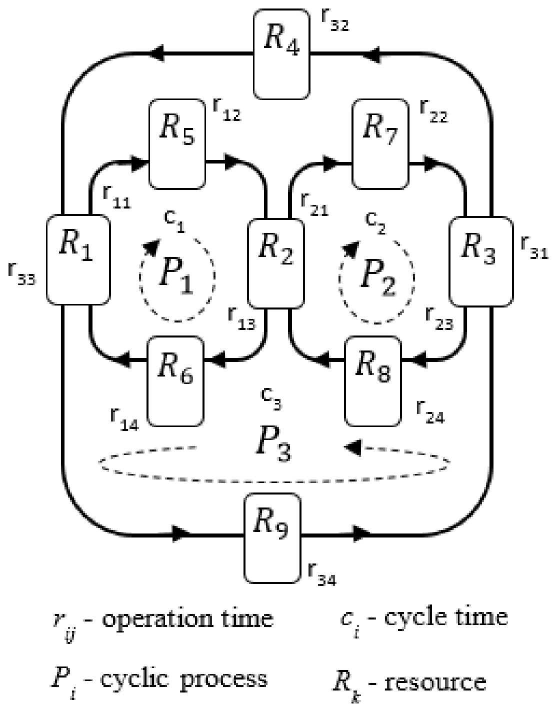


Fig. 4. A system of cyclic processes sharing resources.

For instance the CPS = (P_1, P_2, P_3) shown in Fig.4 uses nine resources. The resources R_1, R_2, R_3 that are used by at least two processes are defined as shared ones. In the opposite, the resources R_4, R_5, R_6, R_7, R_8, R_9 are unshared ones because each one is exclusively used by only one process. The processes P_1, P_2, P_3 are executing operations: $ZO_1 = (O_{11},O_{12},O_{13},O_{14})$, $ZO_2 = (O_{21},O_{22},O_{23},O_{24})$, $ZO_3 = (O_{31},O_{32}, O_{33},O_{34})$, using resources given by the sequences: $Z_1 = (R_{11},R_{12},R_{13},R_{14}) = (R_1,R_5,R_2,R_6)$, $Z_2 = (R_{21},R_{22},R_{23},R_{24}) = (R_2,R_7,R_3,R_8)$, and $Z_3 = (R_{31},R_{32},R_{33}, R_{34}) = (R_3,R_4,R_1,R_9)$.

Let $R(O_{ij}) \in RE$ denotes a resource that is used by the operation $O_{ij} \in OP$. A directed graph $G = (V, E)$, where $V = RE$ is a set of vertices, and $E \subseteq V \times V$ is a set of edges such that for any R_a, $R_b \in V$ occurs: $(R_a,R_b) \in E$ iff $(\exists i \in \{1,\ldots,n\})$

$(\exists j\in\{1,...,k(i)-1\})\{\ (\ \exists O_{ij},O_{i(j+1)}\in OP(i)\ )[\ R_a=R(O_{ij})\ \&\ R_b=R(O_{i(j+1)})\ ]$ or $(\ \exists O_{ik(i)},O_{i1}\in OP(i)\ )[R_a=R(O_{ik(i)})\ \&\ R_b=R(O_{i1})]\}$, defines the structure of the process system, i.e. the structure of the processes resource requests (Fig.4).

In this work are considered systems of processes such that their G-graphs do not contain cycles composed of shared resources only, i.e. the subgraphs of G limited to vertices belonging to the RES set are acyclic. This condition guarantees that the considered systems of processes are structurally deadlock-free (Banaszak et al., 1990; Wójcik, 2018). Also, we assume that each shared resource is used by two processes only. This latter assumption allows us to reduce the number of constraints that guarantee the existence of no-wait cyclic schedules to the number of shared resources included in the set RES. The proposed method of system analysis is an extension of the method presented for cascade-like CPSs, whose G-graphs may contain cycles composed exclusively of shared resources (Wójcik et al., 2019).

One can note that CPS may be analysed as a composition of extended subsystems composed of two cyclic processes (2-PE) sharing one resource (Fig.3). In the case of the system shown in Fig.4, we have the following 2-process subsystems: 2-PE$_{1(13)}$ - the subsystem of processes SP_1 = $\{P_1,P_3\}$ sharing R_1, and 2-PE$_{2(12)}$ - the processes SP_2 = $\{P_1,P_2\}$ sharing R_2, and 2-PE$_{3(23)}$ - the processes SP_3 = $\{P_2,P_3\}$ sharing R_3 (where indices $i(kl)$ of the subsystem 2-PE$_{i(kl)}$ denote: i – shared resource number; kl – numbers of processes sharing resource R_i, and SP_i - a set of processes sharing R_i).

There are cyclic processes in the CPS (Fig.4) that use more than one shared resource (e.g. P_1 uses R_1 and R_2, Fig.4), i.e. resources used in different 2-PE subsystems. Then, in the CPS process system, the following situation may occur: the P_1 process, which uses the R_1 resource and requires R_2 resource for the next operation, and the P_2 process, which uses the R_2 resource and requires R_1 resource for the next operation, are completing the first operations at the same time. In this case, none of the processes waited for the resource to operate; however, the processes cannot perform subsequent operations, because they block each other's access to the resources necessary to perform these operations (i.e. P_1 blocks R_1 and P_2 blocks R_2), which leads to the system's deadlock. In the considered class of CPSs, this not happens, as they are structurally deadlock-free. Thus, the necessary and sufficient condition for a no-wait cyclic schedule to exist in a CPS concerns the property of the initial state of the processes (i.e. the start times of the jobs). It should be such that none of the 2-PE subsystems of the CPS using the shared resource of the RES set may have a process waiting for resources. The appropriate condition uses the conditions necessary and sufficient for the existence of no-wait cyclical schedules of basic 2-P systems (Fig.2) developed in our previous works (Wójcik, 2001, 2018; Wójcik et al., 2019).

The problem considered in this paper is as follows: For a cyclic process system (CPS) with a fixed structure set by the system's parameters determine the start times of cyclical processes (if any) for which there is no need for processes to wait for the beginning of the operations. The no-wait condition must hold for each of 2-PE subsystems of the CPS.

3. DESIGN METHOD USING DECOMPOSITION

The process system CPS=$\{P_1, \ldots, P_n\}$, which is structurally deadlock-free, is analysed as a composition of 2-PE = $\{P_i,P_j\}$, subsystems where $P_i,P_j\in$CPS are sharing one resource (Fig.3). The necessary and sufficient conditions for the existence of no-wait cyclic schedules for the 2-PE subsystem, in which the shared resource $R=R_{ia}=R_{jb}$ & $R_{ia},R_{jb}\in RES$ is used by two processes $\{P_i,P_j\}$, where $Z_i = (R_{i1}, \ldots, R_{ia}, \ldots, R_{ik(i)})$, and $Z_j = (R_{j1}, \ldots, R_{jb}, \ldots, R_{jk(j)})$, will be determined on the basis of appropriate conditions developed for the 2-P subsystem (Wójcik, 2001, 2018; Wójcik et al., 2019).

3.1 Conditions for no-wait execution of cyclic processes

A basic 2-P subsystem $SP_R = \{P_i,P_j\}$ is a composition of two cyclic processes sharing a resource R (Fig.2). The processes P_i and P_j execute periodically sequences Z_i, Z_j of the operations using unshared resources O_i, O_j, and shared resource R (Fig.2). Let us assume that the processes start with the operations using the shared resource R. $ZT_i = (r_i, o_i)$, where r_i, $o_i\in N$, define the operation times, and a cycle time of P_i is equal to $c_i = r_i + o_i$. Let $x_i(k)$, $x_j(l)\in N\cup\{0\}$, k,l=0,1,2,...,∞ denote the times of starting operations using the shared resource R in subsequent iterations of the processes P_i, P_j. We assume that $0 \le x_i(0)$ and $0 \le x_j(0)$ are the process start-up times at the initial state of the system (i.e. starting times of the first operations of the first cycles). In case of a no-wait 2-P subsystem processes can be analysed as if they were performing independently and start times of operations using the shared resource can be calculated according to formulas: $x_i(k) = x_i(0) + k\cdot c_i$ and $x_j(l) = x_j(0) + l\cdot c_j$ (Fig.5).

Let $t_{ij}(l)\in N\cup\{0\}$ define a local start times $t_{ij}(l)\in[0,c_i)$ (1) of process P_j, calculated in relation to resource allocation times $x_i(k)$ of P_i, where k,l=0,1,2,...,∞. The process P_j starts an operation with the resource R at the moment $x_j(l) = x_i(k) + t_{ij}(l)$, such that $x_j(l)\in[x_i(k), x_i(k)+c_i)$, and the following starting times are defined by $x_j(l+1) = x_j(l) + c_j$ (Fig.5). It can be shown (Wójcik, 2001, 2018; Wójcik et al., 2019) that in a no-wait basic system of two cyclic processes $SP_R = \{P_i,P_j\}$, sharing the resource R, process P_j may start to use shared resource only at times $x_j(l) = x_i(k) + t_{ij}(l)$, such that local times $t_{ij}(l)\in[0,c_i)$ can be expressed by a relationship:

$$t_{ij}(l) = f_{ij}\cdot D_{ij} + y_{ij} \qquad (1)$$

where: $D_{ij}=D_{ji}=gcd(c_i,c_j)$ & $c_i=D_{ij}m_{ij}$ & $c_j=D_{ji}m_{ji}$ &
& $gcd(m_{ij},m_{ji})=1$ & m_{ij}, $m_{ji}\in N$ & $f_{ij}\in\{0,1, \ldots, m_{ij}-1\}$ &
& $y_{ij}\in[r_i, D_{ij} - r_j]$ & gcd - the greatest common divisor.

The constraint (1) ensures that resource requests $x_j(l)$ of process P_j, concerning the shared resource, may occur only within some intervals $[x_i(k)+r_i, x_i(k)+c_i-r_j]$, such that $r_i \le t_{ij}(l) \le c_i - r_j$, which guarantees that no processes wait for the resource.

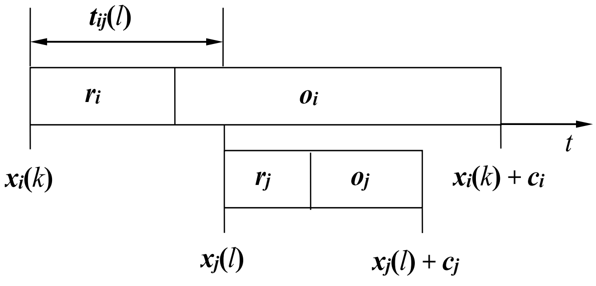


Fig. 5. Resource allocation times.

It can be shown (Wójcik, 2001, 2018) that if the relation (1) holds then also formula describing local start times $t_{ji}(k)\in[0,c_j)$ of the process P_i at time intervals $x_i(k)\in[x_j(l), x_j(l)+c_j)$, and $x_i(k) = x_j(l) + t_{ji}(k)$, holds:

$$t_{ji}(k) = f_{ji} \cdot D_{ji} + y_{ji} \tag{2}$$

where: $D_{ij} = D_{ji} = gcd(c_i,c_j)$ & $c_i = D_{ij}m_{ij}$ & $c_j = D_{ji}m_{ji}$ & & $f_{ji} \in \{0,1, \ldots, m_{ji}\text{-}1\}$ & $y_{ji}\in[r_j, D_{ji} - r_i]$ & $y_{ji} = D_{ij} - y_{ij}$.

The values y_{ij} (1) and y_{ji} (2) exist if the intervals $[r_i, D_{ij} - r_j]$ and $[r_j, D_{ji} - r_i]$ are not empty, i.e. when $r_i \le D_{ij} - r_j$, and $r_j \le D_{ji} - r_i$, which is equivalent to the necessary and sufficient condition for existence of a no-wait cyclic schedule for the subsystem (P_i,P_j) such that:

$$r_i + r_j \le D_{ij} \quad \& \quad r_i, r_j\in N \tag{3}$$

It can be also noticed that in a no-wait 2-P system of cyclic processes $SP_R = \{P_i,P_j\}$, a time distance $dX_{ij} = x_j(l) - x_i(k)$ between any start times $0 \le x_i(k) \le x_j(l)$ is such that the following relations hold:

if ($x_i(k) \le x_j(l)$) then

$$y_{ij} = dX_{ij} \bmod D_{ij} \quad \& \quad y_{ij}\in[r_i, D_{ij} - r_j], \tag{4}$$

where *mod* - modulo operation.

The last condition is due to the fact (1), and $x_j(l+w) = x_j(l) + w\cdot c_j = x_i(k) + t_{ij}(l) + w\cdot c_j$, where $w\in N\cup\{0\}$, and the relation $dX_{ij} = x_j(l+w) - x_i(k) = (t_{ij}(l) + w\cdot c_j) = f_{ij}\cdot D_{ij} + y_{ij} + w\cdot D_{ji}m_{ji} = (f_{ij} + w\cdot m_{ji})\cdot D_{ij} + y_{ij}$, hence $dX_{ij} \bmod D_{ij} = y_{ij}$.

Similarly, a time distance $dX_{ji} = x_i(k) - x_j(l)$ between any start times $0 \le x_j(l) \le x_i(k)$ is such that the following holds:

if ($x_j(l) \le x_i(k)$) then

$$y_{ji} = dX_{ji} \bmod D_{ji} \quad \& \quad y_{ji}\in[r_j, D_{ji} - r_i] \tag{5}$$

In the case of 2-PE subsystem composed of a set of processes $SP_R = \{P_i,P_j\}$ sharing the $R\in RES$ resource, where each process may carry out more than two operations (Fig.3). Let us assume that R is used by the O_{ia} operation in the P_i process, and by O_{jb} in the P_j process, i.e. the following dependencies are fulfilled $R(O_{ia})=R_{ia}=R$ and $R(O_{jb})=R_{jb}=R$, and $Z_i = (R_{i1}, \ldots, R_{ia}, \ldots, R_{ik(i)})$, and $Z_j = (R_{j1}, \ldots, R_{jb}, \ldots, R_{jk(j)})$. Suppose that the starting times for operations in subsequent process cycles are defined by $x_{ia}(k)$, $x_{jb}(k)\in N\cup\{0\}$, $k=0,1,2, \ldots,\infty$. Let the starting times of the operations O_{ia} and O_{jb} at the initial state of the CPS are known, i.e. $x_{ia}(0) = x_{ia}$, and $x_{jb}(0) = x_{jb}$, and x_{ia}, $x_{jb}\in N\cup\{0\}$. Taking into account, the relations (4) and (5) the following theorem presents the conditions for the existence of a no-wait cyclical schedule for the CPS composed of the 2-PE subsystems sharing resources.

Theorem 1. Let a CPS$=\{P_1, \ldots, P_n\}$ be given. Assume, that the start times of the processes operations are equal to $X_i(0) = (x_{i1}(0), \ldots, x_{ik(i)}(0))$, where $x_{i1}(0)\in[0, c_i)$, $x_{i2}(0)=x_{i1}(0)+r_{i1}, \ldots, x_{ik(i)}(0)= x_{i(k(i)-1)}(0)+r_{i(k(i)-1)}$, and $(i=1,2,\ldots,n)$. Let 2-PE be a subsystem of the CPS that consists of processes $SP_R = \{P_i,P_j\}$ sharing the resource $R\in RES$, and $R(O_{ia})=R_{ia}=R$, and $R(O_{jb})=R_{jb}=R$, and starting times $x_{ia}(0) = x_{i1}(0)+\Sigma_{h=1,\ldots,a-1}r_{ih}$ & $x_{jb}(0) = x_{j1}(0)+\Sigma_{h=1,\ldots,b-1}r_{jh}$ are calculated taking into account $x_{i1}(0)\in[0,c_i)$ and $x_{j1}(0)\in[0,c_j)$. The necessary and sufficient condition for existence of a no-wait cyclic schedule for the CPS with the initial state given by $x(0) = (x_{11}(0), x_{21}(0), \ldots, x_{n1}(0))$ and $x_{i1}(0)\in[0,c_i)$ is defined as follows:

- for each subsystem 2PE, such that $0 \le x_{ia}(0) \le x_{jb}(0)$, and $x_i(k) = x_{ia}(0)$, and $x_j(l) = x_{jb}(0)$, the relation (4) holds;

- for each subsystem 2PE, such that $0 \le x_{jb}(0) \le x_{ia}(0)$, and $x_j(l) = x_{jb}(0)$, and $x_i(k) = x_{ia}(0)$, the relation (5) holds.

Theorem 1 results in a procedure enabling quick verification whether a no-wait cyclic schedule exists for a CPS with a fixed structure and a fixed initial state, and also determining all possible initial states for which such a schedule exists. One may notice that a cycle time of the no-wait schedule is equal to $T=lcm(c_1, \ldots, c_n)$, where lcm denotes the least common multiple, and c_i - cycle time of process P_i.

3.2 Procedure of no-wait cyclic schedule determination

Consider a CPS$=\{P_1, \ldots, P_n\}$ system with fixed parameters and a fixed initial state $x(0) = (x_{11}(0), x_{21}(0), \ldots, x_{n1}(0))$, where $x_{i1}(0)\in[0,c_i)$, that meets the required constraints concerning its structure defined by its G-graph. The following steps must be taken to verify whether it is possible to construct a no-wait cyclic schedule for the CPS:

1) For each shared resource $R\in RES$ determine sets of processes using this resource $SP_R = \{P_i, P_j\}$, and additionally for each process determine the operations O_{ia} and O_{jb}, using this resource, i.e. those for which $R(O_{ia})=R_{ia}=R$, and $R(O_{jb})= R_{jb}=R$.

2) For each 2-PE subsystem, composed of set of processes $SP_R = \{P_i,P_j\}$, calculate the process cycle times c_i, c_j, and $D_{ij}=D_{ji}=gcd(c_i, c_j)$. Check if the conditions (3) necessary for the existence of a no-wait cyclic schedule are met for each subsystem 2-PE.

3) If conditions (3) are satisfied for each 2-PE, determine the start times of the operations $x_{ia}(0) = x_{i1}(0) + \Sigma_{h=1,\ldots,a-1}r_{ih}$ & $x_{jb}(0) = x_{j1}(0) + \Sigma_{h=1,\ldots,b-1}r_{jh}$, where $x_{i1}(0)\in[0,c_i)$ and $x_{j1}(0)\in[0,c_j)$.

4) For starting times such that $0 \le x_{ia}(0) \le x_{jb}(0)$, calculate (4) $y_{ij} = dX_{ij} \bmod D_{ij}$, where $dX_{ij} = x_{jb}(0) - x_{ia}(0)$. If $y_{ij}\in[r_i, D_{ij} - r_j]$,

then there is no-waiting for the shared resource R in the subsystem 2-PE. In the same way, for starting times such that $0 \le x_{jb}(0) \le x_{ia}(0)$, calculate (5) $y_{ji} = dX_{ji} \ mod \ D_{ji}$, where $dX_{ji} = x_{ia}(0) - x_{jb}(0)$. If $y_{ji} \in [r_j, D_{ji} - r_i]$, then there is no-waiting for the shared resource R in the subsystem 2-PE.

5) All 2-PE subsystems shall be checked in the same way. If the conditions of Theorem 1 are fulfilled in the initial state $x(0)$ under consideration for each 2-PE subsystem, then this state belongs to a no-wait cyclic schedule of the CPS.

The example of applying of the method is presented in the next section. It is shown the procedure of finding all initial states $x(0) = (x_{11}(0), \ldots, x_{n1}(0))$, where $x_{i1}(0) \in [0, c_i)$, of the processes for which no-wait cyclic schedules exist.

4. EXAMPLE OF SCHEDULING

Let us consider the CPS=(P_1,P_2,P_3), shown in Fig.4, defined by the following relations: $ZO_1 = (O_{11},O_{12},O_{13},O_{14})$, $Z_1 = (R_{11},R_{12},R_{13},R_{14}) = (R_1,R_5,R_2,R_6)$; $ZO_2 = (O_{21},O_{22},O_{23},O_{24})$, $Z_2 = (R_{21},R_{22},R_{23},R_{24}) = (R_2,R_7,R_3,R_8)$, and $ZO_3 = (O_{31},O_{32},O_{33},O_{34})$, and $Z_3 = (R_{31},R_{32},R_{33},R_{34}) = (R_3,R_4,R_1,R_9)$. The set of shared resources $RES = \{R_1, R_2, R_3\}$. The operation times: $ZT_1 = (r_{11}, r_{12}, r_{13}, r_{14}) = (2,4,2,1)$; $c_1=9$; $ZT_2 = (r_{21}, r_{22}, r_{23}, r_{24}) = (1,1,3,1)$; $c_2=6$; $ZT_3 = (r_{31}, r_{32}, r_{33}, r_{34}) = (1,2,1,8)$; $c_3=12$. According to (1): $D_{12}=D_{21}= gcd(c_1,c_2)=3$, $m_{12}=3$, $m_{21}=2$, and $c_1=D_{12}m_{12}$, $c_2=D_{21}m_{21}$; $D_{13}=D_{31}= gcd(c_1,c_3)=3$, $m_{13}=3$, $m_{31}=4$, and $c_1=D_{13}m_{13}$, $c_3=D_{31}m_{31}$; $D_{23}=D_{32}=gcd(c_2,c_3)=6$, $m_{23}=1$, $m_{32}=2$, and $c_2=D_{23}m_{23}$, $c_3=D_{32}m_{32}$. We can decompose the CPS into three 2-process subsystems that are sharing one resource: 2-PE$_{1(13)}$ - the subsystem of processes $SP_1 = \{P_1,P_3\}$ sharing resource R_1, and 2-PE$_{2(12)}$ - the processes $SP_2 = \{P_1,P_2\}$ sharing resource R_2, and 2-PE$_{3(23)}$ - the processes $SP_3 = \{P_2,P_3\}$ sharing R_3 (for $i(kl)$ - i denotes a number of shared resource, and (kl) the numbers of processes. The following relations hold: $R(O_{11})=R(O_{33})=R_1$; $R(O_{13})=R(O_{21})=R_2$; $R(O_{23})=R(O_{31})=R_3$, and for SP_1 we have $r_{11}+r_{33} = 2+1 = 3 \le D_{13}=3$; for SP_2 there is $r_{13}+r_{21} = 2+1 = 3 \le D_{12}=3$; and for SP_3 there is $r_{23}+r_{31} = 3+1 = 4 \le D_{23}=6$. The last relations guarantee that the condition (3) necessary for no-wait execution of the 2-PE subsystems is satisfied. We define the initial state of the system $x(0) = (x_{11}(0), x_{21}(0), x_{31}(0))$, where $x_{11}(0) \in [0,c_1)$, $x_{21}(0) \in [0,c_2)$, $x_{31}(0) \in [0,c_3)$, and using it we calculate all starting times of the operations: $x_{12}(0)$, $x_{13}(0)$, $x_{14}(0)$, and $x_{22}(0)$, $x_{23}(0)$, $x_{24}(0)$, and $x_{32}(0)$, $x_{33}(0)$, $x_{34}(0)$. For the set $SP_1 = \{P_1,P_3\}$ we calculate the time distance (4), (5) between start times $x_{11}(0)$, $x_{33}(0)$ of the operations using the resource R_1, i.e. $dX_{1(13)} = (x_{33}(0) - x_{11}(0))$, and $dX_{1(31)} = (x_{11}(0) - x_{33}(0))$, where $dX_{i(kl)}$ denotes the time distance between operations of the processes P_k, P_l using resource R_i. In the following we determine the conditions guaranteeing that there is no waiting for the resource R_1 in the subsystem 2-PE$_{1(13)}$: if ($x_{11}(0) \le x_{33}(0)$) then $\{y_{1(13)} = dX_{1(13)} \ mod \ D_{13}$ & $y_{1(13)} \in [r_{11}, D_{13} - r_{33}]\}$, otherwise if ($x_{33}(0) \le x_{11}(0)$) then $\{y_{1(31)} = dX_{1(31)} \ mod \ D_{31}$ & $y_{1(31)} \in [r_{33}, D_{31} - r_{11}]\}$. Similarly, we define the no-wait constraints for the subsystem 2-PE$_{2(12)}$: if ($x_{13}(0) \le x_{21}(0)$) then $\{y_{2(12)} = dX_{2(12)} \ mod \ D_{12}$ & $y_{2(12)} \in [r_{13}, D_{12} - r_{21}]\}$, otherwise if ($x_{21}(0) \le x_{13}(0)$) then $\{y_{2(21)} = dX_{2(21)} \ mod \ D_{21}$ & $y_{2(21)} \in [r_{21}, D_{21} - r_{13}]\}$, and for the subsystem 2-PE$_{3(23)}$: if ($x_{23}(0) \le x_{31}(0)$) then $\{y_{3(23)} = dX_{3(23)} \ mod \ D_{23}$ & $y_{3(23)} \in [r_{23}, D_{23} - r_{31}]$ $\}$, otherwise if ($x_{31}(0) \le x_{23}(0)$) then $\{y_{3(32)} = dX_{3(32)} \ mod \ D_{32}$ & $y_{3(32)} \in [r_{31}, D_{32} - r_{23}]\}$. Performing calculations using IBM ILOG CPLEX or just Microsoft Excel tool, in the present case, there are $c_1 \cdot c_2 \cdot c_3 - (c_1-1) \cdot (c_2-1) \cdot (c_3-1) = 208$ different initial states $x(0) = (x_{11}(0), x_{21}(0), x_{31}(0))$, such that $x_{11}(0) \in [0,9)$, $x_{21}(0) \in [0,6)$, $x_{31}(0) \in [0,12)$, and ($x_{11}(0)=0$ or $x_{21}(0)=0$, or $x_{31}(0)=0$), but only 10 of them satisfies the conditions for no-wait execution of the processes given by the Theorem 1. All these states (i.e. (0,2,2), (0,2,8), (0,5,5), (0,5,11), (1,0,0), (1,0,6), (4,0,0), (4,0,6), (7,0,0), (7,0,6)) belong to the same cyclic schedule with $y_{1(13)}=2$, $y_{2(12)}=2$, and $y_{3(23)}=4$. A Gantt chart in Fig.6 presents a no-wait cyclic schedule obtained for the initial state $x(0) = (x_{11}(0)=1, x_{21}(0)=0, x_{31}(0)=0)$. Its cycle time: $T = lcm(9,6,12) = 36$. The numbers shown in the diagram represent the resources used by a given process to carry out a specific operation. Each number is at the beginning of the operation (e.g. in the case of the P_1 process, the sequence of numbers (1,5,2,6) mean resources (R_1, R_5, R_2, R_6), which are necessary to carry out subsequent operations).

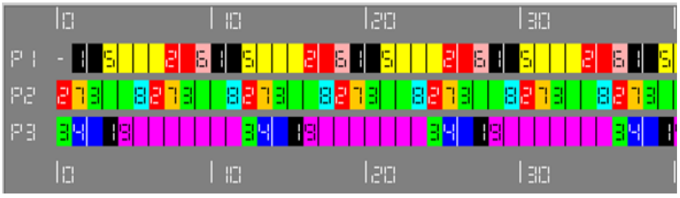


Fig. 6. Example of a no-wait cyclic schedule of the CPS = (P_1,P_2,P_3) for process starting times $x(0) = (1, 0, 0)$, where (1, 5, 2, 6) denote numbers of the resources used by the process P_1, (2, 7, 3, 8) – by P_2, and (3, 4, 1, 9) by P_3.

5. CONCLUSIONS

The paper presents a method for determining no-wait cyclic schedules for cyclical process systems with resource blocking where each resource is used by at most two processes. Additionally, the graph describing the structure of resource requests specified by process operations does not contain cycles that may lead to system deadlock. The developed method enables quick verification of whether there are process start times in a system of processes with a fixed structure belonging to a no-wait cyclic schedule. Thanks to the introduced constraints on the structure of processes, the condition of the existence of a cyclical no-wait schedule for a process system being in a given initial state is equivalent to the conjunction of the necessary and sufficient conditions for the existence of no-wait schedules for each extended 2-PE subsystem consisting of two processes sharing a single resource. If at least one of the 2-PE subsystems does not meet the conditions, which may be verified in the time depending on the number of resources in the system, then it is also not possible to construct a no-wait cyclic schedule for the whole system. Future research may concern the analysis of systems in which multiple processes can share a single resource, and then the development of a method for designing cyclical no-wait schedules for job-shop process systems with the possibility of deadlock.

REFERENCES

Abadi, I. N. K., Hall, N. G., Sriskandarajah, C. (2000). Minimizing cycle time in a blocking flowshop. *Oper. Research*, 48, 177-180.

AitZai, A., Benmedjdoub, B., Boudhar, M. (2012). A branch and bound and parallel genetic algorithm for the job shop scheduling problem with blocking. *Int. J. Oper. Res*., 14, (3), 343-365.

Allahverdi, A. (2016). A survey of scheduling problems with no-wait in process. *European Journal of Operational Research*, 255, (3), 665–686.

Alpan, G., Jafari, M. A. (1997). Dynamic Analysis of Timed Petri Nets: a Case of Two Processes and a Shared Resource. *IEEE Trans. on Robotics and Aut.*, 13, (3), 338-346.

Alpan, G., Jafari, M. A. (1998). Synthesis of sequential controller in the presence of conflicts and free choices. *IEEE Trans. on Robotics and Aut.*, 14, (3), 488-492.

Aschauer, A., Roetzer, F., Steinboeck, A., Kugi, A. (2017). An Efficient Algorithm for Scheduling a Flexible Job Shop with Blocking and No-Wait Constraints. *20th IFAC World Congress*. IFAC-PapersOnLine, 50, (1), 12490-12495.

Banaszak, Z., Krogh, B. (1990). Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows. *IEEE Trans. on Robotics and Automation*, volume (6), no.6, 724-734.

Bocewicz, G., (2014). Robustness of Multimodal Transportation Networks. In: *Eksploatacja i Niezawodność - Maintenance and Reliability*, 16, (2), 259–269.

Bocewicz, G., Nielsen, I., Banaszak, Z. (2014). Automated Guided Vehicles Fleet Match-up Scheduling with Production Flow Constraints. *Engineering. Applications of Artificial Intelligence*, volume (30), 49-62.

Bocewicz G., Nielsen I., Banaszak Z. (2016). Production flows scheduling subject to fuzzy processing time constraints. *International Journal of Computer Integrated Manufacturing*, 29, (10), 1105-1127.

Bocewicz, G., Nielsen, P., Banaszak, Z. (2019). Declarative Modeling of Milk-Run Vehicle Routing Problem for Split and Merge Supply Streams Scheduling. *Advances in Intelligent Systems and Computing*, 853, 157–172.

Brucker, P., Kampmeyer, T. (2008). Cyclic job shop scheduling problems with blocking. *Ann. Oper. Res*., 159, (1), 161-181.

Hall, N. G., Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44, (3), 510-525.

Kamoun, H., Sriskandarajah, C. (1993). The complexity of scheduling jobs in repetitive manufacturing systems. *European Journal of Oper. Research*, 70, 350–364.

Kampmeyer, T. (2006). Cyclic scheduling problems. *Ph.D. Dissertation,* Mathematik/Informatik, Universität Osnabrück.

Kumar, S., Bagchi, T. P., Sriskandarajah, C. (2000). Lot streaming and scheduling heuristics for m-machine no-wait flowshops. *Comput. Indust. Eng*., 38, 149-172.

Levner, E., Kats, V., Alcaide, D., Pablo, L., Cheng, T.C.E. (2010). Complexity of Cyclic Scheduling Problems: A State-of-the-Art Survey. *Computers and Industrial Engineering*, volume (59) (2), 352-361.

Louaqad, S., Kamach, O., Iguider, A. (2018). Scheduling for job shop problems with transportation and blocking no-wait constraints. *Journal of Theoretical and Applied Information Technology*, 96, (10), 2782-2792.

Mascis, A., Pacciarelli D. (2002). Job-shop scheduling with blocking and no-wait constraints. *European Journal of Oper. Research*. 143, (3), 498-517.

Patel, D., Patel, M. B., Vadher, J. A. (2014). Implementation of milk run material supply system in vehicle routing problem with simultaneous pickup and delivery. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, 3, (11), 122–124.

Pinedo, M. (2005). Planning and scheduling in manufacturing and services. New York: Springer.

Schuster, Ch. J., Framinan, J. M. (2003). Approximative procedures for no-wait job shop scheduling. *Operations Research Letters*, 31, (4), 308–318.

Schuster, Ch. J. (2006). No-wait Job Shop Scheduling: Tabu Search and Complexity of Subproblems. *Mathematical Methods of Operations Research*, 63, (3), 473–491.

Sitek, P., Wikarek, J. (2019). Capacitated Vehicle Routing Problem with Pick-up and Alternative Delivery (CVRPPAD) – model and implementation using hybrid approach. *Annals of Operations Research*, 273, (1-2), 257–277.

Smutnicki, Cz. (2009). Minimizing cycle time in the manufacturing system based on the flow of various jobs. *13th IFAC Symposium on Inf. Control Problems in Manufacturing*. IFAC Proc. Vol. 42, (4), 1137-1142.

Wójcik, R. (2001). Towards strong stability of concurrent repetitive processes sharing resources. *Systems Science*, volume (27), no. 2, 37-47.

Wójcik, R. (2018). Designing a No-Wait Cyclic Schedule for a Class of Concurrent Repetitive Production Processes. *16th IFAC Symp. on Information Contr. Problems in Manufact.* IFAC-PapersOnLine, 51, (11), 1305-1310.

Wójcik, R., Pempera, J. (2019). Designing Cyclic Schedules for Streaming Repetitive Job-Shop Manufacturing Systems with Blocking and No-Wait Constraints. *13th IFAC Workshop on Intelligent Manufacturing Systems, IMS 2019.* IFAC- PapersOnLine , 52, (10), 73–78.

Zaremba, M. B., Jędrzejek, K. J., Banaszak Z., A. (1998). Design of steady-state behavior of concurrent repetitive processes: an algebraic approach. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28, (2), 199-212.

Zhang, J., Ding, G., Zou, Y., Qin, S., Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *J. Intell. Manuf.*, 30, (4), 1809–1830.